Enhancing Transparency and Consent in the Internet of Things

Victor Morel

N°d'ordre NNT : 2020LYSEI073

THESE de DOCTORAT DE L'UNIVERSITE DE LYON
opérée au sein de
Insa de Lyon

Ecole Doctorale N° 512
Infomaths

Spécialité/ discipline de doctorat :
Informatique

Soutenue publiquement le 24/09/2020, par :
Victor Morel

Enhancing Transparency and Consent in the Internet of Things

Devant le jury composé de :

Schneider, Gerardo - Professor at the University of Gothenburg - Rapporteur
Serrano Alvarado, Patricia - Maître de conférences HDR à l'Université de Nantes - Rapporteure

Rivano, Hervé - Professeur des universités à l'Insa de Lyon - Examinateur
Vallet, Félicien - Docteur ingénieur au sein du service de l'expertise technologique de la CNIL - Examinateur

Le Métayer, Daniel - Directeur de recherche Inria - Directeur de thèse
Castelluccia, Claude - Directeur de recherche Inria - Co-directeur de thèse

ENHANCING INFORMATION AND CONSENT IN THE INTERNET OF THINGS

VICTOR MOREL

Final Version

2020 – Classic Thesis version 4.2

ABSTRACT

In an increasingly connected world, the Internet permeates every aspect of our lives. The number of devices connected to the global network is rising, with prospects foreseeing 75 billions devices by 2025. The Internet of Things envisioned twenty years ago is now materializing at a fast pace, but this growth is not without consequence. The increasing number of devices raises the possibility of surveillance to a level never seen before.

A major step has been taken in 2018 to safeguard privacy, with the introduction of the General Data Protection Regulation (GDPR) in the European Union. It imposes obligations to data controllers on the content of information about personal data collection and processing, and on the means of communication of this information to data subjects. This information is all the more important that it is required for consent, which is one of the legal grounds to process personal data. However, the Internet of Things can pose difficulties to implement lawful information communication and consent management.

The tension between the requirements of the GDPR for information and consent and the Internet of Things cannot be easily solved. It is however possible. The goal of this thesis is to provide a solution for information communication and consent management in the Internet of Things from a technological point of view.

To do so, we introduce a generic framework for information communication and consent management in the Internet of Things. This framework is composed of a protocol to communicate and negotiate privacy policies, requirements to present information and interact with data subjects, and requirements over the provability of consent.

We support the feasibility of this generic framework with different options of implementation. The communication of information and consent through privacy policies can be implemented in two different manners: directly and indirectly. We then propose ways to implement the presentation of information and the provability of consent. A design space is also provided for systems designers, as a guide for choosing between the direct and the indirect implementations.

Finally, we present fully functioning prototypes devised to demonstrate the feasibility of the framework's implementations. We illustrate how the indirect implementation of the framework can be developed as a collaborative website named Map of Things. We then sketch the direct implementation combined with the agent presenting information to data subjects under the mobile application CoIoT.

RÉSUMÉ

Dans un monde de plus en plus connecté, Internet s'infiltre dans tous les aspects de nos vies. Le nombre d'appareils connectés au réseau mondial ne cesse d'augmenter, certaines perspectives prédisant 75 milliards d'appareils d'ici 2025. L'Internet des Objets envisagé il y a 20 ans se matérialise à une vitesse soutenue, mais cette croissance n'est pas sans conséquence. Le nombre croissant d'appareils suscite en effet des possibilités de surveillance jamais vu auparavant.

Un cap a été franchi en 2018 pour la protection de l'intimité numérique (*privacy*), avec la mise en application du Réglement Européen sur la Protection des Données (RGPD) dans l'Union Européenne. Il impose des obligations aux responsables de traitements sur le contenu de l'information à communiquer aux personnes concernées à propos de la collecte et du traitement de leurs données personnelles, ainsi que sur les moyens de communiquer cette information. Cette information est d'autant plus importante qu'elle est une condition préalable à la validité du consentement, une base légale de traitement. Cependant, l'Internet des Objets peut poser des difficultés pour mettre en place la communication de l'information nécessaire à la validité légale d'un traitement, ainsi qu'à la gestion du consentement.

La tension entre les exigences du RGPD à propos de l'information et du consentement et l'Internet des Objets n'est pas chose facile à résoudre. Ce n'est cependant pas impossible. Le but de cette thèse est de fournir une solution pour la communication de l'information et la gestion du consentement dans l'Internet des Objets.

Pour ce faire, nous proposons un cadre conceptuel générique pour la communication de l'information et la gestion du consentement dans l'Internet des Objets. Ce cadre conceptuel est composé d'un protocole de communication et de négociation des politiques de protection de la vie privée (*privacy policies*), d'exigences pour la présentation de l'information et l'interaction avec les personnes concernées, ainsi que d'exigences pour la démonstration du consentement.

Nous soutenons la faisabilité de ce cadre conceptuel générique avec différentes options de mise en oeuvre. La communication de l'information et du consentement peut être effectuée de deux manières : directement et indirectement. Nous proposons ensuite différentes manières de mettre en oeuvre la présentation de l'information et la démonstration du consentement. Un espace de conception (*design space*) est aussi proposé à destination des concepteurs de systèmes, afin d'aider à choisir entre différentes options de mise en oeuvre.

Enfin, nous proposons des prototypes fonctionnels, conçus pour démontrer la faisabilité des options de mise en oeuvre du cadre conceptuel. Nous illustrons comment la communication indirecte de l'information peut être mise en oeuvre au sein d'un site web collaboratif appelé Map of Things. Nous présentons ensuite la communication directe de l'information et du consentement combinée à un agent présentant l'information aux personnes concernées à travers une application mobile nommée CoIoT.

PUBLICATIONS

Some ideas and figures have appeared previously in the following publications:

CONFERENCE AND WORKSHOP ARTICLES

[1] Claude Castelluccia, Mathieu Cunche, Daniel Le Metayer, and Victor Morel. "Enhancing Transparency and Consent in the IoT." In: IWPE. 2018, p. 4. DOI: 10.1109/EuroSPW.2018.00023. URL: https://hal.inria.fr/hal-01709255.

[2] Mathieu Cunche, Daniel Le Métayer, and Victor Morel. "A Generic Information and Consent Framework for the IoT." In: TRUSTCOM. 2019, p. 9. URL: https://hal.inria.fr/hal-02166181.

[3] Mathieu Cunche, Daniel Le Métayer, and Victor Morel. "DEMO: CoIoT: A Consent and Information Assistant for the IoT." In: WISEC. 2020, p. 3.

[4] Victor Morel and Raúl Pardo. "SoK: Three Facets of Privacy Policies." In: *WPES*. Aug. 24, 2020. DOI: 10.1145/3411497.3420216. URL: https://hal.inria.fr/hal-02267641 (visited on 09/25/2019).

BOOK CHAPTER

[1] Silvia Giordano et al. "UPRISE-IoT: User-Centric Privacy & Security in the IoT." In: *SPIoT*. 2019, p. 17. URL: https://pdfs.semanticscholar.org/4451/8fed85347ce0a7e734d7a9c34affde8e881e.pdf.

ACKNOWLEDGEMENTS

La conduite d'une thèse de doctorat est un processus long, que je n'aurais pu mener à bien sans le support de nombreuses personnes que je souhaite remercier ici. Plutôt que dresser une liste exhaustive, j'addresserais dans cette page quelques mots à différents groupes de personnes sans forcément les nommer explicitement.

Je remercie tout d'abord mes directeurs de thèse, Claude Castelluccia et Daniel Le Métayer, pour leur soutien durant ces quatre années, leur expertise et leur pédagogie.

Je remercie les autres membres de mon jury, à savoir Gerardo Schneider, Patricia Serrano Alvarado, Hervé Rivano et Félicien Vallet, pour leurs remarques pertinentes sur mes travaux.

Je remercie tous les membres de l'équipe Privatics pour les moments passés au bureau, dans la cafétéria ou sur les pistes de ski.

Je remercie les personnes de l'antenne d'Inria, pour les repas partagés ensemble, ponctués de discussions philosophico-politiques toutes plus intéressantes les unes que les autres et conclus par la désormais traditionnelle cérémonie du café *french press*.

Je remercie mes colocataires qui m'ont quotidiennement supporté, tant dans le sens de *soutenir* que d'*endurer*, durant les phases les plus intenses de cette thèse de doctorat.

Je remercie les membres de la FELINN, notre jeune association de défense des libertés à l'heure du numérique, tant pour leurs conseils scientifiques qu'humains.

Je remercie ma famille, et particulièrement ma soeur, ma mère et mon père, pour leur confiance en moi qui m'a poussé à continuer les études malgré des passages quelque peu chaotiques.

Enfin, je remercie toutes les autres personnes que j'ai pu côtoyer d'une manière ou d'une autre, à savoir les copaines du lycée et de l'internat, des éclés, de la grimpe, de la chorale, et de mon groupe de musique, pour les bons moments passés ensemble qui m'ont permis de garder le moral.

CONTENTS

I BACKGROUND

II CONTRIBUTIONS

III CONCLUSION

IV APPENDIX

LIST OF FIGURES

LIST OF TABLES

LISTINGS

ACRONYMS

CoIoT	Consent and information in the Internet of Things
CSS	*Consent Storage Server*
DC	Data controller
DCD	Data controller device
DCG	Data controller gateway device
DCP	Data controller privacy policy
dcr	Data Communication Rule
DS	Data subject
DSD	Data subject device
DSP	Data subject privacy policy
DSG	Data subject gateway device
dur	Data Usage Rule
MoT	Map of Things
PDC	*Personal Data Custodian*
PPNP	*Privacy Policy Negotiation Protocol*
TR	Transfer Rule
UI	User Interface

CNIL	Commission Nationale Informatique et Libertés
DPA	Data Protection Agency
EU	European Union
GDPR	General Data Protection Regulation
ICANN	Internet Corporation for Assigned Names and Numbers
OECD	Organisation for Economic Co-operation and Development
WP29	Working Party 29

CMP	Consent Management Provider
CMU	Carnegie Mellon University
FPF	Future of Privacy Forum
IAB	Interactive Advertising Bureau
IRR	Internet of Things Resource Registry
PEP	Privacy Enforcement Point
PPA	Personalized Privacy Assistant

ANPR	Automatic Number Plate Recognition
CCTV	Closed-Circuit Television
ICT	Information and communications technology
IoT	Internet of Things
VANET	Vehicular Adhoc Network
WHAN	Wireless Home Automation Network
WSN	Wireless Sensors Network

AP	Advertising Packets
API	Application Programming Interface
ATT	Attribute Protocol
BLE	Bluetooth Low Energy
CHF	Cryptographic Hash Function
GATT	Generic Attribute Profile
HTTP	Hypertext Transfer Protocol
IE	Information Element
IP	Internet Protocol
JSON	JavaScript Object Notation
MAC	Media Access Control
MHT	Merkle Hash Tree
P2P	Peer-to-Peer
REST	Representational State Transfer
XML	Extensible Markup Language

Part I

BACKGROUND

Arguing that you don't care about the right to privacy because you have nothing to hide is no different than saying you don't care about free speech because you have nothing to say.

Edward Snowden

1 INTRODUCTION

In an increasingly connected world, the Internet permeates every aspect of our lives. We are now uninterruptedly accompanied by our smartphone, our homes are equipped with smart thermostats, and our streets with smart sensors. The number of devices connected to the global network is rising, with prospects foreseeing 75 billions devices by 2025. The Internet of Things envisioned twenty years ago is now materializing at a fast pace.

This tantalising trend to digitalise the world has advantages: public transportation can be optimized by analysing attendance, home temperature can be set remotely, and we can access numerous services from devices fitting in our pocket. But the growth of the Internet of Things is not without consequence. The increasing number of devices raises the possibility of surveillance to a level never seen before. It is indeed child's play to track bystanders through their smartphones, and aggressive targeted advertising may become the norm if privacy is not seriously considered.

A major step has been taken in 2018 to safeguard privacy, with the introduction of the General Data Protection Regulation (GDPR) in the European Union (EU). This text now regulates under what conditions personal data* [1] can be collected and processed. The GDPR puts strong requirements on data controllers*, and enhances rights of data subjects*. However, this text can be difficult to interpret and to implement, and could be toothless if not complemented with appropriate technology. For instance, the GDPR emphasizes requirements over the information about personal data collection and processing to be communicated to data subjects before the collection of such data. [2] It imposes obligations on the content of information, and on the means of communication of this information. [3] This information is all the more important that it is required for consent, which is one of the legal grounds* to process personal data. Consent must be freely given, without ambiguity, and must be revocable, among other things. [4]

The Internet of Things can pose difficulties to implement lawful information communication and consent management. Internet of Things devices are indeed ubiquitous, *i.e.*, they physically surround

[1] Words marked with an asterisk are described in the Glossary, see Appendix A.

[2] In the following, any mention of *information* refers more specifically to the information about personal data collection and processing to be communicated to data subjects unless stated otherwise.

[3] The content and the means of communication of this information is described in Section 1.1.2.2.

[4] Section 1.1.2.3 describes the requirements of the GDPR on consent with more depth.

data subjects; various, *i.e.*, they collect different types of data or use different communication protocols; are often endowed with low computational power; and rarely possess appropriate interfaces to present information. Information is traditionally conveyed as privacy policies* — long and verbose texts difficult to understand for lay-users. This state of affairs is considered unacceptable by some scholars [102], and the application of such a mechanism of "notice and choice" [35] is not more appropriate in the Internet of Things than it is on the Web. It is questionable whether having to read and consent to the privacy policy of every Internet of Things device encountered is realistic and acceptable in order to use the service associated.

The tension between the requirements of the GDPR for information and consent and the Internet of Things cannot be easily solved. It is however possible. The goal of this thesis is to provide a solution for information communication and consent management in the Internet of Things from a technological point of view. More specifically, it strives to answer the following question: *Is it possible to intelligibly inform data subjects about personal data collection and processing in the Internet of Things, and to manage consent in a privacy-preserving way, while facilitating lawfulness for data controllers?* To introduce the topic, we first present the context in Section 1.1, with the Internet of Things, then elements of legal background, and finally the issues arising from the tension between the Internet of Things and legal requirements. In Section 1.2, we finally expose the objectives of this thesis to address the issues presented at the end of Section 1.1.

1.1 CONTEXT

In order to grasp the importance to enhance information and consent in the Internet of Things, this section first presents the Internet of Things in Section 1.1.1, including the threats it poses to privacy. We then present a brief overview of the GDPR, the legal framework which provides a context to our contributions, in Section 1.1.2. Finally, we present the issues to be addressed to enhance information and consent in the Internet of Things in Section 1.1.3.

1.1.1 *Internet of Things*

The Internet of Things is not a well-defined notion. However, it is possible to adopt the following definition, derived from its designation: *the Internet of Things is the network encompassing any digital device whose interface allows for an Internet connection.*

Its origin is to be found in Ubiquitous Computing, term coined in 1991 by Weiser [156]. We shortly describe three common areas of application of the Internet of Things: wearables in Section 1.1.1.1, smart homes in Section 1.1.1.2, and smart cities in Section 1.1.1.3.

Appendix B provides a more exhaustive overview of the Internet of Things.

1.1.1.1 *Wearables*

Some Internet of Things devices can be worn by data subjects, and are therefore denoted *wearables*. Tehrani & Andrew define wearables as "*electronic technologies or computers that are incorporated into items of clothing and accessories which can comfortably be worn on the body*"[140]. This incorporation of devices into the outfit invisible them. For instance, glasses can record audio and video without bystanders noticing it [77]. Smartwatches have the possibility to retrieve the pulse rate, for which they are designed most of the time, and can also disclose the activity of the holder: whether she is running, walking, or sitting for example [71].

1.1.1.2 *Smart homes*

The smart home, also known as Wireless home automation networks (WHANs) [59], is the enhancement of the home with connected devices in order to facilitate the life of data subjects. Among all devices, smart speakers embedded with virtual assistants made a notable breakthrough. Major tech companies launched their own smart speaker. Smart speakers continuously listen, and react to a wake word which then triggers voice control. Data subjects command built-in virtual assistants, which interact with other smart devices in the home. The smart home is entangled with wearables which react and communicate with the rest of the infrastructure. All these devices communicate with various technologies, such as ZigBee, 6LoWPan, or Bluetooth when wearables are considered.

1.1.1.3 *Smart city*

The third and last part of the Internet of Things presented in this section is smart cities. Smart cities encompass many devices and use cases. It is for instance common to find Wireless Sensor Networks [21] (WSN) to measure air quality. Air pollution is considered a major issue in big cities, and such WSN can help to monitor pollutant emissions. Data about air quality is not personal, and does not threaten privacy. However, other data sources are less innocuous, such as smart grids [41] and smart meters (see Appendix B.3 (under the term *smart energy*)). Smart meters can reveal the devices used, and even the activities performed — such as whether an occupant is cooking, sleeping, or using a laptop — if the collection granularity is fine enough. They can be combined in a smart grid, which can provide "the opportunity to reduce energy consumption enabled by detailed, personalized energy consumption reports and consequent monetary savings" [41]. Cameras have spread out in the recent years, and they are now connected to Internet. They allow for a real-time and continuous monitoring, not to mention the improvement made in facial recognition [30, 39]. At the heart of a city resides its different means of transportation. It is no surprise to find smart mobility as a component of a smart city [18]. In Barcelona for instance, the combination of a smart bus network designed according to a data analysis of traffic flow with smart traffic lights optimize the overall transportation system [1]. In addition, the traffic lights are set to free the road for emergency vehicles in case of a fire or an accident. Tomorrow's smart mobility may very well be composed of Vehicular Ad-hoc Networks (VANETs). Connected vehicles form a VANET, and are able to communicate together, and with the roadside [161]. This communication can prevent accidents by warning equipped vehicles in the vicinity, or can improve traffic efficiency.

Many self-claimed smart cities chose to put some data retrieved from these sensors online and publicly available, when this data is not personal. This information is labelled Open Data [15]. Open Data denotes the set of public data opened to the public for scrutiny and research [69]. Because of the public nature of Open Data, the data available must not be personal, and common examples include transportation network information, locations of bicycle-sharing docks and their availability, Wi-Fi hotspots, air quality information, and data unrelated to the smart city aspect such as trees locations or city budgets [152] *etc.*

Summary

Capturing the different applications of the Internet of Things is difficult, as the term encompasses various types of devices, types of uses, and types of data collected. Moreover, these devices communicate with numerous protocols, and their sole common point may be their connection to Internet. The Internet of Things differs from ubicomp in that ubicomp is an historic term, rooted in academia; while the Internet of Things is more of a current and commercial designation, although now seized by academia. However, it does not mean than the Internet of Things does not have a physical existence. Quite the contrary: the Internet of Things is part of our selves, homes, and cities, and is not less threatening to privacy than ubicomp.

1.1.2 *Legal background*

In Europe the deployment of IoT systems must comply with different legal frameworks, the most recent of which is the General Data Protection Regulation (GDPR), enacted in May 2018 in the European Union. It has an extraterritorial scope, which makes it important for data controllers to consider it, regardless of their location. Amongst other things, the GDPR requires specific modalities of information, and of consent retrieval. Section 1.1.3 lists the issues arising from the tension between the GDPR and the Internet of Things, but we first introduce the legal framework and its main principles in Section 1.1.2.1. Sections 1.1.2.2 and 1.1.2.3 then respectively describe the requirements over information and consent, as well as clarifications of the WP29. [5]

1.1.2.1 *GDPR*

The Regulation 2016/679 of the European Parliament and of the council 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, often shortened in GDPR) was adopted on the 27th of April 2016 and became enforceable on the 25th of May 2018. With the growth of Internet use, the rise of the Internet of Things, and the risks posed by the application of machine-learning techniques to ever-growing personal data, the previous Directive needed an update. The GDPR is now the main text regulating personal data collection and processing in the EU, and many non-european countries consider it as well since it has an extraterritorial scope. Whereas the Directive had implicit principles (*e.g.*, proportionality and transparency, which do not appear as is in the text), the GDPR provides its principles explicitly in Article 5. Personal data shall be processed *lawfully*, *fairly*, and *transparently*; the purpose of collection shall be specified, explicit and legitimate (*purpose limitation*); only required data shall be collected (*data minimization*); data shall be *accurate*; its storage has to be *limited*; and personal data must be stored securely (*integrity and confidentiality*). On top of this bundle of principles, the data controller must be able to demonstrate compliance with these principles (*accountability*).

The regulation also emphasizes the implementation of data protection by design and by default in Article 25 (often denoted as *privacy by design* and *privacy by default*). Implementing privacy by design means that the data controller must consider "the state of the art, the cost of implementation and the nature, scope, context and purposes of processing [...] both at the time of the determination of the means for processing and at the time of the processing itself". Implementing privacy by default means that the data controller "shall implement appropriate technical and organisational measures for ensuring that, by default, only personal data which are *necessary* for each specific purpose of the processing are processed". Finally, Article 35 requires data protection impact assessments in certain situations. Such assessment must prove the proportionality and necessity of processing, in line with the stated purposes. The GDPR provides Data Protection Agen-

[5] WP29 stands for Working Party 29, a European advisory board, now European Data Protection Board (EDPB)

cies (DPAs) with a strong economic leverage to enforce its obligations: DPAs can impose fines up to 4% of the annual turnover. [6]

1.1.2.2 *Information under the GDPR*

The GDPR requires data controllers to inform data subjects when they collect and process personal data, regardless of the legal ground of processing. It specifies the content of that information, as well as the means to communicate this information.

Art. 13 and 14 specify the content to provide when informing:

- the identity of the data controller and its contact
- the type of data collected
- its purpose
- the legal ground
- the recipient of data
- the 3rd parties involved
- the retention time
- the rights of the data subject

Art. 12 requires that information must be *inter alia*: concise, transparent, intelligible and easily accessible; expressed in clear and plain language; where appropriate, communicated by electronic means.

The communication of information participates in the obligation of transparency specified by the GDPR. The WP29 has detailed in guidelines [155] its interpretation of transparency. For instance, it clarifies the notion of the "appropriate measures" with which data controllers must inform. Data controllers have to consider the timing of information: before the collection when data is collected directly from the data subject; the modalities of information: data controllers must take active steps to inform data subjects and not the contrary. The WP29 also recommends the use of layered notices to avoid information fatigue: it recommends information to be presented in consistent layers, whose first one must provide the most impactful information. It encourages icons for Internet of Things environments, as a complement with "information necessary for the exercise of a data subject' rights". These icons must be standardized, and machine-readable where presented electronically.

1.1.2.3 *Consent under the GDPR*

Consent is one of the six legal grounds to process personal data under the GDPR. A consent is considered as valid under the GDPR (Art. 4(11)) if it fulfils the following conditions:

- It has to be freely given
- It has to be specific

[6] See the 4 million euros penalty of the CNIL against Google LLC [27].

- It has to be informed
- It has to be unambiguous

In addition to these conditions, the data controller must be able to demonstrate consent (Art. 7(1)), and the data controller must ensure that consent can be withdrawn (Art. 7(3)). The Working Party 29 has detailed in guidelines [154] its interpretation of this definition. We summarise their interpretation below:

FREE The free character of consent implies that the data subject has a real choice and control [84] over the decision. Imbalance of power has to be taken into account, and the service provided must not be degraded if consent is not given. Bundling unnecessary consent with the acceptance of a required contract is not acceptable: a choice must always be offered (the WP29 denotes it "conditionality"). Free also means granular, and data subjects should be free to choose the purposes to which they consent. It must be possible to refuse consent without detriment.

SPECIFIC Similar to the notion of granularity, consent has to be specific to each purpose of processing.

INFORMED Information must be provided to data subjects prior to consent. We refer to Section 1.1.2.2 for the information required and its modality of communication.

UNAMBIGUOUS Last, consent has to be given unambiguously by a "statement from the data subject or a clear affirmative act". The WP29 specifies that pursuing the browsing of a website cannot constitute consent, and so are opt-out facilities — *i.e.*, solutions considering that data subjects accept conditions by default: data subjects should explicitly mention their disagreement, further discussed in Section 2.2.2.

1.1.3 *Issues related to information and consent*

We saw in Section 1.1.1 that the Internet of Things encompasses various application domains — *i.e.*, wearables, smart homes, and smart cities. This variety of contexts makes arduous to retrieve **information** related to personal data collection and processing. The invisibility of devices combined with their proliferation exacerbates the difficulty of being informed. Weiser wrote about this invisibility in his well-known article "The Computer for the 21st Century" [156]: *The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it.*

Information is all the more important that **consent** heavily relies on it. Data collection and processing must rely on a legal ground to be lawful, and a common legal ground is consent (see Section 1.1.2.3). Interactions are necessary to communicate consent, but it can be difficult with devices in the Internet of Things.

The most hurdling characteristics of Internet of Things devices for information and consent can be grouped into four features: ubiquity, variety, low computational power, and inappropriate interface.

UBIQUITY The ubiquity of Internet of Things devices means they have pervaded our everyday lives: Internet of Things devices are everywhere.

VARIETY Ubiquity often goes with variety. A smartwatch does not function as a smart thermostat, does not collect the same type of data, nor communicate using the same protocols.

LOW COMPUTATIONAL POWER Numerous devices do not have high computational power. It is notably the case for WSNs. Such devices are fit for a limited number of tasks, which often amounts to collecting data and forwarding it to a central node (named the *sink*). Some devices literally lack the communication means to interact with data subjects, as they were only devised to listen surrounding wireless communications: they are *passive*.

INAPPROPRIATE INTERFACE Correlated with their small size and due to the fact that they are battery-powered, most Internet of Things devices are not endowed with user interfaces. When they are, it is often inappropriate interfaces to convey consent.

Information communication and consent management are far from being implemented by design in Internet of Things environments, which causes threat for privacy. We describe the issues pertaining to information in Section 1.1.3.1, and pertaining to consent in Section 1.1.3.2.

1.1.3.1 *Information issues*

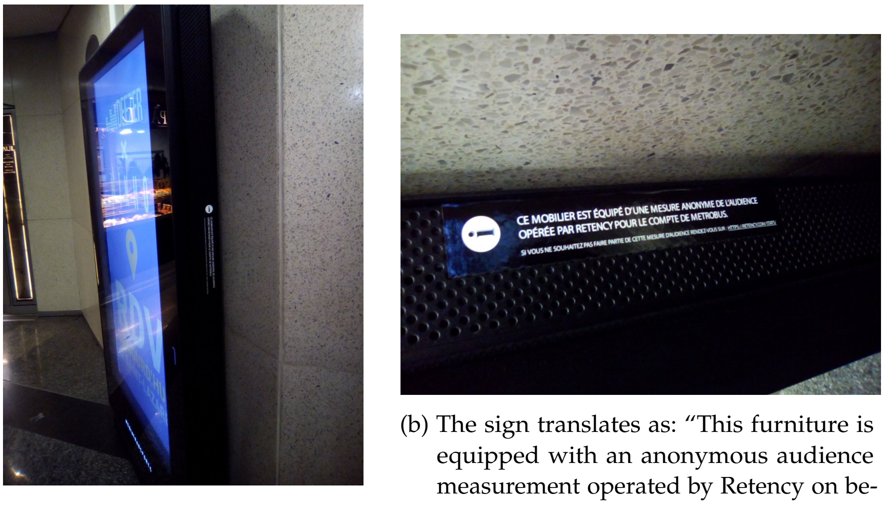
Data subjects are generally unaware of devices, and of what data is collected, for which purposes of processing.

Their **low computational power** and **inappropriate interface** make it difficult for devices to declare themselves, *i.e.*, to announce their presence, and to inform of their data practices. It contrasts with the web where websites can provide their own privacy policy. Devices in the Internet of Things are often limited by their capacities, *i.e.*, they cannot always communicate information electronically nor through means of display, resulting in ineffective declaration (as described below) or no declaration at all.

Because of their **variety**, it is difficult to design a single solution which would permit the communication with all Internet of Things devices. Section 1.1.1 presented the variety of Internet of Things situations, involving different types of devices. These different types of devices do not always function according to the same communication protocols: they are not always equipped to communicate with data subjects' devices.

The **absence or the inappropriateness of interfaces** results in the difficulty to intelligibly present information related to data collection and processing to data subjects. The web is all about interactions with users: either on the screen of a desktop computer or on a smartphone, the web is meant to be *seen*; whereas the Internet of Things is meant to be *invisible*. Moreover, the intelligibility of the information conveyed in privacy policies is questionable. Privacy policies in their current format are hard to understand [31], for lay users [102] as for experts [128]. Presenting verbose and ambiguous information is not acceptable to inform data subjects. A last obstacle for information is therefore the actual presentation of information.

As a consequence, the current situation is made of wall signs or stickers, which are ineffective information means in most situations. For example, tracking panels were installed in the Parisian subway in early 2019, with a small sticker (a few centimetres large) placed vertically on the side of the panels [89] as the only mean of information (see Figure 1). Data collection and processing in the Internet of Things happens unbeknownst to data subjects, and it is uncertain that the few means of information deployed actually convey this information to end-users, all the more intelligibly.


CE MOBILIER EST ÉQUIPÉ D'UNE MESURE ANONYME DE L'AUDIENCE
OPÉRÉE PAR RETENCY POUR LE COMPTE DE METROBUS.
SI VOUS NE SOUHAITEZ PAS FAIRE PARTIE DE CETTE MESURE D'AUDIENCE RENDEZ-VOUS SUR : HTTPS://RETENCY.COM/STATS/

(a) The sticker is placed vertically at 90° on the side of the panel.

(b) The sign translates as: "This furniture is equipped with an anonymous audience measurement operated by Retency on behalf of Metrobus."

Figure 1: Example of an ineffective information in the Internet of Things.

1.1.3.2 *Consent issues*

Even when they are aware of data collection and processing surrounding them, data subjects have difficulties with the communication of their choices in terms of personal data collection and processing.

Data subjects do not necessarily have the means to express their choices in the first place, for instance whether they agree to a specific type of data to be collected, and processed for certain purposes *etc.* This is partly due to the **ubiquity** of Internet of Things devices: there is no easily accessible point of access, which results in the difficulty of expressing a choice. [7] Moreover, when a choice is available, it is often a binary one.

Symmetrically to the reception of information, the communication of choices is problematic due to the **variety** of devices. This may also be due to the passivity of devices, to the extent that a data subject cannot establish two-way communications with passive devices.

As far as consent is concerned, the current situation is far from acceptable. It is common to find systems considering opt-out as consent, and even when opt-in is considered, it rarely goes beyond a yes-or-no choice. To illustrate, some Wi-Fi hotspots consider that accepting the terms of use and the privacy policy — required to access Wi-Fi — also provides consent from data subjects to be geolocalised for marketing purposes [58]. This arguably goes against the *free* requirement of consent described in Section 1.1.2.3. It is then possible to opt-out from this processing by sending by email the device's MAC address. [8] This situation is not convenient for data subjects, whose data is collected without proper consent; but it has drawbacks for data controllers too, who risk considerable sanctions if they cannot demonstrate consent.

Summary

Amongst information and consent issues, the Internet of Things appears to be a perilous terrain for privacy. It is difficult to inform data subjects about data practices due to the variety, ubiquity, and lack of appropriate interface; and managing consent — *i.e.*, expressing and communicating choices of data subjects so that data controllers can demonstrate consent — is also challenging because of the variety and the low computational power of Internet of Things devices. The issues presented can be put in the form of a question as follows: *Is it possible to intelligibly inform data subjects about personal data collection and processing in the Internet of Things, and to manage consent in a privacy-preserving way, while facilitating lawfulness for data controllers?*

[7] This issue is however not a specificity of the Internet of Things, as it can happen on the Web.

[8] The MAC address is a unique identifier of a digital device.

1.2 OBJECTIVES

The Internet of Things raises numerous privacy concerns, and is especially challenging when it comes to information communication and consent management. And while we can rely on data protection laws to the extent of their application, they must be complemented with the appropriate technology not to be toothless.

The current situation is not acceptable: posters and wall signs cannot guarantee the reception of information by data subjects, and requesting consent from data subjects for every device encountered results in user fatigue.

To tackle the issues presented in Section 1.1.3, we present the goal set for this thesis. We presented it in the beginning of the document as *providing a solution for information communication and consent management in the Internet of Things from a technological point of view*. With the elements brought in by the analysis of the specific issues of information and consent in the Internet of Things, it can now be more specific: the goal is to design *a generic framework to communicate information and manage consent in the Internet of Things*.

The global approach uses privacy policies to communicate information and to manage consent. Data controllers privacy policies are a commitment to conditions of processing. Data subjects privacy policies refer to the choices of data subjects regarding their personal data. Both types of privacy policies are machine-readable, and as a matter of fact, the entire framework relies on electronic setups: data subjects must possess an electronic device, and data controllers must declare their privacy policies with electronic devices. More details and conditions over these privacy policies and devices will be given in the following chapters.

We now detail the objectives such a framework must satisfy in order to communicate information and manage consent. These objectives can be presented according to the two notions discussed so far: information and consent. They echo the issues aforementioned.

1.2.1 *Objectives with respect to information*

As far as information is concerned, an Internet of Things system collecting personal data must satisfy the following objectives:

Objective 1 (Systematic declaration). *Data controllers must declare all devices collecting personal data, with all the information required by the GDPR. In addition to legal requirements, this declaration must be automatic, and communicated by electronic means. A systematic declaration is required to automatize information and to enable its processing by data subjects' electronic devices.*

Objective 2 (Reception of information). *The information declared must be received by the device of any data subject about whom personal data can be collected. Similar to the declaration of information, information must be received electronically. Ensuring the reception of information electronically is complementary with the declaration of this information.*

Objective 3 (Intelligible presentation to data subjects). *The information received must be presented to data subjects in forms and at times that should minimize information fatigue and maximize the likelihood that data subjects will not miss any useful information. Misleading, repetitive, or truncated information is not acceptable, and the information presented must be carefully designed.*

The objectives of systematic declaration, of reception of information, and of intelligible presentation tackles the lack of appropriate information in the Internet of Things. They must be addressed with machine-readable and user-friendly privacy policies. These privacy policies must be received and presented on data subjects devices.

1.2.2 *Objectives with respect to consent*

As far as consent is concerned, an Internet of Things system collecting personal data must provide the following facilities:

Objective 4 (Expression of choices). *Data subjects must be able to express their choices with respect to personal data collection and processing, in forms and at times that should minimize their fatigue and maximize the likelihood that they make appropriate decisions regarding the protection of their personal data. It must be possible for data subjects to reflect on their choices at a different time than the collection of data. The expression of choices must encompass the expression of consent in the sense of the GDPR, and its withdrawal.*

Objective 5 (Communication of consent). *Consent must be received by any data controller able to collect personal data. Data controllers must not collect the data (or must immediately delete it) if this consent is not consistent with their data controller policy, i.e., of this consent does not respect data subjects' choices. Consent must be communicated electronically, and seamlessly if the data subject has expressed choices consistent with the data controller policy.*

Objective 6 (Demonstration of consent). *Data controllers must have the possibility to store the consents obtained from data subjects so they can demonstrate GDPR compliance regarding consent, in particular that it has been provided by the data subject on whom data is held. The demonstration of consent must encompass the identity of the data subject, the conditions of processing, and its lawfulness: i.e., its free, specific, informed, and unambiguous character.*

The objectives of expression of choices, of communication of consent, and of demonstration of consent tackles the difficulty of consent management in the Internet of Things. They must be addressed with machine-readable privacy policies set by data subjects on their device. In particular, consent must be demonstrated with the help of electronic facilities.

1.3 THESIS OUTLINE

Chapter 1 introduced the Internet of Things; the legal requirements in terms of information about personal data collection and processing and consent; and the problems resulting from the combination of the Internet of Things and these requirements. The chapter concludes on the objectives set to address these challenges.

Chapter 2 will provide an overview of the state of the art related to information and consent in the Internet of Things. The section related to information will successively present manners to communicate information aimed for humans in a user-friendly way, then for machines in a machine-readable format. The section related to consent will detail existing solutions on privacy assistants for the Internet of Things, opt-out facilities, cookie management systems, and solutions for the secure storing of consent.

We will pursue in Chapter 3 with the main contribution of this thesis: a generic framework for information communication and consent management in the Internet of Things. The main idea behind this framework is its generic character. In addition to a set of assumptions about its scope of application, this framework is composed of: a protocol to negotiate privacy policies, that we denote *PPNP* for *Privacy Policies Negotiation Protocol*; requirements to present information and interact with data subjects as a *PDC* (for *Personal Data Custodian*); and high-level requirements over the provability of consent, necessary for its demonstration.

In order to prove the feasibility of the framework, we will present in Chapter 4 different options of implementation. The communication of information and consent through privacy policies can be implemented in two different manners: directly and indirectly. We will then continue on ways to implement the *PDC* and the *ledger*. A design space is then provided for systems designers, as a guide for choosing between the direct and the indirect implementations. The chapter concludes with illustrating scenarios.

Chapter 5 will present the prototypes devised to demonstrate the feasibility of the framework's implementations. A first section will illustrate how the indirect implementation of the framework can be developed as a collaborative website named Map of Things. A second section will sketch the direct implementation combined with the *PDC* under the mobile application CoIoT.

Chapter 6 will close the document with a summary of contributions, discussions about design choice and designers' responsibilities, limitations of the work presented, and avenues for future research.

2 STATE OF THE ART

This chapter provides a state of the art of the solutions to inform about personal data collection and manage consent in the Internet of Things. The work presented in this chapter relates to the objectives enumerated in Section 1.2. We present solutions to inform data subjects in Section 2.1. An overview of solutions to manage consent is then provided in Section 2.2.

Only a handful of solutions exists and is presented in this chapter. It denotes both the challenge to inform and to manage consent in the Internet of Things, as well as the need to tackle the issues presented in Section 1.1.3.

2.1 INFORMATION

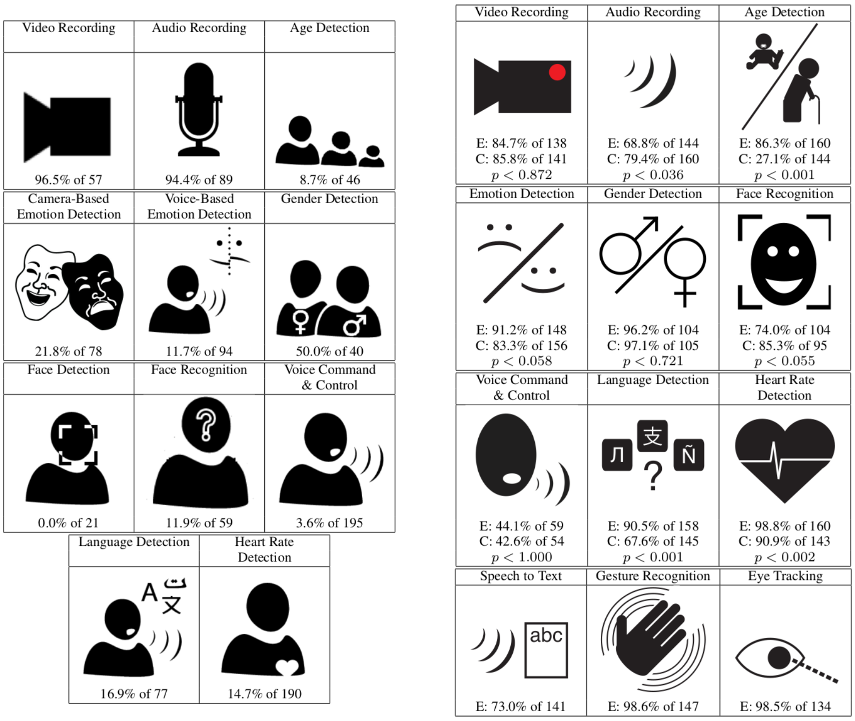
The objectives presented in Section 1.2 require information to be systematic (Objectives 1 and 2), as well as intelligibly presented to data subjects (Objective 3). A number of solutions focus on conveying intelligible information in the Internet of Things, as sets of icons, standardized notices [35], or as a risk analysis tool, and we present them in Section 2.1.1. Other proposals are motivated by the systematization of information in the Internet of Things: privacy languages, and they are described in Section 2.1.2. The latter can also participate in fulfilling Objective 4, to the extent that choices expressed by data subjects have to be in a format understandable by machines.

These two types of solutions to communicate information — respectively user-friendly and machine-readable — can be seen as means to express privacy policies. The first type conveys what we denote **graphical privacy policies**, while the second type conveys what we denote **machine-readable privacy policies**.

2.1.1 *User-friendly information*

A limited number of solutions provide user-friendly information for Internet of Things environments. Egelman *et al.* [49] developed a set of icons for the Internet of Things, later refined with crowdsourcing. The final set of icons (see Figure 2b) focuses only on the type of data collected — voice, gesture, image — and its purposes — detection of gender, emotion, language.

