Empowering IoT Developers with Privacy-Preserving End-User Development Tools

ATHEER ALJERAISY, Cardiff University, UK and Majmaah University, Saudi Arabia
OMER RANA, Cardiff University, UK
CHARITH PERERA, Cardiff University, UK

Internet of Things applications (IoT) have the potential to derive sensitive user data, necessitating adherence to privacy and data protection laws. However, developers often struggle with privacy issues, resulting in personal data misuse. Despite the proposed Privacy by Design (PbD) approach, criticism arises due to its ambiguity and lack of practical tools for educating software engineers. We introduce Canella, an integrated IoT development ecosystem with privacy-preserving components leveraging End-User Development (EUD) tools Blockly@rduino and Node-RED, to help developers build end-to-end IoT applications that prioritize privacy and comply with regulations. It helps developers integrate privacy during the development process and rapid prototyping phases, offering real-time feedback on privacy concerns. We start by conducting a focus group study to explore the applicability of designing and implementing PbD schemes within different development environments. Based on this, we implemented a proof-of-concept prototype of Canella and evaluated it in controlled lab studies with 18 software developers. The findings reveal that developers using Canella created more privacy-preserving applications, gained a deeper understanding of personal data management, and achieved better privacy compliance. Our results also highlight Canella's role in educating and promoting privacy awareness, enhancing productivity, streamlining privacy implementation, and significantly reducing cognitive load. Overall, developers found Canella and its privacy-preserving components useful, easy to use, and easy to learn, which could potentially improve IoT application privacy. Watch the demo video.

CCS Concepts: • **Security and privacy** → *Usability in security and privacy*; • **Human-centered computing** → *Empirical studies in ubiquitous and mobile computing*; • **Software and its engineering** → *Integrated and visual development environments.*

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

ACM Reference Format:
Atheer Aljeraisy, Omer Rana, and Charith Perera. 2024. Empowering IoT Developers with Privacy-Preserving End-User Development Tools. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 8, 3, Article 90 (September 2024), 47 pages. https://doi.org/10.1145/3678588

1 Introduction

The Internet of Things (IoT) involves interconnected physical objects, or "things," that gather, process, and exchange data with other systems [85, 109]. This ecosystem, including wearable devices monitoring users' activities and health data, presents unique privacy challenges [70, 109]. Unlike traditional applications, IoT systems excel in data collection, device variety, and complex data pathways, significantly amplifying privacy and compliance challenges [85]. The widening accessibility of IoT development has expanded the developer

Authors' Contact Information: Atheer Aljeraisy, aljeraisya@cardiff.ac.uk, Cardiff University, Cardiff, UK and Majmaah University, Majmaah, Saudi Arabia; Omer Rana, ranaof@cardiff.ac.uk, Cardiff University, Cardiff, UK; Charith Perera, PereraC@cardiff.ac.uk, Cardiff University, Cardiff, UK.

ACM 2474-9567/2024/9-ART90
https://doi.org/10.1145/3678588

base but also increased the risk of privacy breaches due to the inclusion of developers with limited privacy expertise [65]. This shift, coupled with the IoT's complexity, complicates privacy protection efforts, necessitating the prioritization of user privacy throughout IoT development to ensure legal compliance.

However, developers face multiple challenges in addressing privacy concerns, leading to the mishandling of personal data in applications for several reasons. Firstly, developers often do not treat privacy as a primary concern [8, 17]. Secondly, they lack a comprehensive understanding of safeguarding users' data [48]. Thirdly, developers limited knowledge of their app's data practices complicates implementing privacy requirements [18]. For instance, when apps use third-party libraries [18, 100] or lack proper documentation [62], developers may not fully grasp data practices. Lastly, there is inadequate support to keep up with evolving legal requirements [63].

In response to the growing importance of user privacy, various countries have enacted privacy and data protection laws, such as the General Data Protection Regulations (GDPR) in Europe and the UK [79], as well as the California Consumer Privacy Act (CCPA) in the United States [61]. Ensuring compliance with these laws and integrating their regulatory requirements into software systems poses significant challenges for software professionals [11, 106]. Additionally, the responsibility for addressing legal implications is typically assigned to legal compliance teams rather than developers [78]. Moreover, these regulations often offer broad, general guidelines for comprehensive coverage at a higher level, making it difficult to apply them to specific software implementation details [31].

To integrate privacy more effectively into software development and avoid treating it as an afterthought, Privacy by Design (PbD) has been advocated [25, 26]. Endorsed by the EU's GDPR, which mandates PbD's application in all data processing systems, a cornerstone of IoT technology, PbD's practicality has nonetheless been questioned for its vagueness and the absence of specific implementation tools [47, 95, 99, 101, 110]. This challenge is heightened in IoT's diverse device landscape, where developers struggle to meet privacy requirements. With increasing penalties for non-compliance, there is a critical need for PbD tools tailored for IoT, bridging the theoretical-practical gap to ensure privacy integration throughout development.

Utilizing End-User Development (EUD) techniques like Blockly and Node-RED can simplify complex IoT applications, making them more manageable [69]. EUD aims to enable users to create systems suitable for their skills and needs [19]. Adding interactivity to these tools can streamline privacy integration and enhance user friendliness. Interactivity can take the form of alerts, real-time feedback, or highlights [37, 54, 68]. When two EUD tools interact, they become a powerful platform, especially valuable in the diverse IoT architecture with varying computational capabilities [20, 81]. This integration can empower users to build customized, privacy-aware IoT applications, leveraging the strengths of different tools while complying with privacy and legal requirements. Adopting such an approach equips developers to effectively navigate the IoT's unique privacy challenges, seamlessly integrating PbD as a fundamental component of IoT application development.

In our prior work [13], we introduced an initial concept and early demonstration of Canella, an integrated IoT development ecosystem with privacy-preserving components. In this paper, we provide a comprehensive and significantly expanded version of our research on Canella. Canella utilizes Blockly@rduino and Node-RED to help developers create privacy-aware IoT applications, ensuring compliance with privacy and data protection laws and offering real-time feedback on potential issues. It promotes privacy-conscious practices by encouraging developers to consider the effects of their applications and offering less risky options for handling personal data. Canella includes a privacy law validator to verify data compliance with regulations when transmitting data from Blockly@rduino to Node-RED. Overall, We believe Canella bridges the gap between software developers and privacy requirements both in operational and implementation aspects, fostering privacy compliance and enhancing awareness among developers. Building upon prior research in privacy-aware software development tools, Canella introduces a novel approach tailored for the IoT context, designed for programming IoT applications. It addresses the unique challenges of IoT by providing developers with tools that integrate privacy-preserving measures and compliance with privacy and data protection laws. This paper's contributions include:

- We conducted a focus group study to explore the design and implementation of specific privacy-preserving techniques of the PbD schemes within various EUD environments. This study provided valuable insights into the design and implementation of Canella's privacy-preserving components.
- We present the design and implementation of Canella, a privacy-compliant integrated IoT development ecosystem. It prompts developers to consider privacy during programming, offering real-time feedback on potential privacy issues. Canella enhances developers' understanding of personal data usage in their applications, introduces alternative privacy options, and increases their knowledge of privacy regulations.
- We provide the results of our lab studies on Canella. Our findings demonstrate that developers using Canella can create more privacy-preserving applications, reduce developers' cognitive load, increase their awareness, streamline privacy implementation, and facilitate adherence to privacy and data protection regulations. Overall, developers consider Canella and its privacy components useful and easy to use.

2 Related Work

2.1 Common Privacy Issues Developers Face

To effectively support developers in the IoT domain, understanding their perceptions and knowledge of privacy is paramount. Prior research underscores a concerning trend: developers often prioritize usability and desired features over privacy considerations when designing applications [8, 17, 48, 67]. This inclination can lead to a neglect of user privacy preferences, particularly evident in the IoT sector, where the pressure to swiftly enter the market often overshadows privacy concerns [14, 17]. Interviews, surveys, and observations have shed light on this phenomenon [18, 62, 92, 96], revealing that while some developers express concern for user privacy, many lack a comprehensive understanding of privacy principles [62]. Moreover, their use of data security terminology, influenced by organizational climates, may further hinder effective privacy implementation [14, 48]. This disconnect between user expectations and developer priorities [96] underscores the urgent need to bridge the gap in understanding and integrate privacy considerations more effectively into app development processes.

In addition to conceptual challenges, developers also face technical hurdles in ensuring privacy protection. One common issue is their limited awareness of potential privacy violations and the corresponding mitigation strategies. For instance, certain third-party libraries used for advertising and analytics may automatically transmit users' personal data to service providers, a fact of which many developers remain unaware due to the complexity of privacy policies and limited exposure to privacy guidelines [18]. This lack of awareness not only impedes the creation of privacy-preserving applications but also complicates efforts to understand the data collected and user privacy concerns within specific application contexts [64, 93]. Therefore, developers must acquire a thorough understanding of privacy concerns and data practices to effectively navigate the complexities of privacy preservation in the development of IoT applications [18]. Addressing both conceptual and technical challenges is crucial for privacy-conscious development and protecting user data in the IoT ecosystem.

2.2 Developers Engagement with Privacy in Practice

Understanding developers' current data practices and the challenges they face is paramount for effectively addressing privacy concerns in software development. Recent studies have investigated developers' engagement with privacy requirements, shedding light on their efforts to safeguard user privacy in software systems [23, 92, 108, 111]. While much prior research focuses on the organizational perspective of privacy practices in software development [48], it is evident that the culture and policies within organizations significantly influence developers' ability to integrate privacy effectively into software applications [10]. Larger organizations typically provide general privacy guidance, while smaller companies face greater challenges in applying privacy measures independently [10]. Clear policies and guidelines are essential for organizations to guide developers in implementing privacy measures effectively [53, 96].

Researchers have highlighted developers' struggles in incorporating privacy into their applications, citing difficulties in understanding privacy requirements and integrating them into software systems [16, 96]. Challenges also arise in estimating privacy risks from the user's viewpoint, particularly in IoT scenarios [74, 92]. Providing guidelines and best practices can empower developers to make privacy-preserving decisions when creating applications, addressing these challenges and fostering a culture of privacy-conscious development [53, 92].

2.3 Privacy and Data Protection Laws and Privacy by Design

As privacy concerns have grown, many countries have implemented privacy and data protection laws. These include Europe's GDPR [79], Canada's PIPEDA [76], the U.S.'s CCPA [61], Australia's APPs [15], and New Zealand's Privacy Act 2020 [75]. Aljeraisy et al. [11] created the Combined Privacy Law Framework (CPLF) to unify principles and rights across these laws. Yet, translating these laws into actionable software requirements poses significant challenges for developers, particularly within the IoT context, where the complexity, heterogeneity, and real-time data processing exacerbate compliance difficulties [14, 85, 106].

To bridge this gap, researchers and regulators have proposed privacy engineering frameworks, notably PbD, which integrates privacy throughout the software development process [25, 26]. This concept has generated various PbD schemes, principles [24–26, 41, 52, 112], strategies [50, 88], guidelines [34, 84], and patterns [28, 29]. However, ongoing debates surround privacy measures' early-stage inclusion to avoid cost and launch delays [32]. Aljeraisy et al. [12] connect privacy schemes with regulations through the CPLF, emphasizing the challenges software developers face in applying these frameworks to IoT's architectures. The necessity for Data Protection by Design and Default (DPbD) [1] becomes evident as developers ensure proper personal data handling across IoT's diverse nodes and resource constraints, balancing privacy with system functionality [11, 42].

Yet, the practical application of PbD remains limited due to a lack of tools that cater to the intricacies of IoT's device ecosystem and data flows, exacerbated by the involvement of developers of varying expertise [10, 12, 47, 65, 95, 99, 101, 110]. This gap, coupled with the potential for significant GDPR penalties of up to 2% of annual turnover [39], underscores the critical need for developing PbD tools specifically designed for the IoT. Such tools must not only align with comprehensive privacy and data protection laws but also ensure that complex legal requirements can be translated into actionable software development practices, accessible to developers across all skill levels. This rationale justifies our effort to enhance developers' tools for effectively integrating privacy measures in IoT applications.

2.4 Privacy Engineering Practices

The emergence of "privacy engineering" aims to enhance privacy practices in software development by drawing from foundational principles and regulatory frameworks [25, 42, 55, 79, 99]. It relies on pivotal Privacy Impact Assessments (PIAs) and Data Protection Impact Assessments (DPIAs) methodology [79] to identify and address privacy concerns in high-risk projects for PbD, DPbD, and GDPR compliance [79]. Challenges arise due to the manual nature of these assessments, especially in data-intensive technologies like IoT, where incorporating them into architecture faces limited adoption. Standards such as NIST provide guidance, but measuring privacy is complex, which can hinder the proactive inclusion of privacy considerations [10]. To encourage developer acceptance, clear privacy integration guidelines are essential, promoting simple techniques for immediate and visible results, especially crucial for IoT developers navigating the intricacies of privacy engineering [92, 94].

[1]DPbD is a legally required evolution of the concept of Privacy by Design. It consists of a set of recommended principles for best practices and highlights the importance of early involvement with data protection. Data Protection Authorities have widely accepted it [30, 42], it is endorsed by the European Commission [1, 44], and it is legally mandated by the GDPR [79]. DPbD advocates for different data protection measures like data minimization, using pseudonyms, ensuring transparency, and implementing appropriate system security.

To address these limitations, privacy engineering integrates protection measures into software development, especially in complex contexts like the IoT. Previous research has included privacy during development, making it more accessible and helping users understand privacy issues [43, 90]. For instance, Li et al. introduced Coconut, an Android Studio plugin that enhances privacy awareness, providing feedback on privacy issues [62]. Honeysuckle helps create in-app privacy notices for Android developers [64]. However, these tools are designed for Android platforms, and there remains a notable gap in meeting the IoT's specific privacy needs. Though Alhirabi et al. presented PARROT, an interactive tool for designing privacy-aware IoT applications [10], our approach enriches developers' tools with privacy-preserving measures, streamlining end-to-end IoT application development with privacy in mind. Canella eliminates the need for coding, enabling developers to effortlessly incorporate privacy guidelines and balance privacy and application functionality. What sets Canella apart is its consideration of IoT's unique characteristics, taking into account diverse computational capabilities and resource constraints. It offers developers an immediate understanding of privacy-related choices and guarantees adherence to privacy laws.

3 Methodology

Our methodology involves conducting an in-depth empirical study that includes various phases and involves participants with varying levels of expertise. The research consists of three distinct phases, as follows (Figure 1):

- **Focus Group Phase:** In the initial phase (Section 4), we conducted a focus group study with 100 participants, divided into 25 groups of software developers to explore the applicability of the design and implementation of the PbD schemes, as they are the mechanism to comply with privacy and data protection laws. The findings provided valuable insights into the development and implementation of the privacy-preserving components of Canella.
- **Designing and Prototyping Phase:** In phase two (Section 5), we constructed the architecture of Canella. Subsequently, we designed and implemented the privacy-preserving components using Blockly@rduino and Node-RED, informed by insights from the focus group study, aligned with the principles of the CPLF and guidelines of the PbD schemes. We created a fitness-tracking IoT application scenario for Canella. Drawing upon our findings, we developed a proof-of-concept prototype of Canella. We then updated the privacy-preserving components to ensure the inclusion of all data types, each having specific criteria for computation. The study was piloted and further tested by three software developers.
- **Evaluation Phase:** In Phase three (Section 6), we conducted in-lab studies with 18 software developers. To ensure objectivity and broad applicability of the privacy assessment criteria, we consulted two privacy researchers and a privacy lawyer well-versed in PbD schemes. Additionally, we included a fitness trainer with seven years of experience to provide insights into how data granularity impacts trainee monitoring. Combining inputs from these privacy experts and the fitness trainer helps mitigate potential bias and ensure impartiality in the privacy assessment. This approach considers the developers' diverse backgrounds in fitness applications and user preferences, aiming to validate the practicality of privacy criteria in a real-world context. For a detailed explanation of the scoring criteria, see Appendix H.

4 Formative Study: Focus Group Study

4.1 Objective

To bridge the existing gap between software developers' theoretical knowledge and the practical application of PbD schemes in IoT application development, we embarked on a focused group study. Despite prior investigations into developers' privacy behaviors [10, 18, 62], little attention has been paid to the practical challenges of implementing privacy-preserving techniques. Our study specifically targets this gap, aiming to understand the applicability and feasibility of embedding PbD in various EUD environments, aligning with the principles and rights of the CPLF [11].

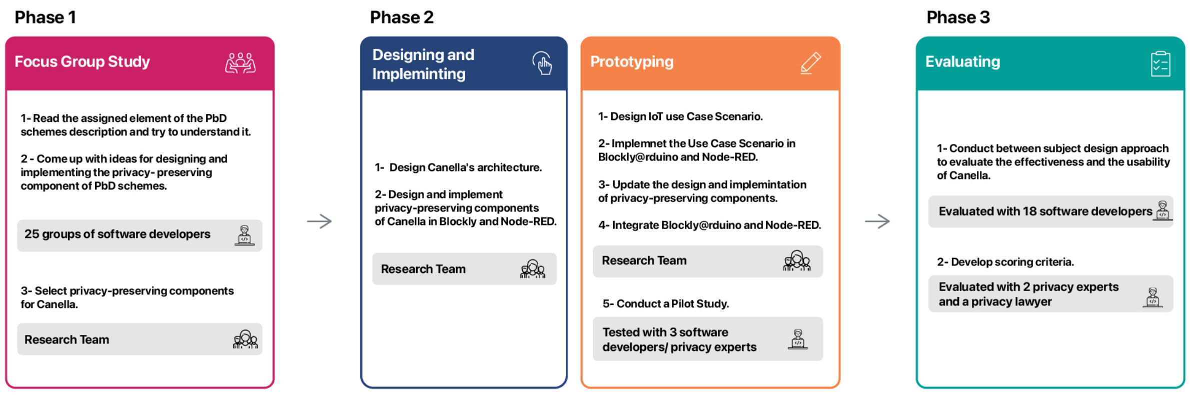

Phase 1
Focus Group Study
1- Read the assigned element of the PbD schemes description and try to understand it.
2 - Come up with ideas for designing and implementing the privacy- preserving component of PbD schemes.
25 groups of software developers
3- Select privacy-preserving components for Canella.
Research Team
Phase 2
Designing and Implementing
1- Design Canella's architecture.
2- Design and implement privacy-preserving components of Canella in Blockly and Node-RED.
Research Team
Prototyping
1- Design IoT use Case Scenario.
2- Implemnet the Use Case Scenario in Blockly@rduino and Node-RED.
3- Update the design and implemintation of privacy-preserving components.
4- Integrate Blockly@rduino and Node-RED.
Research Team
5- Conduct a Pilot Study.
Tested with 3 software developers/ privacy experts
Phase 3
Evaluating
1- Conduct between subject design approach to evaluate the effectiveness and the usability of Canella.
Evaluated with 18 software developers
2- Develop scoring criteria.
Evaluated with 2 privacy experts and a privacy lawyer

Fig. 1. This figure represents the methodology we followed to build a proof-of-concept for Canella. The methodology consists of three phases discussed in detail in Sections 4, 5, and 6.

4.2 Participants

We conducted a focus group study with computer science bachelor's students, a common practice in software engineering studies where students often serve as proxies for software engineers in industry settings [51, 89]. Our decision to involve students instead of professional software developers was supported by prior research showing consistent results when utilizing student participants, ensuring both methodological consistency and practical relevance [51, 89]. Specifically, we targeted students who were IoT app developers or mobile app developers utilizing sensors in their apps, such as location tracking. This choice was further justified by the scarcity of experienced IoT developers due to the field's novelty and rapid evolution [2, 57, 103]. Participants were selected based on their engagement in Emerging Technologies and Communication Networks courses, ensuring they possessed a foundational understanding of IoT concepts and practical experience through coursework projects. Recruitment was conducted via mailing lists, attracting 100 interested participants—25 second-year and 75 final-year students of BSc. Voluntary participation was crucial to our study and was not linked to grading.

While the study focuses on privacy in the IoT, it is situated within the broader context of software engineering. It aligns with software engineering research by investigating the integration of privacy considerations within the IoT software development process, reflecting the core principles of privacy engineering [97]. In our formative study, the focus is on the conceptual understanding and application of PbD in software development rather than deep, industry-specific experiences [9]. This endeavor involves essential software engineering tasks such as designing, implementing, and evaluating privacy-adherent solutions, highlighting the practical application of PbD schemes. Specifically, we explore how novice software developers, equipped with relevant education, perceive and plan to tackle these privacy challenges in IoT development.

4.3 Procedure

Our study comprised four sessions within a single day, engaging participants in intensive discussions on privacy-preserving techniques in IoT development. To accommodate diverse privacy concepts and class sizes, participants were divided into small groups of 4–5. Prior to the sessions, participants were briefed on the study's objectives and provided with writing materials, including Li et al.'s paper [62] to ground discussions in relevant context.

Participants engaged with a comprehensive set of guiding questions and PbD elements to foster active engagement and practical solution brainstorming. The focus was on privacy patterns [28, 29] for their concrete nature and a set of 30 privacy guidelines proposed by Perera [82, 84]. Each group tackled a distinct PbD element, encouraging diverse discussions on design and implementation within various EUD environments like Arduino IDE, Blockly, and Node-RED. Participants could choose another privacy-preserving technique to better respond to the discussions and provide effective suggestions and solutions. A researcher guided the focus groups, conducted in English to ensure participant understanding and promote idea sharing through discussions, pseudo-codes, and flowcharts. Participants explained their proposed approaches after each session, and these were recorded with their consent. Although written responses were collected for analysis, not all discussions were recorded due to time constraints.

4.4 Focus Group Study Analysis and Results

The focus group study was designed to explore how privacy-preserving techniques can be effectively integrated into IoT development environments to aid developers in adhering to privacy regulations. Specifically, the focus group study sought to answer the following main question: "To what extent can a specific privacy idea be designed and implemented in a development environment?" We analyzed participants' responses, combining sheet data, transcripts, and moderator notes, using a thematic analysis methodology [22]. The analysis revealed three key themes regarding the PbD schemes' design and implementation in IoT environments: elements of PbD schemes, proposed solutions, and ease of implementation.

- **Elements of PbD Schemes:** In the first session, participants were introduced to privacy patterns and found them difficult to understand and confusing. However, in the subsequent sessions with privacy guidelines, participants responded more positively due to their clarity and relevance to the IoT domain.
- **Proposed Solutions:** Discussions yielded diverse solutions, including flowcharts, interfaces, pseudocode, trade-offs, use cases, and proactive motivation, such as warning messages, tailored to the specifics of PbD schemes. Some of the practical solutions are visualized in Figure 11 in Appendix B, illustrating the range of proposed solutions. It is important to note that there are interconnections between these PbD guidelines, which implies interconnected solutions for addressing them.
- **Ease of Implementation:** Feedback varied, with some elements seen as straightforward and others as challenging, indicating a gap in understanding that necessitates further educational efforts. Participants struggled to grasp PbD elements and apply them in IoT contexts, echoing prior research findings [48, 63, 94]. However, our findings offer valuable insights into the development of privacy-preserving components, laying the foundation for the subsequent design phase of Canella (Section 5).

The results of the focus group study guided the design and implementation of Canella's privacy-preserving components based on the PbD guidelines suggested by Perera et al. [82]. These guidelines proved clearer in their specifications compared to other elements within the PbD schemes, particularly given their specificity to the IoT domain. After analyzing all participants' responses, we excluded the guidelines that were not properly supported by useful feedback. The resulting guidelines, presented in Figure 2, each align with one or more of the CPLF principles. For instance, the integration of 'Minimize Data Storage' into an IoT application corresponds to the CPLF's Data Minimization Principle. In this paper, we emphasized guidelines aligning with the Data Minimization Principle of the CPLF as foundational for designing and implementing privacy-preserving components. These include: (1) Reduce Location Granularity; (2) Minimize Raw Data Intake; (3) Category-Based Aggregation; and (4) Minimize Data Storage. Although it is vital to acknowledge fundamental principles like obtaining consent, in Hoepman's privacy design strategies for the software development lifecycle [50], data minimization was identified as the most critical principle. Additionally, data minimization is highlighted by all privacy and data protection laws of the CPLF (e.g., Article 5 in GDPR, Principle 4 in PIPEDA, 1798.100. (b) in CCPA, App 3 in the Australian Privacy Act, and Principle 1 in the New Zealand Privacy Act 2020) [12].

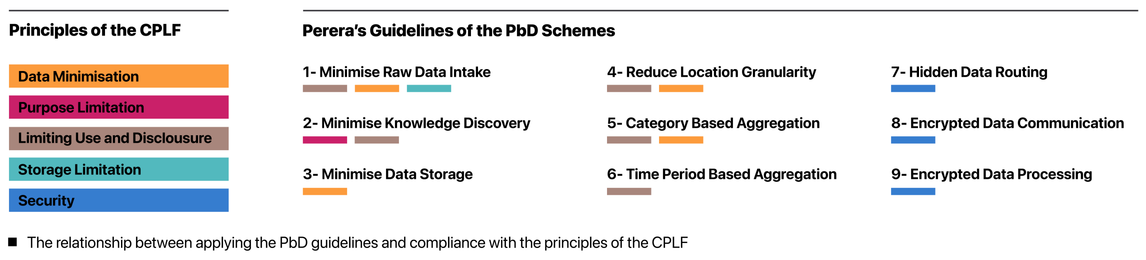

Principles of the CPLF
Data Minimisation
Purpose Limitation
Limiting Use and Disclousure
Storage Limitation
Security
Perera's Guidelines of the PbD Schemes
1- Minimise Raw Data Intake
2- Minimise Knowledge Discovery
3- Minimise Data Storage
4- Reduce Location Granularity
5- Category Based Aggregation
6- Time Period Based Aggregation
7- Hidden Data Routing
8- Encrypted Data Communication
9- Encrypted Data Processing
The relationship between applying the PbD guidelines and compliance with the principles of the CPLF

Fig. 2. The relationship between applying these PbD guidelines and compliance with the principles of the CPLF.

5 Canella Design and Implementation

The goal of Canella is to provide support to software developers who may lack expertise in privacy or encounter resource and time constraints during programming and rapid prototyping. To achieve this goal, we defined the following key objectives:

- **Create Privacy-Preserving Components:** This objective focuses on the development of components that can be integrated seamlessly into IoT applications, promoting privacy prioritization and compliance.
- **Foster a Privacy-Centric Mindset:** This objective aims to encourage developers to consider privacy an integral aspect throughout the data lifecycle of an IoT application.

In the following sections, we introduce Canella's architecture, detail the design and implementation of the privacy-preserving components, discuss key features and design rationale, present a hypothetical use-case scenario, explain the Canella prototype, and demonstrate its role in helping developers integrate privacy components for privacy-aware IoT applications.

5.1 Architecture of Canella

The architecture of Canella aligns with IoT application architecture, where data moves from sensors to gateways and then to the cloud [83], following the centralized pattern commonly used in IoT development [87]. This pattern encompasses IoT devices, gateways, and cloud platforms, each with diverse computational capabilities [46, 82].

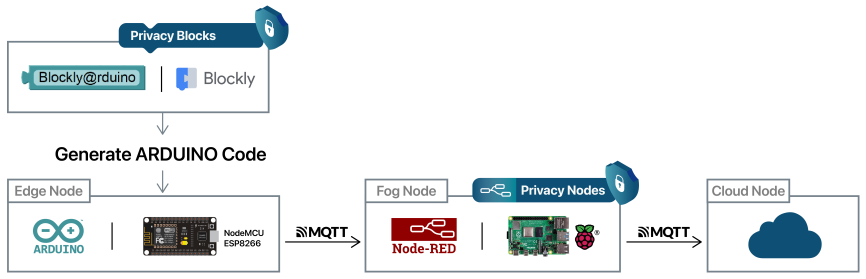


Fig. 3. Canella's architecture.

Canella utilizes two widely used visual programming languages with large user bases and active communities [40, 72, 91, 105, 107]: Blockly [3] and Node-RED [6]. Blockly, though prevalent in educational settings, also has applications beyond education, serving multiple purposes similar to Node-RED. Both tools offer accessible platforms for individuals lacking expertise in traditional programming paradigms, such as those with limited technical proficiency. This accessibility is crucial for facilitating the development of privacy-preserving applications, as it allows users to focus on designing and implementing privacy features without needing deep programming skills. Additionally, these tools support rapid prototyping and idea exploration, benefiting both novice and experienced developers. By incorporating built-in privacy features, Blockly and Node-RED illustrate the potential for integrating privacy considerations into IoT development tools. Their growing popularity and widespread adoption underscore their value in showcasing proof-of-concept implementations and encouraging further advancements within the IoT community [3, 6].

Figure 3 depicts Canella's architecture, which particularly utilizes Blockly@rduino [4]. It is a web-based visual programming editor for Arduino. It is based on Google's Blockly, which has been forked to generate Arduino's C/C++ code by visual dragging-and-dropping blocks. It represents the edge of an IoT application in Canella, using a NodeMCU ESP8266 microcontroller for data collection, processing, and storage. The Node-RED represents the fog in Canella's IoT application, running on a Raspberry Pi, a single-board microcomputer with complex computational capabilities. The Node-RED can access data from sensors connected to the ESP8266 board in several ways, including serial ports or WiFi, using the Message Queuing Telemetry Transport (MQTT) protocol. The Node-RED then forwards analytical summary information to the cloud via WiFi using MQTT for archiving.

Previous studies [11, 82] have emphasized the importance of taking specific privacy protection measures for each distinct IoT component based on its characteristics. This aligns with Canella's aim to assist developers in addressing privacy issues on edge and fog nodes in IoT applications to mitigate privacy risks associated with personal and sensitive data before it is transmitted to the cloud node. Accordingly, we created privacy blocks in Blocky@rduino and privacy nodes in Node-RED (Figure 3). These privacy-preserving components collaboratively assist developers in adhering to privacy requirements across the data lifecycle and constructing privacy-conscious end-to-end IoT applications. This approach corresponds to the five data lifecycle phases outlined by Perera et al., [82], promoting a systematic approach to data flow and privacy protection on each device or node.

The architecture of Canella primarily centers on WiFi-based IoT nodes due to the widespread adoption and familiarity of WiFi technology in IoT development. However, we acknowledge the diversity of the IoT landscape, encompassing technologies like LoRa, Zigbee, and cellular networks (e.g., 4G/LTE), each with distinct use cases and requirements, resulting in different privacy implications [58, 65]. To address this diversity, we envision Canella as not just a solution but as a catalyst for the broader IoT development community, encouraging the creation and integration of privacy components tailored to various IoT architectures.

5.2 Design and Implementation of Privacy-Preserving Components of Canella

We begin by presenting an overview of selected PbD guidelines from Perera's framework [84]. Subsequently, we demonstrate how Canella's privacy-preserving components have been developed to align with these PbD guidelines. Each privacy block and node within Canella corresponds to a specific PbD guideline, adhering to the principles of the CPLF [11]. The design and implementation of these privacy-preserving components were informed by proposed solutions from focus groups, as depicted in Figure 11 in Appendix B. Compliance details are in Appendix C.2.

5.2.1 ***Reduce Location Granularity:*** Granularity in data refers to its level of detail, with high granularity being more detailed and low granularity offering a summarized view [84].

Design and implementation: The focus groups (G1, G13, G14, and G19) suggested solutions with some presented in Figure 11 in Appendix B, that informed the development of the 'Reduce Location Granularity' guideline. In Blockly@rduino, we designed a custom block as a left output value block that generates a single value for transmission to other blocks. This block features a drop-down field for GPS coordinate options (e.g., postcode, city name, country name) and a text input for the required Google Maps API key. It performs reverse geocoding, converting latitude and longitude received by a GPS sensor into a human-readable address. This allows developers to select suitable GPS coordinates that meet their IoT app's needs while addressing privacy concerns. A corresponding node in Node-RED includes an edit dialog for selecting location granularity, an API key input, and custom node labeling. With a single input and output connection, developers can manage sensor location data while maintaining reduced granularity. Both the block and node are color-coded orange for consistency.

5.2.2 ***Minimise Raw Data Intake:*** IoT applications should minimize raw data collection into secondary context data, which is recommended for IoT platforms to avoid privacy violations [85].

Design and implementation: The suggested solutions from the focus groups (G15, G16, G17, and G20), with some depicted in Figure 11 in Appendix B, inspired the development of the 'Minimize Raw Data Intake' guideline. We designed its block in Blockly@rduino with a left output, which reduces raw data by calculating the average of sensor data over a specified time frame. Users can choose data types (e.g., heart rate, speed) and time units (seconds, minutes, hours) using a drop-down field and input the relevant time value. Figure 12 in Appendix C.1 illustrates the block returning the average sensor data at fixed 5-second intervals. In Node-RED, the corresponding "Minimize Raw Data Intake" node averages sensor data values as it receives data after a user completes a workout. Its edit dialog includes a data type drop-down and a label text input. The node, with a turquoise color, maintains consistency with the block's design (Figure 12 in Appendix C.1).

5.2.3 ***Category-based Aggregation:*** This guideline suggests reducing the detail in raw data [33]. For example, instead of exact values (e.g., 160 cal), calories burned can be represented as a range (e.g., 150–200 cal) [84].

Design and implementation: The focus groups suggested solutions (G19, G20, and G23), with some shown in Figure 11 in Appendix B, inspired the development of the "Category-based Aggregation" guideline. In Blockly@rduino, we designed a custom block as a left-output value block with a drop-down menu for data type selection and specific category options. For instance, for heart rate data, options include "status" (low, normal, high), with values below 60 as low, 60–100 as normal, and above 100 as high [5], and "range" (60–80 BPM). Similarly, in Node-RED, we designed a node that mirrors the Blockly@rduino block's functionality. It features a text input for node labeling, a drop-down menu for data type selection, and dynamic category options, allowing developers to select the appropriate category for their specific use case. This node maintains consistency with the block by sharing the same pink color (Figure 12 in Appendix C.1).

5.2.4 ***Minimise Data Storage:*** This guideline recommends minimizing stored data in an IoT application [99], including removing unnecessary information like raw data once secondary contexts are obtained [84].

Design and implementation: The focus groups' suggested solutions (G15, G19, G21, and G23), with some presented in Figure 11 in Appendix B inspired the development of the "Minimise Data Storage" guideline. In Blockly@rduino, we designed a custom value block with top and bottom connections and a drop-down menu for tasks like deleting or updating primary data with secondary data. This block is useful for updating stored data after deriving secondary contexts, particularly when adding privacy-preserving components to the IoT application's data flow, using commands like EEPROM.write and EEPROM.update. Figure 12 in Appendix C.1 illustrates how developers can use this block to delete or update stored data. In Node-RED, however, we designed its node to serve a distinct purpose by simply storing data in a file, maintaining a constant reminder for developers to prioritize data minimization. It includes a single input connection for storing received data and is color-coded in the same green shade as the Blockly@rduino counterpart for visual consistency (Figure 12 in Appendix C.1).

5.3 Designing Canella to Encourage the Use of Personal Data While Preserving Privacy

In this section, we outline Canella's key features, highlighting their alignment with the findings of the focus group study, established best practices, and adherence to privacy laws. Our focus is on the data minimization principle, emphasizing the importance of only collecting personal data that is relevant, adequate, and necessary for the intended purpose [11]. It is imperative for developers to articulate the purpose of collecting and processing personal data clearly, ensuring precision, explicitness, and legality. For compliance and fostering trust with data subjects, developers need to define the purpose of data practices: ***collection***, ***storage***, and ***sharing***.

5.3.1 Canella Supports Developers with Privacy-Preserving Components. Privacy-preserving components are the core feature of Canella, helping developers overcome the challenges of privacy integration and legal compliance during development (referenced in Sections 2.2 and 2.3). Existing guidelines often neglect technical requirements, creating a gap between legal aspects and developers' practical needs [11, 66, 101]. To tackle this, Canella bridges this gap by offering privacy-preserving components aligned with the CPLF, correlating privacy guidelines with the principles of the CPLF [11]. By doing so, Canella educates developers on best practices and enhances their awareness of privacy techniques, ensuring effective privacy measures and compliance with laws. Considering the implementation and maintenance costs associated with privacy techniques, a significant concern for developers discussed in prior research [11], Canella addresses this by abstracting low-level details from developers. This reduces both implementation and maintenance efforts, which is crucial for enhancing developer acceptance [64].

5.3.2 Canella Recommends Alternative Values for Users' Data. Previous studies examining developers' attitudes towards privacy revealed their limited awareness of viable, less invasive alternatives (discussed in Section 2.3). Tahaei [104] emphasized the technical complexity arising from a lack of understanding of privacy-preserving techniques for mitigating privacy risks. To address this issue, Canella's privacy-preserving components offer various options for safeguarding users' data, enabling developers to protect personal data while aligning with the goals of IoT applications, as recommended by nine focused groups (G1, G3, G11, G13, G14, G19, G20, G22, and G25). These components enable developers to select data values through block configuration in Blockly@rduino and node's edit dialog in Node-RED, as illustrated in Figure 12 in Appendix C.1. For instance, as detailed in Section 5.2.1, the Reduce Location Granularity block and node offer various levels of location granularity, prompting consideration of application implications and raising awareness of the adoption of lower-risk options for personal data. As a result, developers can make informed design decisions that prioritize privacy.

5.3.3 Canella offers Explanations of Privacy-preserving Components and their Compliance with Privacy Laws. To enhance developers' privacy knowledge (discussed in Section 2.1), we utilized tooltips in Blockly@rduino and the help feature in Node-RED. These features offer detailed information about each privacy-preserving component, its intended functionality, and compliance with privacy laws (Figure 12 in Appendix C.1). This empowers developers to better understand privacy techniques and their applicability to IoT application data flow, so they can make informed decisions on personal data protection. For instance, hovering over a block displays a pop-up box explaining its purpose and usage. This interactive feature increases developers' awareness of privacy techniques and facilitates integration. Additionally, the compliance information in the box specifies adherence to CPLF principles, building user trust, and enhancing adoption. In Node-RED, we utilized the data-help-name property that guides users on a node's purpose, usage, and compliance, educating developers and promoting a deeper understanding of their apps' data practices for informed decision-making.

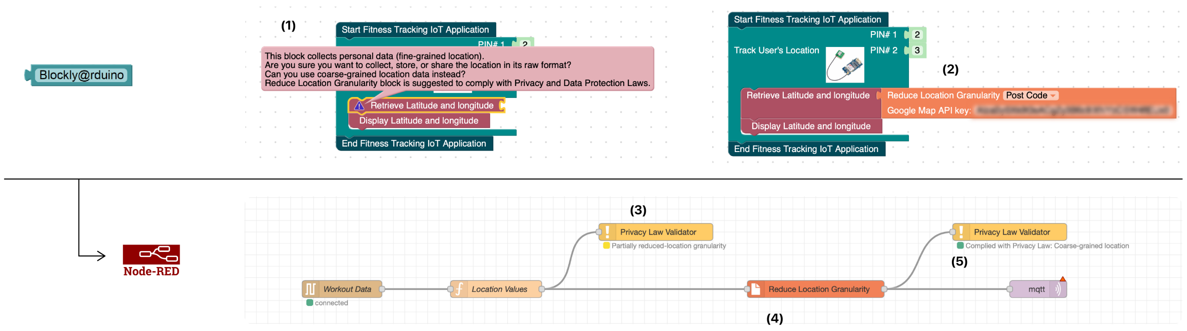
5.3.4 Canella Offers Real-Time Feedback on Potential Privacy Concerns. Canella offers developers real-time warnings during programming, notifying them of any potential privacy concerns pertaining to the processing of personal data in compliance with the CPLF, including the GDPR [79], PIPEDA [76], CCPA [61], Australian Privacy Act [15], and New Zealand Privacy Act 2020 [75]. These laws define the processing of personal data as a broad range of activities, including the collection, storage, use, disclosure, and deletion of personal data. The warnings focus on data minimization, urging developers to collect only necessary data for the intended purpose. Developers must be aware of the specific category of personal data processed, which the CPLF regulations define as any data relating to an identified or identifiable individual. The GDPR introduces safeguards for "special categories" of personal data (Article 9) [79]. The Australian Privacy Act [73] and PIPEDA (Principle 4.3.4) [77] require extra safeguards for sensitive information. The New Zealand Privacy Act 2020 emphasizes sensitivity considerations. While the CCPA [61] does not categorize data, its amended version, the CPRA, introduces "sensitive personal information." Therefore, in the warning messages, we included information about data categories (refer to Figure 15 in Appendix D.3), such as considering location data as personal data under all CPLF regulations and heart rate data as sensitive data. This educates developers on different data categories, raises privacy awareness, and encourages consideration of the specific privacy implications associated with each data type.

To assist developers in addressing privacy concerns during data processing, warning messages are integrated within blocks collecting personal or sensitive data. This approach is supported by prior research advocating timely privacy reminders [62] and suggestions from 11 focused groups (G2, G3, G8, G9, G10, G11, G13, G19, G21, G22, and G24). These messages promote privacy compliance at various stages, guiding the evaluation of data collection, storage, and sharing while adhering to IoT application requirements. Furthermore, they prompt developers to incorporate suitable privacy-preserving components into IoT data flows, as recommended by six focus groups (G2, G8, G10, G11, G13, and G19), encouraging the assessment of potential impacts and adoption of less risky alternatives for managing personal data. This proactive approach helps alleviate the cognitive load. Additionally, the warning messages disappear when developers connect the appropriate privacy component to data-collecting blocks, providing interactive guidance and facilitating a compliant development environment.

5.3.5 Canella Recognizes Personal Data Practices to Facilitate Review. Canella features a Privacy Law Validator, verifying data practices when sending data from Blockly@rduino to Node-RED. It checks the compliance status of the received data, displaying three indicators: a red circle for privacy issues, yellow for partial compliance, and green for full compliance. We implemented this node using regular expressions to ensure data format adherence to privacy laws. Developers connect the node to specific received data to identify and address privacy concerns. For instance, the location status can be classified as (a) "Privacy issues: fine-grained location" for latitude and longitude; (b) "Partial compliance" for postcode or city name, indicating the possibility for added privacy layers; and (c) "Complied with privacy and data protection laws" for country format, where additional privacy measures are not necessary (Figure 4. However, once data is processed according to necessity and aligned with the user's intended purpose, privacy compliance is achieved, regardless of the Privacy Law Validator's status.