A notable example of privacy icons for mobile phones are the android permissions [60], created by Google. They present icons combined with simple natural language (see Figure 3). For each application



(a) “Is this thing on?” initial icons

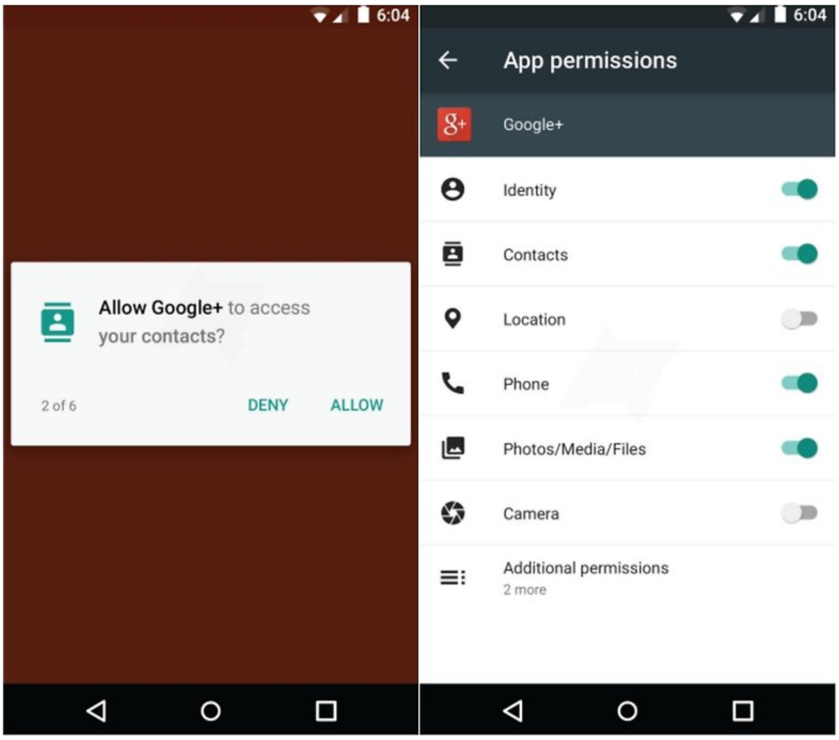
(b) “Is this thing on?” final icons

Figure 2: “Is this thing on?” icons

installed on a mobile phone running Android, the permission manager presents a short graphical policy. Only limited information is presented (the type of data collected, and processing in recent versions, but not the purpose for instance), and data subjects have to look into the application’s natural language privacy policies in order to find more information. This proposition is presented in a standardized manner, and heavily uses icons.

The only example of genuine standardized notice is the work of Emami-Naeini *et al.* [50], who conducted a survey in order to rank the factors of Internet of Things devices purchase. They determined that security and privacy were among the most important factors of purchase, and consequently developed an Internet of Things privacy label to improve information visualization (see Figure 4).

In [124], Pardo & Le Métayer present a web interface to inform data subjects about the potential risks of their privacy policies. The interface is composed of a user-friendly form for data subjects to input their privacy policies, and a set of risk analysis questions, *e.g.*, “Can company X collect my data?” (see Figure 5). Data subjects simply need to click on “Analyze” to automatically obtain the answer to the questions. Additionally, data subjects may introduce risk assumptions in order to specify possible misbehaviours that malicious entities could perform.



6:04

Allow Google+ to access your contacts?

2 of 6

DENY ALLOW

6:04

App permissions

Google+

Identity ●

Contacts ●

Location ○

Phone ●

Photos/Media/Files ●

Camera ○

Additional permissions
2 more

Figure 3: Android permissions

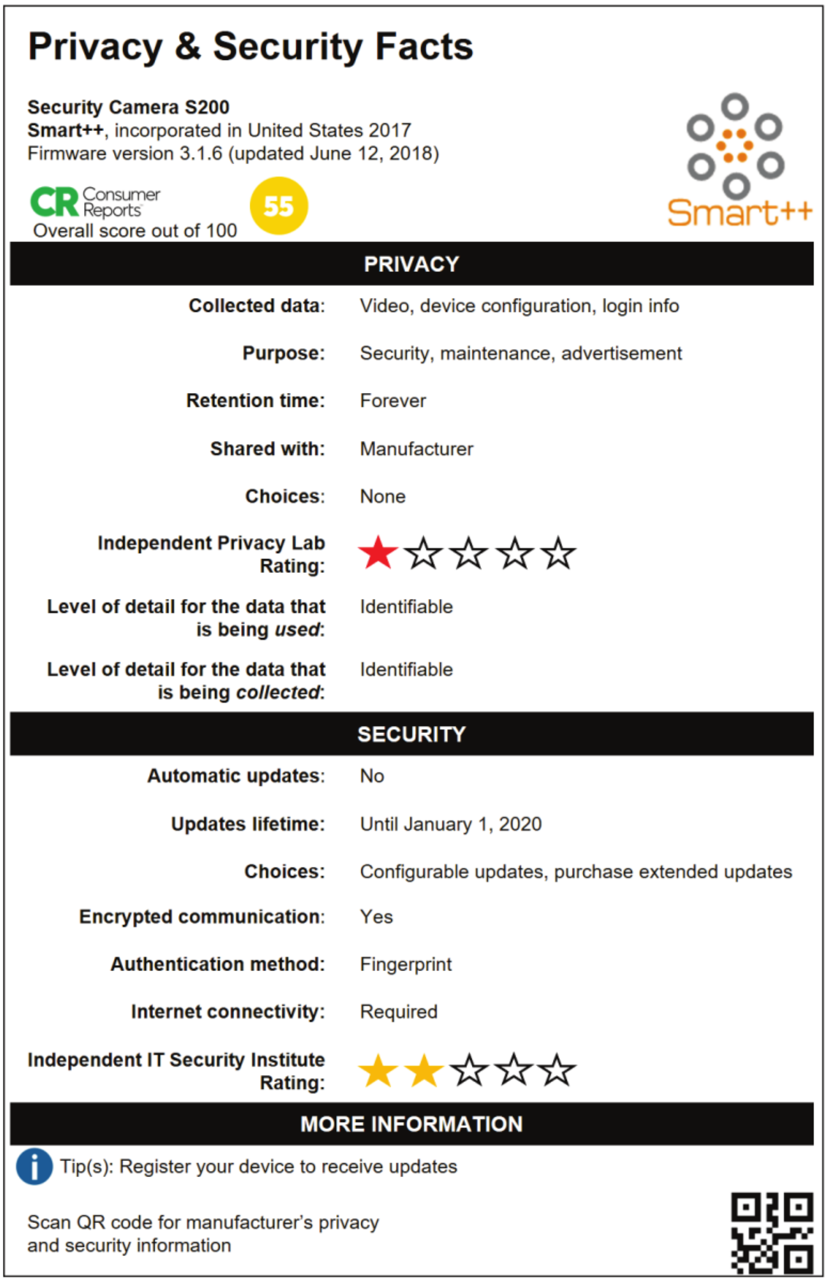
2.1.1.1 *Benefits and limitations*

Graphical privacy policies have a notable benefit: to foster understanding of lay-users. This is required to fulfil Objective 3. But they are however not exempt of limitations, as they are often ambiguous, and can rarely convey all the information required by law.

BENEFITS Many solutions aim to provide intelligible information to lay-users. Attempts were made to analyse what icons were recognisable and to measure their reliability. Egelman *et al.* [49] crowdsourced privacy indicators for the Internet of Things. In their study, they found out that some icons are well-recognized (for example, the camera symbol was recognized by more than 95% of participants as representing *video recording*), while others do not (only 3.6% recognized the *voice command & control* icon).

These solutions often pertain to "Legal Design". Legal design can be defined as "[the] application of design-thinking (processes by which design concepts are developed by designers) principles to the practice of law, to make legal systems, products, services and processes more useful, useable, understandable and engaging for all" [133].

LIMITATIONS Though accessible to lay-users, graphical privacy policies may be interpreted in different ways, thus leading to ambiguities. The same icon can be interpreted in different ways according to the differences in culture, education level, or context *etc.* For instance, a



Privacy & Security Facts

Security Camera S200
Smart++, incorporated in United States 2017
Firmware version 3.1.6 (updated June 12, 2018)

CR Consumer Reports 55
Overall score out of 100

Smart++

PRIVACY

Collected data:	Video, device configuration, login info
Purpose:	Security, maintenance, advertisement
Retention time:	Forever
Shared with:	Manufacturer
Choices:	None
Independent Privacy Lab Rating:	★☆☆☆☆
Level of detail for the data that is being *used*:	Identifiable
Level of detail for the data that is being *collected*:	Identifiable

SECURITY

Automatic updates:	No
Updates lifetime:	Until January 1, 2020
Choices:	Configurable updates, purchase extended updates
Encrypted communication:	Yes
Authentication method:	Fingerprint
Internet connectivity:	Required
Independent IT Security Institute Rating:	★★☆☆☆

MORE INFORMATION

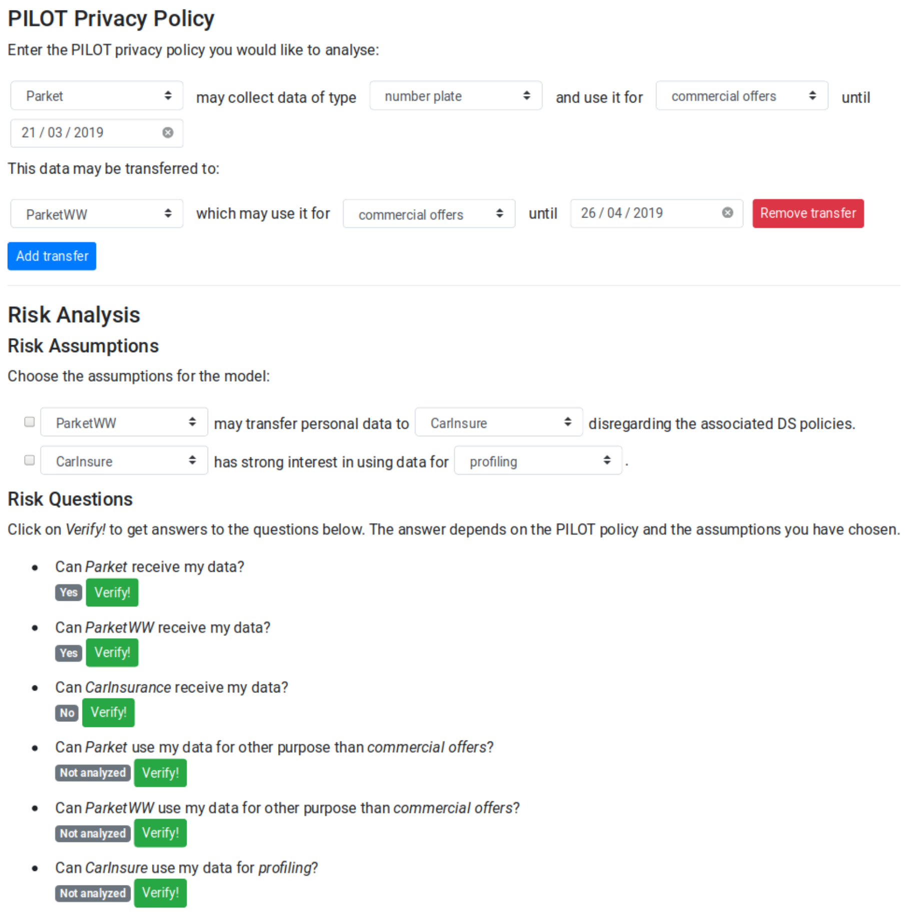
Tip(s): Register your device to receive updates

Scan QR code for manufacturer's privacy and security information

Figure 4: Prototype IoT label

euro symbol € can represent the commercial use of collected data, or that data subjects will be paid for having their data collected. Nothing has been done to produce a reasonably recognized set of privacy icons for the Internet of Things— *e.g.*, validated by a user study — despite the attempts of [49] to measure icons recognition.

Graphical privacy policies are also limited by their restricted scope. Existing graphical privacy policies are not very expressive, due to the limited number of icons available. Some aspects required by law, and



PILOT Privacy Policy

Enter the PILOT privacy policy you would like to analyse:

Parket may collect data of type number plate and use it for commercial offers until 21 / 03 / 2019

This data may be transferred to:

ParketWW which may use it for commercial offers until 26 / 04 / 2019

Risk Analysis

Risk Assumptions

Choose the assumptions for the model:

□ ParketWW may transfer personal data to CarInsure disregarding the associated DS policies.

□ CarInsure has strong interest in using data for profiling .

Risk Questions

Click on *Verify!* to get answers to the questions below. The answer depends on the PILOT policy and the assumptions you have chosen.

- Can *Parket* receive my data?
  Yes
- Can *ParketWW* receive my data?
  Yes
- Can *CarInsurance* receive my data?
  No
- Can *Parket* use my data for other purpose than *commercial offers*?
  Not analyzed
- Can *ParketWW* use my data for other purpose than *commercial offers*?
  Not analyzed
- Can *CarInsure* use my data for *profiling*?
  Not analyzed

Figure 5: Input Forms of Risk Analysis Web Application.

notably by the GDPR, are rarely mentioned, others only in complementary text and not in the graphical part of the policy.

2.1.2 *Machine-readable information*

Few solutions have been developed to communicate information in the Internet of Things systematically. These solutions often take the form of *machine-readable privacy policies* — *i.e.*, privacy policies that can be automatically processed by computers. Most of these solutions were devised by academics, and result in what has been called *privacy languages*. According to Kasem *et al.* [72], a privacy language is "a set of syntax and semantics that is used to express policies". Many privacy languages have been proposed in the past twenty years (cf. [72, 148]), but most are not tailored to the Internet of Things.

A pioneer project in this area was the "Platform for Privacy Preferences" (P3P) [36]. P3P was conceived as a policy language for websites. It allows clients to declare their privacy preferences, and online service providers (mostly websites) to inform how they use customers' data. P3P policies are specified in XML format, and include notions such as purpose, retention time and *conditions*. Conditions may be opt-in or opt-out choices for data subjects, or preferences based on data subjects' credit or service usage. Many extensions to P3P have been proposed [5, 11, 81], where its syntax has been extended — for instance, E-P3P [11] extends P3P's syntax with obligations. P3P has notably been used by Langheinrich [80] to systematically inform users in ubiquitous environments.

Another line of work is that of formal privacy languages (*formal languages* in the sequel). Formal languages are languages which have their syntax and semantics defined by means of mathematical definitions. More precisely, they use formal languages such as *Linear Temporal Logic* [65], *First-Order Logic* [65] or *Authorization Logic* [2]. The only example of formal privacy language for the Internet of Things is Pilot [123]. Pilot makes it possible to include spatio-temporal conditions which allow data subjects and data controllers to describe when, where, and by which devices data may be collected. Formal languages give meaning to their privacy policies by means of *formal semantics*. Typically, these semantics define what events may be executed depending on the privacy policies selected by the actors interacting in the system, *e.g.*, data subjects and data controllers.

For some languages, algorithms have been devised to automatically compare policies. The goal is to determine, given two policies, which one is more restrictive. For example, a policy that allows data processing for research purposes during 7 days is more restrictive than a policy that allows data processing for advertisement *and* research during 90 days. Comparison is necessary to make it possible to mechanize consent. If the policy of a data controller is more restrictive than that of a data subject, then the data subject privacy preferences are satisfied. This step, although insufficient, is necessary for consent to be legally valid (see Section 1.1.2.3).

Formal languages can also come with tools to perform different types of automatic analyses. Pilot uses model-checking [14] to perform risk analysis. Given a data subject privacy policy, and a set of risk assumptions such as "Company X may transfer data to Company Y", it is possible to automatically answer questions such as "Can Company Z use my data for advertisement?" or "Can my data be collected by Company Z?".

2.1.2.1 *Benefits and limitations*

Machine-readable privacy policies have several benefits, which make them partly suitable to fulfil Objectives 1 and 4. But they also possess an important limitation, *i.e.*, their low understandability, which require them to be complemented with other solutions such as graphical privacy policies.

BENEFITS As opposed to natural language or graphical policies, machine-readable policies can be automatically enforced. Policy languages have means to guarantee that data is accessed according to the policies. Formal languages often provide stronger guarantees as they define how data is processed by all the parties after data collection. For example, they enable the definition of unambiguous rules to ensure that data is only used for purposes in the policies, or that data is only transferred to allowed entities.

Machine-readable privacy policies enable the possibility of auditing whether data is being handled according to their respective privacy policies. This functionality is of great value for DPAs. Auditing mechanisms are typically implemented as logs that record the operations performed on data. Ensuring the integrity of the logs is an orthogonal issue which is crucial for the legal validity of the auditing mechanism [17, 132].

The lack of ambiguity in policy languages endowed with formal semantics makes it possible to precisely reason about their correctness, *i.e.*, that data is handled as stated in the privacy policies. This is specially true for formal languages. It is important to remark that there exists a gap between the formal semantics and its implementation — technical details not modelled in the semantics may lead to unforeseen violation of the properties. Therefore, formal languages should be complemented with auditing mechanisms.

Machine-readable privacy policies enable the possibility of automating certain procedures such as information communication and consent management. Data subjects can automatically receive information on devices able to present it intelligibly, if the communicating entities have been programmed to interpret the format under which the privacy policies are communicated. Automatic information communication facilitates transparency by making data subjects more aware of how their data is being handled — notably in ubiquitous systems where passive data collection is common.

This automatic information is a first step in the design of automatized consent management (consent has to be informed, see Section 1.1.2.3). Automatic communication of privacy policies also makes possible a negotiation of privacy choices: data controllers and data subjects can interact more quickly by means of machines. For instance, instead of refusing an entire privacy policy, a data subject could propose to negotiate a restricted set of purposes of processing. The machine-readability of privacy policies hence greatly facilitates the communication in such cases.

Machine-readable privacy policies can also be beneficial on the web, where they facilitate enforcement [6]. As a matter of fact, they enable an automatic verification of the privacy settings of data subjects and can solve conflicts if there is any, even if the number of stakeholders is important and their interactions complex. Machine-readable privacy policies can similarly be used in P2P systems [70]. Here, the machine-readability of privacy policies enables automatic access control over the purpose of use in P2P environments.

LIMITATIONS The main limitation of machine-readable privacy policies is their lack of usability. As adoption relies among other things on human understandability, understandable and usable policies seems to be a condition *sine qua non* for their adoption and efficiency.

One of the most recurring criticism of machine-readable privacy policies is their lack of human understandability. Pilot takes into account readability requirements: it includes a natural language version of each policy, but many other do not [107]. However, it is questionable whether they can actually be understood.

To put things into perspective, the OECD [120] conducted a study which shows that two third of adults from developed countries cannot conduct a medium-difficulty task related to ICT environments. Although privacy management was not mentioned in the OECD study, it is a medium-difficulty task, and solutions tackling privacy management must consider information-illiteracy.

Summary

Few solutions have been devised to provide the necessary information to the data subjects before the collection of their data in the Internet of Things. Some of them can convey intelligible information, but they are often ambiguous. Other solutions can perform automatic information, at the expense of a poor understandability by lay-users. The interested reader may find more details in a joint paper with Raúl Pardo titled "SoK: Three Facets of Privacy Policies" [107].

2.2 CONSENT

The objectives presented in Section 1.2 include the expression of choices (Objective 4), the communication of consent (Objective 5), and the demonstration of consent (Objective 6). Few solutions have been developed precisely on consent management in the Internet of Things, and can therefore satisfy these objectives. In this section, we successively present: privacy assistants in Section 2.2.1, that could be considered as the closest technical achievements to a consent management system; opt-out facilities in Section 2.2.2; Section 2.2.3 analyses cookie management systems, a thriving research area; and finally we describe how a secure ledger can store and help demonstrate consent in Section 2.2.4. Privacy assistants, opt-out facilities, and cookie management systems address Objectives 4 and 5, while ledgers address Objective 6.

2.2.1 *Privacy Assistants*

Solutions emerged in the 2000s to centralize the management of personal data collection in the Internet of Things from a data subject point of view. These solutions often take the form of privacy assistants. They do not refer specifically to the idea of consent as presented in Section 1.1.2.3, but come close to what could be defined as a consent management system, in the sense that they offer a choice to data subjects regarding personal data collection. With respect to the definition of consent, this choice does not always fulfil all the requirements set forth by the GDPR. For example, the choice is often free, but does not always consider processing (only collection). It can be specific or informed, but ambiguity is not always considered.

Privacy assistants, and their framework, can be coarsely defined as agents interacting on behalf of data subjects, and communicating privacy preferences in a structured format to an environment. The environment is the set of surrounding devices able to collect personal data or to communicate with the agent. Devices able to communicate with the agent can be named *proxies*, *gateways*, or *registries*. An *assistant* usually represents in a user-friendly manner the environment, its intents in terms of data collection, and permits the data subject to set her privacy preferences. The enforcement of privacy preferences is not always taken into account, and when it is, it is often done on a *privacy enforcement point* (PEP): an entity which controls what data can be collected, to which purpose, according to the privacy preferences communicated.

2.2.1.1 *PawS*

Langheinrich presents in [80] a privacy awareness system: PawS. This pioneer work is composed of: i) privacy proxies, with a *personal* proxy for the data subject, which communicates with one or more *service* proxy of data collectors. These proxies refer to both agents (for data subjects) and gateways (for data controllers); ii) machine-readable privacy policies, to codify data practices and to allow automated processes to read such policies and "take actions on them"; iii) policy announcement mechanisms, to notice users with either implicit or active announcement; and iv) privacy-aware databases, on which are stored privacy preferences as privacy policies, as well as collected data. An overview of the system is illustrated in Figure 6.

2.2.1.2 *Framework for informed consent*

Neisse *et al.* describe a framework for informed consent in the Internet of Things in [111]. Their framework provides pseudonymity through a privacy-preserving authentication using Idemix, to register users in services. Privacy preferences are denoted *informed consent policy rules*,

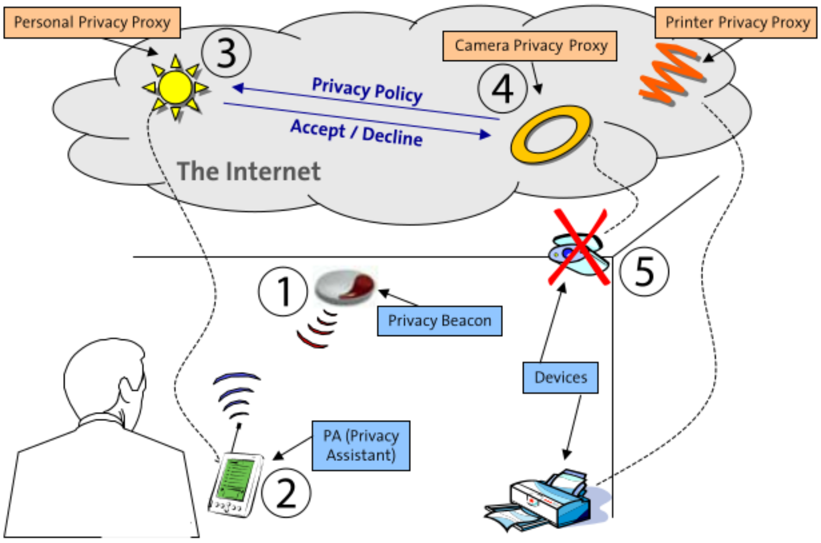


Figure 6: Overview of PawS

and use Model-Based Security Toolkit [112], but these rules do not consider purpose and could rather be categorized as access control rules rather than privacy policies. A *service user* stores *policy rules* in a *security gateway* through her smartphone, these rules are compared to a *security policy* of a *service provider* in a PEP, which selectively disclose data to service providers if a consent has been given.

2.2.1.3 *PrivacyBat*

In a more specific setting, Cha *et al.* [32] propose a user-friendly privacy framework named PrivacyBat to achieve consent using Bluetooth Low Energy devices (see Section 4.1.1.1). The framework allows the declaration of devices as well as their data practices. Privacy preferences of data subjects are communicated through what they call Privacy Preference Expression GATT services.

2.2.1.4 *CMU's Privacy Assistants*

CMU conducts a project named Privacy Assistants, spanning over several years of research and multiple articles. [1] They developed three components for a IoT privacy infrastructure [40], described thereafter. The interaction among the components is presented in Figure 7.

INTERNET OF THINGS RESOURCE REGISTRY (IRR) An IRR allows for its owners — data controllers— to declare devices in Internet of Things environments. Such IRRs can participate in legal compliance by declaring data collection and processing, and can also help advertising Internet of Things resources (their functionalities and services for

[1] https://privacyassistant.org/

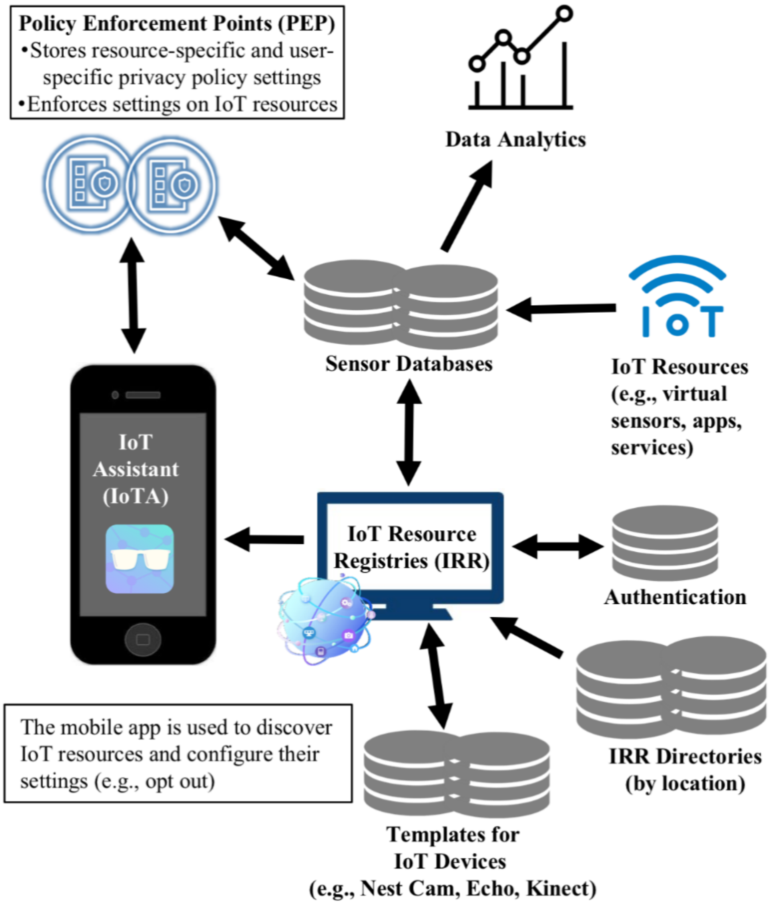

Policy Enforcement Points (PEP)
•Stores resource-specific and user-specific privacy policy settings
•Enforces settings on IoT resources
Data Analytics
IoT
Sensor Databases
IoT Resources (e.g., virtual sensors, apps, services)
IoT Assistant (IoTA)
IoT Resource Registries (IRR)
Authentication
IRR Directories (by location)
The mobile app is used to discover IoT resources and configure their settings (e.g., opt out)
Templates for IoT Devices (e.g., Nest Cam, Echo, Kinect)

Figure 7: Illustrative diagram of CMU Personalized Privacy Assistants

instance). The declaration is operated by "IoT resource owners" and administrators on a web portal, and consists in the type of data, the granularity of collection, the purpose of processing, *etc.* Several IRRs can operate within the same geographical zone, and can be discovered through directories. Because the level of trust between IRRs may not be the same, they propose a hierarchical management of IRRs similar to ICANN and the regulation of domain names.

PERSONALIZED PRIVACY ASSISTANT (PPA) A PPA is an app on a data subject's smartphone in charge of discovering nearby IoT devices (IoTA in Figure 7). It requests IRRs to *notice* data subjects about data practices. Data subjects have the possibility to set their privacy preferences on their PPA, and these preferences are adapted according to machine learning models. They therefore contend that data subjects should be informed of what interest them, as opposed to what they *must* legally be informed of.

PRIVACY ENFORCEMENT POINT (PEP) A PEP is the logical entity responsible for controlling the collection and processing of personal data. It does so according to privacy preferences communicated by a PPA through APIs. A PEP stores the privacy preferences of data subjects, and maintains them in a database. If a data subject decides to use the service of an Internet of Things device, the device first has to request the PEP to verify whether data collection can happen.

2.2.2 *Opt-out facilities*

Certain solutions offer a weak version of consent management: opt-out facilities. Opt-out is not considered by the GDPR as a valid form of consent, but it has been extensively used by data controllers before the enaction of the GDPR. These solutions are often used to opt out of tracking systems. They typically require logging into a portal or an access point, which can either automatically retrieve the identifier used for tracking or request the said identifier to the data subject.

For instance, the Future of Privacy Forum (FPF) uses a solution named Smart Places [134] as an opt-out solution. Data subjects have to manually retrieve their Wi-Fi or Bluetooth MAC address, in order to copy/paste it on a dedicated website. Instructions are provided to locate the addresses.

Another example is the experimental tracking system Wombat [99], which involves an opt-out access point (see Figure 8). Data subjects have to connect to a Wi-Fi access point explicitly denoted "Opt-Out Wi-Fi tracking" for instance. The access point stores the address of the device connected, and adds it to a black list. The devices whose identifiers are in the blacklist are not tracked any more.

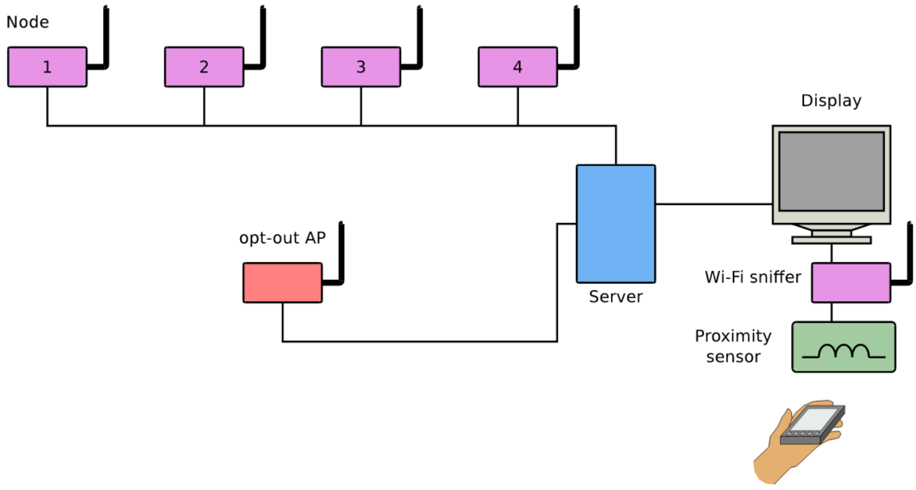


Figure 8: Architecture of Wombat

The settings of opt-out facilities may be considered as too technical for lay-users when a manual input is required, and data subjects can be tracked without having been informed of the processing.

2.2.3 *Cookies management*

A notable example of consent management is offered by cookies. Recent work analysed the rightfulness of cookie consent notices in a post-GDPR era. Cookie consent notices (also named cookie banners) are the prime choice for notifying data subjects of data collection on the web, and should therefore comply with legal requirements regarding consent. These notices are gaining interest in research, both from a usability [78] perspective and in implementation, as we present in this section. Cookie banners are all the more important that the GDPR requires an *informed* consent. They are therefore becoming an important component for valid consent, and 62.1% of websites in Europe display these notices [42].

Degeling *et al.* [42] examine topsites from January to October 2018 — pre- and post-GDPR. They checked the existence of privacy policies, their updates, and their references to the GDPR. More importantly, they analysed cookie consent notices. They observed a significant increase post-GDPR (+16%). But these cookie banners largely lack proper manners to provide choice, and sometimes no choice at all. They extend this analysis in [146] where they argue that most cookie consent notices do not comply with transparency requirements. Indeed, cookie banners tend to use Dark Patterns* to lure data subjects into giving their consent. Utz *et al.* [146] explore the influence of position, the type of choice, and content framing on consent in web cookies context. They found out that nudging obviously influences decision, but so is the position of the cookie banner: data subjects are most likely to interact with consent notices placed at the bottom left position in the browser window.

Recent work by Matte and Bielova [98] analysed Consent Management Providers (CMP) of cookie banners, with a focus on IAB Europe's Transparency and Consent Framework. A CMP provides a cookie banner with the possibility to share the retrieved consents between third parties. They found out at least one violation in 56% of the websites studied . They identify the following four GDPR and ePrivacy violations:

CONSENT STORED BEFORE CHOICE The CMP stores a consent before any choice from the data subject. It violates the requirement of prior consent.

NO WAY TO OPT OUT The data subject cannot refuse consent, and is only informed of data collection. This goes against the requirement of unambiguous consent.

PRE-SELECTED CHOICE The CMP provides already selected boxes or slides, and the data subject has to actively deselect them if she refuses consent.

NON-RESPECT OF CHOICE A consent is stored and shared even though the data subject explicitly refuses consent.

They conclude that although the GDPR led data controllers to provide better consent notices for cookie management, the situation is still far from satisfactory.

2.2.4 *Storing consent*

The GDPR introduces an obligation for data controllers to be able to demonstrate that they have received a valid consent, when consent is the legal basis for processing. A strong way to demonstrate consent should involve a proof of the identity of the data subject for whom consent has been retrieved, and the proof of a successful storage. Ledgers can help store consent. We mean here by *ledger* or registry "a digital, auditable, and trustable log".

Securely storing data has been a topic of interest for many years. The literature has been extensively discussed by the Special Privacy Project in [137]. Their study considers ledgers for storing processing activities and personal data transactions. They list the expected functionalities of a ledger. We only present the functionalities of interest for the present work:

COMPLETENESS All data processing and sharing events should be recorded in the ledger.

IMMUTABILITY The log should be immutable such that it is not possible to go back and modify the history.

INTEGRITY The log should be protected from accidental and/or malicious modification.

NON-REPUDIATION When it comes to both data processing and sharing events it should not be possible to later deny that the event took place.

RECTIFICATION & ERASURE It should be possible to rectify errors in the stored personal data and/or delete data at the request of the data subject. This feature concerns the data but not the ledger itself.

They distinguish three options to achieve these requirements, each option being more appropriate for certain features:

- A local ledger maintained by a single company provides performance and scalability
- A global ledger with a trusted third party provides traceability through event trails

- A global ledger with a peer-to-peer network does not require to trust any third party

Special Privacy [137] references many solutions, but Bellare [17] and Schneier & Kelsey [132] are relevant for our goals. As integrity is difficult to achieve in practice, [17] proposes a property that states that existing data cannot be modified without detection, even if the machine storing data is compromised. This property is denoted the *forward integrity* security property. He also shows how this property can be of use for auditable logs. [132] proposes a method to make log entries unreadable and impossible to undetectably modify on a compromised machine, if they have been generated before the compromise. Both propositions are local ledgers, and provide strong integrity guarantees. However, their implementation is far from trivial due to the complexity of their proposal.

More recently, [117] proposed a distributed and append-only *register* named Hypercore, whose contents are cryptographically hashed and signed. Their ledger was built to prove integrity of distributed content. Their solution replicates i) content on different nodes, ii) metadata about this content, and iii) the ledger which ensures integrity of content. The granularity of the replication can be parametrized. The ledger uses Merkle Hash Trees [103, 109]. The ledger also uses a history to keep trace of modifications, making undetectable modifications difficult. However, an attacker can compromise the ledger if she gets access to the machine storing the ledger. This proposition relies on a global distributed ledger, either based on a peer-to-peer network or a single trusted third party. Although it seems to provide lower guarantees at first sight, due to the possibility of compromising the ledger, its distributed nature makes the possibility to modify content without detection difficult: the attacker would need to gain access to all machines.

SUMMARY

This chapter provided an overview of research dealing with information communication and consent management in the Internet of Things. The solutions surveyed here do not take into account the generic character of the Internet of Things. They often attempt to answer a specific question, or are tailored to a specific setting.

The solutions informing data subjects can be categorized into those providing user-friendly information — through graphical privacy policies— and those providing machine-readable information — through machine-readable privacy policies. Both the former and the latter are scarce, and only a combination of user-friendliness and automation can convey information as mandated by our objectives.

Solutions managing consent are even less numerous. While privacy assistants come close in terms of goals, opt-out facilities and cookie management systems are often misleading, and do not consider consent in the sense of the GDPR (see Section 1.1.2.3). We concluded this chapter with a presentation of ledgers, which can provide guarantees for the storage of consent.

Part II

CONTRIBUTIONS

You know, anyone who wears glasses, in one sense or another, is a cyborg.

Evgeny Morozov

3 FRAMEWORK

Chapter 2 provided a state of the art of existing solutions to communicate information and manage consent in the Internet of Things, and concluded with the fact that few solutions have been devised to these ends in the Internet of Things. Today, no existing solution is satisfactory with respect to the issue presented in the first chapter, *i.e.*, no solution communicates information in the Internet of Things about data collection and processing to data subjects in a user-friendly manner, and provides a fieldable, privacy-protective and facilitating consent management solution for data subjects and data controllers. Moreover, existing solutions are often specific, and cannot address the heterogeneity of the Internet of Things. To fill this gap, this chapter presents a generic framework for information and consent in the Internet of Things. The framework is generic in the sense that it does not depend on specific communication protocols, channels or types of devices, nor fielding configurations: it provides a high-level solution of information communication and consent management applicable to any Internet of Things environment.

The framework involves three complementary components:

INFORMATION The means for data controllers to communicate information about personal data collection and processing to data subjects.

CONSENT The means for data subjects to communicate their consents, and for data controllers to securely store consents received from data subjects.

INTERACTION The means for data subjects to interact in a friendly manner with devices in charge of information retrieval and consent management.

The framework provides the basis for a **user-friendly** solution, addresses **legal compliance**, and achieves these properties **without requiring extensive modifications** of existing infrastructures. These features are discussed along the chapter, and some are more thoroughly described in Chapter 5.

Needless to say, we do not claim that any implementation of the framework proposed here would be legally compliant, or even GDPR compliant, as such compliance involves many legal principles and practical considerations. However, one of our goals is to ensure that the frameworks facilitates GDPR compliance regarding information and consent in the Internet of Things. Moreover, the framework does not consider the enforcement of privacy policies. Enforcement is out of the scope of this work, which focuses on information and consent management.

We present in Section 3.1 the assumptions of the framework, and the notation used. We then pursue with the presentation in Section 3.2 of a protocol for information and consent in the Internet of Things: the *Privacy Policy Negotiation Protocol* (*PPNP*). *PPNP* mostly aims to fulfil Objectives 1, 2, and 5. Section 3.3 presents guidelines for designing a privacy managing agent in the Internet of Things, that we denote *Personal Data Custodian* (*PDC*). The *PDC* mostly addresses Objectives 3 and 4. Finally, Section 3.4 describes how a proof of consent can be achieved using high-level requirements. Objective 6 is satisfied by these requirements, which can be seen as functions a system must be able to perform. However, each component of the framework does not exclusively address and fulfil the objectives aforementioned, and each component participates in numerous ways to their completion.

Each component also possesses *mandatory* and *optional* requirements: the framework cannot be implemented if the former are not met, and the latter provide refined features. A restrictive setting of the framework can only consider mandatory requirements, but optional requirements are recommended for a full-fledged implementation. The two types of requirements are summarised at the end of the chapter in Tables 1 and 2.

3.1 ASSUMPTIONS AND NOTATION

The framework proposed in this section relies on a set of hypotheses: it requires **devices** (Section 3.1.1) possessing certain **features**, which communicate by means of specific **messages** (Section 3.1.2). We assume that we are in a ubiquitous environment where data controllers deploy devices that can collect personal data on data subjects. We refer the reader to the glossary in Appendix A for a summary of the acronyms used. [1]

3.1.1 *Devices*

Figure 9 shows the different devices and their possible interactions. In this figure, *information and consent* refers to *PPNP* (Section 3.2); *interaction* and *bond* refer to the human-computer interactions (Section 3.3); *consent storage* refers to the proof of consent (Section 3.4); *potential data collection* and *consent list update* illustrate the collection process and the enforcement of privacy policies.

Any device owned by a data controller is denoted *Data Controller Device*, or *DCD*. Any device owned by a data subject is denoted *Data Subject Device*,

[1] Note that the appendix also present important terms which are not necessarily acronyms.

or *DSD*. We describe in what follows the specific roles of these devices. A summary of the devices required is provided in Tables 1 and 2 under **Devices**.

3.1.1.1 *Mandatory devices*

We denote *DCG*, for *Data Controller Gateway*, a device controlled by a data controller, able to communicate information (the content of which is described thereafter), and to retrieve consent. The *DCG* therefore participates in Objectives 1 and 5. The consents received are stored on a component denoted *CSS*, for *Consent Storage Server*. This component contributes to Objective 6.

We denote *DSG*, for *Data Subject Gateway*, a device controlled by a data subject and able to communicate information and consent messages.

FEATURES A *DSG* must possess a *Data Subject Policy* (*DSP*), and a *DCG* must possess a *Data Controller Policy* (*DCP*). [2] The privacy policies must be machine-readable, *i.e.*, in a structured format interpretable by a machine. More precise requirements over privacy policies are described in Section 3.1.2.

A *DSG* must be able to hash files, to store cryptographic keys, and to sign messages (see Appendix D). Requirements over cryptographic capabilities are needed for the proof of consent (see Section 3.4).

Similarly, a *CSS* must be able to hash files, to store cryptographic keys, and to sign messages. The *CSS* must also have the capacities to *archive* consents (see Section 3.4).

A *DSG* must have an appropriate interface. We mean by appropriate interface: *a touchscreen or the combination of a screen and a keyboard, allowing for the human-computer interactions described in Section 3.3* (consultation of privacy policies, management of the *DSP*, and notifications to the data subject).