The Privacy Law Validator encourages developers to prioritize privacy in IoT architecture, informs about data practices, and ensures protection from edge to cloud. It addresses concerns highlighted in Section 2.1, balancing usability and privacy considerations [8, 17]. Following Li et al.'s recommendation [62], the Validator provides timely reminders for privacy-related tasks. By offering real-time feedback on data compliance, the Validator facilitates collaboration between developers working in different environments, such as Blockly and Node-RED. For instance, when a developer using Blockly@rduino sends data to Node-RED, the Validator informs the Node-RED developers about the privacy status of the data. This feedback helps Node-RED developers understand the data practices implemented by their colleagues, promoting better coordination and informed decision-making. Consequently, privacy concerns are consistently addressed throughout the development process.

In summary, we encourage developers to prioritize privacy by adhering to the data minimization principle, which is interconnected with other privacy principles. Canella encourages and guides developers to consider the purpose of data processing by showing real-time feedback on potential privacy issues, striking a balance between application requirements and user privacy. Developers should consider the responsible entity for data management, particularly third-party hosting providers, to anticipate and address potential risks like unauthorized access to personally identifiable information (PII) from sensors. Understanding data access and its specific purposes is vital for developers to effectively manage privacy risks.


Blockly@rduino
(1)
Start Fitness Tracking IoT Application
Retrieve Latitude and longitude
Display Latitude and longitude
End Fitness Tracking IoT Application
(2)
Start Fitness Tracking IoT Application
Track User's Location
PIN# 1 2
PIN# 2 3
Retrieve Latitude and longitude
Reduce Location Granularity Post Code
Google Map API key: [redacted]
Display Latitude and longitude
End Fitness Tracking IoT Application
Node-RED
(3)
Privacy Law Validator
Partially reduced-location granularity
Privacy Law Validator
Compiled with Privacy Law: Coarse-grained location
(5)
Workout Data
connected
Location Values
Reduce Location Granularity
mqtt
(4)

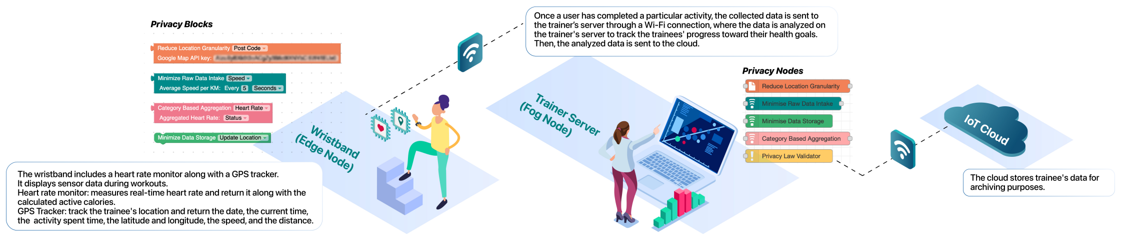
This block collects personal data (fine-grained location).
Are you sure you want to collect, store, or share the location in its raw format?
Can you use coarse-grained location data instead?
Reduce Location Granularity block is suggested to comply with Privacy and Data Protection Laws.

Fig. 4. Integrating the Reduce Location Granularity block and node into the flow of location data in the fitness tracking IoT application involves the following steps: **(1)** A warning message informs developers about a potential privacy issue. **(2)** The Reduce Location Granularity block is integrated to convert latitude and longitude into a postcode form before sending the location data to Node-Red. **(3)** The Privacy Law Validator checks the compliance status of the received data. **(4)** The Reduce Location Granularity node is used to convert the location from a postcode form to a city or country name form before sending the data to the cloud. **(5)** As a result, the user's data is now protected in accordance with privacy laws.

5.4 Prototyping: Design and Implementation of Canella

In this section, we present a hypothetical scenario to demonstrate how software developers can integrate privacy-preserving components into the data flow of an IoT application during the development process. We also provide an overview of the design and implementation of the Fitness Tracking IoT application within Blockly@rduino and Node-RED and the integration of these EUD environments. Further details on the prototyping of Canella are available in Appendix D.2. For detailed insights into the Canella tool, the source code and documentation are available on GitLab at the Canella Repository.

5.4.1 Canella IoT Use Case Scenario. **Fitness Tracking IoT Application:** Figure 5 demonstrates the system's components: a wristband (with GPS and a heart rate monitor), a trainer server, and a cloud. The trainer center manages the wristband and enables trainers to monitor trainees' activities. Additionally, the trainer server registers trainees for activity monitoring. The cloud is managed by a third party. In this scenario, a wristband tracks a user's activity, capturing real-time data such as heart rate, calories burned, distance covered, and location. Managing sensitive personal data requires adherence to privacy regulations. Monitoring devices, like wristband sensors and smart monitoring objects, enable observation of the user, heart rate tracking, and location monitoring [45]. Figure 13 in Appendix D.1 shows the Business Process Model Notation (BPMN) diagram that describes the system's process workflow. It also describes the components of the fitness-tracking IoT application.


Privacy Blocks
Reduce Location Granularity Post Code
Google Map API key: [redacted]
Minimize Raw Data Intake Speed
Average Speed per KM: Every 5 Seconds
Category Based Aggregation Heart Rate
Aggregated Heart Rate: Status
Minimize Data Storage Update Location
Wristband (Edge Node)
Trainer Server (Fog Node)
Privacy Nodes
Reduce Location Granularity
Minimise Raw Data Intake
Minimise Data Storage
Category Based Aggregation
Privacy Law Validator
IoT Cloud

Once a user has completed a particular activity, the collected data is sent to the trainer's server through a Wi-Fi connection, where the data is analyzed on the trainer's server to track the trainees' progress toward their health goals. Then, the analyzed data is sent to the cloud.

The wristband includes a heart rate monitor along with a GPS tracker.
It displays sensor data during workouts.
Heart rate monitor: measures real-time heart rate and return it along with the calculated active calories.
GPS Tracker: track the trainee's location and return the date, the current time, the activity spent time, the latitude and longitude, the speed, and the distance.

The cloud stores trainee's data for archiving purposes.

Fig. 5. Fitness Tracking IoT Application: Canella Use Case Scenario

5.4.2 *Design and Implementation of Canella's IoT Use Case Scenario.*

- **Blockly@rduino.** As detailed in Section 5.1, Blockly@rduino is the edge of the IoT application in Canella. For our fitness-tracking IoT application, we integrated two sensors: a GPS (Air530) and a finger-clip heart rate sensor into the wristband, as shown in Figure 5. All necessary building blocks for this application are accessible in Blockly@rduino's toolbox under "Fitness Tracking IoT Application." These blocks follow guidelines outlined on the Blockly website, are designed using JavaScript, and employ a natural language style for user-friendliness [80] (refer to Figure 14 in Appendix D.3).
- **Node-RED.** As the Node-RED is the fog of the IoT application in Canella as explained in Section 5.1, we have listed all the privacy-preserving components explained in Section 5.2 under "Privacy Nodes".
- **Integrated IoT Development Ecosystem.** To build a privacy-aware ecosystem, we have integrated two EUDs: Blockly@rduino and Node-RED. This integration empowers developers to effectively address privacy concerns for each unique IoT component, considering its specific characteristics. By combining the capabilities of these EUDs, we enable the creation of an end-to-end privacy-preserving application.

5.4.3 *Demonstration: Applying Canella's Privacy-Preserving Components to Fitness Tracking IoT Application.* Figure 5 illustrates the intricate data journey within the Fitness Tracking IoT application. Unlike traditional centralized systems, where data collection, processing, and storage typically occur within a single or a few central locations, IoT networks gather diverse data from a multitude of sensors embedded in devices distributed across various nodes [86]. This decentralized data collection from numerous sensor-equipped devices is a key characteristic that distinguishes IoT systems from traditional systems. This application manages the collection, storage, and transmission of personal and sensitive information, such as users' locations and heart rates. Such data could potentially reveal a user's identity and health status, thus presenting significant privacy challenges.

Implementing privacy measures throughout the IoT architecture is crucial for user protection and legal compliance. Our paper highlights Canella's pivotal role in addressing these challenges by seamlessly integrating privacy features into the app, with a specific focus on data minimization as discussed in Section 5.3. We illustrate Canella's effectiveness through a scenario featuring trainer Sara and trainee Amy (see Appendix G), emphasizing high-level functional requirements such as consent for specific data usage, automatic data storage for a month, and secure data transfer, aligning with relevant principles [12]. We detail how Canella facilitates the integration of privacy-preserving components within the IoT application's data flow, including real-time warning messages, seamless interaction between Blockly@rduino and Node-RED, and the role of the Privacy Law Validator.

Data Collection— To develop the Fitness Tracking IoT application, developers need to retrieve and display various data, including date, time, activity duration, location, heart rate, and active calories. When developers add certain blocks, a warning message alerts them to privacy concerns and suggests relevant privacy-preserving components (Figure 15 in Appendix D.3). In our use case, Amy wants to visualize all her workout data while exercising (purpose limitation). Therefore, developers must retrieve and display all this information.

Data Storage in the Wristband— Amy is not interested in viewing historical data or the timing of her activities (purpose limitation). Following the Minimize Data Storage guideline [84], developers should avoid storing non-essential data on the wristband, ensuring compliance with the Data Minimization principle. The *Minimize Data Storage* component can erase the date and time data upon workout completion (see Figure 14 in Appendix D.3). Amy's preference for accessing her workout history for other data means developers need to store this data on the wristband (purpose limitation). Given the short data retention period and direct user access, additional privacy measures are not required.

Data Sharing with the Trainer Server— Sharing workout data with the trainer's server requires caution, particularly regarding third-party services [62]. Trainer Sara does not need detailed speed values (purpose limitation), suggesting the use of privacy blocks like *Minimize Raw Data Intake* or *Category-Based Aggregation*. Heart rate values, sensitive under CPLF regulations, require additional privacy measures. Incorporating blocks like *Minimize Raw Data Intake* or *Category-Based Aggregation* helps mitigate this (Figure 14 in Appendix D.3). Location data, also sensitive, can be protected using the *Reduce Location Granularity* block to share appropriate coordinates like postcodes (see Figure 14 in Appendix D.3). These measures prevent privacy violations and ensure compliance. Regarding activity time, distance, and calorie data, which are crucial for fitness tracking and are accumulative (single value), there is no immediate need for extra privacy measures on the wristband before sharing them with the trainer. As the trainer has no interest in monitoring the specific dates and timing of Amy's activities (purpose limitation), developers should avoid sharing this information with the trainer.

Data Storage in the Trainer Server— To analyze and track user workout data, developers must store it on the trainer server, assuming deletion after a month. Since Sara does not specify storage requirements and privacy measures have been applied before the data was sent to the trainer server, it can be stored in received format.

Data Sharing with the Cloud— Sharing data with the cloud for archival purposes involves third-party access, raising privacy concerns [10]. To mitigate risks, developers should prioritize privacy on the trainer server before cloud sharing. Using the *Privacy Law Validator* provides insights into data practices, aiding informed decisions and ensuring compliance with data minimization principles. Figure 14 in Appendix D.3 highlights a privacy issue due to the absence of privacy-preserving measures at the data's edge. Sharing raw data is unnecessary; thus, privacy measures should be applied before transmission. The *Category-Based Aggregation* node reduces data granularity, ensuring compliance. For location data, using the *Reduce Location Granularity* node to opt for a city or country format enhances privacy. The *Privacy Law Validator* confirms compliance for speed and heart rate values, making them ready for cloud storage, as seen in Figure 14 in Appendix D.3.

In summary, while there is no universal approach for applying privacy to IoT data flow, Canella aids developers in integrating privacy-preserving components. It offers options for data values and helps implement privacy requirements. Developers must consider their application's needs and user preferences when making decisions.

5.5 Pilot Study

During Canella's development, we conducted a pilot study involving three developers with varied expertise in Blockly@rduino and Node-RED, as well as different levels of privacy understanding. One developer is an industry professional, works as a freelancer, and is pursuing a PhD in related fields. This developer was selected based on their extensive practical experience and industry insights. The other two participants are PhD students specializing in IoT and privacy research. Their valuable suggestions influenced the design of privacy-preserving components, block shapes, and the evaluation process, all of which were integrated into the initial prototype. Additionally, based on the received feedback, we refined the study evaluation procedure by adjusting elements such as task timing and instructions to enhance participant clarity. These adjustments will be further detailed in the study process for evaluating Canella in Section 6.3.

6 Evaluation

In this study, we conducted an evaluation using methods inspired by previous research, particularly Coconut [62]. Our evaluation involved a case-based technique [82] and employed a between-subject design approach [62].

6.1 Objective

The aim of this study is to evaluate the usability and effectiveness of Canella and gain insights into how software developers handle users' personal data with and without Canella. In addition, we sought to understand developers' perceptions of Canella's features, their ease of use, and their perceived usefulness. Our evaluation includes both qualitative discussions to explore participants' thoughts and ideas as well as quantitative data analysis. To achieve this, we address the following research questions, drawing inspiration from Coconut [62] and Honeysuckle [64]:

- RQ1: What is the impact of Canella on developers' time performance compared to development without it?

- RQ2: What is the impact of Canella on developers' cognitive load when integrating privacy measures, in contrast to not using Canella?
- RQ3: How do developers perceive the usability and the usefulness of Canella in facilitating the integration of privacy into their IoT applications?
- RQ4: To what extent does Canella contribute to the creation of more privacy-preserving applications by software developers compared to not using Canella?
- RQ5: How does Canella impact developers' integration of privacy into IoT applications, including awareness, practices, attitudes, challenges, and perceptions, in comparison to developers who do not use Canella?

6.2 Participants

Invitations were sent via the university mailing list and Twitter, resulting in 18 participants, including developers and students experienced in IoT or mobile app creation with sensors. In software engineering studies, students are commonly used as participants rather than industry software engineers [36, 51, 89]. The participants' count aligns with a previous similar study [62] and our sample size of 18 participants aligns with the outcome of a statistical power analysis [2] [56], suggesting a minimum sample size of 16.7 participants.

After obtaining ethical approval, participants were contacted via email to schedule their participation. They received an information sheet, a consent form, and a link to a pre-study survey for demographic data collection, which they completed electronically. Most participants (three male, five female) fell in the 25–34 age range, with varied software engineering experience: three over 10 years, three 5–10 years, four 3–5 years, five 1–3 years, and three less than a year. In IoT app development, four created over five apps, four developed two, and six had experience with one. Regarding privacy training, seven had none, while others had formal training, including two with 3 years and two with 4 years of privacy research experience. Five held master's degrees in cybersecurity considered official privacy training based on Coconut's study [62]. Familiarity with Blockly and Node-RED also varied among participants. Detailed demographic information is available in Appendix A, Table 1.

We employed a between-subjects design, consisting of two groups: a control group that did not use Canella and an experimental group that utilized Canella. To address potential disparities in factors such as software engineering experience, IoT app development, and familiarity with Blockly and Node-RED, we used randomization techniques [59]. This ensured an unbiased allocation of participants, reducing potential bias related to their backgrounds, IoT experience, and tool familiarity, thereby preserving the integrity of the study.

6.3 Study Procedure

We conducted our evaluation through in-lab studies, providing participants with macOS, a screen window, necessary hardware components, and installed software tools (Blockly@rduino, Node-RED, and Arduino IDE). Participants were assigned three programming tasks: a warm-up task, a primary task, and a final task, aimed at creating a fitness-tracking IoT application inspired by popular fitness tracker applications. The task design considered three key factors: the ability to complete each task within a timeframe (as suggested by prior research studies [62, 64]), coverage of various aspects of the data minimization principle of the CPLF, and real-world applicability. Participants received a £40 voucher as compensation.

The study employed a between-subjects design, with both groups completing a warm-up task without utilizing Canella's feature. The experimental group utilized Canella for the primary and final tasks, while the control group did not. Participants in both groups received instruction sheets detailing the study's purpose, the architecture of Canella (Figure 5.1) for the experimental group, and the architecture of Canella without privacy-preserving components for the control group. The sheets also included a fitness tracking use case (Section 5.4.1), the specific scenario of the trainer Sara and trainee Amy (Appendix G), and illustrations of the application's interfaces.

[2]Statistical power analysis determines the required sample size to detect a significant effect with specified confidence levels.

Before the evaluation tasks, the experimenter outlined the study objectives. For participants unfamiliar with the tools, the experimenter provided a quick demonstration. The experimenter also explained the hardware setup. Participants were then guided through the use-case scenario. The study focused on three programming tasks:

Warm-up Task: The warm-up task introduced developers unfamiliar with Blockly@rduino to the programming environment, evaluating their skills in developing personal data IoT applications. This task aimed to build confidence and ensure smooth interaction during the study. Both groups started with this task, simulating part of the Fitness Tracking IoT application. Both groups had 10 minutes; they had to establish a connection with a GPS sensor and retrieve some information for display on the screen. The goal was to acquaint participants with the environment and assess both groups' performance under the control condition for comparison.

Primary Task: In the primary task, developers extended the fitness-tracking IoT wristband simulation by retrieving additional information such as speed, distance, real-time heart rate, and active calorie display. They were also tasked with storing workout data locally and sharing it with the trainer's server for analysis via Node-RED. Both groups had one hour for this task, and developers were explicitly guided to consider privacy throughout the development process as if creating an app for real users. The experimental group had specific instructions on incorporating privacy blocks into Blockly@rduino and adding privacy nodes to Node-RED. The control group was provided with various options for incorporating privacy, including modifying code blocks using Blockly's visual programming interface or creating custom privacy-preserving functions. We provided clear comments for each code line to facilitate understanding, although understanding might have been easier due to the alignment with specific blocks in the Blockly@rduino workspace.

Manually creating privacy measures makes it an unsuitable baseline for comparison due to time and energy requirements. To ensure a fair comparison between the control and experimental groups, two forms of support were provided. The primary task was conducted in two rounds: the first round involved developers considering privacy during application development, and their practices were documented. In the second round, privacy guidelines were provided (presented in Section 5.2), and adjustments to their initial decisions were observed. This support guided the control group toward data minimization and assessed how privacy guidelines influenced decision-making and data practices (Table 6 in Appendix 4). The second form of support involved the control group writing pseudocode, creating a more equitable basis for evaluation and allowing a comparison of proposed privacy-preserving measures by both groups, focusing on assessing developers' behavior in applying privacy.

Final Task: In the final task, developers had 20 minutes to visualize and store workout data on the trainer's server for progress monitoring. This data would be sent to the Thingsboard cloud platform for archival purposes. Privacy considerations were emphasized, and the control group had flexibility in incorporating privacy-preserving measures using various nodes or writing pseudocode if preferred. Contrary to the previous task, no additional rounds were conducted, assuming familiarity with the privacy guidelines provided in the second task.

Throughout each task, we employed the think-aloud strategy [38], prompting participants to vocalize their thoughts. This aimed to reveal their cognitive processes, understand their reasoning, and identify challenges. An experimenter closely observed participants, documenting their actions, search queries, and verbalized thoughts. Participants received support to clarify task requirements, assist with block and node connections, suggest available resources, and provide code instructions (for the control group). If participants struggled to complete a task within the designated time, hints for improvement were offered to both groups. The control group had the option to use web resources and receive general explanations about privacy guidelines if needed.

The study, lasting approximately 2 to 2.5 hours, focused on programming tasks, a survey, and interviews. Participants could take breaks during tasks. Screen recording was initiated before the tasks for later analysis. After task completion, participants filled out an exit survey with four sections. The first section covered factual questions about personal data practices. The second section used a NASA-TLX questionnaire to assess cognitive load. The experimental group had two extra survey sections for Canella feedback (see Appendix F.2.1) and the

Technology Acceptance Model Scale (TAMS) [35] questionnaire to evaluate Canella's perceived usefulness and usability. Participants could refer to their work when answering questions during the survey and interviews.

Finally, the experimenter conducted voice-recorded interviews with participants. The interviews included questions about privacy considerations during app development, factors influencing their application of privacy techniques in Blockly@rduino and Node-RED, perceptions of Canella's features (experimental group only), opinions about the privacy guidelines (control group only), and reasons behind specific data practice behaviors. The interviews also explored recommendations for additional Canella features (experimental group only).

7 Findings and Results

7.1 Quantitative Results

The quantitative findings of the between-subject study demonstrate that developers in the experimental group (using Canella) received better ratings compared to those in the control group. Below, we present the outcomes related to the first four research questions.

7.1.1 Time on Task (RQ1). Table 3 in Appendix I provides an overview of the time taken for the warm-up task, primary task, and final task in both control and experimental groups. All participants in both groups completed the warm-up task with a comparable success rate, taking an average of 6 and 7 minutes. The success rate was similar in both groups, showing balanced app development skills. The experimental group spent an average of 38 minutes on the primary task and 22 minutes on the final task, while the control group spent an average of 40 minutes on the primary task and 18 minutes on the final task. Notably, some participants in both groups adjusted their time allocation based on privacy implementation preferences, particularly prioritizing privacy measures in the cloud over the edge. The control group, assigned to write pseudocode instead of full code, demonstrated diverse approaches to data practices. Some participants postponed privacy measures to the cloud, sending raw data, while others refrained from data collection, compromising application requirements. In contrast, the experimental group invested additional time addressing privacy issues from warning messages, integrating privacy-protecting elements like the privacy law validator node, and learning their usage.

7.1.2 Cognitive Load (RQ2). We assessed the six dimensions of cognitive load using the NASA-TLX [49] on a rating scale ranging from 1 to 100, where lower scores indicate better performance. The Shapiro-Wilk test [7] was conducted to assess the normality of the data in both the experimental and control groups [3]. Consequently, it indicates that the data in both groups appears to be normally distributed. Given this observation, we applied an unpaired t-test (p-value = 0.0433 < 0.05, t = -2.3134), which revealed a statistically significant decrease in cognitive load facilitated by Canella. The effectiveness of Canella in alleviating cognitive load is illustrated in Figure 6.

7.1.3 Perceived Usefulness and Usability (RQ3). Developers may incur higher costs by integrating privacy-preserving components. We sought the descriptiveness and time consumption of Canella's features using a 7-point Likert scale. Figure 8 depicts the findings of the questions as a set of diverging stacked bars. Participants were satisfied with Canella and its features, not finding them disruptive or time-consuming. The survey revealed developers mostly found Canella's features very useful, as shown in Figure 7. Two participants rated the Node-RED help tab information on privacy nodes and legal compliance as "neutral." They did not utilize it as they had already learned about privacy-preserving components through the on-hover properties of blocks in Blockly@rduino. The complete questions of how developers perceive Canella can be found in Appendices F.2.1 and F.2.2.

Furthermore, we use the TAMS [35] to examine the perceived usability and ease of use of Canella, rating it on 7 Likert scales (1 = strongly disagree, 7 = strongly agree). The results of the TAM questions revealed that

[3]The p-values obtained were 0.04644 for the experimental group and 0.8583 for the control group. Based on a significance level of 0.05, the results indicate that there is not enough evidence to reject the null hypothesis of normality for either group.

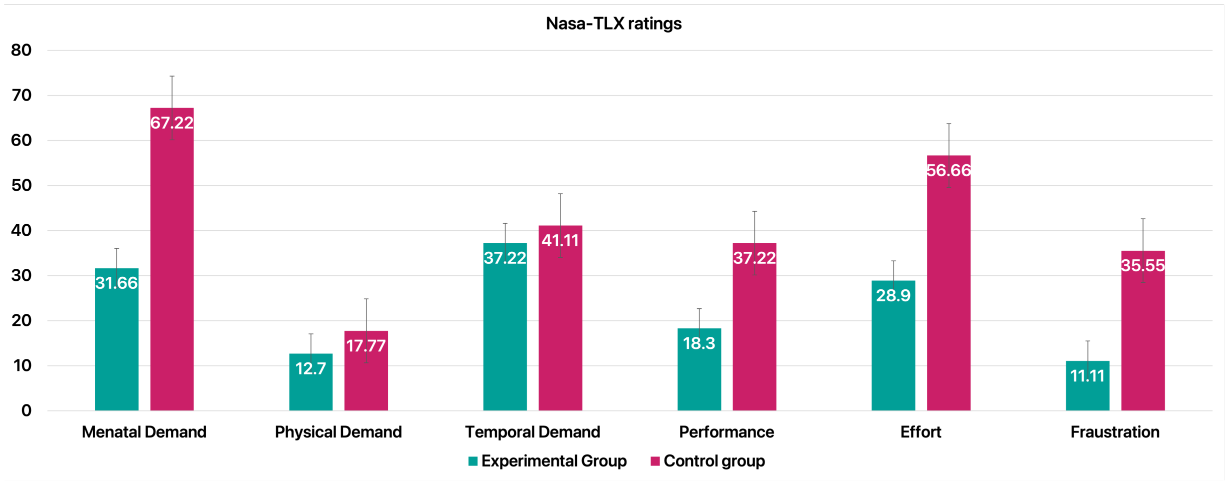


Fig. 6. NASA-TLX ratings for Canella, where lower ratings indicate better performance. Developers in the experimental group experienced a significant decrease in their overall cognitive load compared to those who did not use Canella when building the Fitness Tracking IoT application. The error bars in the figure represent the 95% confidence intervals.

developers considered Canella to be very useful and easy to use, as shown in Figure 9. In addition, the positive response from all participants indicated their keen interest in incorporating Canella into their future projects. Overall, they considered that Canella would improve task efficiency, performance, productivity, effectiveness, and ease of work. They also found it useful for their job and easy to handle privacy with Canella.

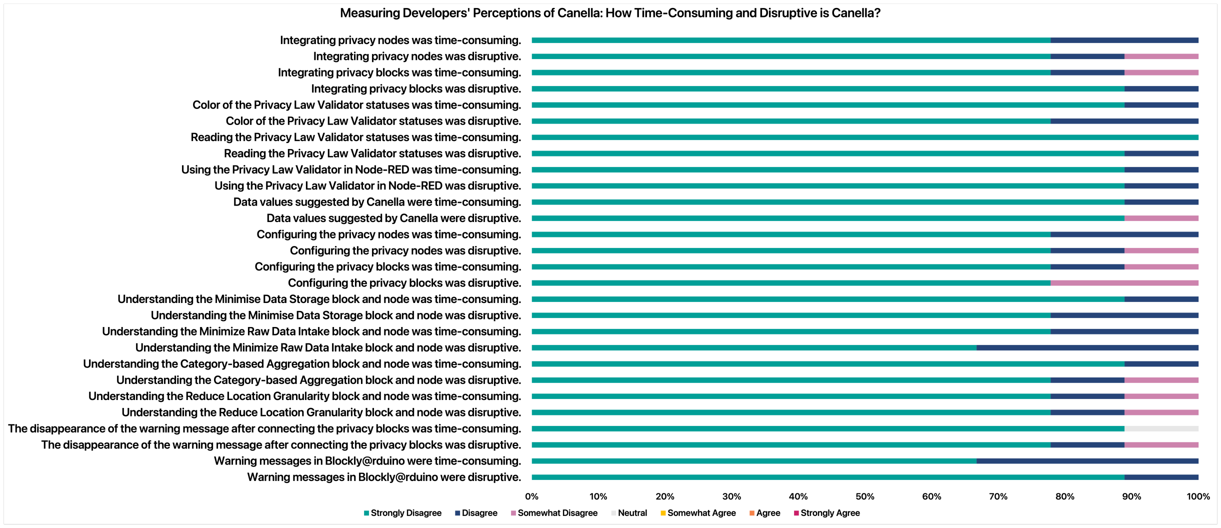


Fig. 7. The results of how much developers perceived Canella to be disruptive and time-consuming were evaluated using a 7-point Likert scale (1 = not disruptive or time-consuming, 7 = very disruptive and time-consuming).

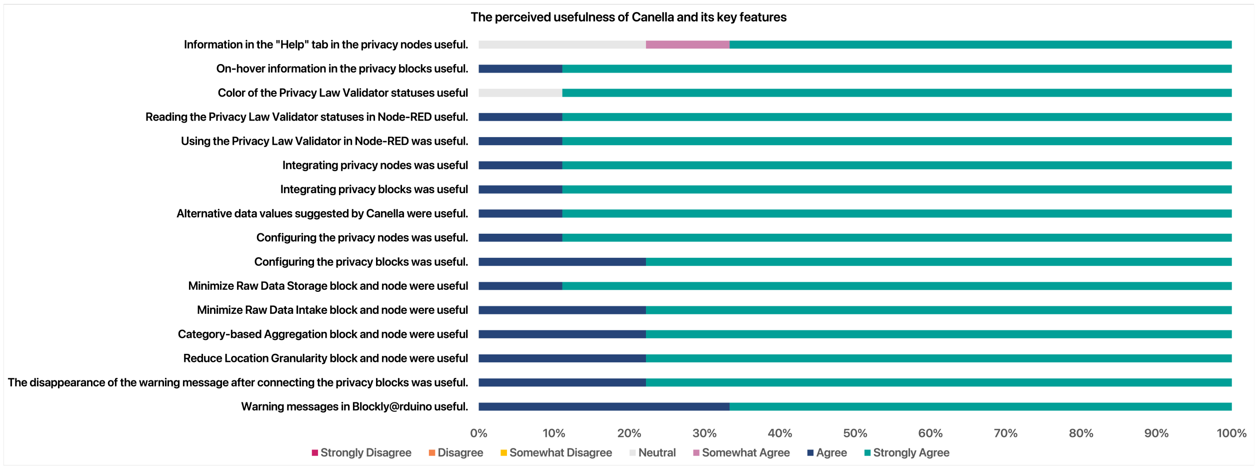


Fig. 8. The perceived usefulness of Canella and its key features were evaluated using a 7-point Likert scale (7 = very useful, 1 = not useful). The majority of the developers found Canella and its key features to be very useful.

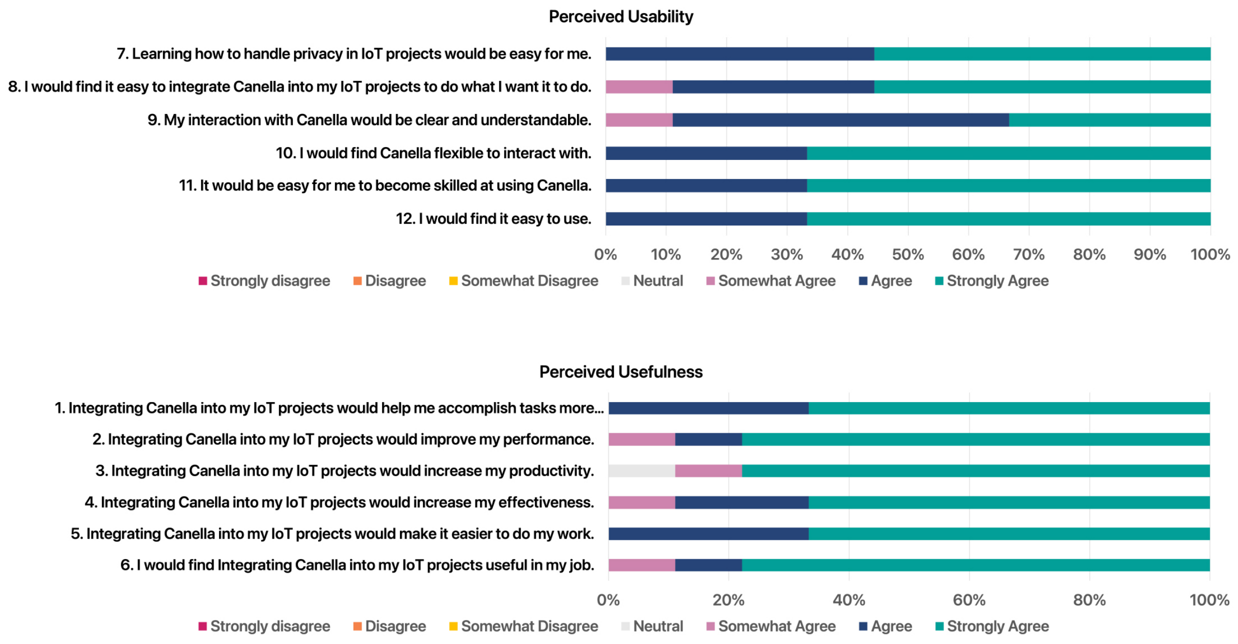
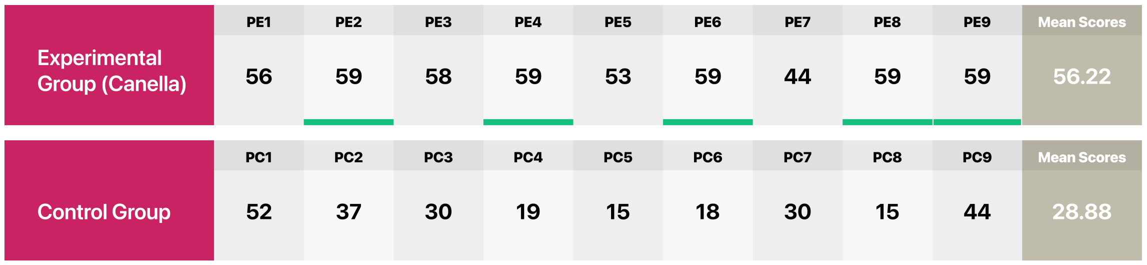


Fig. 9. Perceived Usefulness and Usability

7.1.4 Canella Can Help Developers Create More Privacy-Preserving Applications (RQ4). In general, comparing the two groups shows a privacy improvement when using Canella. A considerable proportion of developers who used Canella created more privacy-preserving applications by minimizing the collection, storage, and sharing of personal data with different nodes of an IoT application. Tables 4 and 5 in Appendix J present the developers' data handling behaviors in the Fitness Tracking IoT application, both in the control and experimental groups.

Table 6 in Appendix J outlines discrepancies in how control group developers managed their application's data between the first and second rounds. It reflects diverse behaviors and decisions with various underlying reasons. Developers processing data in its raw format despite privacy guidelines fell into categories such as finding the guidelines challenging (PC3, PC5, PC6, PC8, PC9), facing implementation issues (PC1, PC4, PC5, PC9), being unaware of the privacy concern (PC3, PC4, PC5, PC6, PC8), and aiming to collect excessive data for maximum user benefit (PC3, PC4). Further analysis follows in the subsequent section.

Result of Privacy Assessment— The Shapiro-Wilk test [7] showed non-normal distributions in both the experimental and control groups [4]. Consequently, the Mann-Whitney U test [71], less affected by non-normality, was chosen for comparing group means. The results revealed a significant difference (p-value = 0.0001 <0.05, t = 5.7368) in mean scores between developers in the experimental group and the control group. Developers using Canella achieved better scores than those who did not. Refer to Figures 10 for visual score representations. Further details on the criteria creation process and a scoring table are available in Appendix H.



	PE1	PE2	PE3	PE4	PE5	PE6	PE7	PE8	PE9	Mean Scores
Experimental Group (Canella)	56	59	58	59	53	59	44	59	59	56.22

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	Mean Scores
Control Group	52	37	30	19	15	18	30	15	44	28.88

Fig. 10. A comparison of privacy scores and mean scores between developers in the experimental and control groups when handling data in the Fitness Tracking IoT application. The total privacy score should amount to 59 points for each participant. Overall, developers in the experimental group obtained a higher mean score compared to developers in the control group.

7.2 Qualitative Feedback

We transcribed interviews, combined them with experimenter notes, and conducted a thematic analysis following the previous guidelines [22] to address our fifth research question (RQ5) on Canella's impact. Two main themes emerged: "Developers' Privacy Practices and Perceptions" and "Canella Impact and Comparative Feedback." The data provided qualitative insights into developers' attitudes, privacy practices, perceptions, and challenges in both the experimental and control groups, as well as the opinions of the experimental group about Canella.

7.2.1 Developers' Privacy Practices and Perceptions. We explore how developers address privacy concerns and their perspectives on privacy guidelines. Within this investigation, several key themes have emerged, offering valuable insights into how developers navigate the trade-offs between data quality, user privacy, and application requirements in the development of IoT applications.

Developers Privacy-Preserving Development Behaviors. During programming, we observed contrasting privacy-preserving practices between developers in the experimental group, using Canella, and those in the control group. The idea of data minimization motivated the majority of developers in the experimental group, who conscientiously limited data collection to what was required for the application. For instance, PE2 vocalized, *"The use case doesn't include emergency requirements for high heart rate values. So, I won't send heart rate in raw*

[4]The p-value for the experimental group was 0.0018 and for the control group was 0.2778

format; I'll share their range format." Additionally, PE3 expressed thoughtful consideration when sharing the total calories with the trainer: "*My wife specializes in sports diets, and I know that burning calories is affected by many factors [..] I believed that the trainer could make an informed decision based on the range of the total burned calories.*" In contrast, the control group exhibited a different behavior, going beyond the application's requirements. For instance, when asked about sharing activity time, not mandated by the application, PC4 reasoned that "*Some people have a meal plan. So they like to plan their meals and day, noting when they went and ate.*"

When we asked developers in the experimental group about their privacy considerations, we found a common theme in their responses, emphasizing decisions based on data purpose and balancing data quality with user privacy. For instance, PE4 prioritized privacy, stating, "*my priority was privacy, so I want the system to be privacy-friendly.*" While most experimental group participants (PE2, PE5, PE6, PE9) prioritize user privacy, considering factors like user preferences (PE2, PE5, PE6, PE9) and application requirements (PE2, PE7), a few (PE2, PE8) also acknowledged the importance of using raw data for fitness-related parameters but emphasized its applicability to cumulative data for informed decisions. In contrast, many control group participants (PC1, PC2, PC3, PC4, PC8) emphasized using raw data for fitness-related parameters to maintain accuracy and meet application goals. Our findings highlight the need for a broader community to prioritize user privacy in IoT development. Encouraging a mindset that aligns with privacy-preserving practices can significantly enhance the overall privacy landscape.

Developers' Data Handling Practices. We observed distinct data handling practices among developers in the experimental and control groups. In the experimental group, all developers shared coarse-grained locations with the trainer server, with some implementing additional privacy measures before transmitting data to the cloud. Conversely, fewer control group participants (PC1, PC2, and PC7) shared postcode locations, and one added additional privacy measures when sharing with the cloud. Some control group participants (PC3, PC5, and PC9) expressed uncertainty regarding the conversion of a fine-grained location to a coarse-grained format. For the heart rate, Canella users processed data before transmission, with one user converting it only when sharing with the cloud. After receiving guidelines, control group participants adjusted their data handling practices, although some (PC3, PC4, PC5, PC6, PC7, PC8) were unaware of the sensitivity of heart rate data. In terms of speed data, most participants in the experimental group shared average or range speed values, whereas raw speed data was predominant in the control group. Additionally, developers in both groups exhibited varying behaviors in handling activity time spent, distance, and calories, influenced by data format and application nature. Refer to Tables 4 and 5 in Appendix J for detailed insights into their respective practices, and Table 6 for comparison between control groups' practices in the first and second rounds. To understand the long-term impact of privacy tools, longitudinal community studies are necessary. These studies can help enhance privacy tools by examining developers' evolving practices and identifying challenges encountered during continuous usage.

Developers' Perceptions of Privacy Guidelines. Developers had varying perceptions of privacy guidelines. PC3 found them challenging to understand, reflecting a broader difficulty with these principles. PC6 encountered language barriers and terminology complexities, struggling to differentiate between "coarse-grained" and "fine-grained" aspects. PC9 admitted to a lack of background knowledge in privacy, making it difficult to interpret and effectively implement the guidelines. In contrast, PC1, PC4, and PC5 found the guidelines beneficial for enhancing understanding. Despite providing positive feedback, they still faced practical implementation challenges. Overall, most control group developers encountered difficulties with the guidelines due to language barriers, a lack of privacy knowledge, and implementation issues. These findings align with prior research, emphasizing the necessity for clearer guidelines and practical tools [47, 95, 99, 101, 110]. Future efforts could focus on collaborative endeavors to establish standardized privacy best practices applicable across diverse IoT applications.

7.2.2 Canella Impact and Comparative Feedback. We explore developers' perspectives on Canella and its impact on privacy integration in applications. Several key themes have emerged highlighting varied opinions and comparisons between experimental and control groups. These insights offer a thorough understanding of how developers handle privacy challenges in IoT development and Canella's impact on their practices and decisions.

Canella's Architecture. The experimental group values Canella's architecture, appreciating its recognition of distinct privacy requirements across IoT nodes. A privacy expert (PE2) expressed surprise: *"To be honest, I am surprised about the architecture of Canella, where it considers the edge, fog, and the cloud,"* emphasizing the need for tailored privacy measures at different IoT architecture levels: *"Canella opened my eyes to that."* Recognizing the diverse IoT development environments, future research could explore customizing privacy tools like Canella to cater to a wide range of platforms and frameworks for broad applicability within the community.

Usefulness. All experimental group participants praised Canella for its benefits in developing privacy-conscious applications. Notably, six of them (PE1, PE2, PE3, PE5, PE6, PE8) explicitly labeled Canella as a "useful tool." Without prompting, PE1 and PE9 expressed eagerness for Canella's public release, emphasizing its potential for personal and widespread use. PE9 specifically mentioned, *"I hope I really see this tool launch so I can use it, and it can be used by everyone."* PE1 echoed this sentiment, stating, *"I hope that I can use it in my projects. It's very nice and useful."* This highlights the limited support available for developers in creating privacy-aware IoT apps.

Canella's Impact on Privacy Education. Developers in the experimental group without a privacy background gained valuable insights into privacy through Canella. Initially, both groups faced confusion about "granularity," but the experimental group (PE1, PE3, PE5, PE6, PE9, PC1, PC3, PC7, PC8, PC9) overcame it with Canella's "Reduce Location Granularity Block," leading to informed decision-making about location data protection. In contrast, the control group (PC4, PC5, PC6, PC8) struggled to grasp the concept, even with a given guideline. Regarding heart rate data sensitivity, while all Canella users were aware, some control group participants were surprised, indicating a lack of awareness. This underscores Canella's role in educating developers about privacy, indicating an opportunity to invest in practical tools for enhancing developers' privacy understanding, potentially through educational programs and workshops for a privacy-aware community.

Privacy Awareness. Several participants (PE2, PE3, PE5, and PE9) highlighted Canella's role, its warning messages, and the privacy law validator in enhancing awareness of privacy concerns and different privacy levels within IoT architecture. PE3 noted, *"Canella has real potential because it can make developers more aware of privacy issues [..] it increases developers' awareness regarding privacy issues."* PE9, who lacked prior privacy knowledge, stated, *"With Canella, I now understand the risks associated with sharing such data [..] it helped me understand privacy as a developer without a privacy background by informing me about specific privacy issues."*.