3.1.1.2 *Optional devices*

Other devices are optional, such as other *DCDs* and *DSDs*. *DCDs* are devices collecting personal data, they are not mandatory for information and consent, but considered to the extent that we assume personal data is collected. For their part, data subjects may own several *DSDs*. They are optional, but our framework supports them. *DSDs* are linked to the *DSG* through a process denoted *bonding* (see Section 3.3.2.2). Other *DCDs* and *DSDs*— *i.e.*, non-gateway devices — , may have weaker computational capacities, and inappropriate or even no interface.

[2] Nothing prevents a data controller from having several *DCGs*, each with a different *DCP*.

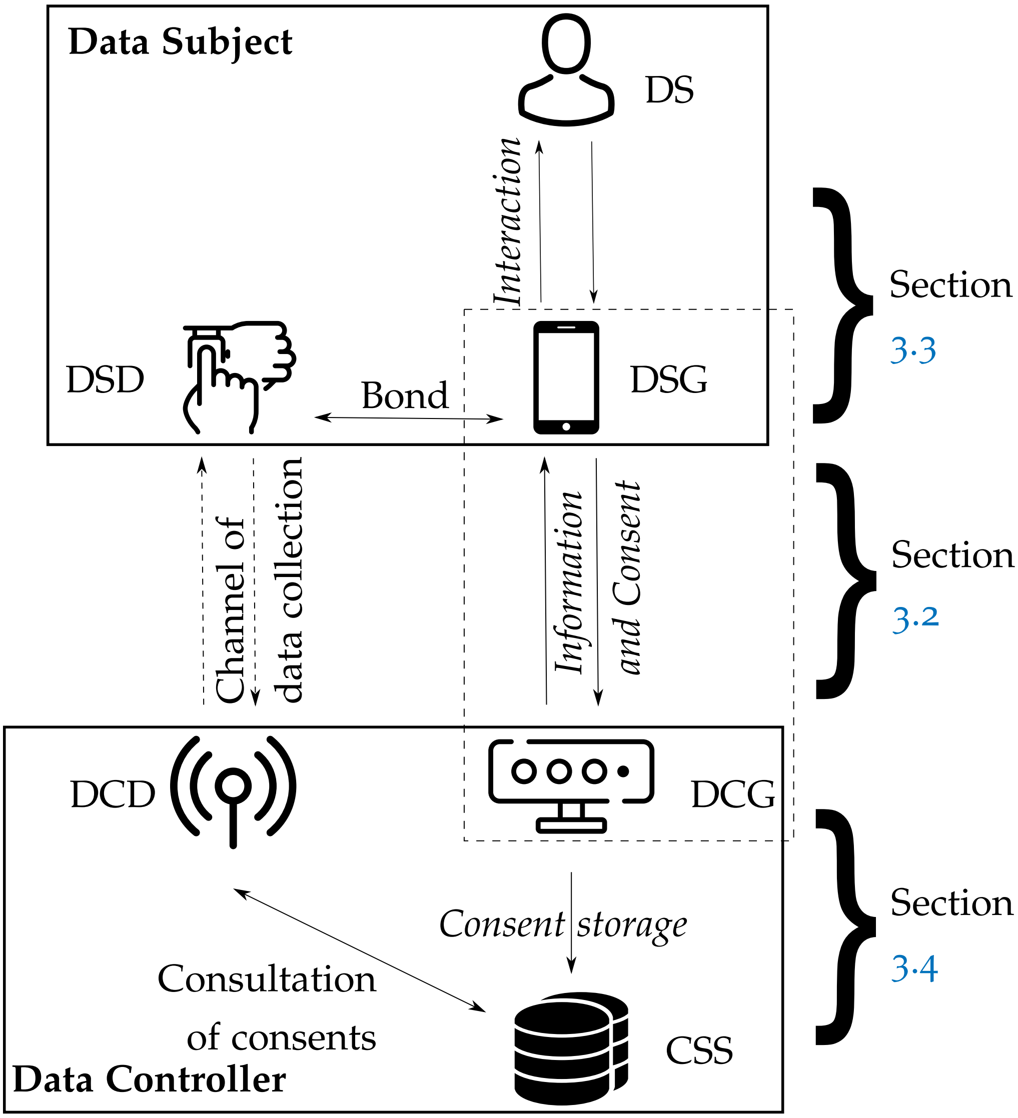


Figure 9: Explanatory diagram of the framework. Devices and their interactions are denoted with normal font, components of the framework are denoted in *italic*, and the two sides of the framework are denoted in **bold**.

3.1.2 *Messages*

Devices can communicate using messages, that we define below. Here, we only consider messages for information and consent. We first consider the privacy policies (Section 3.1.2.1) — on which operations may be applied — then the other types of messages (Section 3.1.2.2).

A summary of the messages required is provided in Tables 1 and 2 under **Messages**.

3.1.2.1 *Privacy policies*

As stated in Section 3.1.1, *DSGs* and *DCGs* must respectively possess a *DSP* and a *DCP*. A *DCP* corresponds to the commitment of a data controller in terms of personal data management. A *DSP* corresponds to the requirements of a data subject for the processing of her data. A *DSP* can be seen as what data one agrees to be collected, and under which conditions of processing. A *DSP* is positive in the sense that it only states what a data subject authorizes and under which conditions. [3]

Following the guidelines provided in [107], we propose a multi-faceted approach to privacy policies. This section presents the machine-readable facet. A suitable privacy language for this facet must meet requirements over the syntax, and over the operations permitted. These requirements are detailed in the rest of the current section. Any language meeting these requirements can be used in the framework.

A summary of the content of privacy policies required is provided in Tables 1 and 2 (pages 62 and 63) under **Privacy policies**. A summary of the required operations permitted over privacy policies is provided in Tables 1 and 2 under **Language**.

MANDATORY REQUIREMENTS Privacy policies must be represented as a set of *rules*, to each of which corresponds one type of data:

$$policy ::= (rule_1, rule_2, \ldots, rule_i)$$

A rule is a set of information specifying for one type of data: a) the commitment of a data controller in terms of personal data management if it belongs to a *DCP*, or b) the requirements of a data subject for the processing of her data if it belongs to a *DSP*.

Content The mandatory part for a rule is:

- the type of data
- the purpose of collection

[3] As opposed to a negative definition, in which every data collection and processing would be authorized, and where a data subject would selectively *opt-out* of the terms of the *DSP*.

- the retention time
- the data controller concerned (*DC*). [4]
- and the 3rd parties involved (which can be empty)

The 3rd parties involved are required by law, but not all data processing include them: this item is required but can be empty. The required content of a rule is a subset of the content required by the GDPR when informing about personal data collection.
See example 1 for clarification.

Example 1. *A DSP rule can be "I agree that my **license plate** is collected for **improvement of service** purposes by **interparking**, and stored no more than **7 days**." Reciprocally, a rule from a DCP would be "**Interparking*** requests *your **license plate** for **improvement of service** purposes, and store it for **7 days**".*

Both *DCP* and *DSP* have the same requirements over content, but they are interpreted differently: a *DSP agrees*, while a *DCP requests*.

Comparison It must be possible to formally compare a *DSP* to a *DCP*, and the result is required to be deterministic. A *DSP* is said to *match* a *DCP* if and only if a $DSP \geqslant DCP$, *i.e.*, the *DSP* is more permissive than or equal to the *DCP*. A consent can be communicated iff $DSP \geqslant DCP$. See example 2 for clarification.

Example 2. *A DSP rule can be "I agree that my **location** is collected for **analytics** purposes by **Villeurbanne**, and stored no more than **30 days**."* [5] *Reciprocally, a rule from a DCP such as "**Villeurbanne** requests your **location** for **analytics** purposes, and store it for **365 days**" will not match the rule above, as the retention time requested by the DCP is more than what I require in my DSP. However, the two policies would match if the same DCP had a retention time of **30 days or less**.*

Intersection Two privacy policies are required to be intersectable. We denote the intersection operation $\cap$. The intersection of two privacy policies should be understood as *the terms on which the parties agree according to their current policies*. In other words, given two policies *DSP* and *DCP*, the result of $DSP \cap DCP$ is the greatest DCP_2 such that $(DSP \geqslant DCP_2) \wedge (DCP \geqslant DCP_2)$. If $DSP \geqslant DCP$, then the intersection $DCP_2 = DCP$. Intersection is required when negotiating as we describe in Section 3.2. See example 3 for clarification.

[4] Only the name of the data controller belongs to the machine-readable facet, and we use the term *DC* to distinguish it. Its contact details belong to the natural language facet.

[5] We assume that no other rule considers location as a type of data for the sake of the example.

Example 3. *A DSP rule can be "I agree that my* ***location*** *is collected for* ***analytics*** *purposes by* ***Villeurbanne****, and stored no more than* ***30 days****."* [6] *Reciprocally, a rule from a DCP such as "****Villeurbanne*** *requests your* ***location*** *for* ***analytics*** *and* ***marketing*** *purposes, and store it for* ***15 days****" will not match the rule above, a purpose of collection (****marketing****) is not authorized by my DSP. However, we can agree on an intersection requiring my* ***location*** *to be processed only for* ***analytics*** *purposes and stored for* ***15 days****.*

OPTIONAL REQUIREMENTS

Content Besides the content necessary for the operation of the protocol, privacy policies can also contain the following information:

- the frequency of collection
- the location of *DCDs*
- the range of collection

Location of device, frequency and range of collection are tailored to the Internet of Things and can provide better insights about devices.

CATEGORIES Among the different items data subjects must consider in the rules of a *DSP*, most could knowingly be defined in advance. However, it may be more convenient for data subjects to define rules for categories of *DC*, data or purposes than on each of them separately. Data subjects must be able to define high-level items, or *categories* of items. See example 4 for clarification:

Example 4. $\mathcal{D}$ = *Identifiers,* d_1 = *Wi-Fi MAC address,* d_2 = *IMEI number.* $d_1, d_2 \in \mathcal{D}$*. Authorizing Identifiers will authorize both Wi-Fi MAC address and IMEI number collection.*

3.1.2.2 *Other types of messages*

The protocol described in the next section involves other types of messages:

MANDATORY MESSAGES

Consent A consent is the authorization from a data subject to a data controller to collect data and use it according to a *DCP*. A consent consists in: a hash of a privacy policy, the set of identifiers of the devices concerned by data collection, [7] and a cryptographic signature

[6] Same comment as above, we assume that no other rule considers location as a type of data for the sake of the example.

[7] Data subjects can consent for as many *DSDs* as they want. Such identifiers can be digital, such as Wi-Fi or Bluetooth MAC addresses, or physical, such as licence plates.

for the authentication. A consent is required to be communicated in plain text and signed — *i.e.*, encrypted with the data subject cryptographic private key. The signature authenticates the data subject, thus ensures the origin of the consent; the hash of the *DCP* ensures its integrity. A *DCP* always produces the same hash, and is the only *DCP* able to produce this specific hash. Hash functions are presumed to be collision-free, *i.e.*, two different *DCPs always* produce two different hashes (see Appendix D).

Dissent A data subject can also withdraw a consent by communicating a *dissent*. A dissent consists in: a hash of a *nil* privacy policy, the set of identifiers of the devices concerned by data collection, and a cryptographic signature for the authentication. A dissent must be communicated in plain text and signed. A data controller who previously retrieved a consent from a specific data subject can then no longer pursue data collection, and must stop data processing. Consent withdrawal does not affect the lawfulness of previous processing [53, Art. 7(3)].

Refusal It is a message sent by a *DSG* to a *DCG* to inform that no agreement can be conducted.

OPTIONAL MESSAGES In order to agree on an intersection policy, data subjects and data controllers undertake what we denote as a *negotiation*. This process requires other messages besides privacy policies and consents:

Deny It is a message sent by a data controller to a *DCG* to inform that the privacy policy intersection is rejected

Accept It is a message sent by a data controller to a *DCG* to inform that the result from the intersection is accepted

3.2 PROTOCOL

The protocol presented in this section is one of the main components of the framework. We denote this protocol *Privacy Policy Negotiation Protocol* or *PPNP* for short. The protocol provides a generic approach to inform data subjects and to manage their consents by communicating machine-readable privacy policies. *PPNP* is generic in the sense that it relies on few technical assumptions. The protocol consists in three main phases: **information**, **consent**, and **interaction**. Negotiation is an *optional* requirement for our framework. The phases information and consent consider machine-to-machine communications, while interaction considers natural persons (data subjects). Note that interactions with data subjects is not the focus of *PPNP*: they are described with more precision in Section 3.3.

Even if it is an optional requirement, negotiation is still an important part of consent. As a matter of fact, the possibility for data subjects to negotiate the terms rebalances power between data controllers and data subjects. Data subjects are no longer constrained to either accept all terms of data collection and processing or refuse them altogether; data controllers can also refuse the new policy, denoted *intersection* policy, proposed by data subjects.

The use of *PPNP* does not in itself guarantee **legal compliance**, but the protocol specifically addresses the notion of informed consent: consent is communicated after the information of the data subject. As a matter of fact, *PPNP* ensures that a consent can be communicated if and only if 1) the *DSG* has received the *DCP*, and 2) the *DSP* matches the *DCP*.

We first present a semi-formal definition of the protocol using state diagrams in Section 3.2.1, and then using sequence diagrams in Section 3.2.2.

A summary of the features required by the protocol is provided in Tables 1 and 2 under **Protocol**.

3.2.1 *State diagrams*

This section presents a view of the *DCG* and the *DSG* through UML state diagrams [144]. These state diagrams provide abstract representations of machines, and can be useful for implementation and reasoning. We consider both mandatory and optional requirements.

Entities are denoted with **lower cases**. Variables are denoted with **UPPER CASES**, other messages with regular **Title Cases**.

We first describe the states and transitions used by the diagrams; then the state diagram of the *DSG* and the *DCG* as directed graphs.

3.2.1.1 *States and transitions*

The states of the *PPNP* protocol include the following information: *DCP*, *DSP*, CONSENT, and DISSENT (see Section 3.1.2 for more details about the content and format of these messages). Similar to Section 3.2.2, we implicitly consider generic entities dealing with the same type of data for the sake of simplicity and explanation. Transitions can be triggered by events, or by conditions. States are represented by rounded rectangles, except the initial state represented as a black circle denoted S1. Choice pseudo-states are represented by diamonds, and address conditions. Transitions are represented by arrows from a state to another.

CONDITIONS Conditions are boolean tests over state variables. An event occurs only if the test evaluates to *true*. For instance, in the

DSG state diagram (see Section 3.2.1.3), the arrow from S2 to a choice pseudo-state means that: if the *DSP* is less restrictive than the received *DCP*, then we go to state S3, and a consent is issued to the *DCG* which communicated the *DCP*; if the *DSP* is more restrictive, then we go to state S4 *etc.* Our systems are deterministic in the sense that there is always only one transition such that its condition evaluates to *true* in a given state.

EVENTS Our state diagrams involve two types of events: internal and external. Internal events trigger external events. In the case where more than one event happen, the events are separated by a "," and it means they happen consecutively.

Internal events Some events in the state diagram of an entity are triggered by the entity itself:

- Send(A, b) where A is an information communicated to another entity b.
  Send(A, b) by entity c triggers a Receive(A, c) in entity b.

- Set timer. A timer set will eventually expire, triggering a change of state

- Prompt(A, b) where A is an information communicated to another entity b. Prompt is used to request a new policy from the data subject (through his *PDC*) or acceptance of a policy from the data controller.

External events Some events in the state diagram of an entity are triggered by another entity, or by the expiration of a timer:

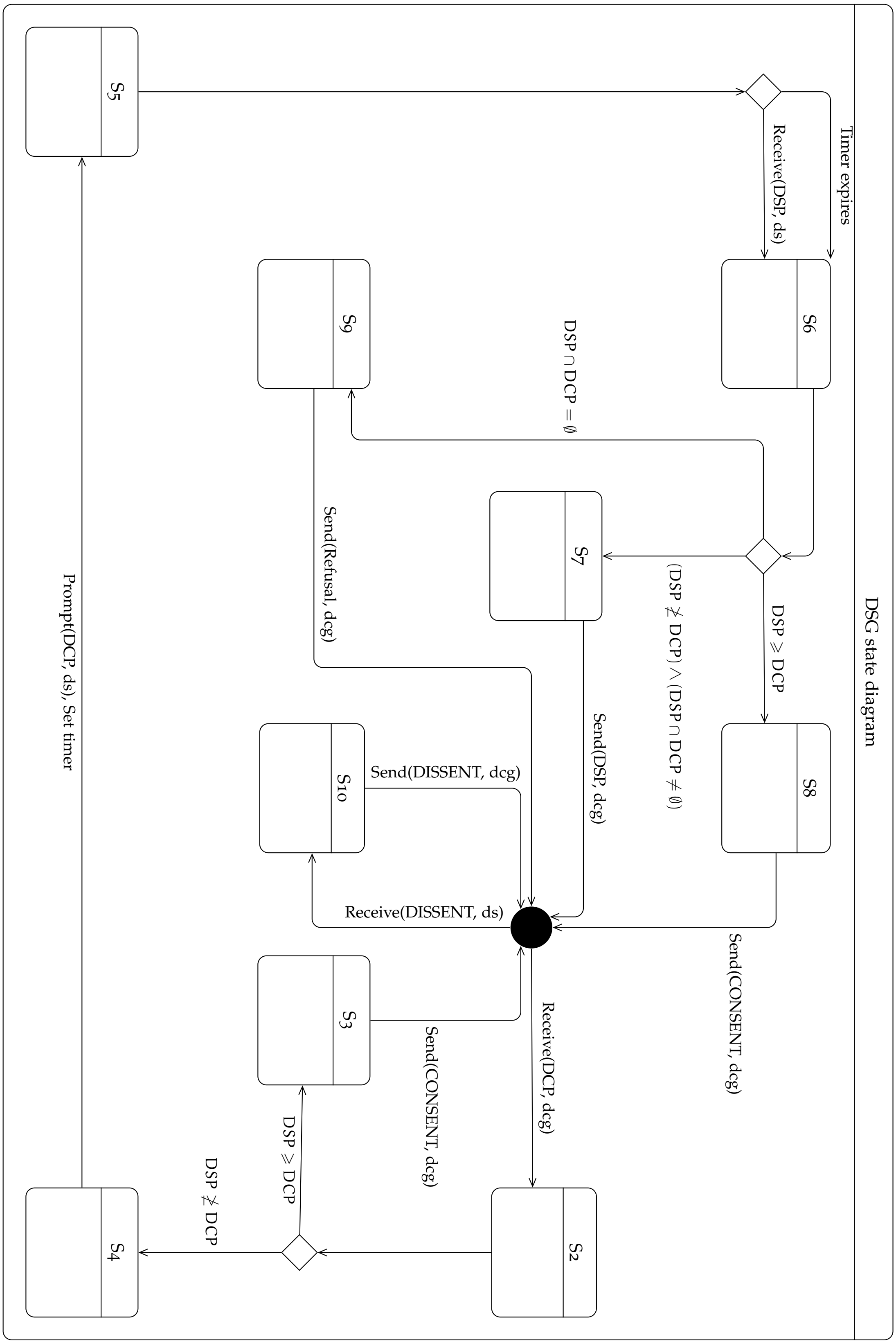
- Receive(A, b) where A is an information received from another entity b. It implicitly changes the configuration of the entity receiving the information

- Timer expires

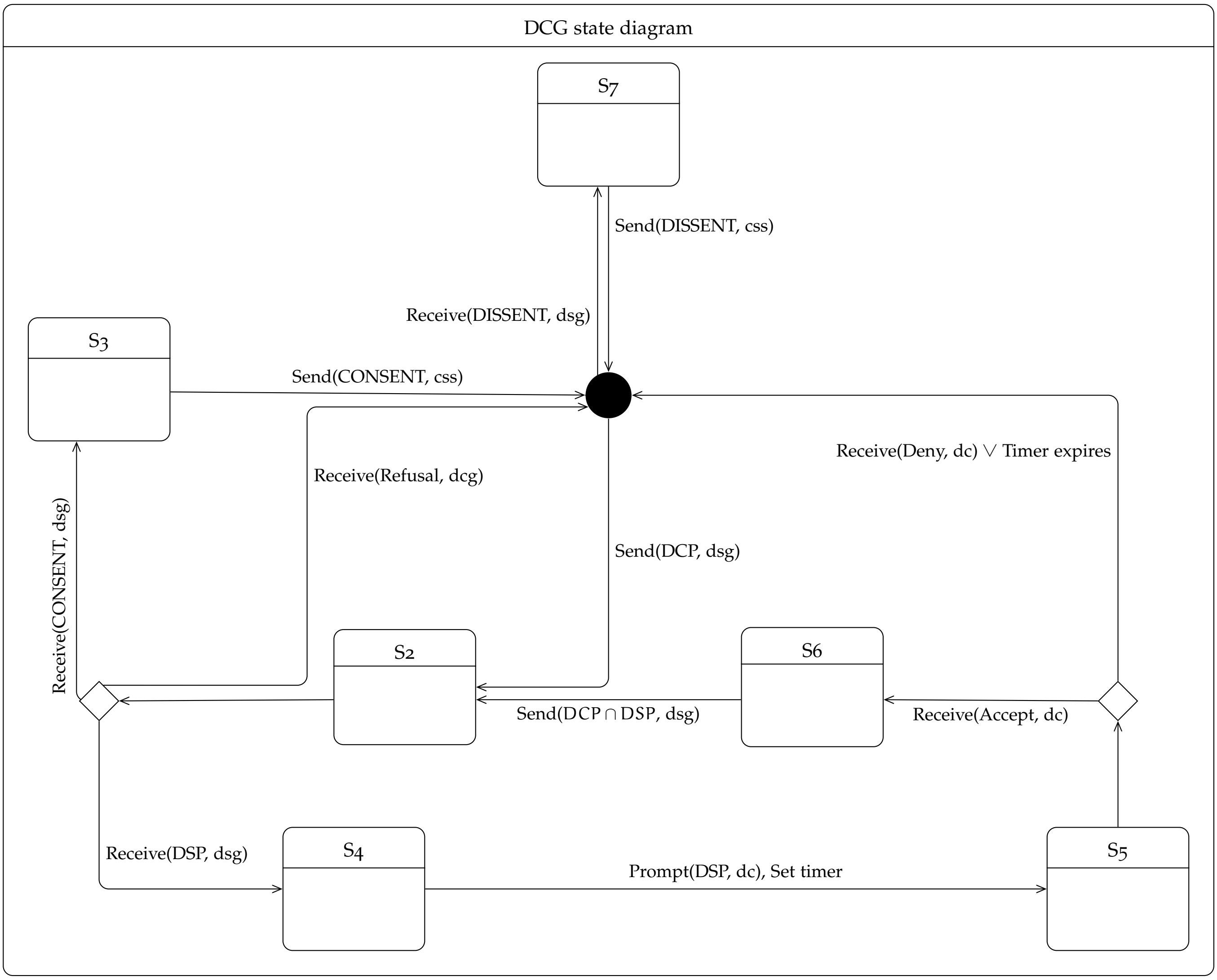
3.2.1.2 *Mandatory and optional requirements*

We refer the reader to Tables 1 and 2 presented at the end of the chapter for a distinction between the mandatory and the optional requirements over the state diagrams.

3.2.1.3 *DSG state diagram*



3.2.1.4 *DCG state diagram*



3.2.2 *Semi-formal definition*

In the following, we use sequence diagrams [144] showing events that happen sequentially from the top to the bottom. These sequence diagrams consider both *mandatory* and *optional* requirements, and are complementary of the state diagrams. We use here the same notation as for the state diagrams when it can be applied; this is not the case for the communication between the *DSD* and the *CSS* for instance, which is not considered in the state diagrams. Devices and actors (data subject and data controller) are represented on the top of the diagram. Continuous arrows from an entity (device or actor) to another are requests. Dashed arrows are answers to requests. Continuous arrows from an entity to itself provide information about the state of the entity, or the result of an operation. The entities considered in the diagrams are *DCDs*, a *CSS*, a *DCG*, a data controller denoted *DC*, a *DSG*, a data subject denoted *DS*, and *DSDs*. We implicitly consider generic entities dealing with the same type of data for the sake of simplicity and explanation. Note that *DCDs* and *DSDs* are presented to illustrate how data collection could happen.

3.2.2.1 *Information (mandatory)*

The *DCG* initiates the communication by providing the *DCP*. The *DCP* is received by the *DSG*. *PPNP* does not consider the actual communication means, and can therefore be implemented on different technologies as shown in Chapter 4. Consequently, we do not consider communication failures, and, as a result, non reception of messages should be dealt with carefully at the implementation level.

3.2.2.2 *Consent (mandatory) and negotiation (optional)*

Upon reception, the *DSG* compares the *DCP* and his own policy (called *DSP*) for the type of data and data controller specified in the *DCP*, and issues a consent message if they match (see Figure 10). We refer the reader to Section 3.1.2.1 for the definition of comparison.

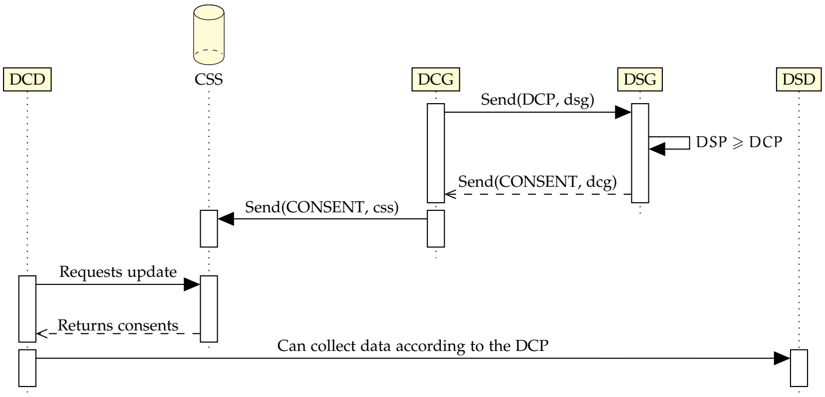


Figure 10: The policies match

If the two privacy policies do not match, the *DSG* prompts the data subject, sets a timer, and awaits for a new *DSP*— denoted *DSP′* — from the data subject (see Section 3.3.1.4). In doing so, we distinguish the machine-to-machine (M2M) communication from the human-computer interaction (HCI). The inputted *DSP′* becomes the new *DSP* for this type of data. Note that the input of a new *DSP′* is important for data subjects to consider purposes of processing or *DC* that were not considered in their original *DSP* (or modalities of processing in a broader meaning).

After the input of the new *DSP′* from the data subject (or the expiration of the timer), the *DSG* checks whether the *DSP′* matches the *DCP*:

- The policies match (*i.e.*, the *DSP′* is less restrictive than the *DCP*, see Section 3.2.1), in that case a **consent** for the *DCP* is issued (see Figure 11).

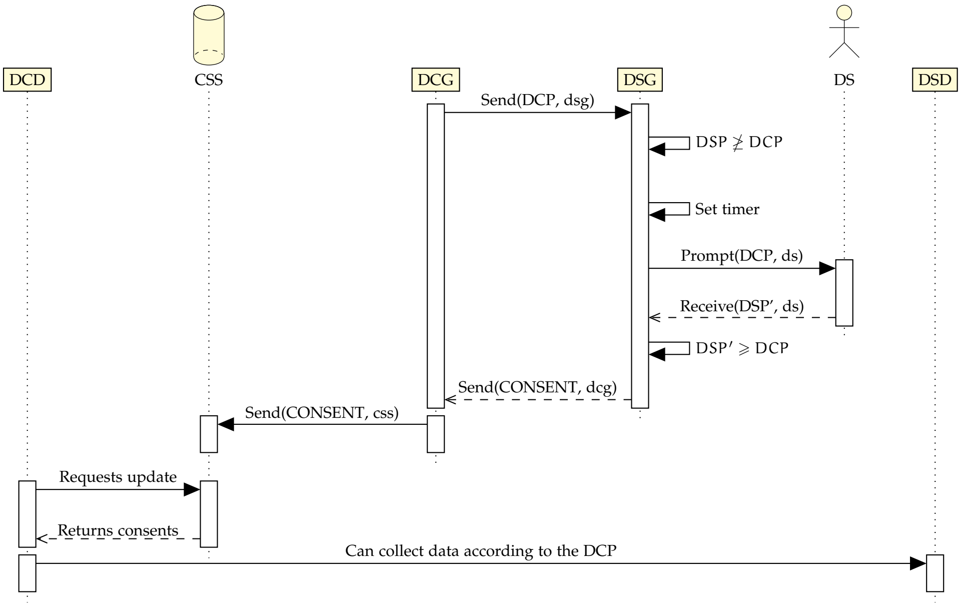


Figure 11: The policies do not match at first, the *DSG* requests an interaction from the data subject. The modification results in a match.

- The policies do not match and their intersection is null (*i.e.*, the data controller and the data subject cannot agree on any data collection according to their current privacy policies, see Section 3.2.1), the *DSG* issues a message of **refusal** (see Figure 12). This means that, based on the current privacy policies, the data controller and the data subject cannot agree on any data collection. As a result, the communication is stopped. A more detailed description is presented in Section 3.2.1.

- (Optional) The policies do not match but their intersection is not null (*i.e.*, they can agree on some terms of data collection and processing), in that case the *DSG* sends the new *DSP′* to the *DCG* (see Figures 13 and 14). The intuition is that the data controller and the data subject can negotiate terms of data collection as long as they have some terms on which they agree.

The *DCG*, after having sent the *DCP*, awaits for answers from the *DSG*:

- If the *DCG* receives a **consent**, this consent is forwarded to the *CSS*.

- If the *DCG* receives a **refusal**, no consent is forwarded to the *CSS*.

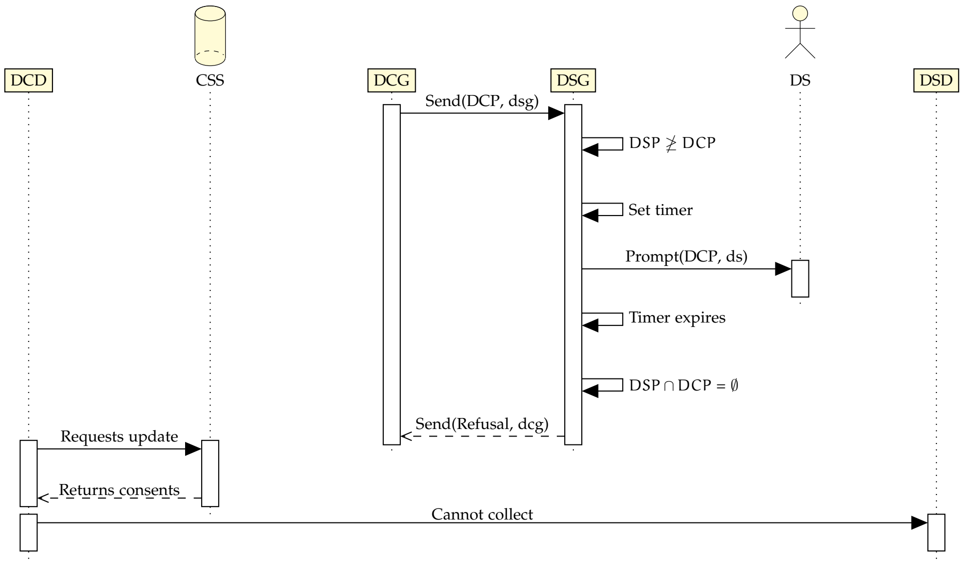


Figure 12: The policies do not match, and the data subject does not interact.

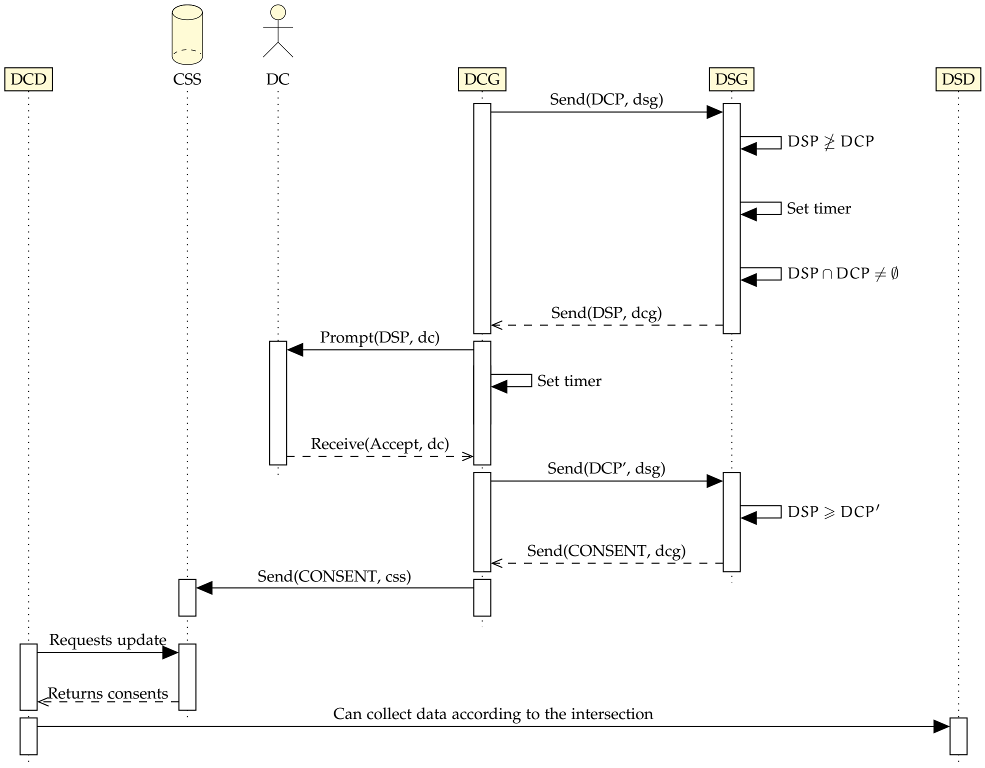


Figure 13: (Optional) The policies do not match, but an agreement is made on the intersection of policies

- (Optional) If the *DCG* receives a *DSP*, it requests its data controller for permission to use the intersection policy between the *DCP* and the *DSP*, and sets a timer. The data controller can either **accept** or **deny** this intersection policy. Accepting the intersection means that the data controller authorizes this new *DCP*—

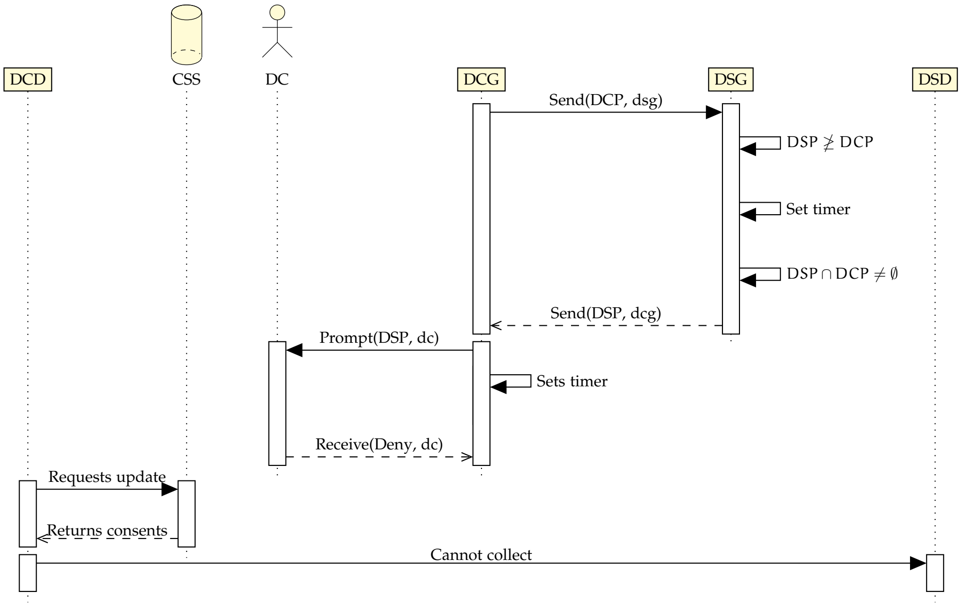


Figure 14: (Optional) The policies do not match, the data subject agrees to a new *DSP*′, but the data controller denies the policy

denoted *DCP*′ — specifically for the collection of data from this data subject. The *DCP* provided originally does not change and will be used for the next requests. If it is accepted, the intersection is sent to the *DSG*, and the *DCG* awaits for a **consent**, a **refusal**, or a new *DSP* (see Figures 13 and 14). Otherwise (a **denial** or the expiration of the timer), the collection of data is not allowed.

The data subject can withdraw her consent at any moment by sending a dissent message to the *DCG* (see Figure 15). The withdrawal of consent corresponds to a requirement from the data subject to stop data collection and further data processing by the data controller (see Section 3.1.2.2). All consents and dissents are forwarded to the *CSS*, which accepts only these two types of messages. The *CSS* provides a list of consents, and dissents if there is any, to his *DCDs* on request. A *DCD* must query the *CSS* regularly — or it can be updated by the *CSS* — in order to maintain an up-to-date list of consents. As mentioned before, the enforcement of the *DCP* by the data controller is out of the scope of this study. The data controller must consider the most recent consent or dissent input (see Section 4.3.3.1).

The information through the *DCP*, the communication of consent and the negotiation can be performed on different channels (see Chapter 4).

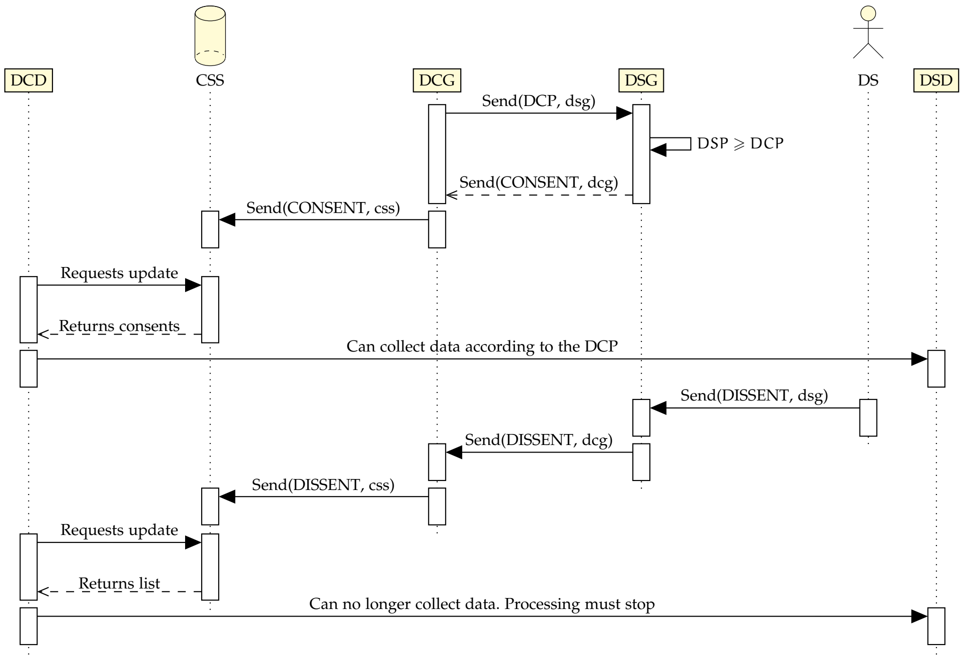


Figure 15: The data subject dissents

3.3 HUMAN-COMPUTER INTERACTIONS

PPNP provides means to inform data subjects, but it does not consider how the information is presented. Similarly, the protocol also provides means to communicate consent, to negotiate, and it assumes that data subjects interact, but it is silent on how these interactions should occur. The present section proposes guidelines to present information and interact with data subjects, introducing the notion of *Personal Data Custodian* (*PDC*).

A *PDC* can be seen as a software agent operating on behalf of data subjects, installed on the *DSG*. The role of the *PDC* is to reduce the burden of personal data management upon data subjects.

A *PDC* must provide certain facilities: presenting *DCPs*, displaying and managing *DSP*, and notifying the data subject when prompted. To each of these facilities corresponds a User Interface (UI), *i.e.*, a screen on the *Personal Data Custodian*. The facilities are described thereafter, and mapped to the UIs, presented as state diagrams in Figure 16.