Usability and learnability. Participants, including PE1-PE9 and PC5, provided insights on Canella's usability and learnability. Many, such as PE2, PE3, PE4, PE5, PE8, and PE9, found it easy to use. PE3 mentioned, "*The privacy blocks and privacy nodes are very user-friendly. Using Blockly and Node-RED is straightforward.*" PE8 added, "*Compared to traditional software, it's quite easy [..] just drag and drop once you know what you want.*" PE2 appreciated the visual icons in the blocks for understanding data collection. PC5 from the control group highlighted how Blockly helped visualize data usage, simplifying the process. Some participants (PE1, PE2, PE6, and PE7) noted that familiarity with tools like Blockly and Node-RED could affect the learning curve. PE4 and PE5, while acknowledging Canella's usability, mentioned complexity and confusion due to unfamiliarity with the tools. Despite initial challenges, participants believed that with practice, Canella could become more intuitive. Overall, they found Canella user-friendly and adaptable, particularly with some familiarity with the development tools involved. This feedback underscores the significance of user-centric design in IoT tool development. Future research could focus on advocating for user-friendly privacy tools that not only integrate privacy-preserving techniques into IoT applications but also other non-functional requirements such as safety.

Cognitive Load. Several participants (PE2, PE3, PE4, and PE9) found Canella effective in reducing cognitive load. PE4 easily identified privacy gaps, stating: *"It's straightforward to [pinpoint] privacy issues and find solutions."* PE2 appreciated visual cues from block shapes, aiding understanding of data flow and privacy implications. PE3 and PE4 highlighted Canella's guidance on privacy techniques and streamlining compliance. PE9 valued its consolidation of privacy laws, simplifying cross-country compliance efforts. Overall, Canella's ability to identify issues, provide visual cues, offer guidance, and ensure compliance alleviates the cognitive load on developers.

Productivity. PE3, PE7, and PE8 highlighted Canella's significant productivity enhancement. PE3 stated, "Canella increases productivity by reducing time spent on privacy considerations, accelerating the process." PE7, aligning Canella with PbD [25], said, *"Canella saves time by integrating privacy regulations [at the application's outset] based on user input rather than addressing them separately later."* This aligns with past research [82], emphasizing the time- and resource-saving benefits of early data protection considerations. In addition, PE8 commended the efficacy of the Privacy Law Validator, *"Uploading data to the cloud often requires manual checks for privacy compliance, but with this node, it's quite easy."*

Easier Implementation. Participants PE4, PE8, and PE9 in the experimental group found privacy implementation using Canella straightforward, with PE4 stating it was "very easy to code" and PE8 appreciating the drag-and-drop approach. PE9 highlighted that Canella simplifies privacy implementation for developers without a background in privacy. In contrast, some control group developers, like PC9 and PC1, faced challenges translating privacy considerations into code due to a lack of specific implementation knowledge. PC1 highlighted, "*The main difficulty was the coding*." PC1, PC3, and PC9 expressed a need for practical code examples or guidance.

Complience with the Privacy Laws. Participants (PE5, PE6, PE8, PE9) praised Canella for ensuring privacy compliance. PE6 felt more comfortable as a developer, knowing they were complying with the law. PE5 emphasized the tool's role in reducing the risk of privacy breaches and the associated consequences. PE9 valued Canella's simplification of privacy laws, providing clarity for implementation across countries. These observations highlight Canella's role in promoting compliance, mitigating risks, and boosting developer confidence.

Challenges in Privacy Decision-Making. While many participants praised Canella, some encountered challenges in making privacy-related decisions. PE1 mentioned the difficulty of deciding which user data to store, collect, or not collect, and PE8 highlighted the challenge of finding a balance between thorough personal data analysis and user reluctance to share certain data. Additionally, PE3 expressed concerns about the ambiguity in privacy laws and regulations, making decision-making uncertain. It is important to note that user consent, as per GDPR and other privacy regulations, does not grant developers unrestricted control over data handling. In the provided use-case scenario, certain aspects of user preferences are intentionally withheld to assess Canella's role in aligning developers with privacy requirements while meeting application needs. Introducing a dedicated 'viewpoint' could significantly facilitate decision-making for privacy compliance. Currently, Canella primarily focuses on guiding and encouraging privacy integration. However, it is worth noting that these participants who faced challenges like PE1, PE3, and PE5 successfully made informed decisions, as shown by their high privacy assessment scores (Figure 10). In contrast, control group developers faced challenges in integrating privacy-preserving applications, including balancing data quality and privacy, limited familiarity with privacy issues, a lack of knowledge about privacy-preserving techniques, and difficulties in implementing privacy measures. These findings highlight the importance of ongoing research within the community to address these challenges and develop practical solutions that support developers in navigating complex privacy landscapes.

Developers' Response to Warning Messages. During our investigation, developers sometimes overlooked warnings initially. Reasons for this behavior varied among participants. PE9 mentioned, *"If I were working independently, I would have noticed the warning easily. The problem arises because I'm within the experiment, adapting to the new environment, and referring to instruction sheets."* Another developer noted that they initially missed warnings due to the small size and colorful blocks, suggesting that uniform block colors could improve warning visibility. Participants' responses on the influence of warning messages differed. Some developers did not feel enforced by warnings. Others initially followed warnings but later took a more thoughtful approach. PE4 mentioned that while warnings assisted, they did not override personal choices. PE6 and PE9 viewed warnings as reminders, guiding without enforcing, as did PE7 and PE8, who made decisions based on personal thoughts, with warnings prompting analysis but not enforcing choices.

Privacy Law Validator Impact. Participants praised Canella's Privacy Law Validator. PE2, for example, lauded its value and status indication, stating, *"It is of great value. The status displayed by the Privacy Law Validator is fantastic."* PE3 found it useful for identifying and addressing privacy issues, describing it as *"a nice node that directly indicates issues."* PE4 simply labeled it as *"an amazing node."* PE8 praised the Validator for simplifying compliance with privacy regulations, especially when uploading data to the cloud. These confirmations underscore the Privacy Law Validator's critical role in facilitating developers' adherence to privacy laws and fostering more privacy-conscious IoT applications. Regarding its impact, participants' responses were consistent with their reactions to warning messages. However, PE5 and PE6 provided unique perspectives, highlighting the Validator's role in refining their decision-making. PE6 mentioned, *"It corrects my decision, especially when dealing with complex IoT data and privacy laws. When you have a piece of info, it can be unclear which law to apply. The Validator helps clarify my choices."* PE5 noted that *"the privacy law Validator does not enforce me, but it changed my thoughts."* PE3 suggested improving the Validator's efficiency by enabling it to process multiple data types simultaneously.

8 Discussion and Future Work

8.1 The Value of Canella

Our study reveals that incorporating privacy features with Canella is beneficial without causing major disruptions to developers. Four key factors illustrate how Canella enhances privacy. Firstly, it simplifies the development of privacy-aware IoT applications, bridging the gap between developers and legal requirements. This proactive approach aligns with the philosophy of [25] that privacy protection should be integral from the beginning, reducing the need for costly privacy-related modifications later on. Canella's Privacy Law Validator streamlines compliance, instilling confidence without legal expertise. Secondly, warning messages raise awareness of privacy issues, among developers, guiding proactive consideration of privacy implications. These messages aid in understanding how applications interact with personal data and facilitate the integration of privacy-preserving techniques, reducing cognitive load. This addresses developers' previous confusion about factors impacting user privacy [62]. Thirdly, developers gain insights into alternatives, such as "Reduce Location Granularity," gaining insights into various options for handling location data and empowering informed design choices. Lastly, Canella educates developers, particularly self-taught ones, about privacy practices, addressing the lack of formal training.

It is noteworthy that Canella currently might offer more benefits to developers in smaller teams with limited resources for addressing privacy requirements. Yet, its impact may be less significant in larger teams with external influences on decisions. Further research in corporate and collaborative settings could explore this aspect.

8.2 The Adaptability of Canella to Guarantee Adherence to Evolving Privacy Regulations.

We explore Canella's scalability in adapting to evolving privacy laws, exemplified by the CPRA as proof of ongoing regulatory shifts [102]. Our attention is on established global privacy laws such as GDPR [79], PIPEDA [76], CCPA [61], Australian Privacy Act [15], and New Zealand Privacy Act 2020 [75], forming the basis of Canella's compliance capabilities. Yet, Canella's flexibility extends its relevance to future laws. Its adaptable architecture, including a user-friendly toolbox suggested by participant PE6, underscores its proactive approach to compliance simplification. This ensures Canella remains a dynamic solution, evolving with legal demands.

8.3 Boosting Motivation for Developers to Enhance Privacy

Empowering software engineers with privacy best practices is vital [26], especially with increasing fines from data protection authorities [27]. Despite legal mandates, implementation gaps persist [14]. Enhancing EUD tools like Canella with privacy-preserving components can catalyze developers' prioritization of privacy and understanding of data risks. Our research confirms Canella's effectiveness in nudging developers to prioritize privacy and comply with laws. Previous studies [62, 82] support that offering tools and techniques aids developers in effectively integrating privacy considerations. Canella not only encourages privacy-conscious thinking but also enhances developers' awareness of privacy issues and personal data practices in applications. With growing interest in supporting developers to maintain privacy while meeting user needs [31], Canella stands out by bridging the gap between developers and the practical application of privacy laws, notably in IoT programming.

8.4 Limitation of the Evaluation Methodology

Recruiting participants for lab studies on programming processes is challenging due to various factors. Firstly, it is difficult to gather developers as their regular income often exceeds study compensation. Secondly, studies can be time-consuming, lasting several hours, which can limit their scale and risk misrepresenting real-world practices [8]. In our research, we engaged 18 developers for a study lasting 1.5 to 2.5 hours, consistent with previous studies [21, 60, 62]. It is essential to note that study duration can vary based on task complexity and participant skill level. Besides, it is important to note that most participants were postgraduate students with field experience but not seasoned professionals. Even though including students in software engineering studies is a recognized practice [51, 89], their potential lack of depth compared to professionals is acknowledged, which may raise concerns about the generalizability of our study findings. Furthermore, the relatively small sample size of participants may impact confidence in drawing study conclusions.

Despite our efforts to design tasks that mirror real-world scenarios and provide clear instructions for task navigation without the necessity of configuring hardware details, lab-based studies inherently have limitations. Participants may engage in less thorough planning and quicker prototyping due to time constraints. Moreover, emphasizing privacy as a study objective during recruitment and task instructions may render participants more privacy-conscious than usual. Ultimately, while initially designed for fitness, Canella may encounter complexities in extending privacy principles to diverse, unregulated domains. However, Canella holds potential for other domains with sensitive data, where we are committed to expanding its utility while safeguarding privacy.

8.5 Future Work

In the future, we aim to expand Canella's impact on privacy from three perspectives. First, we plan to broaden the tool's coverage of privacy aspects and associated concerns. Currently, we mainly focus on data minimization, which interrelates with principles like purpose and storage limitations. However, we intend to incorporate other fundamental privacy principles, like security and anonymity. Second, we aim to extend Canella's coverage to include the privacy and data protection laws of other countries. Our study showed the value of integrating privacy regulations, allowing developers to improve compliance with various laws. The final perspective involves enhancing Canella's role as an educational tool. Our research highlighted its potential to educate developers, bridging a gap in available tools for teaching PbD techniques, especially in university-level courses. Despite existing research in this field [28, 29, 98], there is currently no established avenue for developers to gain practical insights into integrating privacy-preserving measures into their app development process. To this end, we affirm our commitment to enhancing Canella's role as an educational tool. This entails expanding Canella to incorporate a wider range of the most commonly taught privacy-preserving techniques worldwide and introduce additional educational content such as tutorials, case studies, and interactive exercises. These improvements can enhance Canella's role as an educational platform, reaching a broader audience, including university-level courses, and improving their understanding of privacy techniques. Our focus going forward is on strengthening Canella's privacy education offerings, making it an essential resource for developers and students. This can address the current gap in practical privacy education, cementing Canella's position as a leader in IoT privacy and compliance. Building on these, we intend to further explore diverse IoT architectures such as LoRa, Zigbee, and cellular networks. This can enhance developers' understanding of privacy-preserving components and enable collaboration to create tailored privacy features for distinct IoT architectures.

9 Conclusion

In this paper, we present Canella, an integrated IoT development ecosystem designed to create privacy-aware IoT applications. Canella utilizes Blockly@rduino and Node-RED to guide developers in integrating privacy-preserving techniques aligned with CPLF regulations, notifying them about potential privacy issues. We evaluated Canella with 18 software developers using a within-subject design and found that Canella enables the development of more privacy-aware applications, a deeper understanding of personal data handling, and reduces cognitive load, ensuring compliance with privacy laws. Canella also helps developers find a balance between privacy and other considerations, offering developers a foundation for learning more about privacy. While adaptable to various applications, Canella primarily focuses on IoT applications and CPLF compliance. Though it does not guarantee the absence of privacy issues in IoT systems, it enhances developers' awareness among developers. Our research points to two promising directions for future work: enhancing Canella's privacy aspects, including user consent and transparency, and supporting additional privacy and data protection laws worldwide.

10 Acknowledgments

This work is partially supported by EPSRC PETRAS (EP/S035362/1). We also acknowledge the scholarship and support provided by Majmaah University. Lastly, the authors wish to thank the anonymous reviewers for their valuable feedback.

References

[1] 2013. Opinion of the European Data Protection Supervisor on the Joint Communication of the Commission and of the High Representative of the European Union for Foreign Affairs and Security Policy on a 'Cyber Security Strategy of the European Union: an Open, Safe. (2013). www.edps.europa.eu

[2] 2017. Why is IoT talent so hard to find? | CIO Dive. https://www.ciodive.com/news/why-is-iot-talent-so-hard-to-find/449576/

[3] 2023. Blockly. https://developers.google.com/blockly

[4] 2023. Blockly@rduino: Create Code with Blocks. https://create.arduino.cc/projecthub/libreduc/blockly-rduino-create-code-with-blocks-b6d3e4

[5] 2023. Calculators | Heart Online. https://www.heartonline.org.au/resources/calculators/target-heart-rate-calculator

[6] 2023. Node-RED. https://nodered.org/

[7] 2023. Shapiro-Wilks Normality Test. https://variation.com/wp-content/distribution_analyzer_help/hs141.htm

[8] 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

[9] Yaqoob Al-Slais. 2020. Privacy Engineering Methodologies: A survey. In *2020 International Conference on Innovation and Intelligence for Informatics, Computing and Technologies (3ICT)*. 1–6. https://doi.org/10.1109/3ICT51146.2020.9311949

[10] Nada Alhirabi, Stephanie Beaumont, Jose Tomas Llanos, Dulani Meedeniya, Omer Rana, and Charith Perera. 2023. PARROT: Interactive Privacy-Aware Internet of Things Application Design Tool. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 7, 1, Article 1 (mar 2023), 37 pages. https://doi.org/10.1145/3580880

[11] Atheer Aljeraisy, Masoud Barati, Omer Rana, and Charith Perera. 2021. Privacy Laws and Privacy by Design Schemes for the Internet of Things: A Developer's Perspective. *ACM Comput. Surv.* 54, 5, Article 102 (may 2021), 38 pages. https://doi.org/10.1145/3450965

[12] Atheer Aljeraisy, Masoud Barati, Omer Reana, and Charith Perera. 2020. *Exploring the Relationships Between Privacy by Design Schemes And Privacy Laws: A Comparative Analysis.* Technical Report June. Cardiff University. 40 pages.

[13] Atheer Aljeraisy, Omer Rana, and Charith Perera. 2023. Canella: Privacy-Aware End-to-End Integrated IoT Development Ecosystem. In *2023 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*. 279–281. https://doi.org/10.1109/PerComWorkshops56833.2023.10150254

[14] Sami Alkhatib, Jenny Waycott, George Buchanan, Marthie Grobler, and Shuo Wang. 2020. Privacy by Design in Aged Care Monitoring Devices? Well, Not Quite Yet!. In *ACM International Conference Proceeding Series*. 492–505. https://doi.org/10.1145/3441000.3441049

[15] Australian Government. 1988. Australian Privacy Principles — OAIC. https://www.oaic.gov.au/privacy/australian-privacy-principleshttps://www.oaic.gov.au/privacy/australian-privacy-principles/

[16] 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. In *Companion of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing* (Portland, Oregon, USA) *(CSCW '17 Companion)*. Association for Computing Machinery, New

York, NY, USA, 135–138. https://doi.org/10.1145/3022198.3026326

[17] Rebecca Balebako and Lorrie Cranor. 2014. Improving App Privacy: Nudging App Developers to Protect User Privacy. In *IEEE Security and Privacy*, Vol. 12. IEEE, 55–58. https://doi.org/10.1109/MSP.2014.70

[18] Rebecca Balebako, Abigail Marsh, Jialiu Lin, Jason Hong, and Lorrie Faith Cranor. 2014. The Privacy and Security Behaviors of Smartphone App Developers. *In Proceedings 2014 Workshop on Usable Security. Internet Society.* October (2014). https://doi.org/10.14722/usec.2014.23006

[19] 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. *Journal of Systems and Software* 149 (2019), 101–137. https://doi.org/10.1016/j.jss.2018.11.041

[20] Barbara Rita Barricelli, Daniela Fogli, and Angela Locoro. 2023. EUDability: A new construct at the intersection of End-User Development and Computational Thinking. *Journal of Systems and Software* 195 (2023), 111516. https://doi.org/10.1016/j.jss.2022.111516

[21] Joel Brandt, Mira Dontcheva, Marcos Weskamp, and Scott R. Klemmer. 2010. Example-Centric Programming: Integrating Web Search into the Development Environment. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Atlanta, Georgia, USA) *(CHI '10)*. Association for Computing Machinery, New York, NY, USA, 513–522. https://doi.org/10.1145/1753326.1753402

[22] Virginia Braun and Victoria Clarke. 2006. Using thematic analysis in psychology. *Qualitative Research in Psychology* 3, 2 (2006), 77–101. https://doi.org/10.1191/1478088706qp063oa

[23] Fei Bu, Nengmin Wang, Bin Jiang, and Huigang Liang. 2020. "Privacy by Design" implementation: Information system engineers' perspective. *International Journal of Information Management* 53 (2020), 102124. https://doi.org/10.1016/j.ijinfomgt.2020.102124

[24] Fred H Cate. 2006. The Failure of Fair Information Practice Principles. *Consumer Protection in the Age of the 'Information Economy'* (2006), 341–377.

[25] Ann Cavoukian. 2009. Privacy by design: The 7 foundational principles. *Information and privacy commissioner of Ontario, Canada* 5 (2009), 1–12. https://iapp.org/media/pdf/resource_center/pbd_implement_7found_principles.pdf

[26] Ann Cavoukian. 2012. Operationalizing Privacy by Design : A Guide to Implementing Strong Privacy Practices. December (2012), 1–72. https://gpsbydesigncentre.com/wp-content/uploads/2021/08/Doc-5-Operationalizing-pbd-guide.pdf

[27] CMS. 2022. GDPR Enforcement Tracker - list of GDPR fines. https://www.enforcementtracker.com/

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

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

[30] Data Protection Commissioners, Privacy, and Data Protection and Privacy Commissioners. 2010. Resolution on Privacy by Design. *Icdppc* (2010), 1–2.

[31] Luca Compagna, Paul Khoury, Alžběta Solarczyk Krausová, Fabio Massacci, and Nicola Zannone. 2009. How to integrate legal requirements into a requirements engineering methodology for the development of security and privacy patterns. *Artificial Intelligence and Law* 17 (03 2009), 1–30. https://doi.org/10.1007/s10506-008-9067-3

[32] 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.* https://doi.org/10.2824/38623 arXiv:1501.03726

[33] George Danezis, Josep Domingo-Ferrer, Marit Hansen, Jaap-Henk Hoepman, Daniel Métayer, Rodica Tirtea, and Stefan Schiffner. 2014. *Privacy and Data Protection by Design - from Policy to Engineering.* https://doi.org/10.2824/38623

[34] Daniel E. O'leary. 1995. Some Privacy Issues in Knowledge Discovery: The OECD Personal Privacy Guidelines. *IEEE Expert-Intelligent Systems and their Applications* 10, 2 (1995), 48–59. https://doi.org/10.1109/64.395352

[35] Fred D. Davis. 1989. Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly: Management Information Systems* 13, 3 (1989), 319–339. https://doi.org/10.2307/249008

[36] Paloma Diaz, 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. In *2008 IEEE Symposium on Visual Languages and Human-Centric Computing*. 203–210. https://doi.org/10.1109/VLHCC.2008.4639087

[37] 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* (L'Aquila, Italy) *(AVI '98)*. Association for Computing Machinery, New York, NY, USA, 124–134. https://doi.org/10.1145/948496.948514

[38] David W. Eccles and Güler Arsal. 2017. The think aloud method: what is it and how do I use it? *Qualitative Research in Sport, Exercise and Health* 9, 4 (2017), 514–531. https://doi.org/10.1080/2159676X.2017.1331501 arXiv:https://doi.org/10.1080/2159676X.2017.1331501

[39] 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. , 89 pages. https://edpb.europa.eu/our-work-tools/our-documents/binding-decision-board-art-65/binding-decision-12021-dispute-arisen_en

[40] Katalin Ferencz and Jozsef Domokos. 2020. Using Node-RED platform in an industrial environment. *Jubileumi Kandó Konferencia* February (2020), 13.

[41] Gina Fisk, Calvin Ardi, Neale Pickett, John Heidemann, Mike Fisk, and Christos Papadopoulos. 2015. Privacy principles for sharing cyber security data. *Proceedings - 2015 IEEE Security and Privacy Workshops, SPW 2015* (2015), 193–197. https://doi.org/10.1109/SPW.2015.23

[42] FL. 2009. *ARTICLE 29 Data Protection Working Party Working Party on Police and Justice The Future of Privacy Joint contribution to the Consultation of the European Commission on the legal framework for the fundamental right to protection of personal data.* Technical Report.

[43] 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. In *2019 23rd International Conference in Information Visualization – Part II*. 47–51. https://doi.org/10.1109/IV-2.2019.00018

[44] Communication From, T H E Commission, T O The, T H E Council, T H E European Economic, T H E Committee, and O F The. 2014. Towards a thriving data-driven economy. *European Commission* COM(2014), 442 (2014).

[45] Hemant Ghayvat, Subhas Mukhopadhyay, Xiang Gui, and Nagender Suryadevara. 2015. WSN- and IOT-based smart homes and their extension to smart buildings. *Sensors (Switzerland)* 15, 5 (2015), 10350–10379. https://doi.org/10.3390/s150510350

[46] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. 2013. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems* 29, 7 (2013), 1645–1660. https://doi.org/10.1016/j.future.2013.01.010 Including Special sections: Cyber-enabled Distributed Computing for Ubiquitous Cloud and Network Services & Cloud Computing and Scientific Applications — Big Data, Scalable Analytics, and Beyond.

[47] Seda F. Gürses, Carmela Troncoso, and Claudia Díaz. 2011. Engineering Privacy by Design. In *Conference on Privacy & Data Protection*, Vol. 14. 25 pages. Issue 3.

[48] 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

[49] Sandra G. Hart and Lowell E. Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. In *Human Mental Workload*, Peter A. Hancock and Najmedin Meshkati (Eds.). Advances in Psychology, Vol. 52. North-Holland, 139–183. https://doi.org/10.1016/S0166-4115(08)62386-9

[50] Jaap-henk Hoepman. 2014. IFIP AICT 428 - Privacy Design Strategies. (2014), 446–459. https://link.springer.com/content/pdf/10.1007/978-3-642-55415-5{_}38.pdf

[51] Martin Host, 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. https://doi.org/10.1023/A:1026586415054

[52] International Organization for Standardization. 2012. ISO/IEC 27032:2012 Information technology — Security techniques — Guidelines for cybersecurity. https://www.iso.org/standard/44375.html

[53] Shubham Jain and Janne Lindqvist. 2014. Should I Protect You? Understanding Developers' Behavior to Privacy-Preserving APIs. https://doi.org/10.14722/usec.2014.23045

[54] 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* (Redmond, Washington, USA) *(SOUPS '10)*. Association for Computing Machinery, New York, NY, USA, Article 14, 12 pages. https://doi.org/10.1145/1837110.1837129

[55] Israel Jerusalem. 2010. Resolution on Privacy by Design. In *In Proceedings of the 32nd International Conference of Data Protection and Privacy Commissioners*.

[56] Hyun Kang. 2021. Sample size determination and power analysis using the G* Power software. *Journal of educational evaluation for health professions* 18 (2021).

[57] Himmet Karadal and A. Abubakar. 2021. Internet of things skills and needs satisfaction: do generational cohorts' variations matter? *Online Information Review* ahead-of-print (02 2021). https://doi.org/10.1108/OIR-04-2020-0144

[58] Charat Khamsaeng and Sophon Mongkolluksamee. 2020. Providing an End-to-End Privacy Preservation over LoRa WanPlatforms. In *2020 - 5th International Conference on Information Technology (InCIT)*. 56–60. https://doi.org/10.1109/InCIT50588.2020.9310934

[59] Barbara Kitchenham, Tore Dybå, and M. Jorgensen. 2004. Evidence-based software engineering. 273– 281. https://doi.org/10.1109/ICSE.2004.1317449

[60] 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* (Vienna, Austria) *(CHI '04)*. Association for Computing Machinery, New York, NY, USA, 151–158. https://doi.org/10.1145/985692.985712

[61] California State Legislature. 2018. Bill Text - AB-375 Privacy: personal information: businesses. https://leginfo.legislature.ca.gov/faces/billTextClient.xhtml?bill_id=201720180AB375

[62] 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

[63] Tianshi Li, Elizabeth Louie, Laura Dabbish, and Jason I. Hong. 2021. How Developers Talk About Personal Data and What It Means for User Privacy: A Case Study of a Developer Forum on Reddit. *Proc. ACM Hum.-Comput. Interact.* 4, CSCW3, Article 220 (jan 2021), 28 pages. https://doi.org/10.1145/3432919

[64] Tianshi Li, Elijah B. Neundorfer, Yuvraj Agarwal, and Jason I. Hong. 2021. Honeysuckle: Annotation-Guided Code Generation of In-App Privacy Notices. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 5, 3, Article 112 (sep 2021), 27 pages. https://doi.org/10.1145/3478097

[65] Huichen Lin and Neil W. Bergmann. 2016. IoT privacy and security challenges for smart home environments. *Information (Switzerland)* 7, 3 (2016). https://doi.org/10.3390/info7030044

[66] Tom Lodge and Andy Crabtree. 2019. Privacy Engineering for Domestic IoT: Enabling Due Diligence. *Sensors* 19, 20 (2019). https://doi.org/10.3390/s19204380

[67] Kai Uwe Loser and Martin Degeling. 2014. Security and Privacy as Hygiene Factors of Developer Behavior in Small and Agile Teams. In *11th IFIP International Conference on Human Choice and Computers (HCC)*, Vol. 431. 255–265. https://doi.org/10.1007/978-3-662-44208-1_21

[68] Roberto Martinez-Maldonado, Andrew Clayphan, Kalina Yacef, and Judy Kay. 2015. MTFeedback: Providing Notifications to Enhance Teacher Awareness of Small Group Work in the Classroom. *IEEE Transactions on Learning Technologies* 8, 2 (2015), 187–200. https://doi.org/10.1109/TLT.2014.2365027

[69] Diego Martín, Ramón Alcarria, Tomás Robles, and Augusto Morales. 2013. A Systematic Approach for Service Prosumerization in IoT Scenarios. In *2013 Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*. 494–499. https://doi.org/10.1109/IMIS.2013.89

[70] 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

[71] Patrick E McKnight and Julius Najab. 2010. Mann-Whitney U Test. *The Corsini encyclopedia of psychology* (2010), 1–1.

[72] Julio Melo, Melquiades Fidelis, Sidney Alves, Ulisses Freitas, and Rummenigge Dantas. 2020. A comprheensive review of Visual Programming Tools for Arduino. *2020 Latin American Robotics Symposium, 2020 Brazilian Symposium on Robotics and 2020 Workshop on Robotics in Education, LARS-SBR-WRE 2020* (2020). https://doi.org/10.1109/LARS/SBR/WRE51543.2020.9307023

[73] OAIC. 2018. Australian entities and the EU General Data Protection Regulation (GDPR) — OAIC. https://www.oaic.gov.au/privacy/privacy-guidance-for-organisations-and-government-agencies/more-guidance/australian-entities-and-the-european-union-general-data-protection-regulationhttps://www.oaic.gov.au/privacy/guidance-and-advice/australian-entities-and-the-eu-general-data-protection-regulation/

[74] Marie Oetzel and Sarah Spiekermann. 2014. A systematic methodology for privacy impact assessments: A design science approach. *European Journal of Information Systems* 23 (03 2014). https://doi.org/10.1057/ejis.2013.18

[75] Office of the Privacy Commissioner. 2020. Privacy Act 2020. https://www.privacy.org.nz/privacy-act-2020/privacy-principles/https://www.privacy.org.nz/privacy-act-2020/privacy-act-2020/

[76] Office of the Privacy Commissioner Canada. 2019. PIPEDA legislation and related regulations - Office of the Privacy Commissioner of Canada. https://www.priv.gc.ca/en/privacy-topics/privacy-laws-in-canada/the-personal-information-protection-and-electronic-documents-act-pipeda/r_o_p/

[77] OPC. 2023. Office of the Privacy Commissioner of Canada - Office of the Privacy Commissioner of Canada. https://priv.gc.ca/en/privacy-topics/privacy-laws-in-canada/the-personal-information-protection-and-electronic-documents-act-pipeda/pipeda-compliance-help/pipeda-interpretation-bulletins/interpretations_10_sensible/https://www.priv.gc.ca/en/

[78] Paul Otto and Annie Antón. 2007. Addressing Legal Requirements in Requirements Engineering. 5–14. https://doi.org/10.1109/RE.2007.65

[79] European Parliament and Council of the European Union. 2016. 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). https://eur-lex.europa.eu/eli/reg/2016/679/oj

[80] Erik Pasternak, Rachel Fenichel, and Andrew N. Marshall. 2017. Tips for creating a block language with blockly. *Proceedings - 2017 IEEE Blocks and Beyond Workshop, B and B 2017* 2017-Novem (2017), 21–24. https://doi.org/10.1109/BLOCKS.2017.8120404

[81] Fabio Paternò and Carmen Santoro. 2019. End-user development for personalizing applications, things, and robots. *International Journal of Human-Computer Studies* 131 (2019), 120–130. https://doi.org/10.1016/j.ijhcs.2019.06.002 50 years of the International Journal of Human-Computer Studies. Reflections on the past, present and future of human-centred technologies.

[82] 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

[83] Charith Perera, Chi Harold Liu, and Srimal Jayawardena. 2015. The Emerging Internet of Things Marketplace From an Industrial Perspective: A Survey. *IEEE Transactions on Emerging Topics in Computing* 3, 4 (2015), 585–598. https://doi.org/10.1109/TETC.2015.2390034

[84] 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

[85] Charith Perera, Arkady Zaslavsky, Peter Christen, and Dimitrios Georgakopoulos. 2014. Context aware computing for the internet of things: A survey. *IEEE Communications Surveys and Tutorials* 16, 1 (2014), 414–454. https://doi.org/10.1109/SURV.2013.042313.00197 arXiv:1305.0982

[86] Ravendra Pratap Rana, Vishal Sharma, and Varsha Agarwal. 2023. An Efficient Technique for Energy Consumption and Network Lifetime by Distributed Data Gathering Method from IoT Nodes. , 01-07 pages. https://doi.org/10.1109/indiscon58499.2023.10270115

[87] Rodrigo Roman, Jianying Zhou, and Javier Lopez. 2013. On the features and challenges of security and privacy in distributed internet of things. *Computer Networks* 57, 10 (2013), 2266–2279.

[88] Martin Rost and Kirsten Bock. 2011. Privacy by Design and the New Protection Goals. *DuD, January* November 2009 (2011), 1–9. https://www.european-privacy-seal.eu/AppFile/GetFile/ca6cdc46-d4dd-477d-9172-48ed5f54a99c

[89] Iflaah Salman, Ayse Tosun Misirli, and Natalia Juristo. 2015. Are Students Representatives of Professionals in Software Engineering Experiments?. In *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, Vol. 1. 666–676. https://doi.org/10.1109/ICSE.2015.82

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

[91] Anthony Savidis, Yannis Valsamakis, and Dimitris Linaritis. 2022. Blockly Toolbox for Visual Programming of Smart IoT Automations. In *Ambient Intelligence – Software and Applications – 12th International Symposium on Ambient Intelligence*, Paulo Novais, Joao Carneiro, and Pablo Chamoso (Eds.). Springer International Publishing, Cham, 93–103.

[92] Awanthika Senarath and Nalin A. G. Arachchilage. 2018. Why Developers Cannot Embed Privacy into Software Systems? An Empirical Investigation. In *Proceedings of the 22nd International Conference on Evaluation and Assessment in Software Engineering 2018* (Christchurch, New Zealand) *(EASE'18)*. Association for Computing Machinery, New York, NY, USA, 211–216. https://doi.org/10.1145/3210459.3210484

[93] Awanthika Senarath and Nalin Asanka Gamagedara Arachchilage. 2019. A data minimization model for embedding privacy into software systems. *Computers and Security* 87 (2019). https://doi.org/10.1016/j.cose.2019.101605

[94] 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 Trans. Priv. Secur.* 22, 4, Article 23 (nov 2019), 30 pages. https://doi.org/10.1145/3364224

[95] Stuart S. Shapiro. 2010. Privacy by Design: Moving from Art to Practice. *Commun. ACM* 53, 6 (jun 2010), 27–29. https://doi.org/10.1145/1743546.1743559

[96] Swapneel Sheth, Gail Kaiser, and Walid Maalej. 2014. Us and Them: A Study of Privacy Requirements across North America, Asia, and Europe. In *Proceedings of the 36th International Conference on Software Engineering* (Hyderabad, India) *(ICSE 2014)*. Association for Computing Machinery, New York, NY, USA, 859–870. https://doi.org/10.1145/2568225.2568244

[97] Laurens Sion, Kim Wuyts, Koen Yskout, Dimitri Van Landuyt, and Wouter Joosen. 2018. Interaction-Based Privacy Threat Elicitation. In *2018 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. 79–86. https://doi.org/10.1109/EuroSPW.2018.00017

[98] 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

[99] Sarah Spiekermann and Lorrie Faith Cranor. 2009. Engineering Privacy. *IEEE Transactions on Software Engineering* 35, 1 (2009), 67–82. https://doi.org/10.1109/TSE.2008.88

[100] Gaurav Srivastava, Kunal Bhuwalka, Swarup Kumar Sahoo, Saksham Chitkara, Kevin Ku, Matt Fredrikson, Jason Hong, and Yuvraj Agarwal. 2017. PrivacyProxy: Leveraging Crowdsourcing and In Situ Traffic Analysis to Detect and Mitigate Information Leakage. (2017). arXiv:1708.06384 http://arxiv.org/abs/1708.06384

[101] Luke Stark, Jen King, Xinru Page, Airi Lampinen, Jessica Vitak, Pamela Wisniewski, Tara Whalen, and Nathaniel Good. 2016. Bridging the Gap between Privacy by Design and Privacy in Practice. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI EA '16)*. Association for Computing Machinery, New York, NY, USA, 3415–3422. https://doi.org/10.1145/2851581.2856503

[102] State of California Department of Justice. 2018. California Consumer Privacy Act (CCPA) | State of California - Department of Justice - Office of the Attorney General. https://oag.ca.gov/privacy/ccpa

[103] Susan Steffee. 2017. IOT HELP WANTED: A lack of Internet of Things knowledge–and skills–leaves businesses struggling to recruit talent. *Internal Auditor* 74, 5 (Oct. 2017), 11+. link.gale.com/apps/doc/A512185039/AONE?u=googlescholar&sid=bookmark-AONE&xid=464fa250

[104] Mohammad Tahaei, Alisa Frik, and Kami Vaniea. 2021. Privacy Champions in Software Teams: Understanding Their Motivations, Strategies, And Challenges. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 693, 15 pages. https://doi.org/10.1145/3411764.3445768

[105] Alexander G. Tamilias, Theodoros J. Themelis, Theodoros Karvounidis, Zacharenia Garofalaki, and Dimitrios Kallergis. 2017. B@SE: Blocks for @rduino in the Students' educational process. *IEEE Global Engineering Education Conference, EDUCON* April (2017), 910–915. https://doi.org/10.1109/EDUCON.2017.7942956

[106] Ying Tang, Morgan L. Brockman, and Sameer Patil. 2021. Promoting Privacy Considerations in Real-World Projects in Capstone Courses with Ideation Cards. *ACM Trans. Comput. Educ.* 21, 4, Article 34 (oct 2021), 28 pages. https://doi.org/10.1145/3458038

[107] Diogo Torres, Joao Pedro Dias, Andre Restivo, and Hugo Sereno Ferreira. 2020. Real-time Feedback in Node-RED for IoT Development: An Empirical Study. *Proceedings of the 2020 IEEE/ACM 24th International Symposium on Distributed Simulation and Real Time Applications, DS-RT 2020* (2020). https://doi.org/10.1109/DS-RT50469.2020.9213544

[108] Miguel Ehecatl Trujillo, Gabriel García-Mireles, Erick Orlando Matla Cruz, and Mario Piattini. 2019. A Systematic Mapping Study on Privacy by Design in Software Engineering. *CLEI Electronic Journal* 22 (04 2019). https://doi.org/10.19153/cleiej.22.1.4

[109] 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

[110] Jeroen van Rest, Daniel Boonstra, Maarten Everts, Martin van Rijn, and Ron van Paassen. 2014. Designing Privacy-by-Design. In *Privacy Technologies and Policy*, Bart Preneel and Demosthenes Ikonomou (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 55–72.

[111] Chamila Wijayarathna, Marthie Grobler, and Nalin Arachchilage. 2019. Software developers need help too! Developing a methodology to analyse cognitive dimension-based feedback on usability. *Behaviour & Information Technology* 40 (12 2019), 1–22. https://doi.org/10.1080/0144929X.2019.1705393

[112] David Wright and Charles Raab. 2014. Privacy principles, risks and harms. *International Review of Law, Computers and Technology* 28, 3 (2014), 277–298. https://doi.org/10.1080/13600869.2014.913874

A Appendix: Background Information of the Participants

Table 1. The background information of the lab study participants regarding software and IoT development includes the following details: Age (their age range), Years of exp (the number of years of experience they have in software engineering), Privacy training (whether the developers have received any formal privacy training and the nature of that training), No. IoT apps (the number of IoT applications they have developed), Blockly and Node-RED (whether they are familiar with the Blockly and Node-RED visual programming development tools), 'N' indicates that the developer has not received any privacy training. 'Neither' indicates that the developer is not familiar with neither Blockly@rduino nor Node-RED. 'Both' means the developers were familiar with Blockly@rduino and Node-RED.

ID	Age	Years of exp	Privacy training	No. IoT apps	Blockly and Node-RED
PE1	[25-34]	3-5	MSc in Cybersecurity	1	Neither
PE2	[35-44]	> 10	Privacy research, 4 years	5	Node-RED
PE3	[25-34]	3-5	MSc in Information Security	> 5	Both
PE4	[25-34]	> 10	Privacy research, 4 years	1	Neither
PE5	[25-34]	< 1	N	1	Neither
PE6	[25-34]	< 1	MSc in Cybersecurity	1	Neither
PE7	[25-34]	1-3	N	3	Node-RED
PE8	[18-24]	1-3	Malware analysis, Computer and Network forensics	2	Blockly
PE9	[25-34]	1-3	N	2	Node-RED
PC1	[25-34]	5-10	Privacy research, 3 years	3	Neither
PC2	[25-34]	> 10	N	>5	Both
PC3	[25-34]	1-3	MSc in Cybersecurity	1	Neither
PC4	[25-34]	3-5	MSc in Cybersecurity	>5	Neither
PC5	[25-34]	3-5	N	2	Node-RED
PC6	[18-24]	<1	N	2	Neither
PC7	[35-44]	5-10	Several times as a developer	3	Both
PC8	[25-34]	1-3	Privacy research, 3 years	1	Neither
PC9	[25-34]	5-10	N	>5	Node-RED

B Appendix: Focus Group Study Responses

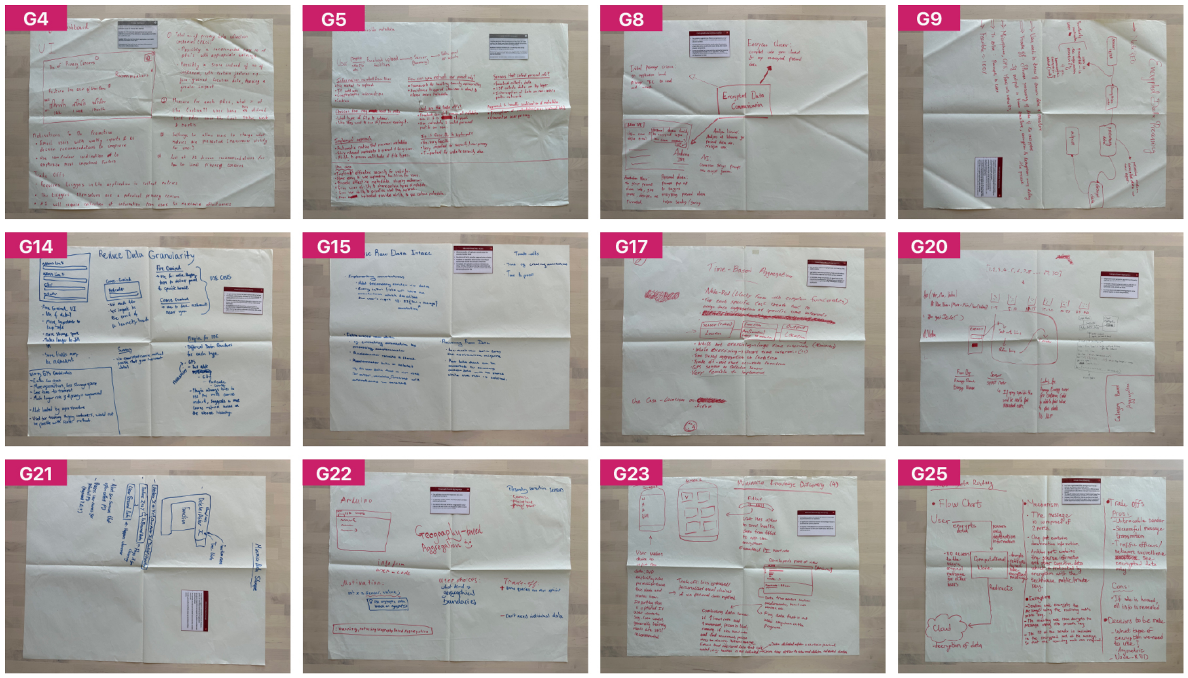
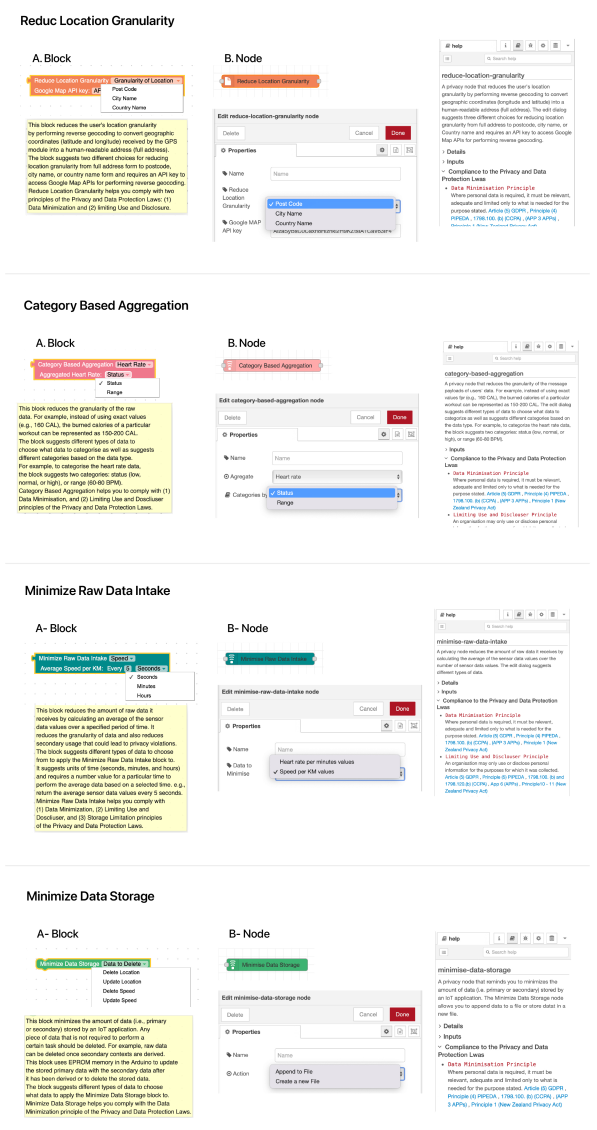


Fig. 11. Focus groups' proposed solutions to different privacy-preserving ideas of the PbD schemes. For a clearer view of each sub-figure, refer to the online repository: Focus Group Solutions.