The *PDC* is **required** to display *DCPs* intelligibly — see UI "*Consult DCP*" (Section 3.3.1.1). This feature is required to fulfil Objective 3. The *PDC* is **required** to provide access to the *DSP*— see UI "*Consult DSP*" —, and control over the creation, modification, and deletion of rules — see UI "*Add/modify/delete*". These features are required to fulfil Objective 4. The *PDC* is **required** to deal with notifications when negotiating — see "*Negotiation*" (Section 3.3.1.4). This feature is required to make a link with *PPNP*. Finally, the *PDC* may **optionally**

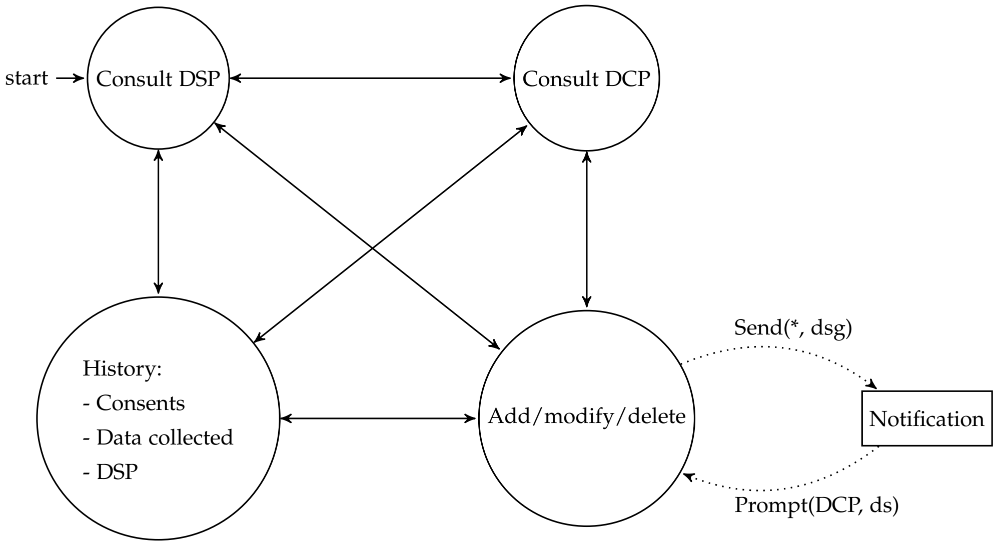


Figure 16: Global functioning of the *Personal Data Custodian*.
"Consult DCP" is a state corresponding to the UI of *DCPs*.
"Consult DSP" is a state corresponding to the UI of the *DSP*.
"Add/modify/delete" is state corresponding to the UI of the modification of the *DSP*.
(Optional) "History" is state corresponding to the UI of consents, a summary of the data collected by a given data controller, and a versioning of the *DSP*.
"Notification" deals with external outputs such as Prompt() requests of *PPNP*, and with answers to these requests using Send().

present a history of important events, such as consents given — see UI "*History*" (Section 3.3.2.4).

A summary of the features required by the *PDC* is provided in Tables 1 and 2 under **PDC**.

3.3.1 *Mandatory requirements*

3.3.1.1 *Consult DCP*

Expressing privacy policies only in natural language is not enough for two reasons: because it makes automatic analysis difficult, and because it is ambiguous and difficult to understand for lay-users (see [107]). Those two issues can be mitigated by adopting a multifaceted approach to the expression of *DCPs*, which combines natural language, graphical, and machine-readable facets of privacy policies.

We proposed a machine-readable format in Section 3.1.2, along with the content required for the framework: type of data, purpose of processing, retention time, *DC*, and 3rd parties. Following the guidelines provided in [107], we now introduce the graphical facet.

The graphical facet of *DCP* must present the mandatory content (see Section 3.1.2) in the foreground. The *PDC* must not present all information on the first UI, but is required to provide access to more details if the data subject requests it.

The standard and mandatory way to present information on a *PDC* is denoted rule-per-rule. A rule-per-rule presentation consists in presenting the content of each rule one after the other in a sequential order. A solution consists in representing each rule in simple natural language, combined with icons (see [107]).
See example 5 for clarification.

Example 5. *A rule-per-rule presents DCPs as follows:*
*"**Elgoog** requests your **Wi-Fi MAC address** for **analytics** and **marketing** purposes, and store it for **30 days**."*
*"**Koobecaf** requests your **Bluetooth MAC address** for **marketing** purposes, and store it for **60 days**."*

The *PDC* must also provide a link to the natural language version in order to be legally compliant (see [107]). In doing so, data subjects can easily access a more detailed version without compromising the usability of the UI. The link to the natural language version is not part of the rules.

3.3.1.2 *Consult DSP*

In addition to the display of *DCPs*, the *PDC* must present the *DSP*. Being able to consult her own *DSP* is a condition *sine qua non* for its management, and its presentation must meet the same requirements as the *DCP*, except for the link to the natural language version, only required for data controllers for accountability purposes [107]. Similar to *DCPs*, the *DSP* must be proposed rule-per-rule.

3.3.1.3 *Add/modify/delete*

Unlike a *DCP*, a *DSP* is likely to be created and modified by a data subject. The modification of a *DSP* must be done rule per rule, hence, a data subject must be able to add, modify, and delete rules. These operations must be conducted from a different UI used to consult the *DSP*, but the addition of a rule must be particularly put forward when consulting the *DSP*.

3.3.1.4 *Notifications*

The *PDC* must handle notifications, especially when an interaction occurs (*Prompt(DCP, ds)* in *PPNP*). The data subject is requested by the *PDC* when the *DSG* receives a *DCP* and the policies do not match. The *PDC* can trigger a physical notification on the *DSG* such as a vibration or a sound, or can be mute and only display the request as a visual notice. Vibrations must be designed carefully as they can tend to over-burden data subjects. The modality of notifications is at the sole designer's discretion. Their content is however a requirement.

The data subject answers with a Send(*, dsg) (according to *PPNP* notation). The answer is a *DSP* if the request was a *DCP*. This *DSP* can be a new one, or the current one if no change is made (illustrated in Figure 12). A data subject can also spontaneously communicate a consent withdrawal (DISSENT) through this medium.

3.3.2 *Optional requirements*

Additionally, the following optional functionalities can alleviate data subjects' burden when managing their *DSPs*.

3.3.2.1 *Alternative presentation*

A sorted presentation consists in selecting an instance of an item, and presenting all the rules associated with the selected instance. See example 6 for clarification.

Example 6. *A presentation sorted by the purpose of collection* ***marketing*** *presents DCPs as follows:*
"DCPs requesting data for ***marketing*** *purposes:*
Elgoog *requests your* ***Wi-Fi MAC address****, and store it for* ***30 days****.* ***Koobecaf*** *requests your* ***Bluetooth MAC address****, and store it for* ***60 days****."*

3.3.2.2 *Bonding other DSDs*

Data subjects must be able to bond other *DSDs*. Here *bonding* refers to the addition of the identifier of a *DSD* to the set of identifiers of a consent (see Section 3.1.2.2). This step consists in either 1) automatically bonding a device. It requires the other *DSD* to be in a bonding state and a communication protocol intended for this purpose; or 2) manually inputting an identifier when automatic bonding is not possible. Bonding another *DSD* does not need a dedicated UI, it can be done on any UI.

3.3.2.3 *Preset items*

When creating or modifying a rule, a data subject should be proposed to choose among preset items to reduce fatigue. For instance, when entering the type of data of a new rule, a data subject must be proposed a list of common or already used types of data. This can be achieved through a drop-down menu, or pull-up list [8]. Preset items can be restricted according to the first characters inputted. Every item may propose a set of preset items.

See example 7 for clarification:

Example 7. *A data subject wants to modify an existing rule. A list containing* "Bluetooth MAC address, Wi-Fi MAC address, Licence plate" *is proposed when the data subject has to input the type of data. The list can be* *restricted to* "Bluetooth MAC address, Wi-Fi MAC address" *if the data subject has already input the letters* "MAC" *in the input field because only these two items contain the substring* "MAC".

3.3.2.4 *History*

The *PDC* should also provide a visual summary of consents previously given. This feature empowers data subjects because it raises awareness about personal data disclosed. Data subjects should be able to access the consents with respect to the *DCP* to which they have consented. Similarly, the *PDC* should be able to display a versioning of the *DSP*. A history could also encompass a summary of the data collected by surrounding devices.

3.4 PROOF OF CONSENT

Proof of consent is a necessary component of our framework. Indeed, demonstration of consent by data controllers is a legal requirement of the GDPR. A lot of research has been conducted on the secure storage (as described in Section 2.2), and we propose a way to reuse and adapt them to our goal: facilitating consent demonstration. Both data subjects and data controllers have an interest in keeping a trace of consents. On the one hand, data subjects wish to ensure their consent cannot be forged, altered nor changed afterwards, and that they can withdraw consent (consent withdrawal is a right under the GDPR); on the other hand, data controllers wish to prove they obtained every consent, and that the proofs cannot be tampered with nor denied.

To verify these properties, the best way is to provide an auditable system. The audit must be conducted by an independent third-party. Such third-party must not have interests colluding with the data controller, and it typically is a DPA. We consider the third-party a DPA in what follows. The auditability of such a system is therefore the most important requirement. However, it is not the only one, and while data controllers have requirements over a system storing consents, data subjects have requirements as well. In this section, we provide details of data subjects and data controllers requirements in Section 3.4.1. These requirements can be achieved using cryptographic properties, described in Section 3.4.2.

A summary of the functions required by the proof of consent is provided in Tables 1 and 2 under **Proof of consent**.

3.4.1 *Requirements*

This section presents the requirements a system must meet in order to securely store consents and to be audited. At this level, these requirements can be seen as functions the system must be able to perform. Each function is composed of one or multiple steps a data controller or a data subject must be able to conduct. Chapter 4 describes how each of these steps can be performed on a technical level.

These functions are not independent of each other, and they must be performed in a certain order. Figure 17 gives an overview of these requirements, and of their interactions. An Audit can be conducted only after the consent has been stored (Archive), this consent would have been verified before being stored (Verification), and the starting point of this temporal sequence is the Generation of the consent by the data subject.

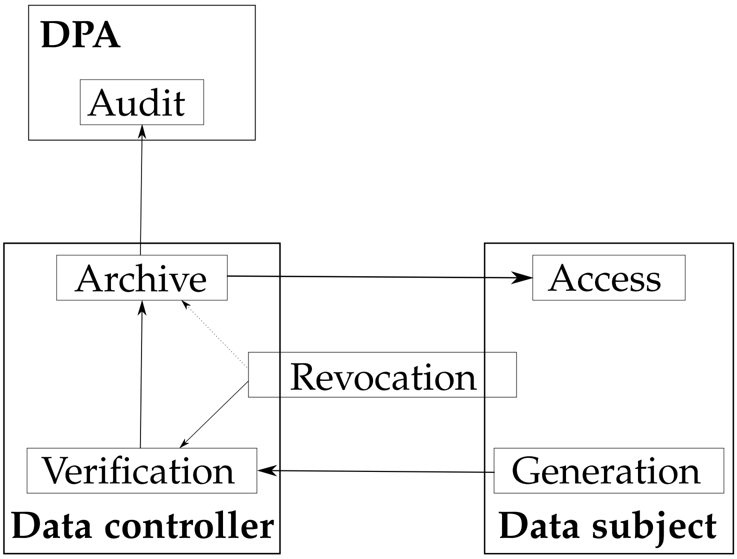


Figure 17: High-level requirements over a system demonstrating the proof of consent. Requirements are inside small boxes. Entities are denoted in **bold** and their range of action is symbolised by the large boxes. The arrows represent a temporal order, except the dashed arrow which represents that the function Revocation impacts the function Archive. Note that Revocation is a requirement for both data controllers and data subjects.

3.4.1.1 *Data controllers requirements*

To proceed to a secure storage of consents compliant with the protocol presented in Section 3.2 and with the GDPR, a data controller must be able to archive consents, verify their well-formedness, accounts for revocation of consents, and enable an audit from a third-party. Note that all data controllers requirements are mandatory.

ARCHIVE We denote *archive* the steps during which consents are stored. The *archive* must provide a **complete** and **tamper-evident** storage of consents. This requirement is needed for the storage of consents.

VERIFICATION We denote *verification* the steps assessing the well-formedness of the consents communicated by the data subject. The *verification* must provide **unforgeability** of the consents communicated, their **non-repudiation** by data subjects, and their **non-impersonation**. This requirement is needed to check the validity of the consents communicated: that they originate from an un-impersonable data subject, and that they correspond to a *DCP*.

REVOCATION We denote *revocation* the steps during which a data controller receives and accounts for a consent withdrawal from a data subject. Accounting for a consent withdrawal means that the data controller modifies the storage of the proofs of consents (the function Revocation impacts the function Archive, see the dashed arrow in Figure 17) so as to stop considering valid the consents previously given by the data subject considered. The *revocation* must provide **unforgeability** of dissents, their **non-repudiation** by data controllers, and their **non-impersonation**. This requirement is needed to provide the withdrawal of consent (as prescribed by the GDPR [53, Art 7.3]).

AUDIT We denote *audit* the steps performed by a DPA on the system. This function can be seen as orthogonal because it is not conducted by the data controller. However, it is the role of the data controller to bring the guarantees necessary to conduct the audit. The *audit* encompasses all properties considered in our context, *i.e.*, it must be possible to verify the compliance of all entries (**completeness**), it must be possible to detect modifications (**tamper-evidence** and **unforgeability**), the entries must be authenticated (**non-impersonation**) and undeniable (**non-repudiation**). This requirement is the highest-level goal of the proof of consent, insofar that the best way to bring guarantees for a technical system is make it auditable.

3.4.1.2 *Data subjects requirements*

For their part, data subjects must be able to generate a consent with certain guarantees, communicate their consent withdrawal, and access a history of their consents for a given data controller. Not all data subjects requirements are mandatory.

GENERATION We denote *generation* the steps during which a data subject produces a consent as well as the guarantees over this consent. The *generation* must provide **unforgeability** of consents, and their **non-impersonation**. This requirement is needed because proofs of consents need to originate from the data subject herself: only the data subject can consent for herself.

REVOCATION We denote *revocation* the steps during which the data subject communicates a consent withdrawal to the data controller (note that this requirement echoes to the data controller requirement of the same name). The *revocation* differs from the data controller side in that the data subject is not involved in the *archive*. However, the *revocation* must provide **non-impersonation** and **unforgeability** (for the *revocation* can only consider consents of a given data subject). This requirement is needed to provide the withdrawal of consent (as prescribed by the GDPR [53, Art 7.3]).

ACCESS (OPTIONAL) We denote *access* the steps enabling a data subject to see her previously given consents. The *access* must provide **non-impersonation**. This optional requirement is needed to provide an overview of the past action of data subject in an empowering perspective.

3.4.1.3 *Mapping with the rest of the framework*

This high-level presentation of the requirements over the proof of consent can be linked with the other parts of the framework. Note that some functions perfectly correspond to components previously described, while others overlap different modules of the protocol.

GENERATION The *generation* of the proof of consent happens on the *DSG*, and it is its communication to the data controller side which maps part of the protocol.

In the proof of consent, the arrow from *generation* to *verification* corresponds to S8 → S1 Send(CONSENT,dcg) in the *DSG* diagram (see Section 3.2.1.3), and to its counterpart S2 → S3 Receive(CONSENT,dsg) in the *DCG* diagram (see Section 3.2.1.4).

REVOCATION The *revocation* of the proof of consent (or more specifically the generation of a proof of *dissent*) happens on the *DSG*, and similar to the *generation*, it is its communication to the data controller side which maps part of the protocol.

In the proof of consent, the arrow from *revocation* to *verification* corresponds to S10 → S1 Send(DISSENT,dcg) in the *DSG* diagram (see Section 3.2.1.3), and to its counterpart S1 → S7 Receive(DISSENT,dsg) in the *DCG* diagram (see Section 3.2.1.4).

ACCESS The *access* by a given data subject to her previous proofs of consent corresponds to a UI described in Section 3.3.2.4. As a matter of fact, the History feature of the *PDC* encompasses the access to previously given consents.

3.4.2 *Properties*

We stated cryptographic properties to achieve the requirements stated above. These properties are more thoroughly detailed in this section. They are drawn from [137] and [149]

COMPLETENESS In our context, completeness means that a *ledger* must store all consents and consents withdrawal received.

TAMPER-EVIDENCE In our context, tamper-evidence (or tamper-detection) is the ability to detect on a ledger any unwilling modification, *i.e.*, any modification which does not correspond to our scheme. Tamper-evidence is required for data controllers because the guarantee of proving consents is the main goal of the *ledger*.

UNFORGEABILITY Unforgeability is the resistance against forgery, *i.e.*, against an attack "trying to fabricate a digital signature for a message without having access to the respective signer's private signing key" [149]. In our context, unforgeability is required against the fabrication of consents. Consents must not be forgeable because it should not be possible to impersonate a data subject (as prescribed by the GDPR [53, Art 7.1]).

NON-IMPERSONATION Impersonation denotes "an attack in which an adversary successfully assumes the identity of one of the legitimate parties in a system or in a communications[*sic*] protocol" [149]. A system with a non-impersonation property is therefore protected against such an attack. This property is typically enforced with a strong authentication. Non-impersonation is required for liability purposes to ensure consents originate from a given data subject.

NON-REPUDIATION Non-repudiation prevents a party from denying the performance of a contract [149]. In our context, non-repudiation means that a data subject cannot deny having sent a consent, and that a data controller cannot deny having received a dissent. Note that it implies that a data subject cannot deny having sent a dissent, and that a data controller cannot deny having received a consent, but we assume these last two events to be against these respective interests.

SUMMARY

We described in this chapter a generic framework for information and consent in the Internet of Things. The framework is composed of a protocol to communicate information and consent, *PPNP* (Section 3.2); guidelines for implementing an agent for human-computer interactions, the *PDC* (Section 3.3); and a solution for proof of consents, the *ledger* (Section 3.4), which focuses on consent. These three components articulate to fulfil the objectives stated in Section 1.2. Tables 1 and 2 respectively summarize our mandatory and optional requirements.

	Rationale	*Entities*	*Features*
Devices	Obj. 1 and 5	*DCG*	1 *DCP*
	Obj. 2, 3, and 5	*DSG*	1 *DSP*
			1 interface
			Hash files / store keys / sign messages
	Obj. 6	*CSS*	Hash files / store keys / sign messages

	Rationale	*Requirements (Content)*
Privacy Policies	Obj. 3 and 4	Req$_{1.1}$ Type of data (cannot be $\emptyset$)
	Obj. 3 and 4	Req$_{1.2}$ Purpose of collection (cannot be $\emptyset$)
	Obj. 3 and 4	Req$_{1.3}$ Retention time (cannot be $\emptyset$)
	Obj. 3 and 4	Req$_{1.4}$ *DC* (cannot be $\emptyset$)
	Obj. 3 and 4	Req$_{1.5}$ 3rd parties (can be $\emptyset$)

	Rationale	*Requirements (Type)*	*Content*
Messages	Functioning of the protocol	Req$_{2.1}$ Consent	Hash of a *DCP*
			Set of identifiers
			Signature
		Req$_{2.2}$ Dissent	Hash of a $\emptyset$ *DCP*
			Set of identifiers
			Signature
		Req$_{2.3}$ Refusal	✗

	Rationale	*Requirements (Features)*
Language	Obj. 5	Req$_{3.1}$ Comparison operation
		Req$_{3.2}$ Intersection operation

	Rationale	*Requirements (High-level features)*	*Edges*
Protocol	Obj. 2	Req$_{4.1}$ *DSG*: Information	S1 → S2: Receive(*DCP*,dcg)
	Obj. 5	Req$_{4.2}$ *DSG*: Consent	S2 → S3: $DSP \geqslant DCP$
			S3 → S1: Send(CONSENT, dcg)
			S6 → S8: $DSP \geqslant DCP$
			S8 → S1: Send(CONSENT,dcg)
			S6 → S9: $DSP \cap DCP = \emptyset$
			S9 → S1: Send(Refusal,dcg)
			S1 → S10: Receive(DISSENT,ds)
			S10 → S1: Send(DISSENT,dcg)
	Obj. 3 and 4	Req$_{4.3}$ *DSG*: Interaction	S2 → S4: $DSP \ngeqslant DCP$
			S4 → S5: Prompt(DCP, ds), Set timer
			S5 → S6: Timer expires
			S5 → S6: Receive(DSP,ds)
	Obj. 1	Req$_{4.4}$ *DCG*: Information	S1 → S3: Send(DCP, dsg)
	Obj. 5 and 6	Req$_{4.5}$ *DCG*: Consent	S2 → S1: Receive(Refusal dsg)
			S2 → S3: Receive(CONSENT, dsg)
			S3 → S1: Send(CONSENT, css)
			S1 → S7: Receive(DISSENT,dsg)
			S7 → S1: Send(DISSENT,css)

	Rationale	*Requirements (Features)*
PDC	Obj. 3	Req$_{5.1}$ Consult DCP
	Obj. 4	Req$_{5.2}$ Consult DSP
	Obj. 4	Req$_{5.3}$ Add/modify/delete
	Link with *PPNP*	Req$_{5.4}$ Notifications
	Legal compliance	Req$_{5.5}$ Natural language
	Obj. 3	Req$_{5.6}$ Rule-per-rule presentation

Table 1: Mandatory requirements for our framework

	Rationale	*Requirements (Functions)*	*Properties required*
Proof of consent	Obj. 6	Req$_{6.1}$ *DC* Archive	Completeness and tamper-evidence
		Req$_{6.2}$ *DC* Verification	Unforgeability, non-repudiation (of consent) and non-impersonation
		Req$_{6.3}$ *DC* Revocation	Unforgeability, non-repudiation (of dissent) and non-impersonation
		Req$_{6.4}$ *DC* Audit	Completeness, tamper-evidence, unforgeability, non-repudiation and non-impersonation
		Req$_{6.5}$ *DS* Generation	Unforgeability and non-impersonation
		Req$_{6.6}$ *DS* Revocation	Unforgeability and non-impersonation

Table 1: Mandatory requirements for our framework.

Rationale denotes either the objectives motivating the requirement or another motivation for a mandatory requirement.

Devices indicates the mandatory devices (under *Entities*) and their mandatory features (under *Features*).

Privacy Policies indicates the mandatory content of privacy policies.

Messages indicates the mandatory messages other than privacy policies. *Content* refers more specifically to their required content if applicable.

Language indicates the mandatory features of a language defining policies.

Protocol indicates the mandatory high-level features for the *DSG* and the *DCG*. *Edges* refers more specifically to the edges of their state diagrams.

PDC indicates the mandatory features of the *PDC*.

Proof of consent indicates the mandatory functions of a secure consent system. *Properties required* refers more specifically to the cryptographic properties required by each requirement.

Devices	*Rationale*	*Entities*
	Multiple *DSDs* support	*DSDs*
	Data collection	*DCDs*

Privacy Policies	*Rationale*	*Requirements (Content)*
	IoT specificities	Req$_{1.6}$ Collection frequency
		Req$_{1.7}$ Location of *DCD*
		Req$_{1.8}$ Collection range

Messages	*Rationale*	*Requirements (Type)*
	Negotiation	Req$_{2.4}$ Deny
	Negotiation	Req$_{2.5}$ Accept

Language	*Rationale*	*Requirements (Features)*
	User-friendliness	Req$_{3.3}$ Categories

Protocol	*Rationale*	*Requirements (High-level features)*	*Edges*
	Fine-grain consent	Req$_{4.6}$ *DSG*: Negotiation	S6 → S7: $(\mathrm{DSP} \not\geq \mathrm{DCP}) \wedge (\mathrm{DSP} \cap \mathrm{DCP} \neq \emptyset)$
			S7 → S1: Send(*DSP*, dcg)
	Fine-grain consent	Req$_{4.7}$ *DCG*: Negotiation	S2 → S4: Receive(*DSP*, dsg)
			S4 → S5: Prompt(*DSP*, dc), Set timer
			S5 → S6: Receive(Accept, dc)
			S6 → S2: Send($\mathrm{DCP} \cap \mathrm{DSP}$, dsg)
			S5 → S1: Receive(Deny, dc) $\vee$ Timer expires

PDC	*Rationale*	*Requirements (Features)*
	Raises awareness	Req$_{5.7}$ History
	User-friendliness	Req$_{5.8}$ Preset items
	Multiple *DSDs* support	Req$_{5.9}$ Bonding
	User-friendliness	Req$_{5.10}$ Sorted presentation

Proof of consent	*Rationale*	*Requirement (function)*	*Property required*
	Raises awareness	Req$_{6.7}$ *DS* Access	Non-impersonation

Table 2: Optional requirements for our framework.
Rationale denotes the motivation for an optional requirement.
Devices indicates the devices optionally required.
Privacy Policies indicates the optional content of privacy policies.
Messages indicates the optional messages other than privacy policies.
Language indicates the optional features of a language defining policies.
Protocol indicates the optional high-level features of the *DSG* and the *DCG*. *Edges* refers more specifically to the edges of their state diagrams.
PDC indicates the optional features of the *PDC*.
Proof of consent indicates the optional functions of a secure consent system. *Properties required* refers more specifically to the cryptographic properties required by each requirement.

4 TECHNICAL OPTIONS

The framework introduced in Chapter 3 is high-level and defines requirements that can be implemented in different ways. In this chapter, we present technical options to implement the framework depending on the context and the capacities of the *DCDs*. We first describe the implementation of the communications between *DCGs* and *DSGs*, which corresponds to the manners to implement *PPNP* (presented in Section 3.2). We consider two approaches for communicating information and consent: direct communication and indirect communication. For each solution, we discuss its compliance with the requirements stated in Chapter 3 and its benefits and limitations.

The main difference between these two approaches is that direct communication uses a peer-to-peer connection while the indirect communication relies on an external medium, *i.e.*, Internet. Another notable difference is for the information phase. Direct information happens through a broadcast channel, and *DCPs* are retrieved passively: the *DCG* initiates the communication. The indirect information happens through an unicast channel and consists in an active request: the *DSG* has to initiate the communication.

Section 4.1 considers direct communications between devices while Section 4.2 focuses on indirect communications. We show in Section 4.3 how the proof of consent can be implemented. Then, we suggest ways to implement the *PDC* (following the requirements presented in Section 3.3) in Section 4.4. We provide an analysis of the design space for information and consent in Section 4.5, and we conclude this chapter with illustrative scenarios in Section 4.6.

At the end of each section, we describe how each technology addresses the requirements set forth in Tables 1 and 2. Direct and indirect communications mostly address the **Protocol** requirements: Information, Consent, and Negotiation. A *ledger* combined with cryptographic signatures addresses the **Proof of consent** requirements. The app for the *PDC* mostly address the **PDC** requirements, but it also details a requirement over the **Proof of consent** (consent signature), and addresses the requirement over the **Protocol** (Interaction). Finally, the language of the *PDC* addresses the **Privacy policies** and the **Operations** requirements. A summary table is provided in each case.

4.1 DIRECT COMMUNICATIONS

A first option to implement information and consent is through direct communications between *DCGs* and *DSGs* (see Figure 18). In

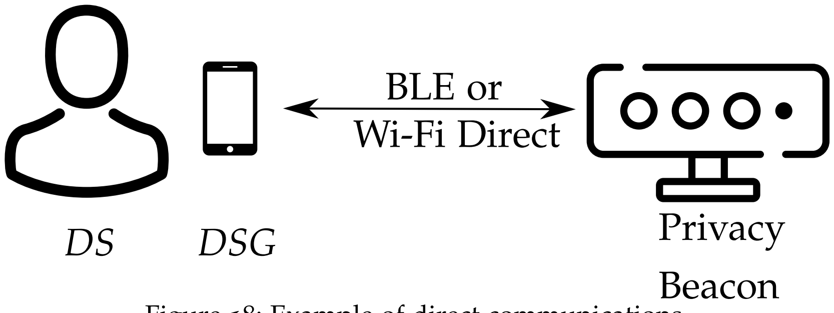


Figure 18: Example of direct communications.

this option (hereinafter "direct communication"), *DCGs* use a direct communication channel to advertise their presence and communicate all the information required, as defined in Section 3.1.2. The same communication channel or a different one can be used by data subjects to transmit their potential consent to the data controller.

4.1.1 *Candidate technologies*

Direct communications can typically be implemented using medium and short range wireless communication technologies, such as Bluetooth or Wi-Fi which are now common place and are embedded in many devices. In addition, their range (several meters to tenths of meters) matches the scale of the area of operation of Internet of Things systems, and their protocol can be leveraged to carry the information required for declaration and consent. As an example, devices such as the Espressif ESP32 [48] can be used as a *DCG*. In this section, we focus on the BLE and Wi-Fi Direct technologies. In what follows, we refer to the *DCG* in direct communications as the *Privacy beacon*. A beacon refers to a passive type of signal in the nautical lexicon, but the term has been used for devices able to actively communicate in the Internet of Things lexicon [142].

4.1.1.1 *BLE protocol*

Bluetooth Low Energy (BLE or LE) features a discovery mechanism that allows the detection and identification of devices as well as the transmission of small amounts of data. This mechanism can be used to implement direct communications between the *DCG* and the *DSG*.

An important distinction to make in the context of the study is the distinction between roles. The Bluetooth system defines a base profile implemented by all Bluetooth devices [20, Vol. 1, Part A, sec. 6.2]. This profile is the Generic Access Profile (GAP), which defines the basic requirements of a Bluetooth device. A device may support multiple LE GAP roles. In LE, GAP defines four specific roles: Broadcaster, Observer, Peripheral, and Central. Only the Central and the Peripheral roles are relevant in our context:

Central The Central role supports multiple connections and is the initiator for all connections with devices in the peripheral role.

Peripheral The Peripheral role is optimized for less complex devices than Central devices.

In the direct communication of consent, the *DCG* endorses the role of Peripheral, while the *DSG* endorses the role of Central. [1] Indeed, the *DSG* has to initiate connections when needed, *i.e.*, when consent is to be communicated. On the other hand, the *DCG*, although possessing more computational capacities, is less complex than the *DSG* as it does not have 1) to sign messages cryptographically, nor 2) to display anything.

ADVERTISING In BLE, device discovery is implemented using *Advertising Packets* [20, Vol. 3, Part C, sec. 11] (AP), that are broadcast at regular intervals and can be received by any device in range with means of communicating in BLE. Those packets are typically used by devices to broadcast information defining their intentions without requiring a connection, and for our needs, they can be configured to carry data necessary for the declaration of *DCDs* (see Section 3.1.2).

A typical manner to convey a *DCP* through these packets is to encode it in a byte format [127]. The byte format must be readable by both *DCG* and *DSG*. A *DSG* in the range of the data controller *Privacy beacon* will thus be able to passively retrieve the declaration data by collecting the Advertising Packets. The *DSG* must then extract the relevant information, *i.e.*, the *DCP*, from the Advertising Packets, and decode that information from a byte format into a text format. This process is illustrated in Figure 19.

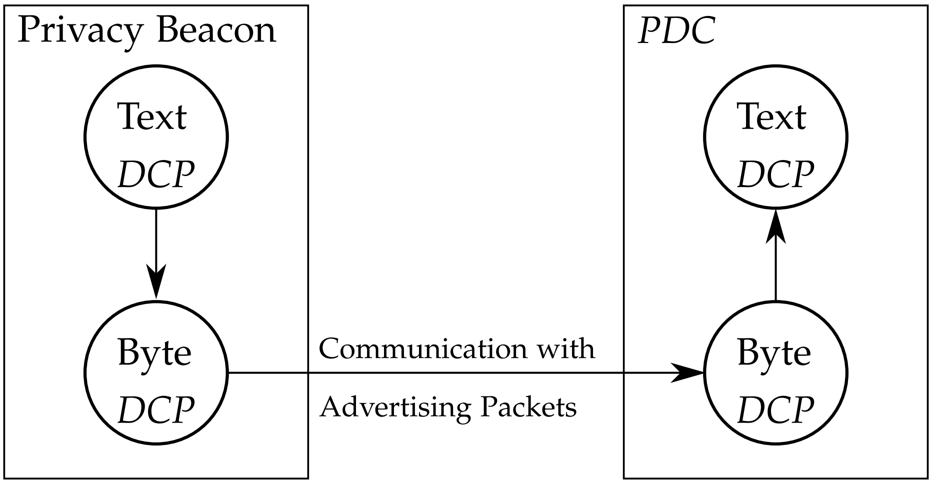


Figure 19: Illustration of the conversion and communication of a *DCP* from a *Privacy beacon* to the *PDC*.

[1] Note that the information phase does not require a connection.

ATTRIBUTE PROTOCOL Another feature offered by BLE is the Attribute Protocol (ATT) [20, Vol. 3, Part F] that allows for the exposure of services and the transmission of small amounts of information through a light-weight connection. ATT defines two roles, the server and the client role. The server defines a set of attributes, and the client can access this set. Note that the same device can encompass the two roles concurrently, but only one server profile shall be defined per device. This feature can be leveraged to implement the communication of consent: the *DSG* connects to the *Privacy Beacon* in a light-weight process and sends the consent data (see Section 3.2) using the ATT protocol.

The *DCG* must possess a Generic Attribute Profile (GATT) to be queried by the *DSG*. GATT is built on top of ATT, it is a "service framework using the Attribute Protocol for discovering services, and for reading and writing characteristic values on a peer device" [20, Vol. 3, Part G]. The *DSG* has then to write messages into a *characteristic* [20, Vol. 3, Part G, sec. 3.3], *i.e.*, a writeable field. This characteristic has to be defined and determined beforehand by both parties. The *DSG* can write strings or bytes in this characteristic. It can therefore communicate a *DSP* in a byte format (see Section 4.1.1.1), or a signed consent in a string format (as required in Section 3.1.2).

4.1.1.2 *Wi-Fi Direct*

Wi-Fi Direct is a protocol developed by Wi-Fi Alliance, whose aim is to provide direct — *i.e.*, one-hop — Wi-Fi communications between devices [158]. It is denoted Wi-Fi Peer-to-Peer (P2P) in its specifications. Similar to BLE, Wi-Fi Direct can be used to implement direct communications between the *DCG* and the *DSG*. Wi-Fi Direct is notably used to communicate with printers.

The protocol also makes a distinction between roles (see [158, Chapter 2]). By default, any device is considered a P2P Device, which can be instantiated into either a P2P Group Owner or a P2P Client.

P2P Group Owner The P2P Group Owner role is similar to an Access Point, on which another P2P Device connects.

P2P Client The P2P Client role is typically for devices to connect to a P2P Group Owner.

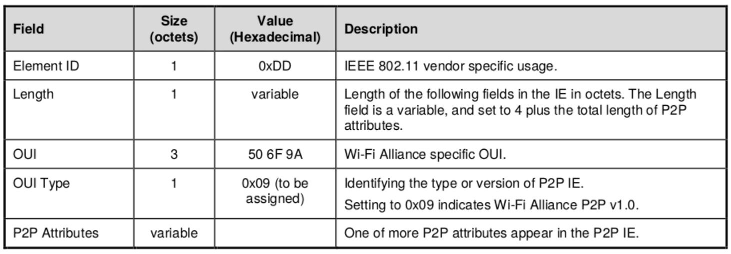
In our case, the *DCG* endorses the P2P Group Owner role, while the *DSG* endorses the role of P2P Client, because the *DSG* can then scan for *Privacy Beacons*.

P2P DISCOVERY In Wi-Fi Direct, the high-level procedure that enables P2P Devices to find each other and form a connection is denoted P2P Discovery (see [158, Chapter 3, Section 1]).

A major component of P2P Discovery is Service Discovery. This feature allows a P2P Device to discover available high-layer services without requiring a connection. The procedure can be used to find a list of services offered by a P2P device, and information about that service. Service Discovery can be used to directly inform data subjects: a *DCG* can broadcast its *DCP* in the Vendor-Specific field (described thereafter) of a service, a *DSG* can then request the list of services, and retrieve the *DCP* if it exists.

Another major component of P2P discovery in Wi-Fi Direct is the Group Formation. This phase is used to determine which device will be the P2P Group Owner. Once the roles have been determined, a connection is formed between the two devices: they can securely exchange data.

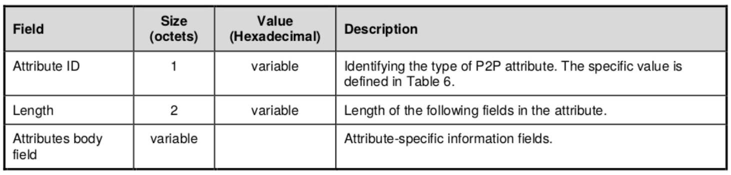
INFORMATION ELEMENT Data can be communicated through P2P Information Element (IE) frames (see Figure 20 and [158, Chapter 4, Section 1]).



Field	Size (octets)	Value (Hexadecimal)	Description
Element ID	1	0xDD	IEEE 802.11 vendor specific usage.
Length	1	variable	Length of the following fields in the IE in octets. The Length field is a variable, and set to 4 plus the total length of P2P attributes.
OUI	3	50 6F 9A	Wi-Fi Alliance specific OUI.
OUI Type	1	0x09 (to be assigned)	Identifying the type or version of P2P IE. Setting to 0x09 indicates Wi-Fi Alliance P2P v1.0.
P2P Attributes	variable		One of more P2P attributes appear in the P2P IE.

Figure 20: P2P IE format

These frames contain a field P2P Attributes (see Figure 21), which itself contains an Attributes body field (see Figure 22). The attribute ID 221 "Vendor specific attribute" can be used for our purposes, *i.e.*, to convey the messages required by *PPNP*.



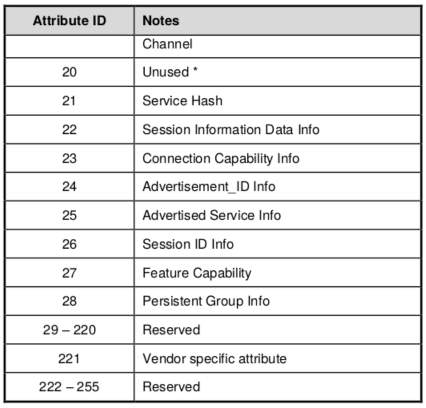
Field	Size (octets)	Value (Hexadecimal)	Description
Attribute ID	1	variable	Identifying the type of P2P attribute. The specific value is defined in Table 6.
Length	2	variable	Length of the following fields in the attribute.
Attributes body field	variable		Attribute-specific information fields.

Figure 21: General format of P2P attribute

4.1.2 *Benefits and limitations*

4.1.2.1 *Benefits*

Direct communications have several benefits: 1) they do not require Internet connectivity: the locality of the communications reduces the risk of further tracking by a remote entity; 2) from the point of view of the data subject, the information part is collected passively by collecting the data transmitted by the *Privacy Beacon*. This means that,



Attribute ID	Notes
	Channel
20	Unused *
21	Service Hash
22	Session Information Data Info
23	Connection Capability Info
24	Advertisement_ID Info
25	Advertised Service Info
26	Session ID Info
27	Feature Capability
28	Persistent Group Info
29 – 220	Reserved
221	Vendor specific attribute
222 – 255	Reserved

Figure 22: P2P Attribute ID definitions

in order to be informed, the data subject does not expose his presence; and 3) the cost of *DCGs* is affordable (the ESP32 mentioned before costs 6 $ apiece).

4.1.2.2 *Limitations*

Direct communications also raise several challenges: 1) all devices should be able to declare themselves. Tracking systems involving passive devices thus need to be enhanced (for example with an additional *Privacy Beacon*) to enable these declarations; 2) the communication protocol should support the communication of the messages described in Section 3.1.2; and 3) in addition, the coverage of the declaration mechanism should match the area in which the data collection is taking place.

4.1.2.3 *Fulfilment of requirements*

Direct communications fulfil the requirements Information, Consent, and Negotiation of the **Protocol** presented in Tables 1 and 2. Table 3 summarizes how direct communications fulfil our requirements.

Name of the requirement	*Type of requirement*	*Technology able to fulfil it*	
		BLE	Wi-Fi Direct
Req$_{4.1}$ and Req$_{4.4}$ Information (*DSG* and *DCG*)	Mandatory	AP	Service Discovery
Req$_{4.2}$ and Req$_{4.5}$ Consent (*DSG* and *DCG*)	Mandatory	ATT	P2P Discovery
Req$_{4.6}$ and Req$_{4.7}$ Negotiation (*DSG* and *DCG*)	Optional	ATT	P2P Discovery

Table 3: Fulfilment of requirements by direct communications

4.2 INDIRECT COMMUNICATION

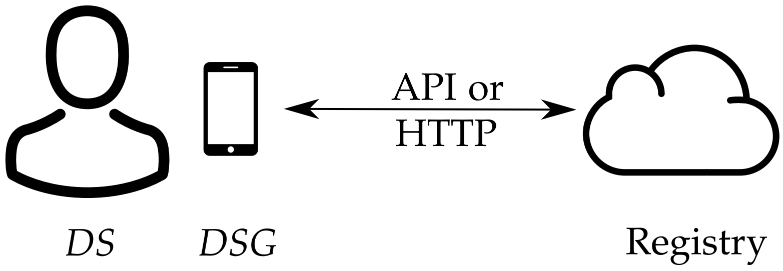


Figure 23: Representation of indirect communications.

Another option to implement information and consent is to use a *registry* (hereinafter "indirect communication") (see Figure 23). *Registries* can be used both by data controllers (to inform data subjects), and by data subjects (to communicate consent). A *registry* is a database freely accessible through the Internet, storing all relevant information about *DCDs*, including the *DCPs*. They must provide the required information in machine-readable format queryable by a *PDC*. They should also include a human-readable version that can be consulted directly by data subjects.

4.2.1 *Candidate technologies*

Unlike a *privacy beacon*, a *registry* can have an interface. The interface can take the form of a website, as it can provide front-ends readable by machines (see Sections 4.2.1.1 and 4.2.1.2) and by humans directly (see Section 4.2.1.3), or as versatile solutions for back-ends such as distributed databases. Figure 24 provides an overview of how these components can be articulated, and we present each component in this section.

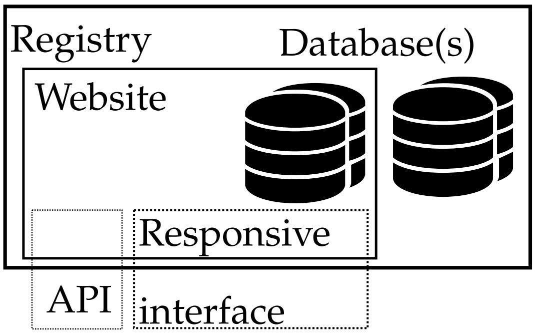


Figure 24: Integration of the different components of a *registry*

4.2.1.1 *HTTP*

The most versatile way to communicate with a website is by the well-known Hypertext Transfer Protocol (HTTP). HTTP was initiated by Tim-Berners Lee [104]. HTTP defines methods to perform different tasks on a resource, and two of them are suitable for our requirements. They were defined as follows:

GET "The GET method means retrieve whatever information (in the form of an entity) is identified".