C Appendix: Privacy-Preserving Components

C.1 Design of Privacy-Preserving Components



This block reduces the user's location granularity by performing reverse geocoding to convert geographic coordinates (latitude and longitude) received by the GPS module into a human-readable address (full address). The block suggests two different choices for reducing location granularity from full address form to postcode, city name, or country name form and requires an API key to access Google Map APIs for performing reverse geocoding. Reduce Location Granularity helps you comply with two principles of the Privacy and Data Protection Laws: (1) Data Minimization and (2) limiting Use and Disclosure.

Edit reduce-location-granularity node

Name: Name
Reduce Location Granularity: [✓ Post Code] City Name Country Name
Google MAP API key: AIzaSyBsC0Caxn8HiznkI2H9KZtsIA1Cav63IF4

reduce-location-granularity

A privacy node that reduces the user's location granularity by performing reverse geocoding to convert geographic coordinates (longitude and latitude) into a human-readable address (full address). The edit dialog suggests three different choices for reducing location granularity from full address to postcode, city name, or Country name and requires an API key to access Google Map APIs for performing reverse geocoding.

› Details
› Inputs
⌄ Compliance to the Privacy and Data Protection Lwas

- `Data Minimisation Principle`
  Where personal data is required, it must be relevant, adequate and limited only to what is needed for the purpose stated. Article (5) GDPR , Principle (4) PIPEDA , 1798.100. (b) (CCPA) , (APP 3 APPs) , Principle 1 (New Zealand Privacy Act)

This block reduces the granularity of the raw data. For example, instead of using exact values (e.g., 160 CAL), the burned calories of a particular workout can be represented as 150-200 CAL. The block suggests different types of data to choose what data to categorise as well as suggests different categories based on the data type. For example, to categorise the heart rate data, the block suggests two categories: status (low, normal, or high), or range (60-80 BPM). Category Based Aggregation helps you to comply with (1) Data Minimisation, and (2) Limiting Use and Doscliuser principles of the Privacy and Data Protection Laws.

Edit category-based-aggregation node

Name: Name
Agregate: Heart rate
Categories by: [✓ Status] Range

category-based-aggregation

A privacy node that reduces the granularity of the message payloads of users' data. For example, instead of using exact values fpr (e.g., 160 CAL), the burned calories of a particular workout can be represented as 150-200 CAL. The edit dialog suggests different types of data to choose what data to categorize as well as suggests different categories based on the data type. For example, to categorize the heart rate data, the block suggests two categories: status (low, normal, or high), or range (60-80 BPM).

› Inputs
⌄ Compliance to the Privacy and Data Protection Lwas

- `Data Minimisation Principle`
  Where personal data is required, it must be relevant, adequate and limited only to what is needed for the purpose stated. Article (5) GDPR , Principle (4) PIPEDA , 1798.100. (b) (CCPA) , (APP 3 APPs) , Principle 1 (New Zealand Privacy Act)
- `Limiting Use and Disclouser Principle`
  An organisation may only use or disclose personal

This block reduces the amount of raw data it receives by calculating an average of the sensor data values over a specified period of time. It reduces the granularity of data and also reduces secondary usage that could lead to privacy violations. The block suggests different types of data to choose from to apply the Minimize Raw Data Intake block to. It suggests units of time (seconds, minutes, and hours) and requires a number value for a particular time to perform the average data based on a selected time. e.g., return the average sensor data values every 5 seconds. Minimize Raw Data Intake helps you comply with (1) Data Minimization, (2) Limiting Use and Doscliuser, and (3) Storage Limitation principles of the Privacy and Data Protection Laws.

Edit minimise-raw-data-intake node

Name: Name
Data to Minimise: Heart rate per minutes values [✓ Speed per KM values]

minimise-raw-data-intake

A privacy node reduces the amount of raw data it receives by calculating the average of the sensor data values over the number of sensor data values. The edit dialog suggests different types of data.

› Details
› Inputs
⌄ Compliance to the Privacy and Data Protection Lwas

- `Data Minimisation Principle`
  Where personal data is required, it must be relevant, adequate and limited only to what is needed for the purpose stated. Article (5) GDPR , Principle (4) PIPEDA , 1798.100. (b) (CCPA) , (APP 3 APPs) , Principle 1 (New Zealand Privacy Act)
- `Limiting Use and Disclouser Principle`
  An organisation may only use or disclose personal information for the purposes for which it was collected. Article (5) GDPR , Principle (5) PIPEDA , 1798.100. (b) and 1798.120.(b) (CCPA) , App 6 (APPs) , Principle10 - 11 (New Zealand Privacy Act)

This block minimizes the amount of data (i.e., primary or secondary) stored by an IoT application. Any piece of data that is not required to perform a certain task should be deleted. For example, raw data can be deleted once secondary contexts are derived. This block uses EPROM memory in the Arduino to update the stored primary data with the secondary data after it has been derived or to delete the stored data. The block suggests different types of data to choose what data to apply the Minimize Data Storage block to. Minimize Data Storage helps you comply with the Data Minimization principle of the Privacy and Data Protection Laws.

Edit minimise-data-storage node

Name: Name
Action: Append to File / Create a new File

minimise-data-storage

A privacy node that reminds you to minimizes the amount of data (i.e. primary or secondary) stored by an IoT application. The Minimize Data Storage node allows you to append data to a file or store datat in a new file.

› Details
› Inputs
⌄ Compliance to the Privacy and Data Protection Lwas

- `Data Minimisation Principle`
  Where personal data is required, it must be relevant, adequate and limited only to what is needed for the purpose stated. Article (5) GDPR , Principle (4) PIPEDA , 1798.100. (b) (CCPA) , (APP 3 APPs) , Principle 1 (New Zealand Privacy Act)

Fig. 12. The figure demonstrates the privacy blocks in Blockly@rduino and privacy nodes in Node-RED, along with their configuration windows. It also includes an explanation of their usage and compliance with privacy laws.

C.2 Compliance with Privacy and Data Protection Laws

If developers use the Reduce Location Granularity, Minimize Raw Data Intake, or Category-based Aggregation block or node and choose the right options for the application, they will be following the CPLF's Data Minimization and Limiting Use and Disclouser Principles. Once developers use Minimize Data Storage, they will comply with the Data Minimization Principle.

(1) **Data Minimization Principle:** "Where personal data is required, it must be relevant, adequate, and limited to what is needed for the purpose stated." Article (5) GDPR, Principle (4) PIPEDA, 1798.100. (b) (CCPA), (APP 3 APPs), Principle 1 (New Zealand Privacy Act) [12].
(2) **Limiting Use and Disclouser Principle:** "An organization may only use or disclose personal information for the purposes for which it was collected." Article (5) GDPR, Principle (5) PIPEDA, 1798.100. (b) and 1798.120. (b) (CCPA), App 6 (APPs), and Principles 10–11 (New Zealand Privacy Act) [12].

D Appendix: Prototyping: Design and Implementation of Canella

D.1 Appendix: Business Process Model Notation of the Fitness Tracking IoT Application

Figure 13 depicts the BPMN of the fitness tracking IoT application, which includes components like the wristband, trainer server, and cloud. The Trainer Center manages the wristband and enables trainers to monitor trainees' activities. Additionally, the trainer server registers trainees for activity monitoring.

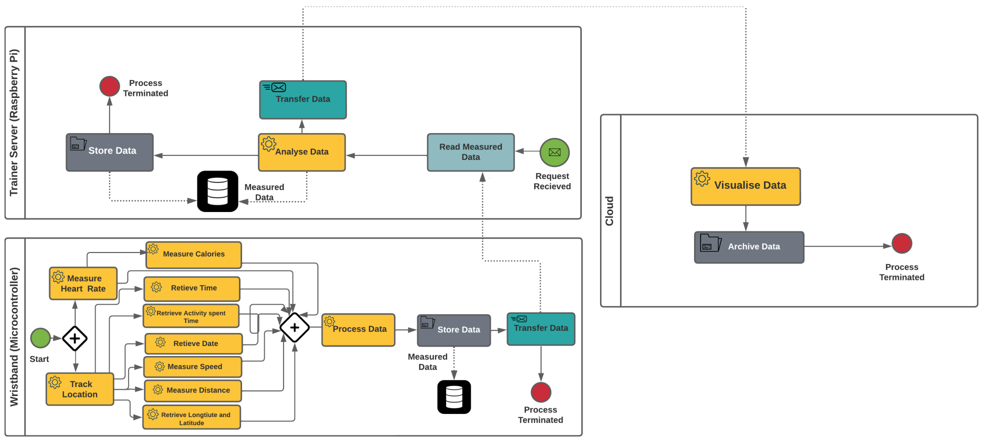


Fig. 13. Business Process Model Notation of the Fitness Tracking IoT Application

Wristband— it includes a heart rate monitor along with a GPS tracker. When a person starts a workout, the wristband displays the current date, the current time, the activity spent time, the latitude and longitude of the user's location, the speed, the distance covered, the real-time heart rate, and the active calories.

- **Heart rate monitor:** It measures the heart rate and returns the real-time heart rate and the active calories.
- **GPS Tracker:** It tracks the location of the user and returns the following information during a particular workout (e.g., running, walking, cycling, etc.): the current date, the current time, the activity spent time, the latitude and longitude of the user's location, the speed, and the distance covered.

Trainer Server— Once a user has completed a particular activity, the collected data is sent to the trainer's server through a Wi-Fi connection to track the trainees' progress toward their fitness and health goals, as seen in

Figure 13. Then, the trainees' data is stored on the trainer's server to analyze the history of their workout. After that, the analyzed data is sent to the cloud for archiving purposes.

Cloud— It receives the trainees' data for archiving purposes, so the trainer and trainees can retrieve data.

D.2 Appendix: Design and Implementation of Canella's IoT Use Case Scenario

D.2.1 Blockly@rduino. All necessary building blocks for this application are accessible in Blockly@rduino's toolbox under "Fitness Tracking IoT Application." These blocks follow guidelines outlined on the Blockly website, are designed using JavaScript, and employ a natural language style for user-friendliness [80]. Sensors are represented with images and pin fields for connections. Data collection blocks feature a right-hand notch for easy integration with privacy-preserving components. Custom blocks for processing specific data types are provided, and two additional structural blocks, "Start of Fitness Tracking IoT Application" and "End of Fitness Tracking Application," facilitate organization. The main blocks, "Track User's Locations" and "Measure Heart Rate," contain nested blocks for GPS and heart rate data processing. The "Retrieve workout data" block sends data to the trainer's server upon workout completion. Tooltip information explains block functionality. Blocks handling specific data types share a consistent color code for clarity and organization.

D.2.2 Node-RED. We utilized the "Dashboard" in the Node-RED Pallet to display the interface of the trainer to monitor their trainees. Furthermore, we used the MQTT out node from the Node-RED Pallet to send the workout data to the cloud. Then, we have integrated two EUDs: Blockly@rduino and Node-RED. The analytical summary information can be sent to the cloud through WiFi using the MQTT protocol for archiving purposes.

D.3 Appendix: Demonstration of Canella

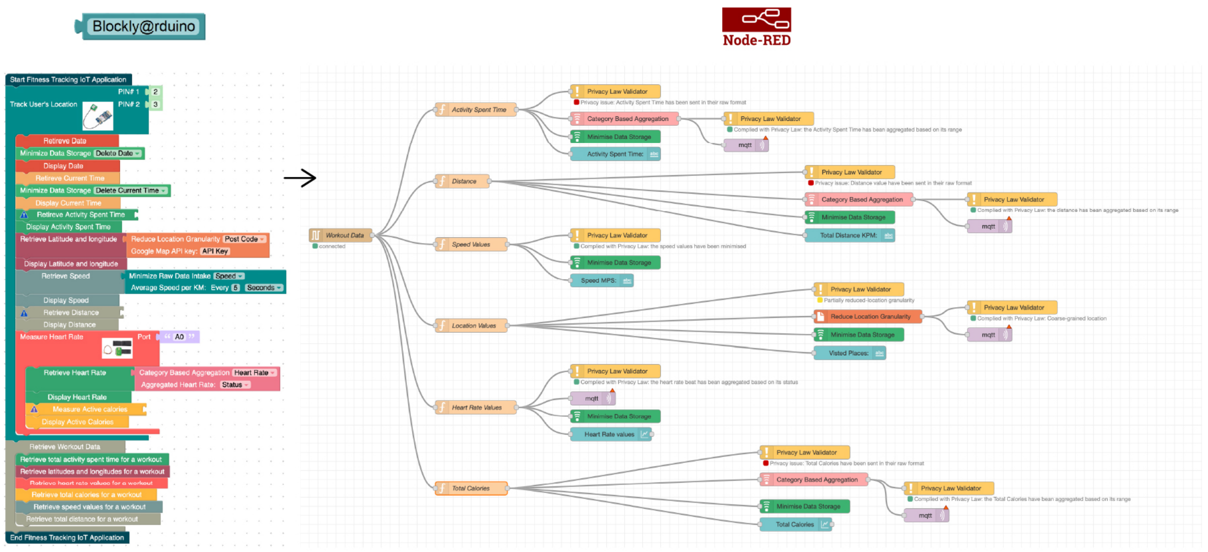
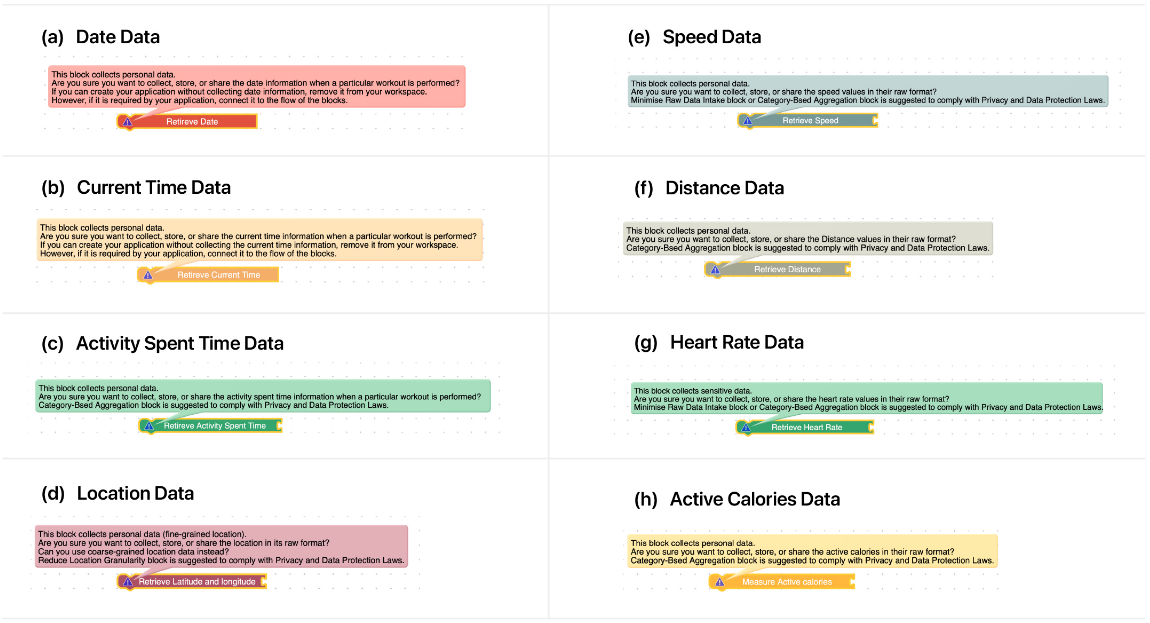


Fig. 14. Demonstration of how Canella can help software developers create privacy-preserving IoT applications.



(a) Date Data

This block collects personal data.
Are you sure you want to collect, store, or share the date information when a particular workout is performed?
If you can create your application without collecting date information, remove it from your workspace.
However, if it is required by your application, connect it to the flow of the blocks.

(b) Current Time Data

This block collects personal data.
Are you sure you want to collect, store, or share the current time information when a particular workout is performed?
If you can create your application without collecting the current time information, remove it from your workspace.
However, if it is required by your application, connect it to the flow of the blocks.

(c) Activity Spent Time Data

This block collects personal data.
Are you sure you want to collect, store, or share the activity spent time information when a particular workout is performed?
Category-Bsed Aggregation block is suggested to comply with Privacy and Data Protection Laws.

(d) Location Data

This block collects personal data (fine-grained location).
Are you sure you want to collect, store, or share the location in its raw format?
Can you use coarse-grained location data instead?
Reduce Location Granularity block is suggested to comply with Privacy and Data Protection Laws.

(e) Speed Data

This block collects personal data.
Are you sure you want to collect, store, or share the speed values in their raw format?
Minimise Raw Data Intake block or Category-Bsed Aggregation block is suggested to comply with Privacy and Data Protection Laws.

(f) Distance Data

This block collects personal data.
Are you sure you want to collect, store, or share the Distance values in their raw format?
Category-Bsed Aggregation block is suggested to comply with Privacy and Data Protection Laws.

(g) Heart Rate Data

This block collects sensitive data.
Are you sure you want to collect, store, or share the heart rate values in their raw format?
Minimise Raw Data Intake block or Category-Bsed Aggregation block is suggested to comply with Privacy and Data Protection Laws.

(h) Active Calories Data

This block collects personal data.
Are you sure you want to collect, store, or share the active calories in their raw format?
Category-Bsed Aggregation block is suggested to comply with Privacy and Data Protection Laws.

Fig. 15. Canella offers warning messages on potential privacy issues and suggests the appropriate privacy-preserving components to be integrated into the flow of data for privacy compliance.

E Appendix: Pre-Lab Study Survey Questionnaire

E.1 Demographic Information and Background Survey

- Email Address:
- Age range:[18-24][25-34][35-44][45-54][55-64][65+]
- How many years of experience do you have in Software Engineering?:[<1][1-3][3-5][5-10][>10]
- Have you had any formal privacy training (for example, a degree in cyber-security, a course, or research)? If yes: What kind of privacy training have you received?
- Have you developed an IoT mobile application with sensors to collect data, or have you ever developed an IoT application using an edge development board (like the Arduino Uno) or a microcomputer (like the Raspberry Pi)?: If yes: How many IoT applications have you developed?: [<1][1-3][3-5][5-10][>10]
- Are you familiar with Blockly and Node-RED visual programming development tools?[Both Blockly and Node-Red][Only Blockly][Only Node-Red][Neither]

F Appendix: Lab Study Survey Questionnaire

F.1 Factual Questions About Developers' Behavior When Handling Application Data

- Have you sent the Date when a user performed a particular activity to the trainer's server for analytical or storage purposes?
  - Yes | No | Other.
- Have you stored the Date when a user performed a particular activity in the wristband?
  - Yes | No | Other.