POST "The POST method is used to request that the destination server accept the entity enclosed in the request as a new subordinate of the resource identified."

HTTP can be secured by using HTTPS, an extension of HTTP encrypted with Transport Layer Security (TLS).

INFORMATION Indirect information can be implemented through the application of a GET to a *DCP* or a list of *DCPs*.

CONSENT Indirect consent can be implemented through the application of a POST to a CONSENT message. Negotiation can be implemented through the application of GET and POST methods to *PPNP*.

4.2.1.2 *API*

For *registries* to be automatically requested by a *PDC*, their front-end can also include an Application Programming Interface (API). APIs can be queried by most devices, since such queries are usually not demanding in terms of computational capacities.

Roy Thomas Fielding presents in [56, Chapter 5] a type of API denoted REST for Representational State Transfer. This type of API architecture is widely used [164] and supported by various tools. A RESTful API can be managed with HTTP methods, which are typically used as follows in a RESTful API:

GET The GET method retrieves a representation of the member resource in the response body.

POST The POST method creates a member resource in the member resource using the instructions in the request body.

INFORMATION Indirect information can be implemented through the application of a GET to a *DCP*. Information must apply to a *DCP* or a list of *DCPs*. Common formats to retrieve data from an API are JSON [23] and XML [129].

consent Indirect consent can be implemented through the application of a POST to a CONSENT message. Negotiation is applying GET and POST methods to *PPNP*.

4.2.1.3 *Responsive interface*

Websites can be accessed from a desktop browser or a mobile browser in addition to a *PDC*. Websites can be queried outside of a *PDC* by different types of devices if they possess a responsive interface. A responsive interface is an interface which takes into account the difference in size of screens (*e.g.*, common sizes of a desktop screen ranges from 13 to 15", while a mobile screen is usually around 5"). A responsive interface can be implemented using frameworks such as Bootstrap [121], or Materialize [46] (based on the Material Design principles [96]) for instance.

information A responsive presentation of information can be communicated to any device endowed with a screen. A responsive interface should however only be used for information. A registry can also be accessed through a *PDC*'s interface using an API as explained in the previous section, as well as through a responsive interface.

consent Consent should only be communicated through the *PDC*, and *not* through a responsive interface. As a matter of fact, only the *PDC* ensures the requirements over consent stated in Section 3.4 (the *generation* first of all). The next section explains how these requirements are met.

4.2.2 *Benefits and limitations*

4.2.2.1 *Benefits*

Indirect communications have several advantages compared to direct communications: 1) they enable the consultation of *DCPs* regardless of the location of data subjects, which means that data subjects can be informed about the collection of data before visiting an area; and 2) they provide a flexible management approach for *DCPs* — they do not require a specific infrastructure or particular capabilities of the devices except for an Internet connection. Therefore, they can be well-suited to passive devices such as cameras.

4.2.2.2 *Limitations*

However, implementation of indirect communications raises several challenges: 1) *DSDs* must always be aware of all surrounding devices: *registries* should be easily accessible; 2) *registries* must be properly managed, up-to-date and accurate in order to meet the requirements defined in the previous section.

4.2.2.3 *Fulfilments of requirements*

Indirect communications fulfil the requirements Information, Consent, and Negotiation of the **Protocol** presented in Tables 1 and 2. Table 4 summarizes how indirect communications fulfil our requirements.

Name of the requirement	*Type of requirement*	*Technology able to fulfil it*
$\mathrm{Req}_{4.1}$ and $\mathrm{Req}_{4.4}$ Information (*DSG* and *DCG*)	Mandatory	HTTP and API (GET), responsive interface
$\mathrm{Req}_{4.2}$ and $\mathrm{Req}_{4.5}$ Consent (*DSG* and *DCG*)	Mandatory	HTTP and API (POST)
$\mathrm{Req}_{4.6}$ and $\mathrm{Req}_{4.7}$ Negotiation (*DSG* and *DCG*)	Optional	HTTP and API (GET and POST)

Table 4: Fulfilment of requirements by indirect communications

4.3 PROOF OF CONSENT

We described our requirements for proof of consent in Section 3.4.1. We propose here a manner to meet this requirements combing a secure and distributed *ledger* and cryptographic signatures, as illustrated by Figure 25.

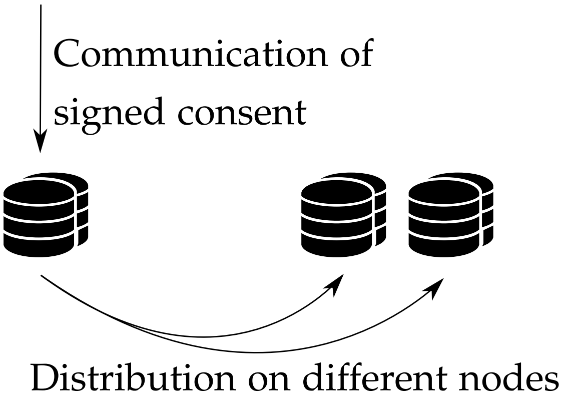

Communication of signed consent
Distribution on different nodes

Figure 25: Distributed ledger

The *ledger* provides the **completeness** and **tamper-evidence** required for the *archive* and *audit*. The signatures provide the **unforgeability**, **non-repudiation** and **non-impersonation** required for the *generation*, *verification* and *revocation*.

4.3.1 *Ledger*

A distributed *ledger* using Merkle Hash Trees, as proposed by [117], provides the required **completeness** and **tamper-evidence** required for *archive* and *audit*.

4.3.1.1 *Data structure*

A Merkle Hash Tree consists in an authenticated binary search tree of a set of elements $x_1, x_2 \dots x_i$, whose leaves contain the hash values $h(x_i)$ of the elements (see Figure 26).

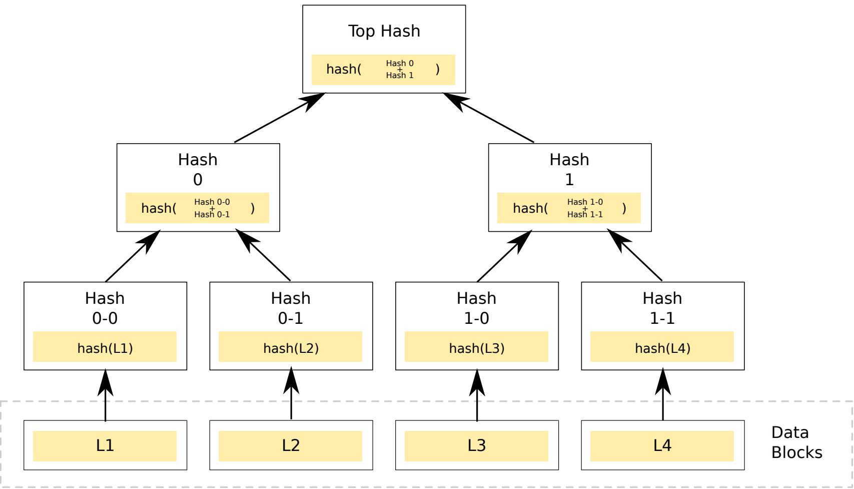


Figure 26: Illustration of a Merkle Hash Tree by Azaghal [2]

The father nodes contain the hash of the concatenation of their children, *e.g.*, a non-leaf node whose children are v_1 and v_2 will contain $h(v_1||v_2)$. The root node is therefore unique with respect to its leaves, it is signed by a private key, and this signature is the only information needed to trust the rest of the tree's content. Merkle Hash Trees are widely used in peer-to-peer networks such as BitTorrent, to verify content integrity.

Such a ledger can provide **completeness** and **tamper-evidence** by distributing the consents (both their plain text and signed versions) on nodes controlled by third parties. Distribution is necessary to ensure the **tamper-evidence**: an entity controlling all the nodes could otherwise modify the content of the *ledger* without being detected. In that case, a third party could be an audit agency, or any other peer: it does not have to be a trusted third party, but it must provide access to competent authorities for accountability purposes. The only requirement about the third party is that it must **not** be controlled nor influenced by a data controller. It typically refers to a *honest-but-curious* scenario in computer security terms [38]. As a matter of fact, the ledger can be compromised if the same actor controls all nodes (which refers to a *malicious users* scenario). It is therefore of prime importance that third parties do not have any interest in providing access to the data controller. The Merkle Hash Tree containing consents is replicated on the nodes, and any undesired modification is detected as soon as the tree is reconstructed.

[2] Own work, CC0, https://commons.wikimedia.org/w/index.php?curid=18157888

4.3.1.2 *Implementation*

An implementation of such a *ledger* is provided by Hypercore, the register of the Dat protocol.

THE DAT PROTOCOL Dat [117] is a protocol for distributed data synchronisation. It has been designed primarily for data-driven science, in order to keep track of changes in large datasets. It is possible to fully or partially replicate the content of a remote Dat repository over different peers. Dat has been implemented in Javascript. The most interesting part in our context is the register used to prove the integrity of data distributed.

HYPERCORE The Dat storage, content integrity, and networking protocols are implemented in a JavaScript module called Hypercore, whose source code is freely available [26]. Hypercore is agnostic to the format of the input data, it operates on any stream of binary data.

Hypercore is made of registers, Hypercore Registers, which are the core mechanism used in Dat. They are binary append-only streams whose contents are cryptographically signed (see Section D.1) and hashed (see Section D.2), and can therefore be verified by anyone with access to the public key of the writer.

Dat uses two registers, content and metadata. The content register contains the files in the repository, and the metadata register contains the metadata about the files including its name, size, last modified time, *etc.* Dat replicates the two registers when synchronizing with another peer.

When files are added to Dat, each file gets split up into a number of chunks depending on the size of the file, and the chunks are then arranged into a Merkle Hash Tree (see Section 4.3.1), which is used later for version control and replication processes.

In our case, the files are consents (defined in page 41). Adding a consent $Consent_1$ to the *ledger* splits the consent into chunks (it may be only one chunk if the size of the consent is small enough). The metadata of $Consent_1$ — such as its name, size, date of last modification *etc.*— are compared to metadata of existing files. If the exact same file already exists, the file is skipped. Otherwise, the chunks of $Consent_1$ are arranged into a Merkle Hash Tree, and distributed over the different nodes hosting the *ledger*. In our context, a tree corresponds to a *ledger*. A tree is used to store consents, but their representation and their order is agnostic of the underlying storage.

Hypercore can be easily and quickly implemented as a JavaScript server using `npm` [116], the Node.js's package manager. A minimal working example can be found in Listing 7 of Appendix C.

4.3.2 *Cryptographic signatures*

We stated in Section 3.4 that consents must be generated in order to be verified before being archived. This step can be done through cryptographic signatures (see Appendix D). Cryptographic signatures provide the **unforgeability**, **non-repudiation**, and **non-impersonation** properties required for the *generation*, the *verification*, and the *revocation* (through the authentication of dissents) requirements over the proof of consent.

A consent signed by a data subject's private key is then guaranteed to originate from the said data subject (**non-impersonation**), to the extent that the private key is not compromised. Moreover, the content signed cannot be altered without modifying the signature. As data controller cannot forge data subjects private keys, cryptographic signatures provide **unforgeability**. Data subjects cannot deny the signature for the same reason: consent signature also provide **non-repudiation**.

4.3.3 *Other requirements*

The above description of the *ledger* combined with cryptographic signatures provides insights about the fulfilment of some requirements for the proof of consent. However, *revocation* and *access* have not been explicitly addressed yet. This section presents how these two requirements are met.

4.3.3.1 *Revocation*

One mandatory requirement cannot be met by merely considering the *ledger* and the signature of consents: *revocation*.

ORDER OF ENTRIES To tackle this issue, the order of entries in the *ledger* has to be considered. In our context, an entry is either a consent or a dissent. As mentioned in Section 3.2, data subjects may object to a previously-sent consent by communicating a *dissent*. A dissent is a consent to a nil *DCP*. Data controllers must therefore always take into account the last input from a data subject in order to ensure consent withdrawal. Note that data controllers are allowed to conduct processing on data before the dissent is received.

PREVENTING REPLAY ATTACKS *Revocation* paves the way for malicious behaviour. While a data controller does not have interest in not storing a consent, it may deny the reception and storage of a dissent in order to continue processing after the reception of a consent.

To prevent replay attacks where the data controller is malicious, and would supersede a withdrawal with a consent previously communicated, it is possible to enhance the protocol with the issuing of a nonce signed along the consent. The *DSG* has to request the nonce before communicating consent, the nonce is issued by the *DCG*. The consent is then signed with the nonce, ensuring the uniqueness of the consent communicated.

Note that a history of consents is optionally kept on the *DSG* to establish the proof of consent communication, and more importantly dissent communication: data controllers cannot deny a consent withdrawal even if they do not store it if a trace is kept on the *DSG*.

4.3.3.2 *Access (Optional)*

The optional requirement of *access* can be fulfilled in two ways. It only requires the **non-impersonation** property.

The first way consists in keeping a local version of the history of consents on the *PDC* (see above). This manner does not require additional authentication measures, and can be conducted by providing access to the logs storing consents communication of the *PDC*.

The second way consists in providing selective access to the *ledger*. This selective access must only encompass entries for which the data subject can be authenticated. Authentication must be as strong as the one required for *generation* and *verification*, *i.e.*, it must use cryptographic proofs.

4.3.3.3 *Fulfilments of requirements*

A combination of a *ledger* and cryptographic signatures fulfils our requirements for the proof of consent. Note that how cryptographic signatures are handled by the *PDC* is explained in Section 4.4.1. Table 5 summarizes how our requirements are fulfilled.

Name of the requirement	*Type of requirement*	*Technology able to fulfil it*
$\text{Req}_{6.1}$ *DC* Archive	Mandatory	Ledger
$\text{Req}_{6.2}$ *DC* Verification	Mandatory	Cryptographic signatures
$\text{Req}_{6.3}$ *DC* Revocation	Mandatory	Cryptographic signatures & Order of entries
$\text{Req}_{6.4}$ *DC* Audit	Mandatory	Ledger
$\text{Req}_{6.5}$ *DS* Generation	Mandatory	Cryptographic signatures
$\text{Req}_{6.6}$ *DS* Revocation	Mandatory	Cryptographic signatures
$\text{Req}_{6.7}$ *DS* Access	Optional	Cryptographic signatures & History of consents

Table 5: Fulfilment of requirements by the different technologies proposed

4.4 PERSONAL DATA CUSTODIAN

Direct and indirect communications only concern machine-to-machine communications, and therefore the implementation of *PPNP*. They should be complemented with communication means between the *PDC* and data subjects. We describe in this section how a *PDC* meeting the requirements defined in Section 3.3 can be implemented (see Figure 27). We first present how it can be implemented as an application ("app" in what follows) for both Android and iOS in Section 4.4.1, we then describe in Section 4.4.2 a privacy policy language fulfilling the requirements set forth in Section 3.1.2.

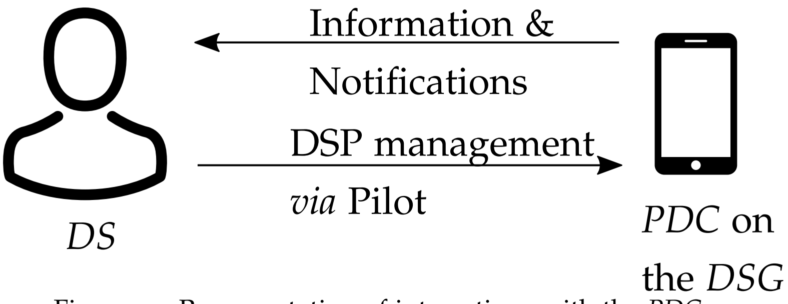


Figure 27: Representation of interactions with the *PDC*.

4.4.1 *App*

A *PDC* is a software agent on a *DSG*. Considering that most adults today own a smartphone [110], with a screen and capacities to run a *PDC*, an app appears to be the most appropriate to implement a *PDC*.

The two main platforms for apps are Android (owned by Google and totalling around 84% of mobile market shares late 2018) and iOS (owned by Apple and totalling around 10% of mobile market shares early 2019). We focus on Android, as the prototype presented in the next chapter has been developed as an Android app. We chose Android because it is the most used mobile operating system [165], and therefore the most promising to reach a large audience. [3] We also present more briefly how a *PDC* can be developed on iOS. We examine how an app is appropriate for our objectives in the next two sections.

4.4.1.1 *Android*

This section describes how Android satisfies our requirements for the *PDC*. Listings of code excerpts are provided in Appendix C. We

[3] It may be argued that Android is not the most privacy-friendly platform because that it is owned by a company known to spy on emails [95], for instance [126]. However, alternative mobile Android OSs such as LineageOS [92] exist. They considerably reduce the dependency to Google because a Google account is not required to make use of all features, and still run Android apps.

successively present how *Activities*, *Notifications*, *BLE*, *Wi-Fi direct*, and *Cryptographic signatures* can be leveraged to satisfy our requirements.

ACTIVITIES On Android, a display corresponds to an *activity*. To be more precise, an activity is a combination of a window and a sheet of code in Java or Kotlin. The window is made of elements — *e.g.*, buttons, text fields, lists, autocomplete — which can trigger functions in the code sheet. Users interact only with the window part, which then interacts with the code sheet, and therefore the back-end of the app. To each UI (required in Section 3.3) must correspond an activity.

Users can navigate (which corresponds to the arrows presented in Figure 16) between activities using *intents*. Intents are messaging objects one can use to request an action from an activity. They trigger activity changes, and can convey information.

As a result, activities are perfectly suitable to implement the different UIs and the navigation between them, as required in Section 3.3.

NOTIFICATIONS If users are solicited by a notification (see Section 3.3.1.4), Android provides facilities to inform them [115]. A notification can appear as four different visual elements, three of which are suitable for our use case:

status bar A notification first appears as an icon in the status bar. Users can swipe down on the status bar to open the notification drawer, where they can view more details and take actions with the notification. Users can drag down on a notification in the drawer to reveal the expanded view, which shows additional content and action buttons, if provided. A notification remains visible in the notification drawer until dismissed by the app or the user.

heads-up Since Android 5.0, notifications can briefly appear in a floating window called a heads-up notification. This behaviour is normally for important notifications that the user should know about immediately, and it appears only if the device is unlocked. The heads-up notification appears the moment an app issues the notification and it disappears after a moment, but remains visible in the notification drawer as usual.

lock screen Since Android 5.0, notifications can also appear on the lock screen.

These three elements — status bar, heads-up, and lock screen notifications — can efficiently represent the notifications required in Section 3.3.1.4. Indeed, when a request actively notifies data subjects, a heads-up notification covering over the current app activity or, less intrusively, a simple status bar notification can be used. A lock screen notification ensures data subjects do not miss a request when not using their *DSG*.

BLE A *PDC* must be able to communicate either directly or indirectly, and ideally both. Whereas communicating indirectly is trivial — it consists in retrieving and posting JSON files — communicating directly is technically more difficult.. We now examine the facilities provided by Android for managing BLE, as it is one of the technology described in Section 4.1.1. Android introduced BLE support in version 4.3 (API 18) [19], and refined it in versions 5.0 (API 21) [130] and 6.0 (API 23) [19]. Most of the excerpts are examples from [19]. The interested reader may find tailored implementations of these concepts in [94].

As far as information is concerned, BLE on Android has packages for scanning peripheral devices which use advertisement packets. For instance, one can scan for BLE devices using the snippet provided in Listing 1 (page 133). For every BLE device found, the Android app (in our case, the *PDC*) triggers a function named a *callback* (see Listing 2 page 134). It is then possible to inspect the content of the advertisement packets communicated, and to retrieve the *DCP*.

As far as consent is concerned, it is also possible to write strings in GATT (see Section 4.1.1.1), and therefore to communicate a *DSP* in order to consent if the policies match, or to negotiate (see Section 3.2). As an example, Listing 3 (page 135) shows how to connect to a GATT server. Listing 4 (page 136) shows how to retrieve a descriptor (an optional attribute nested in a characteristic that describes the specific value and how to access it, see Section 4.1.1.1), and write a consent in a GATT characteristic using this descriptor (see [13] for a full working example, and [94] for the source code of the prototype presented in a next chapter).

WI-FI DIRECT We presented in Section 4.1.1.2 another protocol to communicate directly: Wi-Fi Direct. Similar to BLE, Wi-Fi Direct enables the communication of information and consent.

Wi-Fi direct has features similar to BLE for directly communicating information and consent. As far as information is concerned, it is possible to discover nearby services and to retrieve information about them (see [145]). As far as consent is concerned, Android provides facilities for secure connections (see [37]).

CRYPTOGRAPHIC SIGNATURES Android provides facilities to create and import private keys, as well as to sign and to verify data using asymmetric keys [7]. The Android keystore system manages different hash functions and algorithms, including AES, RSA, and SHA512. The RSA algorithm is appropriate for our scenario, notably with a sufficiently long key size (up to 4096 bits). For instance, it is possible to create a pair of keys using the simple Listing 5 (page 136).

Signing data is straightforward, as shows Listing 6 (page 137). Note that consents must be communicated both in plain text and in their signed version as stated in Section 3.1.2.2.

4.4.1.2 *iOS*

The most used mobile operating system after Android is iOS. In this section we succinctly present some important components of iOS app which, without getting into the details, demonstrate the same ability to communicate than Android.

WINDOWS On iOS, a UI corresponds to a *window*. A window is a container for the content of an app, and displays the content managed by a view controller [159]. A window is made of elements such as text — lists, autocomplete *etc.*—, buttons, which can trigger functions in the controller.

Similar to Android's activities, iOS's windows are suitable to implement the different UIs required.

NOTIFICATIONS Notifications are handled natively in iOS [114]. A notification can appear in one of these two styles:

Banner "Appears at the top of the screen for a few seconds while the device is in use, then disappears."

Alert "Appears at the top of the screen while the device is in use and stays there until manually dismissed."

Notifications can also appear on the lock screen.

BLE iOS can manage BLE communications through its Core Bluetooth framework [3]. An iOS app can scan for advertising BLE devices, and communicate by writing in GATT [125].

KEYCHAIN iOS manages keys and passwords through an API coined Keychain [74]. It is possible to create and import keys [75], as well as sign and verify data.

4.4.1.3 *Fulfilments of requirements*

An app developed either for Android or iOS fulfils the requirements of the **PDC** presented in Tables 1 and 2, as well as the consent signature of the **Proof of consent**. It fulfils the Interaction requirement of the **Protocol** as well. Table 6 summarizes how an Android or an iOS app fulfils our requirements.

4.4.2 *Language*

As discussed in the introduction, consent is valid in the sense of the GDPR only if it is freely given, specific, informed and unambiguous. Each of these conditions brings forward strong requirements on the *PDC*, more particularly on the language used to express privacy policies:

Name of the requirement	*Type of requirement*	*Technology able to fulfil it*	
		Android	iOS
Req$_{5.1}$ Consult *DCP*	Mandatory	Activities (text and lists)	Windows (text and lists)
Req$_{5.2}$ Consult *DSP*	Mandatory	Activities (text and lists)	Windows (text and lists)
Req$_{5.3}$ Add/modify/delete	Mandatory	Activities	Windows
Req$_{5.4}$ Notifications	Mandatory	Notifications	Notifications
Req$_{5.5}$ Natural language	Mandatory	Activities (text and lists)	Windows (text and lists)
Req$_{5.6}$ Rule-per-rule presentation	Mandatory	Activities (text and lists)	Windows (text and lists)
Signatures (see Section 4.3)	Mandatory	Android signatures	Keychain
Req$_{4.3}$ *DSG* interaction	Mandatory	Activities and notifications	Windows and notifications
Req$_{5.7}$ History	Optional	Activities (text and lists)	Windows (text and lists)
Req$_{5.8}$ Preset items	Optional	Activities (autocomplete)	Windows (autocomplete)
Req$_{5.9}$ Bonding	Optional	Activities (buttons)	Windows (buttons)
Req$_{5.10}$ Sorted presentation	Optional	Activities (text and lists)	Windows (text and lists)

Table 6: Fulfilment of the PDC and the Protocol (Interaction) requirements by Android and iOS apps

- *Consent must be freely given:* any personal data and privacy policy communicated should reflect the genuine choices of the data subjects.
- *Consent must be specific:* the privacy policy language must be rich enough to allow data subjects to express granular choices, for example about the types of data, data controllers or authorized purposes.
- *Consent must be informed:* the *PDC* must not disclose personal data to a *DCD* that has not communicated its *DCP*.
- *Consent must be unambiguous:* in order to avoid any ambiguity, the privacy policy language should be endowed with a formal semantics and the interface used to interact with the data subject should not be misleading.

A privacy policy language meeting these requirements is described in [123]: Pilot. Pilot was designed to enhance informed consent and consequently meets the above requirements. The syntax of Pilot policies is defined as follows:

$$\begin{aligned}
\text{Pilot}\, \textit{Privacy Policy} \quad &::= \quad (datatype, dcr, TR) \\
\textit{Data Communication Rule}\ (dcr) \quad &::= \quad \langle condition, entity, dur \rangle \\
\textit{Data Usage Rule}\ (dur) \quad &::= \quad \langle Purposes, retention\_time \rangle \\
\textit{Transfer Rules}\ \ (TR) \quad &::= \quad \{dcr_1, dcr_2, \ldots\}
\end{aligned}$$

The abstract syntax of Pilot therefore consists in a *Pilot Privacy Policy*, which comprises a *datatype*, a *dcr* (for Data Communication Rule) and a *TR* (which stands for Transfer Rule). A *TR* a set of *dcr*. A *dcr* comprises an *entity*, a *dur* (for Data Usage Rule), and possibly *condition*. Finally, a *dur* is composed of one or more *Purposes* as well as a *retention time*. Figure 28 provides a visual explanation of Pilot's structure.

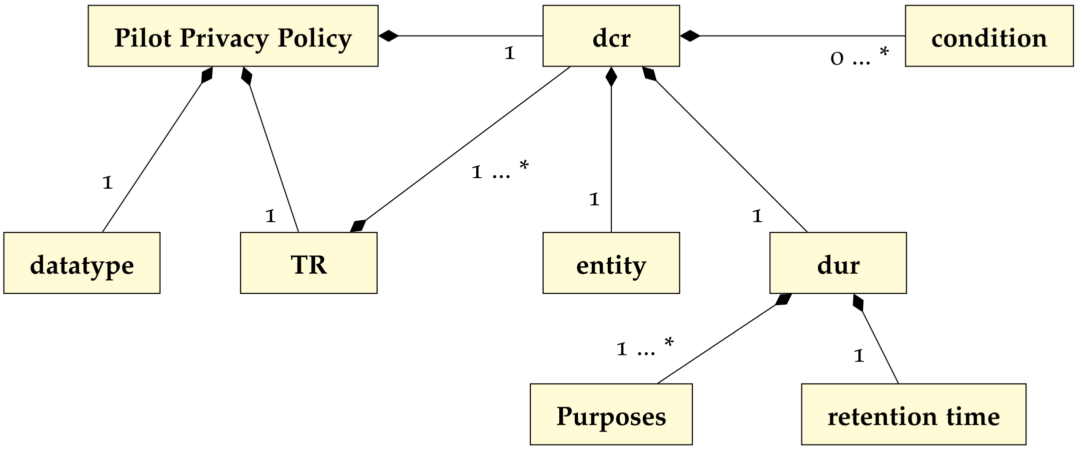


Figure 28: Pilot high-level structure in a UML fashion. Entities are joined by composition links. Multiplicity should be understood as follows: *1* denotes a requirement for the framework, *0...1* and *0...∗* denotes an option for the framework, note that an empty set of Purposes corresponds to an empty policy, *dcr* stands for Data Communication Rule, *dur* stands for Data Usage Rule, *TR* stands for Transfers Rules.

In Pilot, what they denote as a Pilot Privacy Policy corresponds to one rule of our privacy policies: [4]

$$rule ::= Pilot\ Privacy\ Policy$$

See example 8 for clarifications:

Example 8. *The following is an example of a DCP rule in* Pilot*:* $rule_1 ::= (datatype = Email\ address, dcr = \langle condition = \emptyset, entity = Villeurbanne, dur = \langle Purposes = \{Audience\ Measurement\}, retention\_time = 15\rangle\rangle, TR = \emptyset)$

4.4.2.1 *Operations*

Two operations are relevant in our context: the Policy Subsumption and the Policy Join. Pilot provides strong guarantees thanks to its formal semantics, we refer the interested reader to [123, Appendix] for the proofs.

[4] As a result, one of our privacy policy can encompass several types of data.

POLICY SUBSUMPTION It is a formal definition of what we denote the Comparison operation. Given two Pilot privacy policies $\pi_1 = \langle t_1, dcr_1, TR_1 \rangle$ and $\pi_2 = \langle t_2, dcr_2, TR_2 \rangle$, Pilot defines a notion of subsumption between policies such that π_1 *subsumes* π_2 is equivalent to $\pi_1 \leqslant \pi_2$ in our terms. Consent is communicated iff the *DCP* subsumes the *DSP*.

POLICY JOIN Pilot also introduces a notion of policy join which is a formal definition of what we denote the Intersection operation. Given two Pilot policies $p = (t_1, dcr_1, TR_1)$ and $q = (t_2, dcr_2, TR_2)$, the result of a policy join is more restrictive than both operands.

4.4.2.2 *Categories*

Pilot defines a partial order over *entities*, *datatypes*, and *purposes*, which implement our optional requirement over categories. For instance, since the company Google belongs to the company Alphabet, a rule defined for the *entity* Alphabet applies to the *entity* Google as well.

4.4.2.3 *Fulfilments of requirements*

The Pilot privacy policy language allows for the expression of all our mandatory and optional content. Table 7 summarizes how Pilot fulfils these requirements.

MANDATORY REQUIREMENTS In Pilot's terms, the framework requires the *datatype* (type of data in our requirements), a *dcr* comprising an *entity* (*DC* in our requirements) and a *dur*, which itself comprises the purpose and the retention time (both terms are denoted the same way in our requirements). *TR* is required but can be empty, they represent 3rd parties when included in a policy (which is not always the case: a data controller may very well never disseminate data). The requirement Comparison is fulfilled by Policy Subsumption.

OPTIONAL REQUIREMENTS In Pilot the *conditions* field is required but it can be the value True. They represent the optional requirements over the content of privacy policies: frequency of collection, location of *DCDs*, and range of collection. The requirement Intersection is fulfilled by the Policy Join. The requirement Categories is fulfilled by Pilot's partial order.

4.5 DESIGN SPACE

As illustrated by the previous sections, a variety of technical solutions can be implemented to make information and consent more effective in Internet of Things environments. However, depending on the precise context, in particular the features of the devices, not all options are

Name of the requirement	*Type of requirement*	*Technology able to fulfil it*
$\mathrm{Req}_{1.1}$ Type of data	Mandatory	Pilot *datatype*
$\mathrm{Req}_{1.2}$ Purpose of collection	Mandatory	Pilot *Purposes*
$\mathrm{Req}_{1.3}$ Retention time	Mandatory	Pilot *retention time*
$\mathrm{Req}_{1.4}$ *DC*	Mandatory	Pilot *entity*
$\mathrm{Req}_{1.5}$ 3^{rd} parties	Mandatory	Pilot *TR*
$\mathrm{Req}_{1.6}$ Collection frequency	Optional	Pilot *conditions*
$\mathrm{Req}_{1.7}$ Location of *DCD*	Optional	Pilot *conditions*
$\mathrm{Req}_{1.8}$ Collection range	Optional	Pilot *conditions*
$\mathrm{Req}_{3.1}$ Comparison operation	Mandatory	Pilot Policy Subsumption
$\mathrm{Req}_{3.2}$ Intersection operation	Mandatory	Pilot Policy Join
$\mathrm{Req}_{3.3}$ Categories	Optional	Pilot partial order

Table 7: Fulfilment of requirements by Pilot

always possible. In this section, we provide some guidance to designers with an outline of the main parameters to be considered and their impact on the available options. The guidelines presented in this section are illustrated through several case studies in the next section.

Table 8 and Table 9 show, for each feature in the first column, the technical options that are possible or not for the implementation of information and consent respectively. In Table 8, the first column refers to the *DCD*, with an exception for *the physical absence of the data subject*, whereas in Table 9 it refers to the *DSD*, the data subject, or the specificities of the scenario. The physical absence of the data subject denotes use cases where a data subject is informed of *DCPs* without being within range of *DCDs*. For the sake of readability, we only show negative answers in the tables and take the convention that empty boxes are interpreted as the fact that the feature does not prevent the technical option. In order to decide if a technical option is possible in a given context, the designer must check that none of the features of this context corresponds to a "✗" in the column representing this option. Occurrences of "(✗)" denote situations in which the feature does not prevent the technical option but the technical option is likely to be either unnecessary (for example, in Table 8, the use of an additional beacon is probably not necessary for sensors endowed with an extensible protocol) or insufficient (for example, in Table 8, indirect communications will probably not be sufficient for moving sensors, such as cameras mounted on vehicles, unless the registers can be updated in real time).

A passive sensor in Table 8 is a sensor, such as a camera, able to collect data but not to communicate a privacy policy. An extensible protocol is defined as a protocol, such as Wi-Fi or Bluetooth (see Section 4.1.1), which can be configured to communicate a privacy

Table 8: Technical options for information as a function of the *DCD*

Features of *DCD*	Direct communications without beacon	Direct communications with beacon	Indirect communications
Passive sensor	✗		
Active sensor with extensible protocol		(✗)	
Active sensor without extensible protocol	✗		
Fixed sensor			
Moving sensor			(✗)
Physical absence of the data subject	✗	✗	

Table 9: Technical options for consent as a function of the *DSD*

Features of the data source or scenario	Direct communications without pairing	Direct communications with pairing	Indirect communications	*A priori* enforcement	*A posteriori* enforcement
Device with extensible protocol		(✗)	(✗)		
Device without extensible protocol	✗				
Device with substantial resources		(✗)	(✗)		
Device with scarce resources	✗				
Systematic collection process				✗	
Selective collection process					(✗)
Pre-existing relationship					
No pre-existing relationship			✗		

policy. The beacons considered in Table 8 are *Privacy beacons* that can be added to a device to allow it to communicate a privacy policy (see Section 4.1.1).

In Table 9, a device with substantial resources is assumed to be a device that can be used to define and manage privacy policies without significant drawbacks as defined in Section 3.3. The difference between a systematic collection process and a selective collection process is the fact that in the first case it is not possible to prevent the collection of certain data while this is possible in the latter. For example, video recording is a systematic collection process. The collection process can be systematic for certain data (such as MAC addresses for Wi-Fi access points) and selective for other data (for example, payload data in a Wi-Fi protocol). When the collection process is systematic, it cannot filter out data for which consent has not been granted: the only solution in this case is to implement consent *a posteriori*, by deleting or anonymizing the data as soon as it is collected. Pre-existing relationships corresponds to the situation where the data subject and the data controller know each other through any type of identifier that can be used to declare their respective privacy policies.

4.6 SCENARIOS

In order to illustrate the versatility of the framework and of its possible implementations, we now present its application to various scenarios. These scenarios range from existing situations to more futuristic ones, and illustrate the different areas of the Internet of Things presented in Section 1.1.1.

The design space introduced in the previous section is used to choose the most appropriate solution in each case.

4.6.1 *Bluetooth-based tracking*

Systems that monitor individuals based on the Bluetooth MAC address of their device are becoming commonplace. They are deployed in various instances of the smart city such as shopping malls [118] and music festival events [83]. Those systems passively collect the MAC addresses [43] found in messages broadcast by portable and wearable devices such as smartphones, smartwatches, wristbands, *etc.* Any person entering the operation area of such a system must be informed and able to provide her consent. This is particularly challenging in open venues like shopping malls where there is generally no existing link between the visitors and the entity operating the system. As a result, the visitors are currently informed of those tracking systems via posters and consent requirement is simply ignored.

Let us consider an area (shopping mall, museum, music festival ...) in which a Bluetooth tracking system is operating, *i.e.*, all active

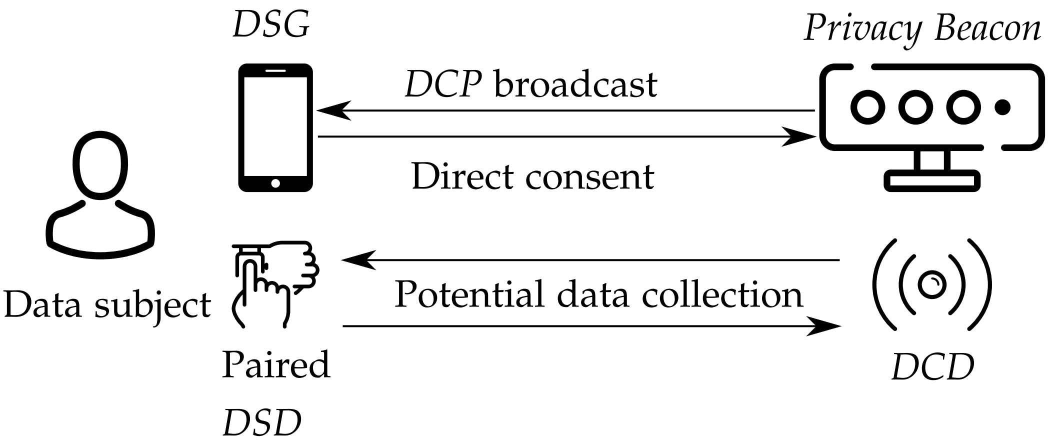


Figure 29: Bluetooth-based tracking

Bluetooth devices in this area can have their MAC address recorded and associated with other data (time, location, *etc.*). This is a situation in which the *DCDs* (Bluetooth sensors) are **fixed**, **passive**, and they implement a **systematic collection process**. Let us further assume that the data controller in charge of the tracking system has no prior link with the data subject: there is **no pre-existing relationship**. This means that, before the data subject enters the area, the data subject and the data subject have not been able to communicate and that any exchange of information must be done on the spot. We consider a data subject equipped with a *DSG* and another *DSD*: a Bluetooth wristband. As of today, wearable devices such as wristband are supporting an *extensible protocol*, but they have **reduced user interface** and **limited energy resources**. They must therefore be **paired** because of their reduced user interface and limited energy resources.

The design guidelines presented in Section 4.5 show that enforcement must be done *a posteriori* and that **direct communications** must apply due to the lack of pre-existing relationship. This case can be addressed with *Privacy Beacons* as illustrated in Figure 29. Such beacons can be deployed in the monitored area and around it in order to inform data subjects as soon as they enter the area. The scenario is the following: when entering the area, the *PDC* will automatically detect the *Privacy Beacon* and directly retrieve the *DCP* of the tracking system. If this *DCP* complies with the *DSP*, the *PDC* automatically sends the consent through the BLE direct communication channel. This consent contains the MAC addresses of both the *DSG* and the wristband, a hash of the *DCP*, and the data subject's cryptographic signature. Once this consent is received and securely stored in a *ledger*, the tracking system is allowed to collect data on the subject identified by the identifiers. By default, the system discards any data for which it has not obtained consent.

4.6.2 *Wi-Fi hotspots*

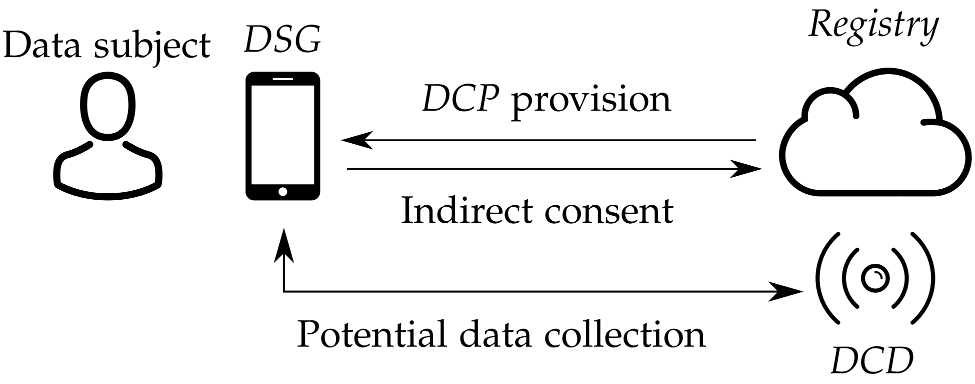


Figure 30: Wi-Fi hotspots

Most medium and large cities are equipped with Wi-Fi hotspots. These hotspots are installed in airports, malls, restaurants, and sometimes along the street [58]. They provide free Wi-Fi for registered users, after they have accepted the terms and conditions. However, some data controllers desire to track users to provide them marketing features such as coupons. While some personal data must be collected to deliver the Wi-Fi service (relying on contract as the legal ground), the use of data for marketing purpose must be based on consent as a legal ground. Data subjects have to be informed of both purposes, but it is important to make a distinction between the two legal grounds and to request consent for marketing purpose. This scenario is typical of the smart city (see Section 1.1.1.3, and Appendix B.4 for more details).

In this scenario, *DCDs* are **fixed** and **passive**, but the collection process is **selective**: hotspots are able to make a difference at the time of collection according to the identifiers (here the Wi-Fi MAC address), even though they need to collect it beforehand. Therefore, enforcement can be conducted *a priori*. We assume data subjects have a *DSG* hosting a *PDC*. In this scenario, data subjects are aware of the existence of hotspots, and even have a **pre-existing relationship** with data controllers.

This use case can be addressed through both direct and indirect communications according to Tables 8 and 9. However, the variety of actors involved and the two purposes favours an **indirect communication**, as presented in Figure 30. A registry can indeed inform of the legal obligation and of the marketing features. This choice of indirect communication also allows all actors to manage their *DCP* themselves while providing a single access point to data subjects. In that setup, access to the registry is known by data subjects, who can lookup the *DCPs*. The registry can be accessed from any device, but only a *PDC* can communicate a consent. Data subjects are informed of all purposes of processing, and can consent *a priori* if their *DSP* matches the *DCP*. By default, hotspots collect but immediately discard data for which they have not obtained consent. Hotspots collect required data, but selectively process it for marketing purposes only if consent has been retrieved.

4.6.3 *Vehicle tracking*

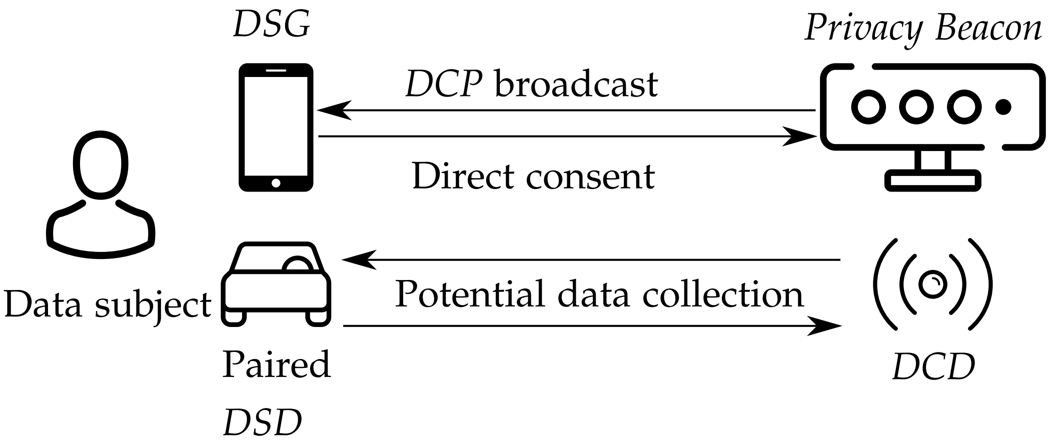


Figure 31: Vehicle tracking

Vehicles may be subject to passive data collection by systems that detect and record their presence. One of the main technologies allowing this data collection is Automatic Number Plate Recognition (ANPR) [47] based on images captured by CCTV cameras. Given the nature of the location where those systems are deployed (road sections, parking lots), data subjects and data controllers have usually no pre-existing relationship. Furthermore, the driving activity requires full attention of the driver, which cannot be disturbed by tasks related to information and consent.

This scenario involves **fixed** and **passive** sensors, with **no pre-existing relationship** and a **systematic collection process**. The situation is therefore similar to Scenario 4.6.1 except that the identification of data subjects relies on number plates rather than MAC addresses: the *DSD* considered is a car.

The constraints put on the driver attention and the lack of pre-existing relationship require a **direct communication**. This scenario is represented in Figure 31. Before entering a monitored area, the *DSG* of the driver (*e.g.*, her smartphone) communicates with the *Privacy Beacons* to retrieve *DCPs* and to return a potential consent including the car plate number. Furthermore, the *PDC* running on the *DSG* implements the consent decision without requiring the driver interaction, preserving its attention for the driving task.

4.6.4 *Smart hotel*

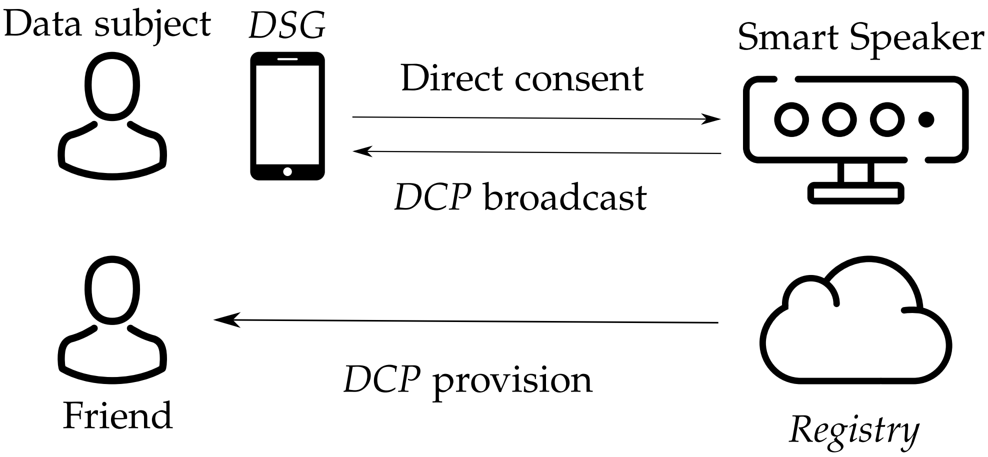


Figure 32: Smart hotel

A more prospective scenario can be imagined in a smart hotel. Some hotels are getting equipped with smart speakers, hosting a virtual assistant [147]. It is now possible to order room service or manage the light and temperature through such speakers, which are continuously listening (see Section 1.1.1.2, and Appendix B.3 for more details). These facilities pose privacy issues with respect to information and consent. For instance, guests may invite friends unregistered at the hotel. As a result, these friends would not be informed of possible data collection, and do not necessarily possess a *PDC*. However, we assume they posses a *DSG*. This scenario is then particularly challenging for privacy, but we can imagine manners to preserve it using our framework.

In that setup, *DCDs* are **fixed** and **active**. Given the fact that they are endowed with speech recognition, we also suppose that they have an **extensible protocol**. The collection process is **systematic**, but it must stop if all the persons present in the room have not given their consent. It is then required for the smart speakers to be able to accurately detect the number of persons in a room. [5] The data collected in that scenario is speech, which also acts as an identifier. We suppose a **pre-existing relationship** for some data subjects, but **not** for all. Guests renting the room are assumed to have a *PDC*, but not their friends.

This challenging use case can be solved using **both direct and indirect communications**, as Figure 32 shows. The smart speaker regularly compares the number of persons in the room with the number of consents stored, and does not listen as long as it has not received a number of consents equivalent to the number of persons detected.

[5] For instance, Hashimoto *et al.* [63] achieved an accuracy of 98% using pyroelectric infrared array detector close to doors.

In a first option, we use the direct communication. When entering the room, the *PDC* detects the smart speaker (acting as a *DCG*). It retrieves the *DCP* and communicates the consent if the two policies match.

In a second option, a hybrid approach is favoured. This approach uses an indirect information through a registry, and a direct communication of consent. Invited friends are informed of the *DCDs* on a local registry operated by the hotel. They may not have a *PDC*, the registry can therefore encourage them to install one through its responsive interface, but they are not encouraged to give their consent without a thorough reflection. The invited friends have to communicate their consent directly if they deem it useful.

summary

We described in this chapter technical options for implementing the framework described in Chapter 3. The machine-to-machine communications can be implemented directly (Section 4.1), or indirectly (Section 4.2). The two options are complementary: it is possible to inform with direct communications, to consent with indirect communications, and *vice-versa*. We showed in Section 4.3 how the requirements presented in Section 3.4 can be implemented as a combination of a *ledger* and cryptographic signatures. We showed here how to implement a *PDC* as an app, by defining a *DSP* using the privacy policy language Pilot (Section 4.4). A design space was proposed to help engineers and designers devising information and consent management solutions (Section 4.5). Finally, we illustrated the different implementations through scenarios in Section 4.6, to show the benefits of the design space and the use of these criteria.

5 THE MAP OF THINGS AND COIOT PROTOTYPES

We proposed in Chapter 3 a framework for enhancing information and consent in the Internet of Things, and technical options to implement it in Chapter 4. These technical options have been implemented as functional prototypes, and these prototypes are presented in the present chapter. The prototypes act as proofs of concept for the framework, and aim to illustrate the different possibilities in which the framework can operate, as well as the design space.

These prototypes, although fully functional, do not aim to be used as is. They rather are the proof that the statements made in the previous chapter are feasible at a low cost and without extensive modifications of hardware and software infrastructures. All the prototypes developed are under free licences and use free licences: we hope that they will be reused thanks to the accessibility of the code, or at least be a source of inspiration for systems designers.

We illustrate the framework with two prototypes: Map of Things and CoIoT. The former is a registry, it implements the indirect communication of information. Map of Things is presented in Section 5.1. The latter is a mobile application fully-fledged for the direct communication of information and consent, and also able to retrieve information indirectly from the registry Map of Things. The application, coined CoIoT, is presented in Section 5.2. We have also developed two complementary components, which can be used in conjunction with the prototype, *Privacy Beacons* and the *ledger*, which are presented more succinctly in Section 5.3.

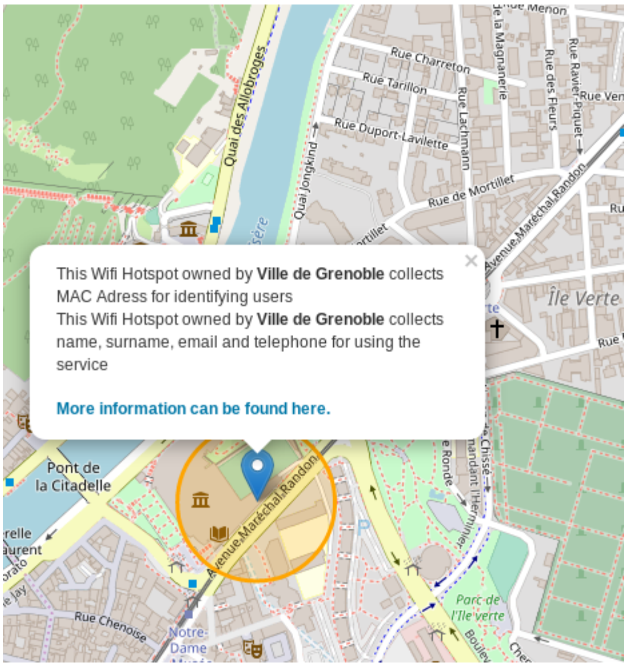
A summary of the requirements fulfilled by these prototypes and the additional components is presented as a table at the end of each prototype, with ✓ meaning that the requirement is fulfilled, and ✗ meaning that the requirement is **not** fulfilled.

5.1 MAP OF THINGS

Map of Things (MoT) is a prototype website for **indirect information** of privacy policies in the Internet of Things. Map of Things has been developed [94] to show that the indirect information can be implemented without any software or hardware modification on *DCDs*. MoT illustrates Scenario 4.6.2: the variety of actors involved in a smart city favours an indirect communication of information. MoT is supported by the framework Laravel [82]. It displays the map of OpenStreetMap [119] using the module Leaflet [85]. It is currently hosted at `https://mapofthings.inrialpes.fr`. The registry is endowed with the Wi-Fi hotspots of the French cities of La Rochelle and Grenoble, associated with their respective privacy policies. Note that it also displays dummy CCTVs for the sake of the example. The following sections present the different features offered by the tool, *e.g.*, a user-friendly map, a list of *DCDs* and of their respective privacy policies. A focus has been put on an experimental version of multi-faceted privacy policies.

5.1.1 *Map*

Its main interface is a map [1] displaying *DCDs*, their range of collection, and a simple notice of the personal data that they can collect (see Figure 33). The simple notice is composed of the type of device, the data controller, the type of data collected, and the purpose of collection.



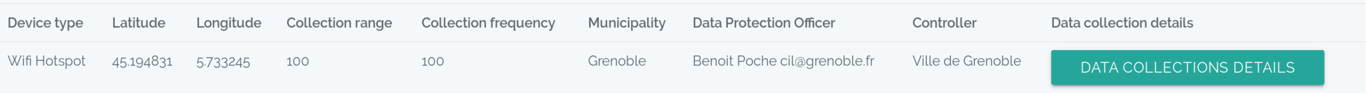
This Wifi Hotspot owned by **Ville de Grenoble** collects MAC Adress for identifying users
This Wifi Hotspot owned by **Ville de Grenoble** collects name, surname, email and telephone for using the service

More information can be found here.

Figure 33: Excerpt of a *DCD* in MoT, its range of collection, and its simple notice.

Clicking on the hyperlink provides more information to data subjects (see Figure 34). [2] This second layer of information informs of the coordinates of the *DCD*, its collection range in meters and frequency in times per second, and the contact of the DPO.

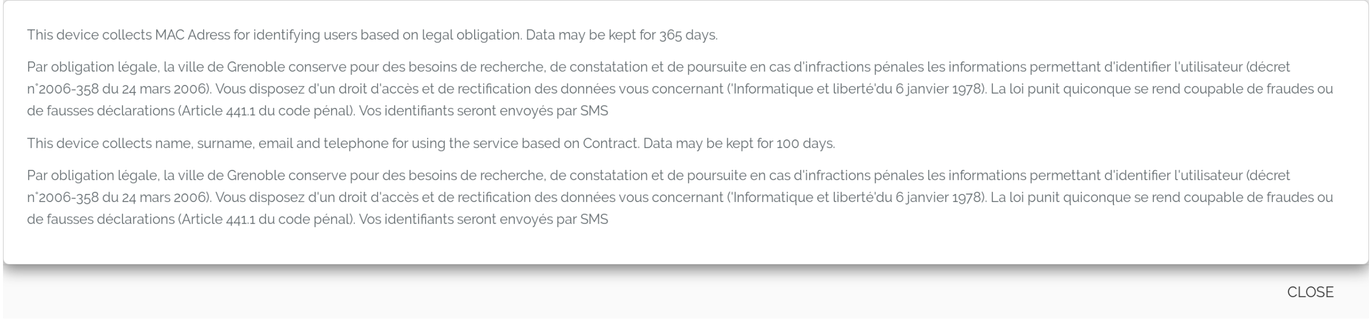
[1] https://mapofthings.inrialpes.fr/map
[2] https://mapofthings.inrialpes.fr/device/52 for instance.



Device type	Latitude	Longitude	Collection range	Collection frequency	Municipality	Data Protection Officer	Controller	Data collection details
Wifi Hotspot	45.194831	5.733245	100	100	Grenoble	Benoit Poche cil@grenoble.fr	Ville de Grenoble	DATA COLLECTIONS DETAILS

Figure 34: Example of a second layer of information in MoT.

The interested user can click on a button labelled *DATA COLLECTION DETAILS*, which triggers a pop-up window (see Figure 35). The pop-up window displays the natural language privacy policy, which additionally informs of the legal basis for processing and of the retention time.



This device collects MAC Adress for identifying users based on legal obligation. Data may be kept for 365 days.

Par obligation légale, la ville de Grenoble conserve pour des besoins de recherche, de constatation et de poursuite en cas d'infractions pénales les informations permettant d'identifier l'utilisateur (décret n°2006-358 du 24 mars 2006). Vous disposez d'un droit d'accès et de rectification des données vous concernant ('Informatique et liberté'du 6 janvier 1978). La loi punit quiconque se rend coupable de fraudes ou de fausses déclarations (Article 441.1 du code pénal). Vos identifiants seront envoyés par SMS

This device collects name, surname, email and telephone for using the service based on Contract. Data may be kept for 100 days.

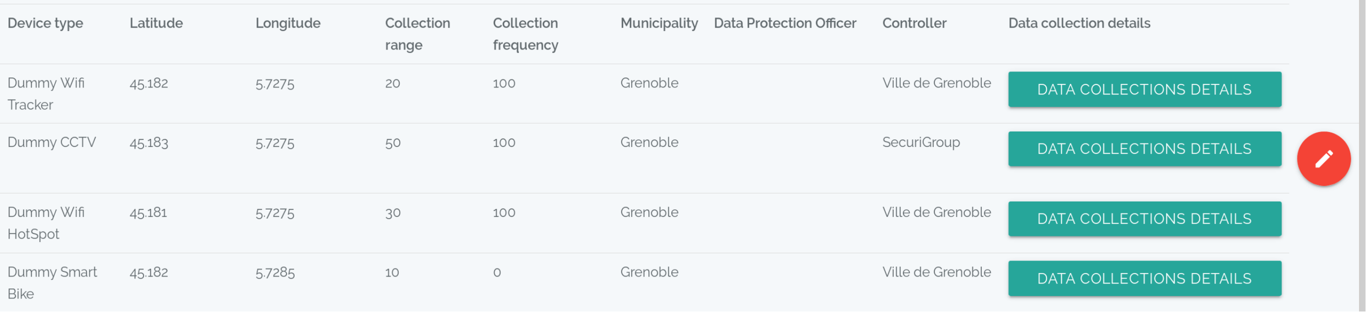
Par obligation légale, la ville de Grenoble conserve pour des besoins de recherche, de constatation et de poursuite en cas d'infractions pénales les informations permettant d'identifier l'utilisateur (décret n°2006-358 du 24 mars 2006). Vous disposez d'un droit d'accès et de rectification des données vous concernant ('Informatique et liberté'du 6 janvier 1978). La loi punit quiconque se rend coupable de fraudes ou de fausses déclarations (Article 441.1 du code pénal). Vos identifiants seront envoyés par SMS

Figure 35: The natural language privacy policy provided by the data controller.

The map focuses on the user-friendliness of the presentation of content (see Section 3.3.1.1), and is fully responsive (see Section 4.2.1.3). It presents the *graphical facet* of privacy policies (see [107]).

5.1.2 *List*

The registry can also present *DCDs* and their privacy policies as a list. [3] The list has the same interface as a single device (see Figure 36). This feature presents the *natural language facet* of privacy policies (see [107]).



Device type	Latitude	Longitude	Collection range	Collection frequency	Municipality	Data Protection Officer	Controller	Data collection details
Dummy Wifi Tracker	45.182	5.7275	20	100	Grenoble		Ville de Grenoble	DATA COLLECTIONS DETAILS
Dummy CCTV	45.183	5.7275	50	100	Grenoble		SecuriGroup	DATA COLLECTIONS DETAILS
Dummy Wifi HotSpot	45.181	5.7275	30	100	Grenoble		Ville de Grenoble	DATA COLLECTIONS DETAILS
Dummy Smart Bike	45.182	5.7285	10	0	Grenoble		Ville de Grenoble	DATA COLLECTIONS DETAILS

Figure 36: List of devices hosted by MoT.

Note that a device has a red button alongside the *DATA COLLECTION DETAILS* button, this functionality is explained in Section 5.1.3.

[3] https://mapofthings.inrialpes.fr/device

5.1.3 *Collaborative edition*

MoT allows for a collaborative edition the *DCDs* and of their respective privacy policies. Registered users [4] can click on a red button located on the bottom-left corner of the map to add a device (see Figure 37), and end up on a new window [5] after a confirmation of the location (see Figure 38).

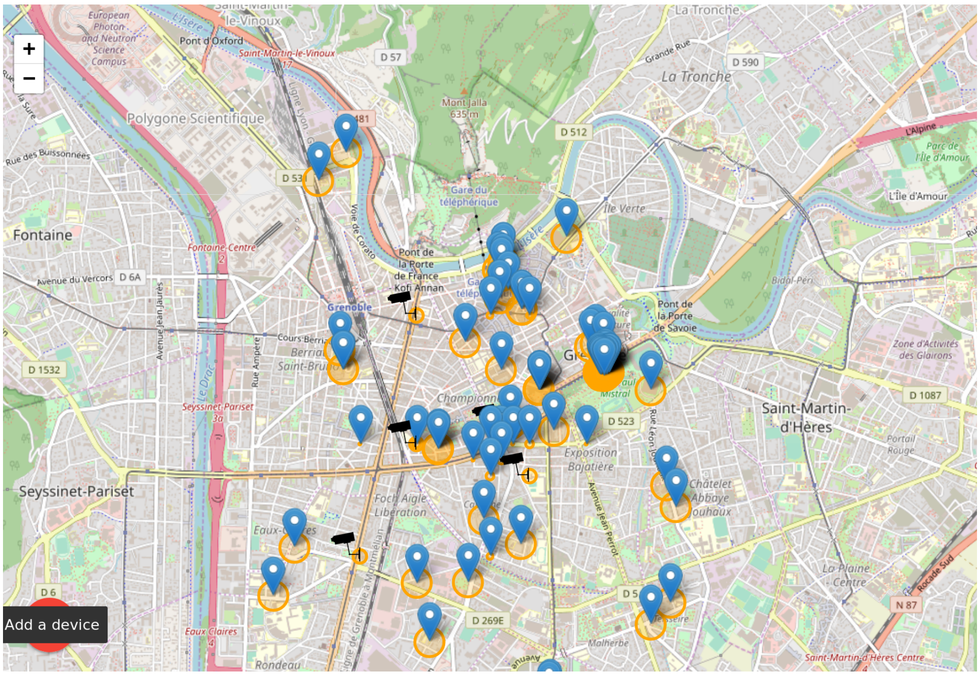


Figure 37: Add a device.

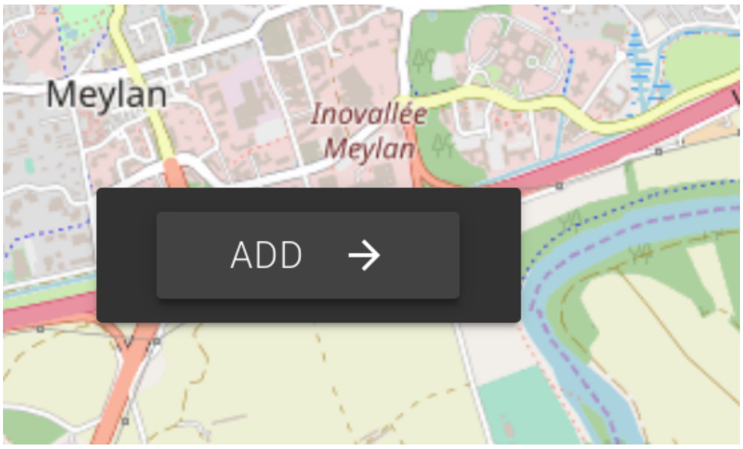
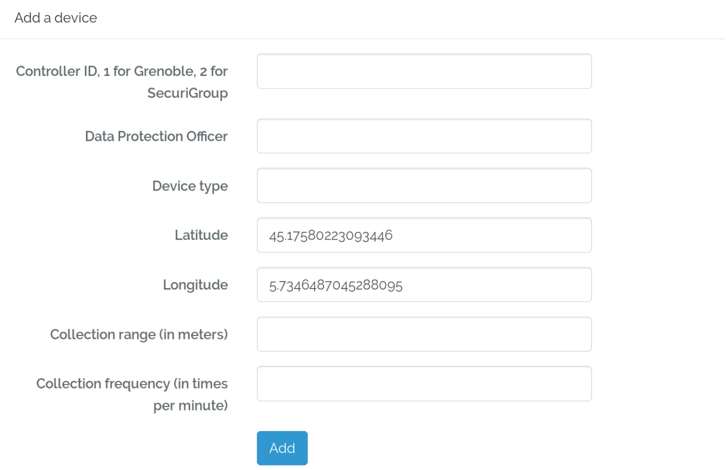


Figure 38: Confirm the location of a device.

[4] https://mapofthings.inrialpes.fr/register

[5] https://mapofthings.inrialpes.fr/device/create?latitude=45.17580223093446&longitude=5.7346487045288095&municipality=Grenoble for instance.

The new window awaits for the input of the information required by the list of devices (see Figure 39). Note that all inputs are controlled before the addition of a *DCD*: coordinates are pre-filled according to its location, and mandatory items (see Section 3.1.2) cannot be left empty. Registered users can modify the *DCPs* of *DCDs* of which they are the creators. This feature is available through a red pencil button located on the right of the devices' list (see Figure 36).



Add a device

Controller ID, 1 for Grenoble, 2 for SecuriGroup	
Data Protection Officer	
Device type	
Latitude	45.17580223093446
Longitude	5.7346487045288095
Collection range (in meters)	
Collection frequency (in times per minute)	

Figure 39: Add the privacy policy of a device.

5.1.4 *API*

The information stored on the registry can also be accessed from an API (see Section 4.2.1.2). [6] The API enables the automatic retrieval of the *DCP* in a machine-readable format (see Figure 52 in the Appendix).

This feature presents the *machine-readable facet* of privacy policies (see [107]).

Fulfilment of requirements

MoT does not fulfil all the requirements of indirect communications. As a matter of fact, communication of consent is not implemented, whereas it is a mandatory requirement since it focuses on information and does not implement consent. In contexts where consent is required, MoT should therefore complemented by additional means such as the CoIot application presented in the next section.

[6] `https://mapofthings.inrialpes.fr/api/devices` for all devices.
`https://mapofthings.inrialpes.fr/api/devices/1` for one device, here for device with id 1.

5.2 COIOT

CoIoT [94] (which stands for Consent and information in the Internet of Things) is a prototype Android app for **direct information and consent**, and **indirect information**, in the Internet of Things. It is our implementation of a *PDC*. CoIoT has been developed to show that seamless consent communication and *DSP* management can be operated without extensive technical knowledge nor efforts from data subjects. CoIoT can illustrate a large range of scenarios such as Scenarios 4.6.1 and 4.6.3, but it can also operate in Scenario 4.6.4 (the technical difficulties would then be put on the *DCG* to count the numbers of persons). [7] CoIoT implements all the requirements from Section 3.3, as shown in the rest of this section.

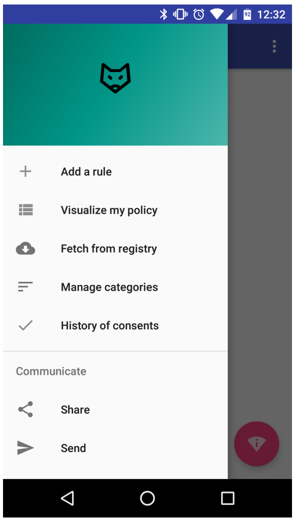
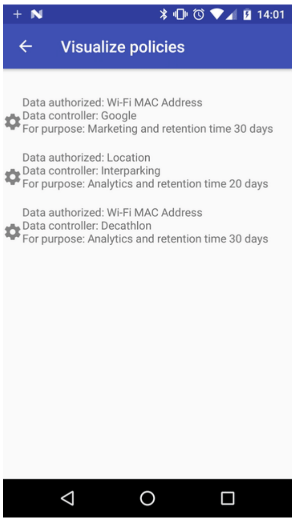


Figure 40: Menu on the left drawer.

5.2.1 *DSP management*

CoIoT is an implementation of a *PDC*. Therefore, it is possible for data subjects to manage their *DSPs* as required in Section 3.3.1.3. It corresponds to the requirements Consult *DSP* and Add/modify/delete. Data subjects can consult their *DSP* (see Figure 41) and modify existing rules (see Figure 42); these features are accessible from the "Visualize my policy" choice on the menu (see Section 40). The *DSP* is presented rule-per-rule. Data subjects can also add new rules (see Figure 43); this feature is accessible from the "Add a rule" choice on the menu (see Section 40). Note that items are preset to alleviate the burden of *DSP* management.

[7] Note that Scenario 4.6.2 cannot be illustrated because indirect communication of consent has not been implemented in CoIot.



14:01

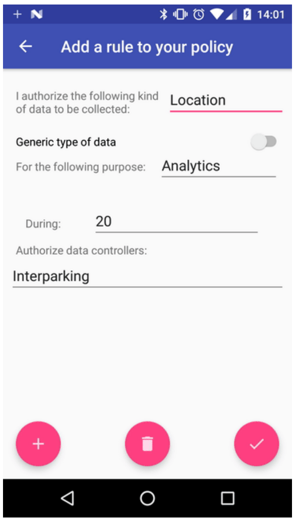
Visualize policies

Data authorized: Wi-Fi MAC Address
Data controller: Google
For purpose: Marketing and retention time 30 days

Data authorized: Location
Data controller: Interparking
For purpose: Analytics and retention time 20 days

Data authorized: Wi-Fi MAC Address
Data controller: Decathlon
For purpose: Analytics and retention time 30 days

Figure 41: Visualizing a *DSP*.



14:01

Add a rule to your policy

I authorize the following kind of data to be collected: Location

Generic type of data

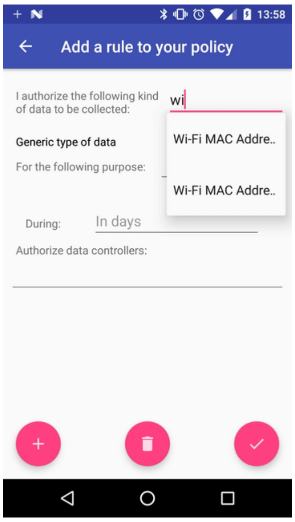
For the following purpose: Analytics

During: 20

Authorize data controllers:

Interparking

Figure 42: Modifying a *DSP*.



13:58

Add a rule to your policy

I authorize the following kind of data to be collected: wi

Wi-Fi MAC Addre..

Wi-Fi MAC Addre..

Generic type of data

For the following purpose:

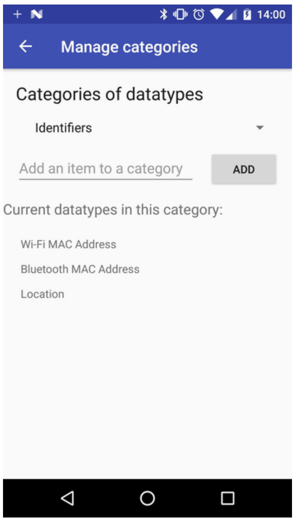
During: In days

Authorize data controllers:

Figure 43: Adding a rule in a *DSP*.

5.2.1.1 *Generic categories*

Data subjects can also define categories of items as required in Section 4.4.2.2. Only the type of data can be defined generically in the current prototype. Clicking on the on/off button when adding or modifying a rule (as seen in Figure 42) results in the definition of a whole category. Note that items are preset according to categories when the button is clicked. Categories can be managed within a separate activity (see Figure 44); this feature is accessible from the "Manage categories" choice on the menu (see Section 40).



14:00

Manage categories

Categories of datatypes

Identifiers

Add an item to a category ADD

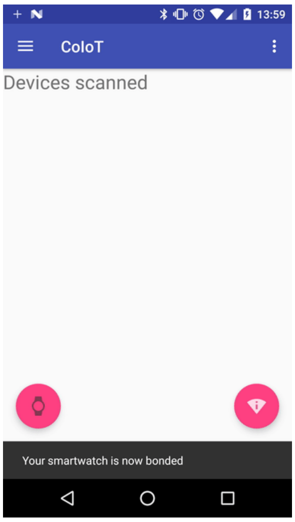
Current datatypes in this category:

Wi-Fi MAC Address

Bluetooth MAC Address

Location

Figure 44: Managing categories.



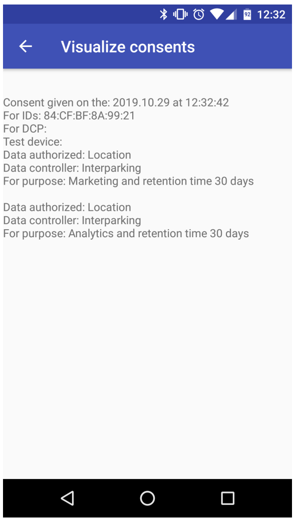
13:59

CoIoT

Devices scanned

Your smartwatch is now bonded

Figure 45: Bonding another *DSD*.



12:32

Visualize consents

Consent given on the: 2019.10.29 at 12:32:42
For IDs: 84:CF:BF:8A:99:21
For DCP:
Test device:
Data authorized: Location
Data controller: Interparking
For purpose: Marketing and retention time 30 days

Data authorized: Location
Data controller: Interparking
For purpose: Analytics and retention time 30 days

Figure 46: Visualizing the history of consents.

5.2.1.2 *Bond another DSD*

Data subjects can bond other *DSDs* as required in Section 3.3.2.2. The bonding is hard-coded and only considers a smartwatch for the demonstration, *i.e.*, it is not possible to choose the identifier added. Data subjects have to click on the bottom-left corner button to bond the *DSD* (see Figure 45).

5.2.1.3 *History of consents*

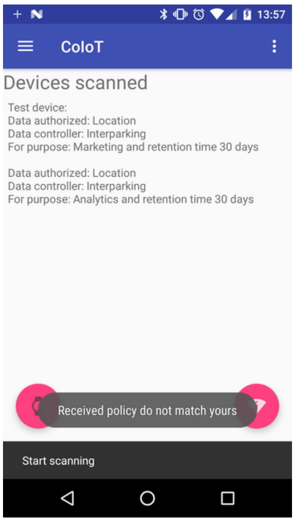
Data subjects can access a visual summary of consents previously given as mentioned in Section 3.3.2.4 (see Figure 46).

5.2.2 *Visualizing DCP*

CoIoT can retrieve information directly through a local scan, or by fetching resources from a registry, such as MoT. In both case, *DCPs* are presented rule-per-rule.

5.2.2.1 *Local scan*

CoIoT scans for *DCPs* on the background. It displays retrieved *DCPs* on the main screen (see Figure 47), and warns if the *DCP* does not match the *DSP* (through a *toast*, and a notification in the drawer marked as a + on the top-left corner, see Figure 47).



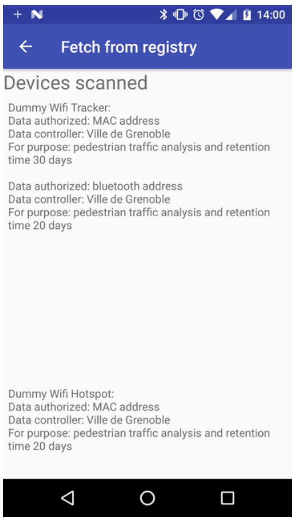
Devices scanned

Test device:
Data authorized: Location
Data controller: Interparking
For purpose: Marketing and retention time 30 days

Data authorized: Location
Data controller: Interparking
For purpose: Analytics and retention time 30 days

Received policy do not match yours

Figure 47: Result of a BLE scan.



Devices scanned

Dummy Wifi Tracker:
Data authorized: MAC address
Data controller: Ville de Grenoble
For purpose: pedestrian traffic analysis and retention time 30 days

Data authorized: bluetooth address
Data controller: Ville de Grenoble
For purpose: pedestrian traffic analysis and retention time 20 days

Dummy Wifi Hotspot:
Data authorized: MAC address
Data controller: Ville de Grenoble
For purpose: pedestrian traffic analysis and retention time 20 days

Figure 48: Fetching *DCPs* from MoT.

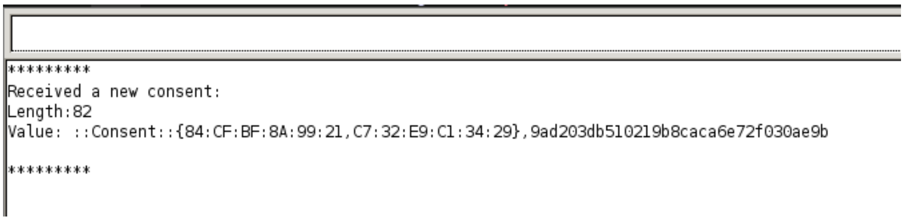
5.2.2.2 *Retrieval from registry*

CoIoT can also retrieve *DCPs* from the registry Map of Things. The feature is accessed from a separate activity (see the menu in Figure 40), and displays the results of the API of MoT (see Figure 48). Note that this feature is experimental insofar that it does not permit a fine-grain tuning, *e.g.*, according to geolocation.

5.2.3 *Consent*

CoIoT can communicate consent directly, it also implements negotiation as specified in *PPNP* (see Section 3.2).

CoIoT automatically sends a signed consent if it retrieves a *DCP* matching the *DSP*. However, if the *DCP* does not match the *DSP*, it sends the *DSP*. An answer composed of a more restrictive *DCP* (see Section 3.1.2.1) results in the communication of a consent for the new *DCP*. The consent and the negotiation are illustrated with Figures 49 and 50 respectively.



```
*********
Received a new consent:
Length:82
Value: ::Consent::{84:CF:BF:8A:99:21,C7:32:E9:C1:34:29},9ad203db510219b8caca6e72f030ae9b

*********
```

Figure 49: This screenshot shows the serial output of the *Privacy Beacon* retrieving a consent from CoIoT. The consent is displayed as a string made of the identifier of the device concerned and a hash of the *DCP*.

Fulfilment of requirements

CoIoT successfully fulfils all mandatory requirements of a direct implementation of the **Protocol**, of the **PDC**, of the **Proof of consent** (the one delegated to the *PDC*), **Privacy Policies**, and operations over the **Language**. Additionally, it also fulfils most optional requirements. Table 10 summarizes which requirements CoIoT fulfils. Note that we consider that bonding is only partially fulfilled due to its hard-coded implementation.

CoIoT			
Category	*Name of the requirement*	*Type of requirement*	*Fulfilled*
Protocol	Req$_{4.1}$ and Req$_{4.4}$ Information (*DSG* and *DCG*)	Mandatory	✓
Protocol (direct)	Req$_{4.2}$ and Req$_{4.5}$ Consent (*DSG* and *DCG*)	Mandatory	✓
	Req$_{4.6}$ and Req$_{4.7}$ Negotiation (*DSG* and *DCG*)	Optional	✓
PDC	Req$_{5.1}$ Consult *DCP*	Mandatory	✓
	Req$_{5.2}$ Consult *DSP*	Mandatory	✓
	Req$_{5.3}$ Add/modify/delete	Mandatory	✓
	Req$_{5.4}$ Notifications	Mandatory	✓
	Req$_{5.5}$ Natural language	Mandatory	✓
	Req$_{5.6}$ Rule-per-rule presentation	Mandatory	✓
	Req$_{4.3}$ *DSG* interaction	Mandatory	✓
	Req$_{5.7}$ History	Optional	✓
	Req$_{5.8}$ Preset items	Optional	✓
	Req$_{5.9}$ Bonding	Optional	(✓)
	Req$_{5.10}$ Sorted presentation	Optional	✗
Proof of consent	Signatures (see Section 4.3)	Mandatory	✓
Privacy Policies	Req$_{1.1}$ Type of data	Mandatory	✓
	Req$_{1.2}$ Purpose of collection	Mandatory	✓
	Req$_{1.3}$ Retention time	Mandatory	✓
	Req$_{1.4}$ *DC*	Mandatory	✓
	Req$_{1.5}$ 3rd parties	Mandatory	✓
	Req$_{1.6}$ Collection frequency	Optional	✗
	Req$_{1.7}$ Location of *DCD*	Optional	✗
	Req$_{1.8}$ Collection range	Optional	✗
Language	Req$_{3.1}$ Comparison	Mandatory	✓
	Req$_{3.2}$ Intersection	Mandatory	✓

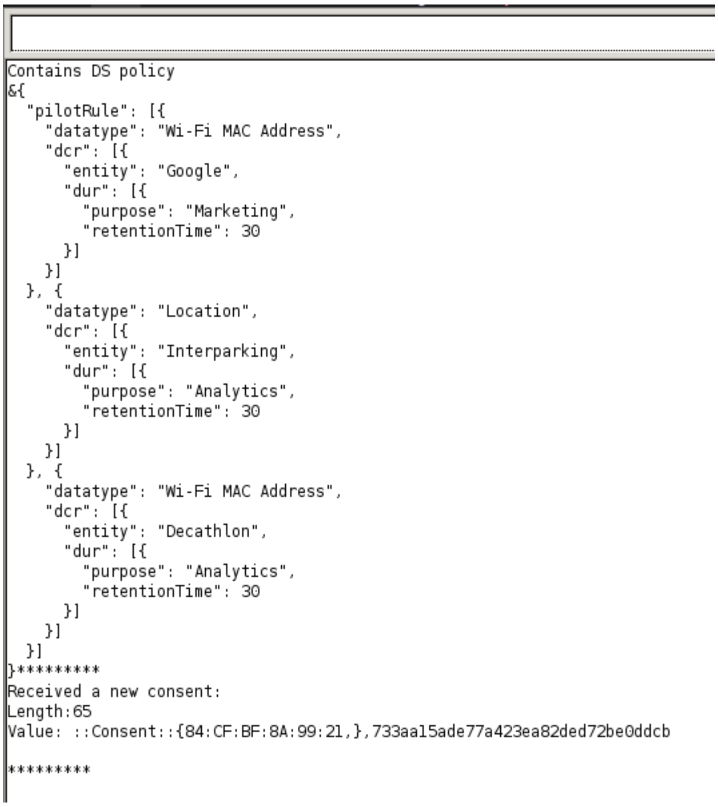
Table 10: Fulfilment of requirements by CoIoT

5.3 PRIVACY BEACONS AND LEDGER

This last section provides short explanations of the last two components of our implementation: *Privacy Beacons* and the *ledger*.

5.3.1 *Privacy Beacons*

To provide a full-fledged implementation of direct communication, a prototype of *Privacy Beacons* has been developed [106]. They are based on ESP32 (see Section 4.1.1). They implement *PPNP* through BLE, and work in tandem with CoIoT. They continually broadcast their *DCP* (encoded in bytes with protobuf [127], see Section 4.1.1.1), and can retrieve consents (see Figure 49) through ATT (see Section 4.1.1.1). They are programmed to accept any negotiation (see Figure 50).



```
Contains DS policy
&{
  "pilotRule": [{
    "datatype": "Wi-Fi MAC Address",
    "dcr": [{
      "entity": "Google",
      "dur": [{
        "purpose": "Marketing",
        "retentionTime": 30
      }]
    }]
  }, {
    "datatype": "Location",
    "dcr": [{
      "entity": "Interparking",
      "dur": [{
        "purpose": "Analytics",
        "retentionTime": 30
      }]
    }]
  }, {
    "datatype": "Wi-Fi MAC Address",
    "dcr": [{
      "entity": "Decathlon",
      "dur": [{
        "purpose": "Analytics",
        "retentionTime": 30
      }]
    }]
  }]
}*********
Received a new consent:
Length:65
Value: ::Consent::{84:CF:BF:8A:99:21,},733aa15ade77a423ea82ded72be0ddcb

*********
```

Figure 50: This screenshot shows the serial output of the *Privacy Beacon* when a negotiation is successfully undertaken. The *DSP* is first received, then the consent. The *DSP* is presented rule-per-rule.