- Have you sent the Current Time when a user performed a particular activity to the trainer's server for analytical or storage purposes?
  - Yes | No | Other.
- Have you stored the Current Time when a user performed a particular activity in the wristband?
  - Yes | No | Other.
- In what format have you stored the Activity Spent Time on the wristband?
  - Raw Format (Activity Spent Time) | Range of the Activity Spent Time | Other.
- What is the format of the Activity Spent Time you have sent to the trainer's server for analytical or storage purposes?
  - Raw Format (The Activity Spent Time) | Range of the Activity Spent Time. | Other.
- In which format have you stored the Activity Spent Time on the trainer's server?
  - Raw Format (The Activity Spent Time) | Range of the Activity Spent Time | Other.
- What is the format of the Activity Spent Time you have sent to the cloud for storage purposes?
  - Raw Format (The Activity Spent Time) | Range of the Activity Spent Time | Other.
- What is the format or granularity of the Location you have stored on the wristband?
  - Longitude and Latitude (fine-grained format) | Postcode | City Name | Country Name (coarse-grained format).
- What is the format or granularity of the Location you have sent to the trainer's server for analytical or storage purposes?
  - Longitude and Latitude (fine-grained format) | Postcode | City Name | Country Name (coarse-grained format).
- What is the format or granularity of the Location you have stored on the trainer's server?
  - Longitude and Latitude (fine-grained format) | Postcode | City Name | Country Name (coarse-grained format).
- What is the format or granularity of the Location you have sent to the cloud for storage purposes?
  - Longitude and Latitude (fine-grained format) | Postcode | City Name | Country Name (coarse-grained format).
- In which format have you stored the Speed values on the wristband?
  - Raw Format (Speed values) | The average speed values | The range of speed values | Other.
- In what format have you sent the Speed to the trainer's server for analytical or storage purposes?
  - Raw Format (Speed values) | The average speed values | The range of speed values | Other.
- What is the format of the Speed values you have stored on the trainer's server?
  - Raw Format (Speed values) | The average speed values| The range of speed values. | Other.
- What is the format of the Speed values you have sent to the cloud for storage purposes?
  - Raw Format (Speed values) | The average speed values | The range of speed values | Other.
- In which format have you stored the Total Distance on the wristband?
  - Raw Format (Total Distance) | Range of the Total Distance | Other.
- In what format have you sent the Total Distance to the trainer's server for analytical or storage purposes?
  - Raw Format (Total Distance) | Range of the Total Distance | Other.
- What is the format of the total distance you have stored on the trainer's server?
  - Raw Format (Total Distance) | Range of the Total Distance | Other.
- What is the format of the Total Distance you have sent to the cloud for storage purposes?
  - Raw Format (Total Distance) | Range of the Total Distance | Other.
- In what format have you stored the Heart Rate values on the wristband?
  - Raw Format (Heart rate values) | Status of Heart Rate (low, normal, high) | Range of Heart Rate (> 60 BPM, 60 - 100 BPM, <100 BPM) | Average of Heart Rate values.

- What is the format of the Heart Rate you have sent to the trainer's server for analytical or storage purposes?
  - Raw Format (Heart rate values) | Status of Heart Rate (low, normal, high) | Range of Heart Rate (> 60 BPM, 60 - 100 BPM, <100 BPM) | Average of Heart Rate values.
- What is the format of the Heart Rate values you have stored on the trainer's server?
  - Raw Format (Heart rate values) | Status of Heart Rate (low, normal, high) | Range of Heart Rate (> 60 BPM, 60 - 100 BPM, <100 BPM) | Average of Heart Rate values.
- What is the format of the Heart Rate values you have sent to the cloud for storage purposes?
  - Raw Format (Heart rate values) | Status of Heart Rate (low, normal, high) | Range of Heart Rate (> 60 BPM, 60 - 100 BPM, <100 BPM) | Average of Heart Rate values.

F.2 Feedback on Canella

Note: (Only for the experimental group, a subjective rating on a 1 to 7 Likert scale, where 1 represents "strongly disagree" and 7 represents "strongly agree").

F.2.1 Measuring how disruptive and time-consuming developers perceive Canella.

- I felt that the warning messages on the blocks in Blockly@rduino were disruptive.
- I felt that the warning messages on the blocks in Blockly@rduino were time-consuming.
- I felt that when connecting the privacy block with the block that collects personal data in Blockly@rduino, the warning message disappeared, which was disruptive.
- I felt that when connecting the privacy block with the block that collects personal data in Blockly@rduino, the warning message disappeared, which was time-consuming.
- I felt that understanding the objective of the Reduce Location Granularity block and node was disruptive.
- I felt that understanding the objective of the Reduce Location Granularity block and node was time-consuming.
- I felt that understanding the objective of the Category-based Aggregation block and node was disruptive.
- I felt that understanding the objective of the Category-based Aggregation block and node was time-consuming.
- I felt that understanding the objective of the Minimize Raw Data Intake block and the node was disruptive.
- I felt that understanding the Minimize Raw Data Intake block and the node was time-consuming.
- I felt that understanding the objective of the Minimise Data Storage block and node was disruptive.
- I felt that understanding the objective of the Minimise Data Storage block and node was time-consuming.
- I felt that configuring the privacy blocks was disruptive.
- I felt that configuring the privacy blocks was time-consuming.
- I felt that configuring the privacy nodes was disruptive.
- I felt that configuring the privacy nodes was time-consuming.
- I felt that the alternative data values suggested by Canella were disruptive.
- I felt that the alternative data values suggested by Canella were time-consuming.
- I felt that using the Privacy Law Validator in Node-RED was disruptive.
- I felt that using the Privacy Law Validator in Node-RED was time-consuming.
- I felt that reading the Privacy Law Validator statuses in Node-RED was disruptive.
- I felt that reading the Privacy Law Validator statuses in Node-RED was time-consuming.
- I felt that the color of the Privacy Law Validator statuses was disruptive.
- I felt that the color of the Privacy Law Validator statuses was time-consuming.
- I felt that integrating privacy blocks into the data flow of the IoT application in Blockly@rduino was disruptive.

- I felt that integrating privacy blocks into the data flow of the IoT application in Blockly@rduino was time-consuming.
- I felt that integrating privacy nodes into the data flow of the IoT application in Node-RED was disruptive.
- I felt that integrating privacy nodes into the data flow of the IoT application in Node-RED was time-consuming.

F.2.2 *Measuring how useful developers perceive Canella and its key features.*

- I found the warning messages in Blockly@rduino useful.
- I found that when connecting the privacy block with the block that collects personal data in Blockly@rduino, the warning message disappeared, which was useful.
- I found that the Reduce Location Granularity block and node were useful.
- I found that the Category-based Aggregation block and node were useful.
- I found that the Minimize Raw Data Intake block and node were useful.
- I found that the Minimize Data Storage Data block and node were useful.
- I found that integrating privacy blocks was useful.
- I found that integrating privacy nodes was useful.
- I found that configuring the privacy blocks was useful.
- I found that configuring the privacy nodes was useful.
- I found that the different data types suggested by Canella were useful.
- I found that using the Privacy Law Validator in Node-RED was useful.
- I found reading the Privacy Law Validator statuses in Node-RED useful.
- I found that the color of the Privacy Law Validator statuses useful.
- I found the on-hover information in the privacy blocks useful.
- I found the information in the "Help" tab in the privacy nodes useful.

F.3 Open-ended questions assessing the participants' comprehension of some of Canella's features

- Please describe in one sentence: What do warning messages on some of the blocks in Blockly mean?
- Please describe in one sentence: What does the yellow color mean in the Privacy Law Validator Node in Node-Red? Skip this question if you don't know what this means.
- Please describe in one sentence: What does the red color mean in the Privacy Law Validator Node in Node-Red? Skip this question if you don't know what this means.
- Please describe in one sentence: What does the green color mean in the Privacy Law Validator Node in Node-Red? Skip this question if you don't know what this means.
- Did you realize that these privacy blocks and privacy nodes comply with the regulations of privacy and data protection laws?
  - Did you realize that these privacy blocks and privacy nodes comply with the regulations of privacy and data protection laws?
- Do you have any ideas for more Canella features that could help developers build privacy-aware IoT applications?

G Appendix: Fitness Tracking IoT Application Use Case Scenario of Trainer Sara and Trainee Amy

Fitness Tracking IoT Application Use Case Scenario of Trainer Sara and Trainee Amy

From the problem owner's perspective, we present a use-case scenario that emphasizes high-level functional requirements. The objective is to develop an IoT application for fitness tracking that analyzes trainees' data and shares their information with trainers. This allows trainers to monitor trainees' progress, encourage them, and support their efforts to improve health and well-being. The scenario includes various privacy challenges that should be considered when building a privacy-aware IoT application.

Sara is a trainer who monitors her trainees' progress, encouraging them to improve their health and wellness. As seen in Figure 5, the system combines many devices and nodes: a wristband (which includes a GPS sensor along with a heart rate monitor), a trainer server, and a cloud. The wristband is managed by the trainer center, which offers trainers who monitor trainees' activities. The trainer's server is also owned by a trainer center, where trainees are registered to allow them to monitor their activities. The cloud is managed by a third party. This use case tracks user activity by getting personal information about the user from a wristband.

Amy is one of Sara's trainees. In the Fitness Tracking IoT application, once a person starts a workout, the wristband displays their workout data. Amy is interested in visualizing all the data during the workout. Once she has completed a particular activity, the collected data is stored on her wristband for a short period of time (a month). She is interested in visualizing the history of her workout data. However, she is not interested in viewing the date and time when she performed the workout.

Then, the collected data is sent to the trainer's server through a Wi-Fi connection. The communication between devices in the fitness tracking IoT application is secure. Sara is interested in visualizing all of Amy's workout data to monitor her progress and encourage her wellbeing. Sara is interested in viewing the different areas where she performed a particular workout. However, Sara doesn't care about the date and time when Amy completed a specific workout. Sara and the trainer center do not have access to the wristband data. For analytical purposes, Amy's data is stored on the trainer's server to analyze the history of her workout for a short period of time (a month). After that, the analyzed data is sent to the cloud for archiving purposes, so Sara and Amy can retrieve workout data.

Imagine you are developing an IoT application for real users and taking privacy into consideration throughout the development process.

H Appendix: Scoring Criteria for Privacy Assessment

We created thorough scoring criteria, outlined in Table 2, to evaluate developers' handling of personal data within the IoT application. These criteria align with data minimization principles, emphasizing the reduction of the collection, storage, and sharing of data to the next node whenever possible. To ensure objectivity and broad applicability, we consulted two privacy experts and a privacy lawyer. In addition, we sought input from a fitness trainer to consider how data in less granular or aggregated formats might affect trainees' well-being. To maintain fairness and minimize bias between the experimental and control groups, we take into account that the experimental group may be more inclined to prioritize privacy. Participants will not receive extra points for unnecessary privacy measures that do not significantly impact the core requirements. We maintain a lenient approach, considering developers' diverse fitness application backgrounds and user preferences. For example, sending cumulative data (e.g., burned calories) in its raw format to the edge node is acceptable, but transmitting the same raw data to the cloud is viewed as a privacy concern during scoring since it's meant for archival purposes. However, sharing sensitive data like heart rate values in their raw format when unnecessary, according to the use-case scenario, results in point deductions during scoring. Our aim is to strike a balance between user data privacy and meeting trainers' specific data needs. In the Fitness Tracking IoT application demonstrated in Section 5.4.3, each participant's total privacy score should amount to 59 points.

Table 2. Scoring criteria set to assess developers' handling of personal data in apps

Score	Criteria
0	If the privacy issue is not identified.
0	If the privacy issue is identified, the data is not collected to meet privacy, and the core requirements are compromised.
1	If the privacy issue is identified but not addressed correctly.
2	If the privacy issue is identified and the issue is addressed correctly, but the data needs additional privacy measures.
3	If the privacy issue is identified and addressed correctly.

I Appendix: Time Spent Results (Table 3)

Table 3. Developers' Time spent on programming tasks in the control and experimental groups

Participant ID	Warm-up tasks	Primary Task	Final task
PE1	"7:49"	"44:29"	"35:12"
PE2	"6:24"	"50:00"	"14:19"
PE3	" 6:24"	"45:51"	"14:19"
PE4	"5:00"	"45:00"	"17:00"
PE5	"8:38"	"53:02"	"22:15"
PE6	"7:15"	"33:30"	"20:23"
PE7	"6:3"	"44:47"	"25:14"
PE8	"5:24"	"36:13"	"15:52"
PE9	"6:36"	"38:40"	"12:45"
PC1	"10:00"	"60:00"	"20:00"
PC2	"10:00"	"55:15"	"20:00"
PC3	15:56"	54:49"	"25:11"
PC4	"8:15"	"36:33"	"24:51"
PC5	"6:42"	"33:40"	"18:44"
PC6	"5:52"	"28:01"	"12:54"
PC7	"6:29"	"29:06"	"12:59"
PC8	"5:52"	"28:01"	"12:54"
PC9	"7:39"	"36:17"	"14:36"

J Appendix: Developers' Behaviour (Table 4, Table 5, and Table 6)

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009

Table 4. Developer's behavior in the experimental group when handling data. (Raw = raw format; PC = postcode; Avg = average)

Type of Data	Processing	PE1	PE2	PE3	PE4	PE5	PE6	PE7	PE8	PE9
Date	Collect	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	Store in wristband	Yes	No	No	No	No	No	No	No	No
	Share with trainer	No	No	No	No	No	No	No	No	No
	Store in the trainer	No	No	No	No	No	No	No	No	No
	Share with cloud	No	No	No	No	No	No	No	No	No
Current Time	Collect	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	Store in wristband	Yes	No	No	No	No	No	No	No	No
	Share with trainer	No	No	No	No	No	No	No	No	No
	Store in the trainer	No	No	No	No	No	No	No	No	No
	Share with cloud	No	No	No	No	No	No	No	No	No
Spent Time	Collect	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	Store in wristband	Raw	Raw	Raw	Range	Raw	Range	Raw	Raw	Raw
	Share with trainer	Range	Raw	Raw	Range	Raw	Range	Raw	Raw	Range
	Store in the trainer	Range	Range	Raw	Range	Raw	Range	Raw	Raw	Range
	Share with cloud	Range	Range	Raw	Range	Range	Range	Range	Range	Range
Location	Collect	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	Store in wristband	PC	Raw	Raw	PC	Raw	PC	Raw	Raw	Raw
	Share with trainer	PC	PC	PC	PC	PC	PC	PC	PC	PC
	Store in the trainer	PC	PC	PC	PC	PC	City	PC	PC	City
	Share with cloud	PC	Country	City	Country	City	City	PC	City	City
Speed	Collect	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	Store in wristband	Raw	Raw	Raw	Range	Raw	Raw	Raw	Raw	Raw
	Share with trainer	Range	Range	Avg	Range	Raw	Range	Raw	Avg	Avg
	Store in the trainer	Range	Range	Avg	Range	Raw	Range	Raw	Avg	Avg
	Share with cloud	Range	Range	Avg	Range	Avg	Range	Avg	Avg	Avg
Distance	Collect	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	Store in wristband	Raw	Raw	Raw	Range	Raw	Raw	Raw	Raw	Raw
	Share with trainer	Range	Raw	Range	Range	Raw	Range	Raw	Raw	Range
	Store in the trainer	Range	Range	Range	Range	Raw	Range	Raw	Raw	Range
	Share with cloud	Range	Range	Range	Range	Range	Range	Range	Range	Range
Heart Rate	Collect	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	Store in wristband	Raw	Raw	Raw	Range	Raw	Status	Raw	Raw	Raw
	Share with trainer	Avg	Avg	Avg	Range	Avg	Status	Raw	Status	Status
	Store in the trainer	Avg	Avg	Avg	Range	Avg	Status	Raw	Status	Status
	Share with cloud	Avg	Avg	Avg	Range	Avg	Status	Range	Status	Status
Calories	Collect	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	Store in wristband	Raw	Raw	Raw	Range	Raw	Raw	Raw	Raw	Raw
	Share with trainer	Range	Raw	Range	Range	Range	Range	Raw	Raw	Range
	Store in the trainer	Range	Range	Range	Range	Range	Range	Raw	Raw	Range
	Share with cloud	Range	Range	Range	Range	Range	Range	Range	Range	Range

Table 5. Developer's behavior in the control group when handling data (Raw = raw format; PC = postcode; Avg = average; NP = not properly applying privacy)

Type of Data	Processing	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9
Date	Collect	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	Store in wristband	No	No	Yes	Yes	No	Yes	Yes	Yes	No
	Share with trainer	No	No	No	Yes	No	No	No	No	No
	Store in the trainer	No	No	No	Yes	No	No	No	No	No
	Share with cloud	No	No	No	Yes	No	No	No	No	No
Current Time	Collect	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	Store in wristband	No	No	Yes	Yes	No	Yes	Yes	Yes	No
	Share with trainer	No	No	No	Yes	No	No	No	No	No
	Store in the trainer	No	No	No	Yes	No	No	No	No	No
	Share with cloud	No	No	No	Yes	No	No	No	No	No
Spent Time	Collect	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	Store in wristband	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw
	Share with trainer	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw
	Store in the trainer	Range	Raw	No	Raw	Raw	Raw	Raw	Raw	Raw
	Share with cloud	Range	Raw	Range	Raw	Raw	Raw	Raw	Raw	Min,Max,Avg
Location	Collect	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes
	Store in wristband	PC	PC	Raw	NP	No	Raw	Raw	Raw	Raw
	Share with trainer	PC	PC	No	NP	No	Raw	PC	Raw	No
	Store in the trainer	PC	PC	No	NP	No	Raw	PC	Raw	No
	Share with cloud	PC	City	No	NP	No	Raw	No	Raw	No
Speed	Collect	Yes	No	Yes	Yes	No	Yes	Yes	Yes	Yes
	Store in wristband	Avg	No	Raw	Raw	No	Raw	Raw	Raw	Raw
	Share with trainer	Avg	No	Raw	Raw	No	Raw	Raw	Raw	Raw
	Store in the trainer	Avg	No	No	NP	No	Raw	Raw	Raw	Avg
	Share with cloud	Avg	No	Range	NP	No	Raw	Avg + Range	Raw	Min,Max,Avg
Distance	Collect	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes
	Store in wristband	Raw	Raw	Raw	Raw	No	Raw	Raw	Raw	Raw
	Share with trainer	Raw	Raw	Raw	Raw	No	Raw	Raw	Raw	Raw
	Store in the trainer	Raw	Raw	No	NP	No	Raw	Raw	Raw	Raw
	Share with cloud	Raw	Raw	Range	NP	No	Raw	Raw	Raw	Min,Max,Avg
Heart Rate	Collect	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	Store in wristband	Status	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw
	Share with trainer	Status	Status	Raw	Raw	Raw	Raw	Raw	Raw	Avg
	Store in the trainer	Status	Status	No	Raw	Raw	Raw	Raw	Raw	Avg
	Share with cloud	Status	Status	Range	Raw	Raw	Raw	Avg + Range	Raw	Avg
Calories	Collect	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes
	Store in wristband	Raw	Raw	Raw	Raw	Raw	Raw	Raw	No	Raw
	Share with trainer	Raw	Raw	Raw	Raw	Raw	Raw	Raw	No	Raw
	Store in the trainer	Range	Raw	No	NP	Raw	Raw	Raw	No	Raw
	Share with cloud	Range	Raw	Range	NP	Raw	Raw	Raw	No	Min,Max,Avg

Table 6. Comparison between the developer's Behavior in the Control Group when handling data in the first and second rounds (Raw = raw format; PC = postcode; Avg = average; NP = not properly applying privacy)

Type of Data	Processing	PC1		PC2		PC3		PC4		PC5		PC6		PC7		PC8		PC9	
		R1	R2	R1	R2	R1	R2	R1	R2	R1	R2	R1	R2	R1	R2	R1	R2	R1	R2
Date	Store	No	No	No	No	No	No	Yes	Yes	No	No	No	No	No	No	No	No	No	No
	Share	No	No	No	No	No	No	Yes	Yes	Yes	No	No	No	No	No	No	No	No	No
Current Time	Store	No	No	No	No	No	No	Yes	Yes	No	No	No	No	No	No	No	No	No	No
	Share	No	No	No	No	No	No	Yes	Yes	Yes	No	No	No	No	No	No	No	No	No
Spent Time	Store	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw
	Share	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw
Location	Store	Raw	PC	PC	PC	Raw	Raw	NP	NP	Raw	No	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw
	Share	Raw	PC	PC	PC	Raw	No	NP	NP	Raw	No	Raw	Raw	PC	PC	Raw	Raw	No	No
Speed	Store	Raw	Avg	Raw	Raw	Raw	Raw	Raw	Raw	Raw	No	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw
	Share	Raw	Avg	Raw	No	Raw	Range	Raw	Raw	Raw	No	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw
Distance	Store	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw	No	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw
	Share	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw	No	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw
Heart Rate	Store	Raw	Status	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw
	Share	Raw	Status	Raw	Status	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Avg
Calories	Store	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw	No	Raw	Raw
	Share	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw	No	Raw	Raw