5.3.2 *Ledger*

Finally, the *ledger* proposed in Section 4.3 has been implemented as well [105]. It runs thanks to a python script that combines the serial display of the ESP32 and the secure storage of consents in Hypercore for demonstration purposes.

SUMMARY

This chapter presented the prototypes implemented as proofs of the concepts proposed in Chapter 4. Section 5.1 presented Map of Things (MoT), a registry for indirect information in the smart city. Section 5.2 provided an overview of CoIoT, a *PDC* able to communicate information directly and indirectly, and consent directly. Section 5.3 succinctly showed the prototypes of the *Privacy Beacons* and of the *ledger*.

Note that some features presented in this chapter are hard-coded for the sake of simplicity. However, the prototypes presented here are mere examples, and can be extended for industrial deployments.

Part III

CONCLUSION

Alas, I have studied philosophy, the law as well as medicine, and to my sorrow, theology; studied them well with ardent zeal, yet here I am, a wretched fool, no wiser than I was before.

Faust in Faust' Johann Wolfgang von Goethe

6 CONCLUSION

We asked in the introduction of this work whether *it is possible to design a generic framework to communicate information and manage consent in the Internet of Things*. This document argues in favour of a positive answer to this question. Indeed, this work proposed a technical infrastructure to inform and manage consent in the Internet of Things.

After a) a contextualization through a definition of the Internet of Things and its privacy issues, an overview of the GDPR requirements in terms of information and consent, and the tension resulting from the application of the latter to the former in Chapter 1; and b) a review of the means to inform and of the means to manage consent in the Internet of Things in Chapter 2; we provided in Chapter 3 a framework for information and consent in the Internet of Things as a main contribution.

This framework is composed of: 1) a protocol to inform and manage consent through the communication of machine-readable privacy policies for both data controllers and data subjects; 2) functionalities for human-computer interactions to display and manage privacy policies; and 3) requirements over the proof of consent.

We proposed in Chapter 4 technical options to implement the framework, a design space to help systems designers, and scenarios illustrating the use of the framework can be used. Some of these technical proposals have been developed as prototypes to show their feasibility, they are presented in Chapter 5.

In this last chapter, we summarize the contributions of this thesis (see Section 6.1), discuss the design choices (see Section 6.2), highlight the limitations (see Section 6.3), and present perspectives for future research (see Section 6.4).

6.1 SUMMARY OF CONTRIBUTIONS

The main contribution of this study is the generic framework presented in Chapter 3. This framework addresses the challenges set forth in the introduction, namely, the need to accommodate a great variety of situations (genericity), to take into account legal requirements (GDPR compliance), to ensure user-friendliness and to facilitate implementations. This framework is also an additional element brought to the debate on the ePrivacy Regulation.

6.1.1 *Generic character*

The framework is generic in the sense that it does not depend on specific communication protocols, channels or types of devices, nor fielding configurations. This generic character is what makes it different from related work, which are usually tailored to specific settings (see Chapter 2). It therefore tackles the heterogeneity of the Internet of Things (see Section 1.1.1).

6.1.2 *Legal compliance*

The framework addresses legal compliance, it has been carefully designed for informed consent according to the GDPR requirements. We presented the requirements for information and consent under the GDPR in Section 1.1.2, and devised the framework to the end.

6.1.3 *User- and privacy-friendliness*

The framework considers user- and privacy-friendliness in many ways.

First, it provides minimum interruptions: while notifications are part of the human-computer interactions, they are only required under specific conditions, *e.g.*, when a negotiation is undertaken.

Second, no data is disclosed by default: unlike opt-out facilities (see Section 2.2.2), our solution favours a *fine-grain opt-in* approach. Data subjects have to selectively and actively consent to data collection processing, in line with the definition of consent in the GDPR, and with the clarifications of the WP29.

Third and last, optional features — such as negotiation or preset items — have been designed so as to make our framework more usable. They can reduce user fatigue, and empower data subjects by providing more choice.

6.1.4 *Implementation*

The framework can easily be implemented, as demonstrated by the multiple technical options presented in Chapter 4. As a matter of fact, the greatest care has been taken to make every aspect of the framework easy to implement. The two options to implement the framework are inexpensive: the direct communication relies on *DCGs* covering a large area for less than 6$, and the indirect communication does not require additional devices (only a website without specific computational needs). The *PDC* can run on devices owned by a large number of persons — smartphones —, and does not require intense computational capacities (for this kind of device). The implementation of the ledger can run on small devices as well. All our prototypes are based on free software, and are under free licences as well.

6.1.5 *ePrivacy Regulation*

Although a large-scale deployment of the implementations of the framework could undoubtedly impact society, this framework shows that consent can be effectively retrieved in the Internet of Things.

A part of personal data in the EU is regulated by the Privacy and Electronic Communications Directive 2002/58/EC on Privacy and Electronic Communications, also known as the ePrivacy Directive. Its scope encompasses mainly metadata, such as traffic data, or cookies. This directive will be superseded by the ePrivacy Regulation, today at the state of proposal. The ePrivacy Directive and the Regulation operates as complements of the GDPR on specific topics. The regulation may impose less stringent requirements over consent: the initial version proposed by the commission [52] did not require consent to geolocate data subjects using their *end-users* terminal equipment (see Art. 8(2) of the initial proposal). This provision has been deleted in the current version of the parliament [54], and new requirements have been set on data controllers, such as the possibility to object to such physical tracking (Art. 8(2a.d) of the parliament report). The current draft can lead to various outcomes, ranging from a privacy protective to a privacy intrusive legislation [79].

We aimed to show with this work that the difficulty to manage and retrieve consent is a fallacious argument, and that consent can be retrieved in a privacy-protective manner for data controllers who respect the legislation.

6.2 DISCUSSION

The framework proposed in this thesis hinges upon certain hypotheses that may raise some questions with respect to design choices. We strive in Section 6.2.1 to provide answers to these interrogations.

The overall approach followed during this document, *i.e.*, creating frictionless interactions between humans and machines for consent retrieval, may also be questioned. Section 6.2.2 attempts to clarify ethical concerns related to the responsibility of designers.

6.2.1 *Design choices*

Some high-level choices made may appear as arbitrary. We discuss two of them below, *i.e.*, the choice to protect privacy by policy, and the necessity for data subjects to own a *DSG*.

6.2.1.1 *Privacy by policy*

Spiekermann & Cranor proposed in [138] a distinction between privacy by architecture and by policy. Building on top of a three-layer model of user privacy concerns (data transfer, storage, and processing), they develop guidelines for building privacy-friendly systems.

Privacy by policy focuses on the implementation of legal requirements, the Fair Information Practices [55] in the article, but it can be generalized to modern legislations. It is the most obvious choice when data can be linked to a natural person. For instance, the Internet of Things is a difficult environment to evolve in without being tracked. Technical anti-tracking measures are difficult to implement, if not impossible. But a system may choose to advertise its data practices, and offer the choice not to track [99].

Privacy by architecture attempts to minimize the digital traces left by diminishing the collection of personal data. This choice is preferred when a system does not require to identify its users, and displaces trust from institutions to technical architectures [131]. As an example, the TOR network implements privacy by architecture: users cannot be identified, or with difficulty [101].

These two approaches do not necessarily oppose each other, and can be complementary. However, we contend that privacy by architecture is intrinsically complex to implement in the Internet of Things, notably when the collection process is systematic. Even though it is possible to limit physical tracking [97], a privacy by policy approach — all the more after the enforceability of the GDPR — can have a better reach than a privacy by architecture approach in that case.

Data subjects may want to use services requiring identification, which makes privacy by architecture, and consequently anonymity, unsuitable. Unlike web browsing, most Internet of Things services — *e.g.*, the smart home or wearables — are tailored to individuals, and therefore cannot be used anonymously. Moreover, it is unusable to protect privacy with an architectural approach in certain setups, such as video recordings. Even though artists have created make-up [28] or masks [143] to prevent facial recognition, these techniques are clearly impractical: privacy by policy is to be favoured in our case.

6.2.1.2 *Gateway device*

The necessity for data subjects to possess a *DSG*, often a smartphone, may be questioned. We stipulated in Section 3.1.1 that data subjects need a *DSG* for the framework to be applicable. Actually, very little data collection can happen if data subjects do not have such a *DSG*. The smartphone is often a condition *sine qua non* for data collection to happen: only video can otherwise be recorded, or the presence detected. The main device subject to data collection becomes the custodian of privacy. Note that we assume data collection cannot happen without consent: a data subject without *DSG* cannot communicate consent in our framework and cannot therefore be subject to data collection. For instance, a data subject possessing a *DSD* but no *PDC* (see Scenario 4.6.4) must in no case have her data collected.

6.2.2 *Ethics*

We proposed in this thesis a facilitating approach to manage consent. While we took care not to encourage data subjects to provide their consent against their interests, [1] this facilitating approach has to be questioned ethically because it may infringe autonomy of data subjects. Here, the responsibility of the designer is engaged insofar that designing Privacy Enhancing Technology (PET) can reduce the latitude of data subjects, for instance when such technology assists decision-making. But is full autonomy always desirable? [139]

"Code is law" said Lessig [90], this assertion implies that system designers can influence actions and have a normative effect. They can design systems ranging from total choice to pre-determined choice. We denote this the *spectrum of choice*. What is the right cursor position on the spectrum of design interaction? Total choice may seem unattainable: every element of a system has an effect on decision-making, albeit minor; and pre-determined choice, although theoretically possible, is rarely found in real systems. Most systems fall in between, and influence users while letting them choose certain settings. In any case, it is a fair assumption to say that design is always manipulative to a certain extent. This manipulation can however be in favour or at the expense of data subjects' interests.

A typical malevolent influence of design are Dark Patterns [61], and a typical benevolent influence can be nudges [4] and soft paternalism. How can we ensure designers are indeed benevolent? Even if they are, can they really pretend to be aware of users' interests? Even though designers may follow a benevolent end, it is uncertain that the effective result would be a choice in favour of users' interests. They may think the system they designed actually reflects everyone's interests, but it is with difficulty that a small set of persons can reasonably determine a large group of persons' interests. Privacy designers are undoubtedly more knowledgeable than most people on privacy topics, but interests differ between individuals [91, 93]. A possible manner for groups of users to verify whether a system matches their interests would be to understand its functioning, its biases, and to be in a position permitting the modification of the said system.

Another important parameter to take into account is the "Rule of default": 95% of users do not change their default settings [45], this choice is therefore crucial.

Based on this parameter, we contend that, by default, users do not wish their data to be collected and processed when the legal ground is consent, *i.e.*, when data collection is not mandatory for the functioning of a service. [2]

[1] Through 1) the negotiation process, 2) the granularity offered when managing consent, in addition to 3) careful interaction design guidelines, see Section 6.1.3

[2] This is in line with the guidance of ICO, the British DPA, on the use of cookies [66].

This choice differs from the CMU approach (see Section 2.2.1) which, on the grounds of more individual freedom, let neglecting users make choices against their interests. This approach is strengthened by the use of machine learning, which reinforces existing carelessness.

6.3 LIMITATIONS

In this section, we present the limitations of the framework in two parts, considering respectively: the theoretical limitations in Section 6.3.1, *i.e.*, the very essence of what the framework cannot cope against; and the technical limitations in Section 6.3.2, *i.e.*, what the prototypes cannot do.

6.3.1 *Theoretical limitations*

Our framework meets three high-level difficulties: 1) the very notion of consent to personal data processing; 2) unlawful data collection, against which it is toothless (see Section 6.2.1); and 3) other legal grounds than consent — *e.g.*, legitimate interest, that may be invoked to process data without having to retrieve consent.

6.3.1.1 *Limitations of consent*

The very idea of consent, and of "notice and choice" [35] have been criticized by numerous scholars. These privacy paradigms are indeed far from being perfect, and we expose their most prominent critiques.

Nissenbaum contends that consent — at least in its current interpretation — should be abandoned [113]. The idea that individuals can effectively manage their own privacy by themselves is, according to her, a fallacy. Nissenbaum advocates for stronger collective protections instead of individual empowerment.

Some have a more balanced point of view with respect to privacy self-management, such as Solove [135]. He contends that existing methods to protect privacy are inherently limited, because of cognitive problems — individuals are not properly informed, and decision-making can be skewed — as well as structural problems — the number of decisions can scale up to becoming unmanageable. Aggregation of data is hard to predict from an individual vantage point, and harm done to individuals can be hard to assess. However, Solove does not go as far as Nissenbaum, and advocates a hybrid solution: rethinking consent and nudges [4] without falling into paternalism, developing partial privacy self-management ("one possible answer might be to find ways for people to manage their privacy globally for all entities rather than one at a time") [3], and more generally what he

[3] Note that it goes in favour of a privacy-by-policy approach

calls “moving towards substance over neutrality”, *i.e.*, the idea that one should not be able to consent to anything.

This idea of proposing a consent limited in scope has also been proposed by Lionel Maurel (also known as Calimaq), a French librarian, jurist, and privacy advocate, in [163]. Lionel Maurel distinguishes what he denotes the “two faces of consent”:

SUBJECTIVE CONSENT is when individuals are mandated to make a decision, whereas they are subject to numerous biases. This interpretation of consent allows data subjects to weaken their own rights, if they choose to do so.

OBJECTIVE CONSENT is about framing the conditions in which data subjects can communicate their consent. According to him, the GDPR emphasizes this face of consent with the precisions brought to its legal validity.

He says about objective consent: “It is then less about giving individuals the power to consent than, on the contrary, to define what they cannot consent to”. [4]

As a result, we can conclude that: 1) consent is difficult to manage, especially when blended with technical systems, and one must be conscious of these difficulties when devising consent management systems, but 2) it can be a valid approach if thoroughly though and carefully designed.

6.3.1.2 *Unlawful data collection*

Data controllers can choose not to comply with their privacy policies [98]. The news often title the non-compliance of some companies with their announced privacy policies [87]. The penalties risked by data controllers— up to 4% of their annual turnover — strongly encourage them to respect the GDPR, but without proper enforcement and audit, a privacy by policy approach is inefficient.

6.3.1.3 *Other legal grounds*

Finally, our framework addresses information, which is required for any personal data collection. [5] But it also considers consent, which is one out of six legal grounds for personal data processing. One legal ground in particular has been in the spotlight due to its unclear application scope: legitimate interest. Claiming legitimate interest for data processing does not require authorization from data subjects. DPAs however clarified that it should not be invoked in place of consent, notably for online advertising [24].

[4] Translation by the author, original sentence: “Il s’agit donc moins en réalité de donner à l’individu un pouvoir de consentir que de définir, au contraire, ce à quoi il ne peut pas consentir.”

[5] The few exceptions do not compromise our approach.

6.3.2 *Technical limitations*

Finally, it is to be noted that the prototypes presented in Chapter 5 suffer from technical limitations due to time shortage and technological obstacles.

The limitations due to time shortage are of little importance for a proof of concept implementation. For instance, the collaborative edition of MoT (see Section 5.1.3) could be restricted according to a level of trust, *i.e.*, trusted users should be marked as such and the privacy policies they upload should not have the same impact as privacy policies uploaded by random users. Similarly, CoIoT only proposes generic categories for the type of data (generic categories is an optional requirement). Note that not all technical options have been implemented: Wi-Fi Direct, the indirect negotiation, and an IoS app have not been developed as prototypes.

The prototypes also suffer from actual technological obstacles, due to the inherent limitations of the technologies used. For example, CoIoT cannot automatically retrieve the MAC addresses of the devices on which it is installed. Indeed, modern versions of Android prevent any app — even with a root access explicitly granted by the data subject — from accessing network information such as MAC addresses.

6.4 PERSPECTIVES

We conclude this document with research perspectives with respect to the work presented along this document.

6.4.1 *Testing and refining*

First, the prototypes, albeit carefully designed, lack user studies to determine their usability, and improve it if necessary. These user studies may encompass a questionnaire, asking feedback to data subjects. Such a questionnaire may answer the following questions: 1) to what extent does the prototype convey information intelligibly? 2) is it usable for every categories of population, notably those lacking digital literacy [120]? 3) could the prototype be ready to be used in the real world? [6] From the results of such tests, the prototypes could be refined, and tested again to measure the improvement in usability.

[6] Note that the deployment of the different contact tracing apps following the COViD-19 pandemic may bring additional insights to the deployment of large scale Bluetooth technology.

6.4.2 *Beyond individual consent?*

This work strived to propose a privacy-preserving approach to consent management, even though consent intrinsically suffers from limitations (see Section 6.3.1.1). Letting individuals choosing whether to disclose their personal data may not always be the best path to empowerment, notably when information is distributed asymmetrically, and the decisions resulting from the disclosure entail more than just the requested individuals.

As a matter of fact, personal data processing has also a collective dimension [29], this fact is notably illustrated with certain types of data such as our DNA or our facial traits. It is for instance possible to identify the quasi-totality of the population using only 2% of the population's DNA [51]: our individual choices can impact more than ourselves. Facial recognition also gives an example of a type of data with a systemic impact [30], as its development in the public space may lead to dreadful outcomes.

These two examples surely are more extreme than a geolocation trace left by a smartphone, and as biometrical data they must be considered under the regime of sensitive data. However, a fact remains: personal data gains value when combined with other personal data, and with that a potential for harm. It is for that reason necessary to consider a large scope when dealing with privacy.

Collective approaches to privacy have thus been proposed by different disciplines. Lawyers advocated that privacy is a right for individuals, but which must be dealt with collectively [22]. Computer scientists proposed a collaborative access control framework for social networks [6]. A research avenue would be to consider more specifically consent as a collective right as well, and to propose technical measures to collectively enforce that right.

With the GDPR now in action, and consent a cornerstone of privacy management, another promising research avenue would be to frame the conditions under which consent must be considered lawful from a Human-Computer Interactions point of view. Toth *et al.* conduct research in that direction [141], with an opinion paper on cookies and other trackers. A set of *privacy-preserving* design guidelines, in UI (User Interface) for the web, or in UX (User eXperience) in a broader sense would participate in that endeavour.

Part IV

APPENDIX

A GLOSSARY

To avoid any ambiguities, we highlight terms considered as important for the reading. We present a glossary to help the reader as it is possible to find redundant terms in the literature, and different concepts expressed under the same word.

COOKIE A cookie is a small piece of data sent from a website and stored on the user's computer by the user's web browser while the user is browsing.

CSS *Consent Storage Server*
A *CSS* is a *role* for a server on which consents are securely stored.

DARK PATTERNS Dark Patterns are "instances where designers use their knowledge of human behaviour (*e.g.*, psychology) and the desires of end users to implement deceptive functionality that is not in the user's best interest" [61].

DATA CONTROLLER According to the GDPR, "[data] controller means the natural or legal person, public authority, agency or other body which, alone or jointly with others, determines the purposes and means of the processing of personal data; where the purposes and means of such processing are determined by Union or Member State law, the controller or the specific criteria for its nomination may be provided for by Union or Member State law". [1] We may name the data controller *DC* in this document.

DCD *Data Controller Device*
A *Data Controller Device* is a device controlled by a data controller.

DCG *Data Controller Gateway*
A *DCG* is a *role* for a *DCD* in charge of declaring other *DCDs*, retrieving *DSPs* and communicating them to the *DSG*.

DCP A *Data Controller Policy* is the privacy policy of a data controller. It is a commitment of the data controller regarding its processing of personal data.

DATA SUBJECT According to the GDPR, a data subject is "an identified or identifiable natural person". [2] We may name the data subject *DS* in this document.

[1] See Article 4 of the GDPR.

[2] See Article 4 of the GDPR, and the definition of *personal data* in this Glossary.

DSD *Data Subject Device*
A *DSD* is a device which belongs to a data subject, usually either worn or carried. It may be subject to data collection.

DSG *Data Subject Gateway*
A *DSG* is a *role* for a *DSD*, usually either worn or carried. It has the capacity to retrieve *DCPs*, and to communicate a *DSP*.

DSP A *Data Subject Policy* is the privacy policy of a data subject. It defines the requirements of the data subject concerning the processing of this data by data controllers.

GRAPHICAL PRIVACY POLICY A *graphical privacy policy* is a privacy policy expressed graphically [107].

ITEM An *item* is a piece of information provided in a privacy policy.

LEGAL GROUND A legal ground is a rational motive or basis for a belief, conviction, or action taken, such as a legal action or argument. The GDPR considers in its Article 6 six legal grounds: a) consent, b) necessity for the performance of a contract, c) necessity for the compliance with a legal obligation, d) necessity in order to protect vital interests, e) public interest, and f) legitimate interests.

MACHINE-READABLE PRIVACY POLICY A *machine-readable privacy policy* is a privacy policy expressed in a format readable my machines [107]. This format is usually derived from a privacy policy language with a well-defined syntax, and a formal semantics in some cases.

NATURAL LANGUAGE PRIVACY POLICY A *natural language privacy policy* is a privacy policy expressed in natural language [107].

PERSONAL DATA According to the GDPR, "personal data means any information relating to an identified or identifiable natural person ('data subject '); an identifiable natural person is one who can be identified, directly or indirectly, in particular by reference to an identifier such as a name, an identification number, location data, an online identifier or to one or more factors specific to the physical, physiological, genetic, mental, economic, cultural or social identity of that natural person". [3] Personal data can identify someone directly and uniquely — *e.g.*, a social security number — with less precision — *e.g.*, a pseudonym — or by combination with other information — *e.g.*, metadata left by online behaviour. Personal data is used in the European legal context.

[3] See Article 4 of the GDPR.

PERSONNALLY IDENTIFIABLE INFORMATION PII refers to personal data in the US legal context.

PRIVACY POLICY A privacy policy is a statement made by data subjects or data controllers to declare respectively their requirements and commitments in terms of personal data management.

PRIVACY POLICY LANGUAGE A *privacy policy language* is a language used to define privacy policies. It can describe *DCPs* as well as *DSPs*.

B INTERNET OF THINGS

The Internet of Things is not a consensual and well-defined notion. However, it is possible to adopt the following definition, derived from its designation: *the Internet of Things is the network encompassing any digital device whose interface allows for an Internet connection.*

This vagueness does not prevent institutional organs to use the term, and the WP29 published an opinion on its recent development [153]. The WP29 refers the Internet of Things in its document to "*an infrastructure in which billions of sensors embedded in common, everyday devices – "things" as such, or things linked to other objects or individuals – are designed to record, process, store and transfer data and, as they are associated with unique identifiers, interact with other devices or systems using networking capabilities.*" These devices and sensors — usually called *smart things* — can collect various types of data: non personal such as the temperature, the light intensity, or the air quality; or personal data. Data is considered personal when it relates to an identified or identifiable natural person (see Appendix A).

The collection and processing of such personal data can have nefarious impacts on society at large [162], insofar that it permits the implementation of a real size Panopticon [57]. The Panopticon is the prison invented by Bentham in which all prisoners can be observed by a single guard without knowing whether they are actually watched. To prevent the advent of such apparatus, it is necessary to define what is meant by the "Internet of Things".

Its origin is to be found in Ubiquitous Computing, term coined in 1991 by Weiser [156], that we present in Section B.1. We then describe three common areas of the Internet of Things, corresponding to three social spheres: [1]

THE SELF With wearables in Section B.2;

THE PRIVATE SPHERE With the Smart Home in Section B.3; and

THE PUBLIC SPHERE With the Smart City in Section B.4.

Note that the three spheres stated are not exhaustively described. For instance, we omitted what is denoted Smart Campus [40] for concision purposes. Additionally, the distinction is not exclusive: certain use cases can overlap multiple spheres. For example, a Smart Building [122] can be encountered in both private and public spheres. The

[1] While the distinction public/private is borrowed from their respective definitions by Arendt in the Human Condition [10], we hear by "*self*" what relates specifically to the individual as a distinct entity (opposed to, *e.g.*, the family, which belongs to the private sphere without being reducible to a single individual).

distinction however corresponds to differences in uses and scenarios: while it is likely to be aware of the presence of devices at home, it is not the case in a public space.

B.1 FROM UBICOMP TO INTERNET OF THINGS

Ubiquitous Computing, often named ubicomp for short, has been envisioned by Weiser [156] in 1991, then head of Xerox PARC. In this pioneer paper, Weiser exposes a vision of "the computer for the 21^{st} century". He draws a picture where technology would disappear in the background, and be used intuitively without having to be though consciously. A user does not have to use a classical trio keyboard/-mouse/screen in this vision, as natural interactions with everyday objects become the norm. Such a paradigm can be fulfilled with three elements: 1) cheap, low-power computers that include equally convenient displays, 2) software for ubiquitous applications, and 3) a network that ties them all together.

This vision turned out to be true, in part. As a matter of fact, we are now surrounded by devices with specialized software and bridged to Internet. From the smartphone to the smart fridge, smart things pervade our daily lives, giving rise to pervasive computing. But the current state of affairs is much more messy than planned [16]. Devices are not always compatible, connections are not as seamless as envisioned, and the single desktop computer has not exactly been replaced by a fleet of invisible machines, but rather reduced in size and enhanced by sensors.

The term Internet of Things exists for more than twenty years — the term was coined by Ashton in a conference in 1999 [12] — but it flourished only recently. In 2003 and 2004, the term was used in non-academic publications such as the Guardian, and the UN's International Telecommunications Union published a report about it in 2005 [67]. We now live in a world where IPv6 [2] allows for the connection of 30 billions devices as of early 2020. This number is growing exponentially: we expect 75 billions devices for 2025 [68]. The following sections present the Internet of Things through three spheres: the self, the private sphere, and the public sphere.

B.2 WEARABLES

Wearable devices are what happen when the Internet of Things is devised for a single individual. They are denoted as such because they are worn by data subjects. Tehrani & Andrew define wearables as

[2] IPv6 is the sixth version of the Internet Protocol (IP), which provides an identification and location system for computers on networks and routes traffic across the Internet. IPv6 replaces IPv4, short of addresses, and enable the direct addressing of an important number of devices.

"*electronic technologies or computers that are incorporated into items of clothing and accessories which can comfortably be worn on the body*"[140]. This incorporation of devices into the outfit invisible them. For instance, glasses can record audio and video without bystanders noticing it [77]. Smartwatch have the possibility to retrieve the pulse rate, for which they are designed most of the time, and can also disclose the activity of the holder: whether she is running, walking or sat for example [71]. However, the growth of wearables does not go without concerns from a privacy perspective [86].

Wearable devices enabled a new trend: that of gathering data about oneself. This trend is often denoted Quantified Self (QS) [88]. The term was coined by Wolf and Kelly in 2007 [157]. QS consists in gathering data such as weight, running scores, the number of steps, or sleep quality, often to fitness ends. This data is predominantly captured by a smartphone, but fitness wristbands are also common in such use cases. Other devices may be used, such as smart scales for weight control. The dominant communication protocol is Bluetooth [20], as most interactions occur within a small physical range. Leibenger *et al.* [88] lists four benefits of QS:

REPORT Reporting health data can be used for trivial purposes or control behaviour

COMPETITION Users can compare their results with other users

CORRELATION Users can analyse the impact of a change in a variable on another, such as the regularity of runs on global health

RESEARCH When considering both different types of data and a sufficiently large number of users, analyse of QS data can provide new insights to researchers

Future wearables will include smart fabric to infer your health or even the mood, smart lenses to analyse your emotions, or smart shoes to finely track your exercise. It is argued that the future may give rise to a new type of networks as wearables meld with the body: the Internet of Bodies [100]. However, not all these predictions are desirable from a privacy point of view, as many of these technologies are too intrusive to answer any kind of legal liability.

B.3 SMART HOME

The private sphere is our next step to present the Internet of Things. The obvious category of Internet of Things in that case is the smart home. The smart home, also known as Wireless home automation networks (WHANs) [59], is the enhancement of the home with connected devices in order to facilitate life of data subjects. Among the possible improvements, Gomez and Paradells [59] list the following:

LIGHT CONTROL Light bulbs can automatically be turned on when the presence of a person is detected, the intensity of light can be proportionate to the luminosity, and light control can be managed from remote, for instance with an smartphone app

REMOTE CONTROL Remote control is not restricted to light, as it can encompass TV, HiFi, or air conditioning

SMART ENERGY The temperature can be controlled remotely, and coupled with a smart thermostat, but the most prominent use case is probably the smart meter: connected to a smart grid, a smart meter can help to manage energy over large area

REMOTE CARE Blended with wearables, remote care benefits to patients, disabled and elderly people. For instance, a smart wrist can detect through its accelerometers when someone falls on the ground and triggers an alarm

SECURITY AND SAFETY Smoke detectors warn firefighters, and motion sensors call the police when a break-in occurs. Surveillance cameras, both to protect from intruders and to watch toddlers, are now connected to Internet and can be accessed remotely.

Among all use cases, smart speakers embedded with virtual assistants made a notable breakthrough. Major tech companies each launched their own smart speaker: Google devised Home, Amazon designed Echo, Facebook offers Portal, Apple created HomePod, and Microsoft launched Invoke. Smart speakers continuously listen, and react to a wake word which then trigger voice control. Data subjects command built-in virtual assistants, which can interact with other smart devices in the home. The smart home can spy on our conversations using smart speakers, notably when guests are invited over: they may not be aware of the data collection happening.

As a result, the smart home mainly considers a private sphere, but it is entangled with wearables which react and communicate with the rest of the infrastructure. All these devices communicate with various technologies, such as ZigBee, 6LoWPan, or Bluetooth when wearables are considered.

B.4 SMART CITY

The third and last part of the Internet of Things presented in this section is the smart city. Whereas the first two parts focused on the self and the private sphere, the smart city considers the public sphere. Many definitions emerged to describe the smart city [34], such as "a city connecting the physical infrastructure, the IT infrastructure, the social infrastructure, and the business infrastructure to leverage the collective intelligence of the city" [62], whose "[...] final aim is to make a better use of the public resources, increasing the quality of the services offered to the citizens, while reducing the operational costs of the public administrations" [160].

A possible manner to present the different types of smart cities is achieved through the prism of its main operators [151]:

TECHNO-CITY The techno-city is driven by the industry, sensors rule the urban development. It results in a centralized city and a top-down approach. A typical example of such a city is Songdo in Korea. Cisco produced most of the equipment, and an operation centre manages city functions. This type of smart city is the least favourable to privacy due to the uncontrolled multiplication of devices.

CONTRIBUTIVE CITY The contributive city is impulsed by citizens and usages of urban space. Such citizens are referred to hackers or makers, and they promote a collaborative and peer-to-peer approach. Some initiatives such as sous-surveillance [136] — a collaborative map to surveillance cameras — illustrates the concept. A contributive city may give birth to a more privacy-protective kind of city due to the collaborative aspect.

E-CITY The e-city is fostered by public institutions, and results in a bottom-up approach. Here, citizens participation (not to be confused with contribute initiatives of the contributive city) is encouraged, and Open Data is fostered. Barcelona belongs to this category with its open participatory platform for deliberation Decidim [9].

Note that these types of smart cities are not mutually exclusive, and Vievard [151] advocates for a discussion between the three dominant operators at the root of the three models — *i.e.*, equipment manufacturers, citizens, and public authorities — in order to model a balanced smart city.

Yet another manner to define the smart city is through its numerous devices and their uses. It is for instance common to find Wireless Sensor Networks [21] (WSN) to measure air quality. Air pollution is considered a major issue in megalopolises, and such WSN can help to monitor pollutant emissions. Data about air quality is not personal, and does not threaten privacy. However, other data sources are less innocuous, such as smart grids [41] and smart meters (see Section B.3 (under the term *smart energy*)). Smart meters can reveal the devices used, and even the activities performed — such as whether an occupant is cooking, sleeping, or using a laptop — if the collection granularity is fine enough. They can be combined in a smart grid, which can provide "the opportunity to reduce energy consumption enabled by detailed, personalized energy consumption reports and consequent monetary savings" [41]. Cameras have spread out in the recent years, and they are now connected to Internet. They allow for a real-time and continuous monitoring, not to mention the improvement made in facial recognition [39]. At the heart of a city resides its different means of transportation. It is no surprise to find smart mobility as a component of a smart city [18]. In Barcelona for instance, the combination of smart bus network designed according to a data analysis of traffic flow and smart traffic lights optimize the overall transportation networks [1]. In addition, the traffic lights are set to free the road for emergency vehicles in case of a fire or an accident. The smart mobility of tomorrow may very well be composed of Vehicular Ad-hoc Networks (VANETs). Connected vehicles form a VANET, and are therefore able to communicate between each other, and with the roadside [161]. This communication can prevent over-accidents by warning equipped vehicles in the vicinity, or can improve traffic efficiency. Many self-claimed smart cities chose to put some data retrieved from these sensors online and publicly available, when this data is not personal. This information is labelled Open Data [15]. Open Data denotes the set of public data opened to the public for scrutiny and research [69]. Because of the public nature of Open Data, the data available must not be personal, and common examples include transportation network information, locations of bicycle-sharing docks and their availability, Wi-Fi hotspots, air quality information, and data unrelated to the smart city aspect such as trees locations or city budgets [152] *etc.*

The smart city thus raises to a whole new level the capacities of surveillance, while providing innovative services for citizens [33]. Certain smart city apparatus considers personal data (*e.g.*, smart grids), but not all (*e.g.*, WSN), and while certain facilities aim to improve services (*e.g.*, Open Data), others strive to monitor citizens (*e.g.*, cameras). Van Zoonen proposes a privacy framework to grasp privacy concerns in the smart city, based on a 2x2 matrix [150]. Challenges are split according to two axes: whether the data collected is personal or impersonal, and whether its purpose is service or surveillance. It is then possible to categorize privacy concerns within this framework: a smart bin provides a service if it simply announces when it is full; but it can also become a tool of surveillance if authentication is required to throw rubbish (see Figure 51).

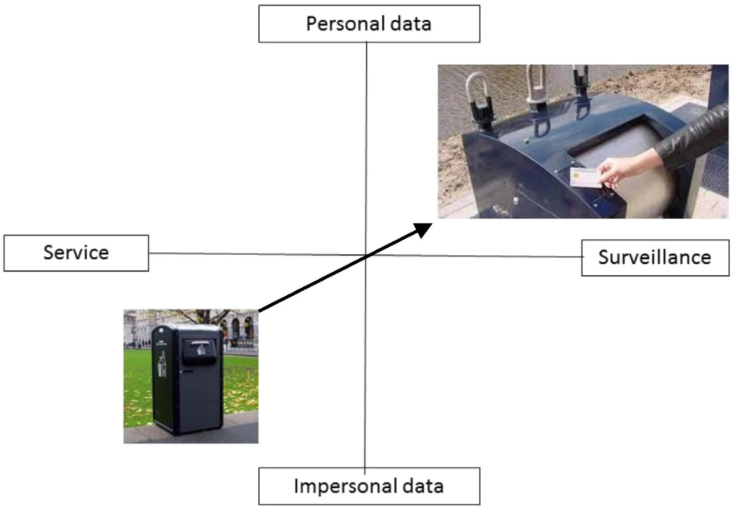


Figure 51: Van Zoonen's privacy framework applied to a smart bin.

The variety of policies applied to the smart city and the number of use cases makes difficult to encompass all variations of the term. However, we observe that the concept of smart city is not purely technical, and that simply adding more technology does not automatically solve urban issues [108]. Authors from social science fields are critical of this label put on urban development. For instance, Hollands [64] contends that the smart city is yet another evolution of the *entrepreneurial* city, prioritizing business interests over social justice, and aggravating gentrification. However, this state of affairs is not inevitable, and the smart city can become progressive and welcoming if all groups of people are included in its development. Kitchin [76] advocates for better consideration of security and privacy in smart cities. He advocates for a multi-pronged approach that blends together market, technical, governance and management, and policy, regulatory and legal solutions.

C CODE EXCERPTS

Listing 1: Minimal implementation of a BLE Scanner in Java for Android

```java
/**
* Activity for scanning and displaying available BLE devices.
*/
public class DeviceScanActivity extends ListActivity
{

    private BluetoothAdapter bluetoothAdapter;
    private boolean mScanning;
    private Handler handler;

    // Stops scanning after 10 seconds.
    private static final long SCAN_PERIOD = 10000;
    ...
    private void scanLeDevice(final boolean enable)
    {
        if (enable)
        {
            // Stops scanning after a pre-defined scan period.
            handler.postDelayed(new Runnable()
            {
                @Override
                public void run()
                {
                    mScanning = false;
                    bluetoothAdapter.stopLeScan(leScanCallback);
                }
            }
            , SCAN_PERIOD);

            mScanning = true;
            bluetoothAdapter.startLeScan(leScanCallback);
        }
        else
        {
            mScanning = false;
            bluetoothAdapter.stopLeScan(leScanCallback);
        }
        ...
    }
    ...
}
```

Listing 2: Minimal implementation of a callback for a BLE Scanner in Java for Android

```java
private LeDeviceListAdapter leDeviceListAdapter;
...
// Device scan callback.
private BluetoothAdapter.LeScanCallback leScanCallback =
new BluetoothAdapter.LeScanCallback()
{
    @Override
    public void onLeScan(final BluetoothDevice device, int rssi,
    byte[] scanRecord)
    {
        runOnUiThread(new Runnable()
        {
            @Override
            public void run()
            {
                leDeviceListAdapter.addDevice(device);
                leDeviceListAdapter.notifyDataSetChanged();
            }
        }
        );
    }
};
```

Listing 3: Minimal implementation of a GATT connection in Java for Android

```java
public class BluetoothLeService extends Service
{
    private final static String TAG = BluetoothLeService.class.
    getSimpleName();
    private BluetoothGatt bluetoothGatt;
    private int connectionState = STATE_DISCONNECTED;
    private static final int STATE_DISCONNECTED = 0;
    private static final int STATE_CONNECTED = 2;
    public final static String ACTION_GATT_CONNECTED =
    "com.example.bluetooth.le.ACTION_GATT_CONNECTED";
    public final static String ACTION_GATT_DISCONNECTED =
    "com.example.bluetooth.le.ACTION_GATT_DISCONNECTED";

    private final BluetoothGattCallback gattCallback =
    new BluetoothGattCallback()
    {
        @Override
        public void onConnectionStateChange(BluetoothGatt gatt,
    int status,
        int newState)
        {
            String intentAction;
            if (newState == BluetoothProfile.STATE_CONNECTED)
            {
                intentAction = ACTION_GATT_CONNECTED;
                connectionState = STATE_CONNECTED;
                broadcastUpdate(intentAction);
                Log.i(TAG, "Connected to GATT server.");
                Log.i(TAG, "Attempting to start service discovery:
    " + bluetoothGatt.discoverServices());
            }
            else if (newState == BluetoothProfile.
     STATE_DISCONNECTED)
            {
                intentAction = ACTION_GATT_DISCONNECTED;
                connectionState = STATE_DISCONNECTED;
                Log.i(TAG, "Disconnected from GATT server.");
                broadcastUpdate(intentAction);
            }
        }

    };
    ...
}
```

Listing 4: Override of methods in order to write a consent on a GATT characteristic.

```
@Override
public void onServicesDiscovered(BluetoothGatt gatt, int status)
{
    ...
    BluetoothGattDescriptor descriptor =
    characteristic.getDescriptor(
    CLIENT_CHARACTERISTIC_CONFIG_UUID);

    descriptor.setValue(
    BluetoothGattDescriptor.ENABLE_NOTIFICATION_VALUE);   gatt.
    writeDescriptor(descriptor);
}

@Override
public void onDescriptorWrite(BluetoothGatt gatt,
    BluetoothGattDescriptor descriptor, int status)
{
    BluetoothGattCharacteristic characteristic =
    gatt.getService(CONSENT_SERVICE_UUID)
    .getCharacteristic(CONSENT_CHARACTERISTIC_UUID);
    characteristic.setValue(consent);
    gatt.writeCharacteristic(characteristic);
}
```

Listing 5: Generating a pair of keys using KeyPairGenerator with KeyPairGeneratorSpec.

```
/*
* Generate a new EC key pair entry in the Android Keystore by
* using the KeyPairGenerator API. The private key can only be
* used for signing or verification and only with SHA-256 or
* SHA-512 as the message digest.
*/
KeyPairGenerator kpg = KeyPairGenerator.getInstance(
KeyProperties.KEY_ALGORITHM_EC, "AndroidKeyStore");
kpg.initialize(new KeyGenParameterSpec.Builder(
alias,
KeyProperties.PURPOSE_SIGN | KeyProperties.PURPOSE_VERIFY)
.setDigests(KeyProperties.DIGEST_SHA256,
KeyProperties.DIGEST_SHA512)
.build());

KeyPair kp = kpg.generateKeyPair();
```

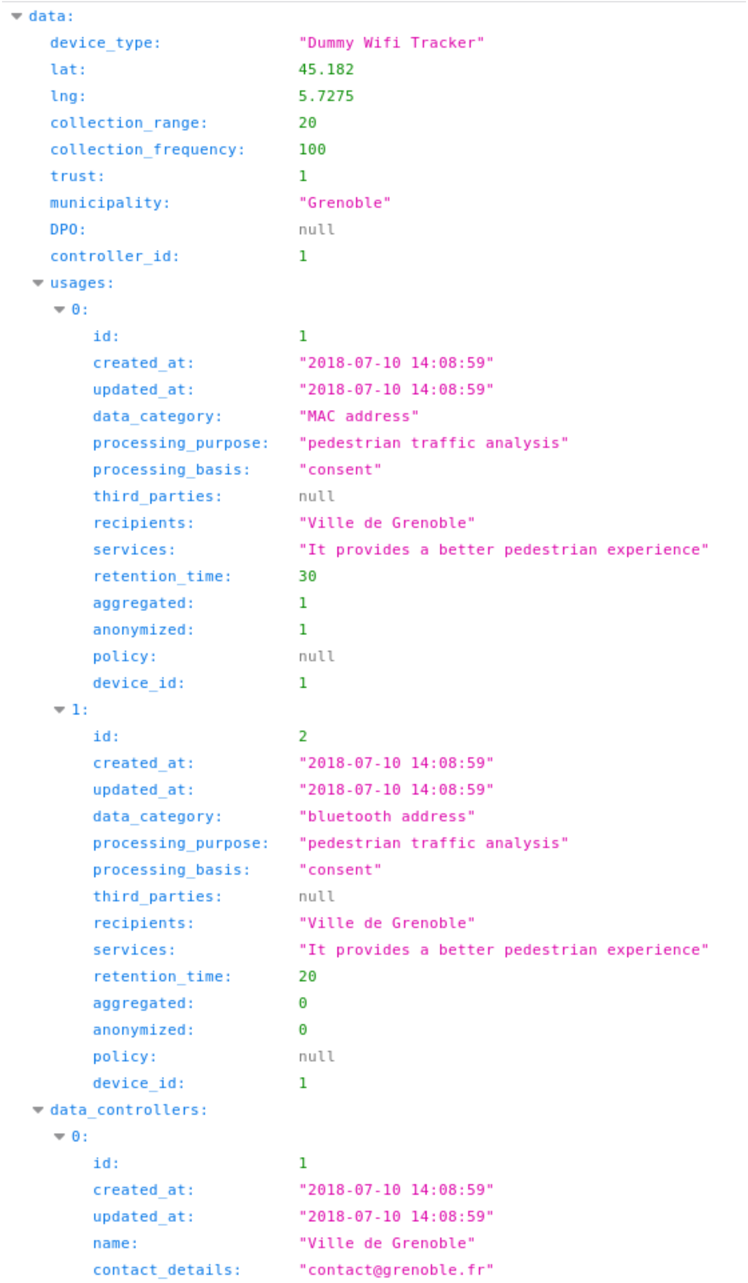
Listing 6: Generating a pair of keys using KeyPairGenerator with KeyPairGeneratorSpec.

```
/*
* Use a PrivateKey in the KeyStore to create a signature over
* some data.
*/
KeyStore ks = KeyStore.getInstance("AndroidKeyStore");
ks.load(null);
KeyStore.Entry entry = ks.getEntry(alias, null);
if (!(entry instanceof PrivateKeyEntry))
{
    Log.w(TAG, "Not an instance of a PrivateKeyEntry");
    return null;
}
Signature s = Signature.getInstance("SHA256withECDSA");
s.initSign(((PrivateKeyEntry) entry).getPrivateKey());
s.update(data);
byte[] signature = s.sign();
```

Listing 7: Minimal working example of Hypercore.

```
var hypercore = require('hypercore')
var feed = hypercore('./my-first-dataset',
{
    valueEncoding: 'utf-8'
}
)

feed.append('hello')
feed.append('world', function (err)
{
    if (err) throw err
    feed.get(0, console.log) // prints hello
    feed.get(1, console.log) // prints world
}
)
```



```
▼ data:
    device_type:                 "Dummy Wifi Tracker"
    lat:                         45.182
    lng:                         5.7275
    collection_range:            20
    collection_frequency:        100
    trust:                       1
    municipality:                "Grenoble"
    DPO:                         null
    controller_id:               1
  ▼ usages:
    ▼ 0:
          id:                    1
          created_at:            "2018-07-10 14:08:59"
          updated_at:            "2018-07-10 14:08:59"
          data_category:         "MAC address"
          processing_purpose:    "pedestrian traffic analysis"
          processing_basis:      "consent"
          third_parties:         null
          recipients:            "Ville de Grenoble"
          services:              "It provides a better pedestrian experience"
          retention_time:        30
          aggregated:            1
          anonymized:            1
          policy:                null
          device_id:             1
    ▼ 1:
          id:                    2
          created_at:            "2018-07-10 14:08:59"
          updated_at:            "2018-07-10 14:08:59"
          data_category:         "bluetooth address"
          processing_purpose:    "pedestrian traffic analysis"
          processing_basis:      "consent"
          third_parties:         null
          recipients:            "Ville de Grenoble"
          services:              "It provides a better pedestrian experience"
          retention_time:        20
          aggregated:            0
          anonymized:            0
          policy:                null
          device_id:             1
  ▼ data_controllers:
    ▼ 0:
          id:                    1
          created_at:            "2018-07-10 14:08:59"
          updated_at:            "2018-07-10 14:08:59"
          name:                  "Ville de Grenoble"
          contact_details:       "contact@grenoble.fr"
```

Figure 52: Excerpt of the API.

D
APPLIED CRYPTOGRAPHY

Some solutions presented in Chapter 4 rely on cryptography: a set of techniques, backed up by mathematical properties, devised to prevent unauthorised entities the access to information. However, uses of cryptography do not only lie in the encryption of data — *i.e.*, the encoding of data in such a way that only authorised parties can access it. It can also help authenticate and verify the integrity of data — respectively prove the identity of the originator of data, and prove data has not been altered according to a previous state. These other aspects of cryptography, authentication and integrity, are presented in this section. Authentication and integrity can be achieved in other ways when applied to physical entities, such as manual signatures and seals respectively. However, cryptography provides mathematical proof, and is more appropriate in a digital context. Explanation of these two aspects provides the reader the basis of an understanding of cryptographic signatures and Merkle hash trees, later used in this document. The content of this section comes from Katz *et al.* [73] if not specified otherwise.

D.1 AUTHENTICATION

A well-known area of cryptography is public-key cryptography, first proposed by Diffie and Hellman in [44]. This part of cryptography consists in the encryption of data by a party and the decryption by another, using a pair of keys named public and private keys. A pair of keys can also provide authenticity of data.

D.1.1 *Public and private keys*

Asymmetric keys consist of pair of keys: public keys which may be disseminated widely, and private keys which are known only to the owner. Each pair of keys is unique: to a public key corresponds only one private key, and *vice versa*. It is practically impossible to counterfeit a private key, the computational cost is way beyond the current capacities of modern computers, and the theoretical limits of non-quantum computing makes it impossible for large size keys such as 256 bits [25]. The public key is used to encrypt data, *i.e.*, made understandable only be authorized parties. Encrypted data is usually denoted as ciphertext, as opposed to plain-text. This ciphertext can only be decrypted by the private key.

d.1.2 *Cryptographic signatures*

Asymmetric keys can also be of use to uniquely sign data, *i.e.*, to ensure the authenticity. In that case, the private key is used, and anyone can verify the resulting signature using the public key. Usually, a hash of the data is signed and not the data itself, due to complexity issues: a hash is always a (relatively) small character string, whereas the data to whom it belongs may not.

d.2 integrity

Another important use of cryptography is the verification of the integrity of data. A typical way to verify data is through the use of hash functions.

d.2.1 *Cryptographic Hash Functions*

A Cryptographic Hash Function (CHF) is a function that maps data of any size to a single and fixed value and that is considered suitable for cryptography. CHF are of use for Merkle hash trees in our case. A function h is required to have certain properties to be considered a CHF suitable for cryptography:

compression A function h maps an input x of arbitrary finite bitlength, to an output $h(x)$ of fixed bitlength n.

ease of computation Given h and an input x, $h(x)$ is easy to compute.

preimage resistance For essentially all pre-specified outputs, it is computationally infeasible to find any input which hashes to that output, *i.e.*, to find any preimage x' such that $h(x') = y$ when given any y for which a corresponding input is not known. This property is also named *one-way*.

2nd-preimage resistance It is computationally infeasible to find any second input which has the same output as any specified input, *i.e.*, given x, to find a 2nd-preimage $x'' = x$ such that $h(x) = h(x')$. This property is also named *weak collision resistance*.

collision resistance It is computationally infeasible to find any two distinct inputs x, x' which hash to the same output, *i.e.*, such that $h(x) = h(x')$. This property is also named *strong collision resistance*.

BIBLIOGRAPHY

[1] May 25 and 2016. *Smart City Spotlight: Barcelona*. URL: https://www.channelfutures.com/msp-501/smart-city-spotlight-barcelona (visited on 10/04/2019).

[2] Martín Abadi. "Logic in Access Control." In: *18th Annual IEEE Symposium of Logic in Computer Science, 2003. Proceedings.* IEEE, 2003, pp. 228–233.

[3] *About Core Bluetooth*. URL: https://developer.apple.com/library/archive/documentation/NetworkingInternetWeb/Conceptual/CoreBluetooth_concepts/AboutCoreBluetooth/Introduction.html (visited on 12/03/2019).

[4] Alessandro Acquisti et al. "Nudges for Privacy and Security: Understanding and Assisting Users' Choices Online." In: *ACM Computing Surveys* 50.3 (Aug. 8, 2017), pp. 1–41. ISSN: 03600300. DOI: 10.1145/3054926. URL: http://dl.acm.org/citation.cfm?doid=3101309.3054926 (visited on 09/20/2017).

[5] Rakesh Agrawal, Jerry Kiernan, Ramakrishnan Srikant, and Yirong Xu. "An XPath-Based Preference Language for P3P." In: *Proceedings of the 12th International Conference on World Wide Web*. ACM, 2003, pp. 629–639. URL: http://dl.acm.org/citation.cfm?id=775241 (visited on 02/01/2017).

[6] Hanaa Alshareef, Raúl Pardo, Gerardo Schneider, and Pablo Picazo-Sanchez. "A Collaborative Access Control Framework for Online Social Networks." In: *Journal of Logical and Algebraic Methods in Programming* 114 (Aug. 2020), p. 100562. ISSN: 23522208. DOI: 10/ggx4tj. URL: https://linkinghub.elsevier.com/retrieve/pii/S235222082030047X (visited on 06/02/2020).

[7] *Android Keystore System*. URL: https://developer.android.com/training/articles/keystore (visited on 08/29/2019).

[8] Apple. *Managing Pop-Up Buttons and Pull-Down Lists*. URL: https://developer.apple.com/library/archive/documentation/Cocoa/Conceptual/MenuList/Articles/ManagingPopUpItems.html (visited on 06/12/2019).

[9] Pablo Aragón, Andreas Kaltenbrunner, Antonio Calleja-López, Andrés Pereira, Arnau Monterde, Xabier E. Barandiaran, and Vicenç Gómez. "Deliberative Platform Design: The Case Study of the Online Discussions in Decidim Barcelona." In: (July 20, 2017). arXiv: 1707.06526 [cs]. URL: http://arxiv.org/abs/1707.06526 (visited on 10/04/2019).

[10] Hannah Arendt. *The Human Condition*. 2nd ed. Chicago: University of Chicago Press, 1998. 349 pp. ISBN: 978-0-226-02599-5 978-0-226-02598-8.

[11] Paul Ashley, Satoshi Hada, Günter Karjoth, and Matthias Schunter. "E-P3P Privacy Policies and Privacy Authorization." In: *Proceedings of the 2002 ACM Workshop on Privacy in the Electronic Society*. ACM, 2002, pp. 103–109. URL: http://dl.acm.org/citation.cfm?id=644538 (visited on 02/01/2017).

[12] Kevin Ashton. "That 'Internet of Things' Thing." In: (2009), p. 1.

[13] Shahar Avigezer. *How To Use Android BLE to Communicate with Bluetooth Devices - An Overview & Code Examples*. URL: https://medium.com/@avigezerit/bluetooth-low-energy-on-android-22bc7310387a (visited on 07/31/2019).

[14] Christel Baier and Joost-Pieter Katoen. *Principles of Model Checking*. MIT Press, 2008. ISBN: 978-0-262-02649-9.

[15] Tuba Bakıcı, Esteve Almirall, and Jonathan Wareham. "A Smart City Initiative: The Case of Barcelona." In: *Journal of the Knowledge Economy* 4.2 (June 2013), pp. 135–148. ISSN: 1868-7865, 1868-7873. DOI: 10/fzjb45. URL: http://link.springer.com/10.1007/s13132-012-0084-9 (visited on 10/04/2019).

[16] Genevieve Bell and Paul Dourish. "Yesterday's Tomorrows: Notes on Ubiquitous Computing's Dominant Vision." In: *Personal and Ubiquitous Computing* 11.2 (Jan. 24, 2007), pp. 133–143. ISSN: 1617-4909, 1617-4917. DOI: 10/fphzkq. URL: http://link.springer.com/10.1007/s00779-006-0071-x (visited on 09/26/2019).

[17] Mihir Bellare. "Forward Integrity For Secure Audit Logs." In: (), p. 16.

[18] Clara Benevolo, Renata Paola Dameri, and Beatrice D'Auria. "Smart Mobility in Smart City." In: *Empowering Organizations*. Ed. by Teresina Torre, Alessio Maria Braccini, and Riccardo Spinelli. Vol. 11. Cham: Springer International Publishing, 2016, pp. 13–28. ISBN: 978-3-319-23783-1 978-3-319-23784-8. DOI: 10.1007/978-3-319-23784-8_2. URL: http://link.springer.com/10.1007/978-3-319-23784-8_2 (visited on 08/28/2017).

[19] *Bluetooth Low Energy*. URL: https://source.android.com/devices/bluetooth/ble (visited on 07/31/2019).

[20] Bluetooth SIG Proprietary. *Bluetooth SIG Specifications*. 2016.

[21] Ahmed Boubrima, Walid Bechkit, and Herve Rivano. "On the Deployment of Wireless Sensor Networks for Air Quality Mapping: Optimization Models and Algorithms." In: *IEEE/ACM Transactions on Networking* 27.4 (Aug. 2019), pp. 1629–1642. ISSN: 1063-6692, 1558-2566. DOI: 10/gf9cdv. URL: https://ieeexplore.ieee.org/document/8750870/ (visited on 10/03/2019).

[22] Danièle Bourcier and Primavera de Filippi. "Vers un droit collectif sur les données de santé." In: (), p. 15.

[23] T. Bray. *The JavaScript Object Notation (JSON) Data Interchange Format*. URL: https://tools.ietf.org/html/rfc7159 (visited on 08/22/2019).

[24] Davis Wright Tremaine LLP-Emily Bruemmer. *Online Advertising Technology: The U.K. ICO Takes an Interest | Lexology*. URL: https://www.lexology.com/library/detail.aspx?g=f85184e7-d17d-4878-a198-4a3b762fe7d5 (visited on 11/08/2019).

[25] *Brute Force Key Attacks Are for Dummies*. URL: https://blog.codinghorror.com/brute-force-key-attacks-are-for-dummies/ (visited on 09/18/2019).

[26] Mathias Buus. *Hypercore Is a Secure, Distributed Append-Only Log.: Mafintosh/Hypercore*. URL: https://github.com/mafintosh/hypercore (visited on 05/23/2019).

[27] CNIL. *The CNIL's Restricted Committee Imposes a Financial Penalty of 50 Million Euros against GOOGLE LLC*. URL: https://www.cnil.fr/en/cnils-restricted-committee-imposes-financial-penalty-50-million-euros-against-google-llc (visited on 03/28/2019).

[28] *CV Dazzle: Camouflage from Face Detection*. URL: https://cvdazzle.com/ (visited on 11/06/2019).

[29] Antonio Casilli. *Quelle Protection de La Vie Privée Face Aux Attaques Contre Nos Libertés Numériques ? | Antonio A. Casilli*. URL: http://www.casilli.fr/2015/02/07/7013/ (visited on 03/11/2020).

[30] Claude Castelluccia and Daniel Le Métayer. "Analyse des impacts de la reconnaissance faciale Quelques éléments de méthode." In: (), p. 34.

[31] F. H. Cate. "The Limits of Notice and Choice." In: *IEEE Security Privacy* 8.2 (Mar. 2010), pp. 59–62. ISSN: 1540-7993. DOI: 10/cgjkcd.

[32] S. Cha, M. Chuang, K. Yeh, Z. Huang, and C. Su. "A User-Friendly Privacy Framework for Users to Achieve Consents With Nearby BLE Devices." In: *IEEE Access* 6 (2018), pp. 20779–20787. ISSN: 2169-3536. DOI: 10/gf3hf7.

[33] Régis Chatellier. *Ville Numérique : Quels Impacts Sur La Vie Privée? | LINC*. Sept. 13, 2016. URL: https://linc.cnil.fr/fr/ville-numerique-quels-impacts-sur-la-vie-privee (visited on 08/22/2017).

[34] Hafedh Chourabi, Taewoo Nam, Shawn Walker, J. Ramon Gil-Garcia, Sehl Mellouli, Karine Nahon, Theresa A. Pardo, and Hans Jochen Scholl. "Understanding Smart Cities: An Integrative Framework." In: IEEE, Jan. 2012, pp. 2289–2297. ISBN: 978-1-4577-1925-7 978-0-7695-4525-7. DOI: 10.1109/HICSS.2012.615. URL: http://ieeexplore.ieee.org/document/6149291/ (visited on 11/14/2016).

[35] Lorrie Faith Cranor. "NECESSARY BUT NOT SUFFICIENT: STANDARDIZED MECHANISMS FOR PRIVACY NOTICE AND CHOICE." In: 10 (), p. 36.

[36] Lorrie Cranor, Marc Langheinrich, Massimo Marchiori, Martin Presler-Marshall, and Joseph Reagle. "The Platform for Privacy Preferences 1.0 (P3P1. 0) Specification." In: *W3C recommendation* 16 (2002). URL: https://elearn.inf.tu-dresden.de/hades/teleseminare/wise0405/Act.%208%20Models%20Languages%20Pierangela/Materials/P3P.pdf (visited on 02/01/2017).

[37] *Create P2P Connections with Wi-Fi Direct*. URL: https://developer.android.com/training/connect-devices-wirelessly/wifi-direct (visited on 12/03/2019).

[38] *Cryptography - Secure Function Evaluation*. URL: https://crypto.stanford.edu/pbc/notes/crypto/sfe.html (visited on 12/13/2019).

[39] Anupam Das, Martin Degeling, Xiaoyou Wang, Junjue Wang, Norman Sadeh, and Mahadev Satyanarayanan. "Assisting Users in a World Full of Cameras." In: (). URL: http://www.cs.cmu.edu/~anupamd/paper/CV-COPS-2017.pdf (visited on 08/31/2017).

[40] Anupam Das, Martin Degeling, Daniel Smullen, and Norman Sadeh. "Personalized Privacy Assistants for the Internet of Things." In: *IEEE PERVASIVE COMPUTING* 2018 (2018). DOI: 10.1109/MPRV.2018.03367733.

[41] Sourya Joyee De and Daniel Le Metayer. "Privacy Harm Analysis: A Case Study on Smart Grids." In: IEEE, May 2016, pp. 58–65. ISBN: 978-1-5090-3690-5. DOI: 10.1109/SPW.2016.21. URL: http://ieeexplore.ieee.org/document/7527754/ (visited on 11/14/2016).

[42] Martin Degeling, Christine Utz, Christopher Lentzsch, Henry Hosseini, Florian Schaub, and Thorsten Holz. "We Value Your Privacy ... Now Take Some Cookies: Measuring the GDPR's Impact on Web Privacy." In: *Proceedings 2019 Network and*

Distributed System Security Symposium (2019). DOI: 10/gfxgxm. arXiv: 1808.05096. URL: http://arxiv.org/abs/1808.05096 (visited on 09/11/2019).

[43] Levent Demir, Mathieu Cunche, and Cédric Lauradoux. "Analysing the Privacy Policies of Wi-Fi Trackers." In: ACM Press, 2014, pp. 39–44. ISBN: 978-1-4503-2825-8. DOI: 10.1145/2611264.2611266. URL: http://dl.acm.org/citation.cfm?doid=2611264.2611266 (visited on 11/14/2016).

[44] W. Diffie and M. Hellman. "New Directions in Cryptography." In: *IEEE Transactions on Information Theory* 22.6 (Nov. 1976), pp. 644–654. ISSN: 0018-9448. DOI: 10/cmtvwm. URL: http://ieeexplore.ieee.org/document/1055638/ (visited on 10/01/2019).

[45] *Do Users Change Their Settings? — Archive of UIE/Brainsparks*. URL: https://archive.uie.com/brainsparks/2011/09/14/do-users-change-their-settings/ (visited on 11/08/2019).

[46] *Documentation - Materialize*. URL: https://materializecss.com/ (visited on 08/22/2019).

[47] Shan Du, Mahmoud Ibrahim, Mohamed Shehata, and Wael Badawy. "Automatic License Plate Recognition (ALPR): A State-of-the-Art Review." In: *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY* 23.2 (2013), p. 15. DOI: 10/gf9z4g.

[48] *ESP32 Overview | Espressif Systems*. URL: https://www.espressif.com/en/products/hardware/esp32/overview (visited on 08/27/2019).

[49] Serge Egelman, Raghudeep Kannavara, and Richard Chow. "Is This Thing On?: Crowdsourcing Privacy Indicators for Ubiquitous Sensing Platforms." In: ACM Press, 2015, pp. 1669–1678. ISBN: 978-1-4503-3145-6. DOI: 10.1145/2702123.2702251. URL: http://dl.acm.org/citation.cfm?doid=2702123.2702251 (visited on 10/18/2017).

[50] Pardis Emami-Naeini, Henry Dixon, Yuvraj Agarwal, and Lorrie Faith Cranor. "Exploring How Privacy and Security Factor into IoT Device Purchase Behavior." In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems - CHI '19*. The 2019 CHI Conference. Glasgow, Scotland Uk: ACM Press, 2019, pp. 1–12. ISBN: 978-1-4503-5970-2. DOI: 10/gf5d6v. URL: http://dl.acm.org/citation.cfm?doid=3290605.3300764 (visited on 07/25/2019).

[51] Yaniv Erlich, Tal Shor, Itsik Pe'er, and Shai Carmi. "Identity Inference of Genomic Data Using Long-Range Familial Searches." In: *Science* 362.6415 (Nov. 9, 2018), pp. 690–694. ISSN: 0036-8075, 1095-9203. DOI: 10/cvr2. pmid: 30309907. URL: https:

//science.sciencemag.org/content/362/6415/690 (visited on 03/11/2020).

[52] European Commision. *Proposal for a REGULATION OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL Concerning the Respect for Private Life and the Protection of Personal Data in Electronic Communications and Repealing Directive 2002/58/EC (Regulation on Privacy and Electronic Communications)*. Oct. 1, 2017.

[53] European Parliament. *General Data Protection Regulation*. Apr. 26, 2016.

[54] European Parliament. *REPORT on the Proposal for a Regulation of the European Parliament and of the Council Concerning the Respect for Private Life and the Protection of Personal Data in Electronic Communications and Repealing Directive 2002/58/EC (Regulation on Privacy and Electronic Communications)*. Oct. 20, 2017. URL: http://www.europarl.europa.eu/doceo/document/A-8-2017-0324_EN.html (visited on 10/14/2019).

[55] Federal Trade Commission. "FAIR INFORMATION PRACTICES IN THE ELECTRONIC MARKETPLACE." In: *PRIVACY ONLINE* (June 2000), p. 208.

[56] Roy Thomas Fielding. "Architectural Styles and the Design of Network-Based Software Architectures." 2000.

[57] Michel Foucault and Elissa Mailänder. *Surveiller et Punir*. Vol. 225. na, 1975.

[58] *'Free' Wi-Fi Hotspots Can Track Your Location Even When You Aren't Connected*. Nov. 1, 2018. URL: https://www.pcworld.com/article/3315197/free-wi-fi-hotspots-can-track-your-location-even-when-you-arent-connected.html (visited on 10/14/2019).

[59] Carles Gomez and Josep Paradells. "Wireless Home Automation Networks: A Survey of Architectures and Technologies." In: *IEEE Communications Magazine* 48.6 (June 2010), pp. 92–101. ISSN: 0163-6804. DOI: 10.1109/MCOM.2010.5473869. URL: http://ieeexplore.ieee.org/document/5473869/ (visited on 11/17/2016).

[60] Google. *Privacy Policy – Privacy & Terms – Google. Effective 22 January 2019*. URL: https://policies.google.com/privacy?gl=en&hl=en-GB (visited on 03/28/2019).

[61] Colin M. Gray, Yubo Kou, Bryan Battles, Joseph Hoggatt, and Austin L. Toombs. "The Dark (Patterns) Side of UX Design." In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. The 2018 CHI Conference. Montreal QC, Canada: ACM Press, 2018, pp. 1–14. ISBN: 978-1-4503-5620-

6. DOI: 10/gfxvpz. URL: http://dl.acm.org/citation.cfm?doid=3173574.3174108 (visited on 04/03/2019).

[62] C. Harrison, B. Eckman, R. Hamilton, P. Hartswick, J. Kalagnanam, J. Paraszczak, and P. Williams. "Foundations for Smarter Cities." In: *IBM Journal of Research and Development* 54.4 (July 2010), pp. 1–16. ISSN: 0018-8646, 0018-8646. DOI: 10/dxrvk6. URL: http://ieeexplore.ieee.org/document/5512826/ (visited on 10/03/2019).

[63] Kazuhiko Hashimoto, Makoto Yoshinomoto, Satoshi Matsueda, Katsuya Morinaka, and Nobuyuki Yoshiike. "Development of People-Counting System with Human-Information Sensor Using Multi-Element Pyroelectric Infrared Array Detector." In: *Sensors and Actuators A: Physical* 58.2 (Feb. 1997), pp. 165–171. ISSN: 09244247. DOI: 10/btjm8g. URL: https://linkinghub.elsevier.com/retrieve/pii/S0924424796014008 (visited on 12/04/2019).

[64] Robert G. Hollands. "Will the Real Smart City Please Stand up?: Intelligent, Progressive or Entrepreneurial?" In: *City* 12.3 (Dec. 2008), pp. 303–320. ISSN: 1360-4813, 1470-3629. DOI: 10.1080/13604810802479126. URL: https://www.tandfonline.com/doi/full/10.1080/13604810802479126 (visited on 08/28/2017).

[65] Michael Huth and Mark Ryan. *Logic in Computer Science: Modelling and Reasoning about Systems*. Cambridge University Press, 2004.

[66] ICO. *Guidance on the Use of Cookies and Similar Technologies*. Dec. 4, 2019. URL: https://ico.org.uk/for-organisations/guide-to-pecr/guidance-on-the-use-of-cookies-and-similar-technologies/ (visited on 12/09/2019).

[67] *Internet of Things (IoT) History*. URL: https://www.postscapes.com/internet-of-things-history/ (visited on 10/02/2019).

[68] *IoT: Number of Connected Devices Worldwide 2012-2025*. 2019. URL: https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/ (visited on 10/02/2019).

[69] Marijn Janssen, Yannis Charalabidis, and Anneke Zuiderwijk. "Benefits, Adoption Barriers and Myths of Open Data and Open Government." In: *Information Systems Management* 29.4 (Sept. 2012), pp. 258–268. ISSN: 1058-0530, 1934-8703. DOI: 10/gf39gb. URL: http://www.tandfonline.com/doi/abs/10.1080/10580530.2012.716740 (visited on 10/04/2019).

[70] Mohamed Jawad, Patricia Serrano Alvarado, and Patrick Valduriez. "Design of PriServ, a Privacy Service for DHTs." In: *Proceedings of the 2008 International Workshop on Privacy and Anonymity in Information Society - PAIS '08*. The 2008 International Workshop. Nantes, France: ACM Press, 2008, p. 21. ISBN:

978-1-59593-965-4. DOI: 10.1145/1379287.1379293. URL: http://portal.acm.org/citation.cfm?doid=1379287.1379293 (visited on 07/01/2020).

[71] Théo Jourdan, Antoine Boutet, and Carole Frindel. "Toward Privacy in IoT Mobile Devices for Activity Recognition." In: *Proceedings of the 15th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services - MobiQuitous '18*. The 15th EAI International Conference. New York, NY, USA: ACM Press, 2018, pp. 155–165. ISBN: 978-1-4503-6093-7. DOI: 10/gf86mj. URL: http://dl.acm.org/citation.cfm?doid=3286978.3287009 (visited on 10/02/2019).

[72] Saffija Kasem-Madani and Michael Meier. "Security and Privacy Policy Languages: A Survey, Categorization and Gap Identification." In: *arXiv preprint arXiv:1512.00201* (2015). URL: https://arxiv.org/abs/1512.00201 (visited on 04/10/2017).

[73] Jonathan Katz, Alfred J Menezes, Paul C Van Oorschot, and Scott A Vanstone. *Handbook Of Applied Cryptography*. URL: http://labit501.upct.es/~fburrull/docencia/SeguridadEnRedes/teoria/bibliography/HandbookOfAppliedCryptography_AMenezes.pdf (visited on 06/05/2019).

[74] *Keychain Services | Apple Developer Documentation*. URL: https://developer.apple.com/documentation/security/keychain_services (visited on 12/03/2019).

[75] *Keys | Apple Developer Documentation*. URL: https://developer.apple.com/documentation/security/certificate_key_and_trust_services/keys (visited on 12/03/2019).

[76] Rob Kitchin. "Getting Smarter about Smart Cities: Improving Data Privacy and Data Security." In: (2016). URL: http://eprints.maynoothuniversity.ie/7242/1/Smart (visited on 08/22/2017).

[77] Katharina Krombholz, Adrian Dabrowski, Matthew Smith, and Edgar Weippl. "Ok Glass, Leave Me Alone: Towards a Systematization of Privacy Enhancing Technologies for Wearable Computing." In: *Financial Cryptography and Data Security*. Ed. by Michael Brenner, Nicolas Christin, Benjamin Johnson, and Kurt Rohloff. Vol. 8976. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 274–280. ISBN: 978-3-662-48050-2 978-3-662-48051-9. DOI: 10.1007/978-3-662-48051-9_20. URL: http://link.springer.com/10.1007/978-3-662-48051-9_20 (visited on 10/02/2019).

[78] Oksana Kulyk, Annika Hilt, Nina Gerber, and Melanie Volkamer. ""This Website Uses Cookies": Users' Perceptions and Reactions to the Cookie Disclaimer." In: *Proceedings 3rd European Workshop on Usable Security*. European Workshop on

Usable Security. London, England: Internet Society, 2018. ISBN: 978-1-891562-54-9. DOI: 10/gf8f9t. URL: https://www.ndss-symposium.org/wp-content/uploads/2018/06/eurousec2018_12_Kulyk_paper.pdf (visited on 09/18/2019).

[79] La Quadrature Du Net. *ePrivacy - LQDN*. URL: https://eprivacy.laquadrature.net/en/ (visited on 10/14/2019).

[80] Marc Langheinrich. "A Privacy Awareness System for Ubiquitous Computing Environments." In: *International Conference on Ubiquitous Computing*. Springer, 2002, pp. 237–245. URL: http://link.springer.com/chapter/10.1007/3-540-45809-3_19 (visited on 11/14/2016).

[81] Marc Langheinrich, Lorrie Cranor, and Massimo Marchiori. "Appel: A P3p Preference Exchange Language." In: *W3C Working Draft* (2002). URL: https://www.w3.org/TR/P3P-preferences/.

[82] *Laravel - The PHP Framework For Web Artisans*. URL: https://laravel.com/ (visited on 10/28/2019).

[83] Jakob Eg Larsen, Piotr Sapiezynski, Arkadiusz Stopczynski, Morten Moerup, and Rasmus Theodorsen. "Crowds, Bluetooth, and Rock-n-Roll. Understanding Music Festival Participant Behavior." In: (June 13, 2013). arXiv: 1306.3133 [cs, stat]. URL: http://arxiv.org/abs/1306.3133 (visited on 10/16/2019).

[84] Christophe Lazaro and Daniel Le Métayer. "Control over Personal Data: True Remedy or Fairytale?" In: *SCRIPTed* 12.1 (June 2015). ISSN: 17442567. DOI: 10.2966/scrip.120115.3. URL: http://script-ed.org/?p=1927 (visited on 12/13/2016).

[85] *Leaflet - a JavaScript Library for Interactive Maps*. URL: https://leafletjs.com/ (visited on 10/28/2019).

[86] Linda Lee, JoongHwa Lee, Serge Egelman, and David Wagner. "Information Disclosure Concerns in The Age of Wearable Computing." In: *NDSS Workshop on Usable Security (USEC)* (2016). DOI: 10.14722/usec.2016.23006. URL: https://www.internetsociety.org/sites/default/files/blogs-media/information-disclosure-concerns-in-the-age-of-wearable-computing.pdf (visited on 11/14/2016).

[87] Timothy B. Lee. *Amazon Admits That Employees Review "Small Sample" of Alexa Audio*. Apr. 11, 2019. URL: https://arstechnica.com/tech-policy/2019/04/amazon-admits-that-employees-review-small-sample-of-alexa-audio/ (visited on 11/08/2019).

[88] Dominik Leibenger, Frederik Möllers, Anna Petrlic, Ronald Petrlic, and Christoph Sorge. "Privacy Challenges in the Quantified Self Movement – An EU Perspective." In: *Proceedings on Privacy Enhancing Technologies* 2016.4 (Jan. 1, 2016). ISSN: 2299-0984. DOI: 10.1515/popets-2016-0042. URL: http://www.

degruyter.com/view/j/popets.2016.2016.issue-4/popets-2016-0042/popets-2016-0042.xml (visited on 11/14/2016).

[89] *Les panneaux de pub du métro tracent-ils les téléphones des usagers ?* URL: https://www.liberation.fr/checknews/2019/03/25/les-panneaux-de-pub-du-metro-tracent-ils-les-telephones-des-usagers_1717316 (visited on 10/15/2019).

[90] Lawrence Lessig. *Code and Other Laws of Cyberspace*. Version 2.0. New York: Basic Books, 2006. 410 pp. ISBN: 978-0-465-03914-2.

[91] Jialiu Lin, Bin Liu, Norman Sadeh, and Jason I. Hong. "Modeling Users' Mobile App Privacy Preferences: Restoring Usability in a Sea of Permission Settings." In: *Symposium On Usable Privacy and Security (SOUPS 2014)*. 2014, pp. 199–212. URL: https://www.usenix.org/conference/soups2014/proceedings/presentation/lin (visited on 11/14/2016).

[92] LineageOS. *LineageOS – LineageOS Android Distribution*. URL: https://lineageos.org/ (visited on 10/17/2019).

[93] Bin Liu, Jialiu Lin, and Norman Sadeh. "Reconciling Mobile App Privacy and Usability on Smartphones: Could User Privacy Profiles Help?" In: *Proceedings of the 23rd International Conference on World Wide Web*. ACM, 2014, pp. 201–212. URL: http://dl.acm.org/citation.cfm?id=2568035 (visited on 11/14/2016).

[94] *MOREL Victor / CoIoT*. URL: https://gitlab.inria.fr/vmorel/coiot (visited on 07/31/2019).

[95] Douglas MacMillan. "Tech's 'Dirty Secret': The App Developers Sifting Through Your Gmail." In: *Wall Street Journal. Tech* (). ISSN: 0099-9660. URL: https://www.wsj.com/articles/techs-dirty-secret-the-app-developers-sifting-through-your-gmail-1530544442 (visited on 10/17/2019).

[96] *Material Design*. URL: https://material.io/design/ (visited on 08/22/2019).

[97] Célestin Matte. "Wi-Fi Tracking: Fingerprinting Attacks and Counter-Measures."

[98] Célestin Matte, Nataliia Bielova, and Cristiana Santos. "Do Cookie Banners Respect My Choice? Measuring Legal Compliance of Banners from IAB Europe's Transparency and Consent Framework." In: (Nov. 22, 2019). arXiv: 1911.09964 [cs]. URL: http://arxiv.org/abs/1911.09964 (visited on 11/27/2019).

[99] Célestin Matte and Mathieu Cunche. "Wombat: An Experimental Wi-Fi Tracking System." In: (), p. 11.

[100] Andrea M Matwyshyn. "The Internet of Bodies." In: *William & Mary Law Review* 61.1 (2019).

[101] Damon McCoy, Kevin Bauer, Dirk Grunwald, Tadayoshi Kohno, and Douglas Sicker. "Shining Light in Dark Places: Understanding the Tor Network." In: *Privacy Enhancing Technologies*. Ed. by Nikita Borisov and Ian Goldberg. Vol. 5134. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 63–76. ISBN: 978-3-540-70629-8 978-3-540-70630-4. DOI: 10.1007/978-3-540-70630-4_5. URL: http://link.springer.com/10.1007/978-3-540-70630-4_5 (visited on 10/17/2019).

[102] Aleecia M. McDonald and Lorrie Faith Cranor. "The Cost of Reading Privacy Policies." In: *ISJLP* 4 (2008), p. 543. URL: http://heinonline.org/hol-cgi-bin/get_pdf.cgi?handle=hein.journals/isjlpsoc4&section=27 (visited on 12/19/2016).

[103] Ralph C. Merkle. "Protocols for Public Key Cryptosystems." In: *1980 IEEE Symposium on Security and Privacy*. 1980 IEEE Symposium on Security and Privacy. Oakland, CA, USA: IEEE, Apr. 1980, pp. 122–122. ISBN: 978-0-8186-0335-8. DOI: 10/bmvbd6. URL: http://ieeexplore.ieee.org/document/6233691/ (visited on 06/05/2019).

[104] Jeffrey C. Mogul, Jim Gettys, Tim Berners-Lee, and Henrik Frystyk. *Hypertext Transfer Protocol – HTTP/1.1*. URL: https://tools.ietf.org/html/rfc2068 (visited on 12/03/2019).

[105] *Morel Victor / DCG_ledger*. URL: http://git.felinn.org/ummon/dcg_ledger (visited on 10/29/2019).

[106] *Morel Victor / Map of Things*. URL: http://git.felinn.org/ummon/map-of-things (visited on 10/29/2019).

[107] Victor Morel and Raúl Pardo. "SoK: Three Facets of Privacy Policies." In: *WPES*. Aug. 24, 2020. DOI: 10.1145/3411497.3420216. URL: https://hal.inria.fr/hal-02267641 (visited on 09/25/2019).

[108] Evgeny Morozov. *To Save Everything, Click Here: The Folly of Technological Solutionism*. Public Affairs, 2013.

[109] Einar Mykletun and Maithili Narasimha. "Providing Authentication and Integrity in Outsourced Databases Using Merkle Hash Tree's." In: (), p. 7.

[110] 1615 L. St NW, Suite 800 Washington, and DC 20036 USA202-419-4300 | Main202-419-4349 | Fax202-419-4372 | Media Inquiries. *Demographics of Mobile Device Ownership and Adoption in the United States*. URL: https://www.pewinternet.org/fact-sheet/mobile/ (visited on 08/22/2019).

[111] Ricardo Neisse, Gianmarco Baldini, Gary Steri, Yutaka Miyake, Shinsaku Kiyomoto, and Abdur Rahim Biswas. "An Agent-Based Framework for Informed Consent in the Internet of Things." In: *Internet of Things (WF-IoT), 2015 IEEE 2nd World Forum On*. IEEE, 2015, pp. 789–794. URL: http://ieeexplore.

ieee.org/xpls/abs_all.jsp?arnumber=7389154 (visited on 12/05/2016).

[112] Ricardo Neisse, Gary Steri, Igor Nai Fovino, and Gianmarco Baldini. "SecKit: A Model-Based Security Toolkit for the Internet of Things." In: *Computers & Security* 54 (Oct. 2015), pp. 60–76. ISSN: 01674048. DOI: 10.1016/j.cose.2015.06.002. URL: http://linkinghub.elsevier.com/retrieve/pii/S0167404815000887 (visited on 06/01/2017).

[113] Helen Nissenbaum. *Why Data Privacy Based on Consent Is Impossible*. URL: https://hbr.org/2018/09/stop-thinking-about-consent-it-isnt-possible-and-it-isnt-right (visited on 09/28/2018).

[114] *Notifications - System Capabilities - iOS - Human Interface Guidelines - Apple Developer*. URL: https://developer.apple.com/design/human-interface-guidelines/ios/system-capabilities/notifications/ (visited on 12/03/2019).

[115] *Notifications Overview*. URL: https://developer.android.com/guide/topics/ui/notifiers/notifications (visited on 07/31/2019).

[116] *Npm/Cli*. npm. URL: https://github.com/npm/cli (visited on 12/04/2019).

[117] Maxwell Ogden, Karissa McKelvey, Mathias Buus Madsen, and Code for Science. *Dat - Distributed Dataset Synchronization And Versioning*. preprint. Open Science Framework, Jan. 31, 2017. DOI: 10.31219/osf.io/nsv2c. URL: https://osf.io/nsv2c (visited on 05/23/2019).

[118] Dieter Oosterlinck, Dries F Benoit, Philippe Baecke, and Nico Van de. "Bluetooth Tracking of Humans in an Indoor Environment: An Application to Shopping Mall Visits." In: (), p. 34. DOI: 10/f9p4t9.

[119] *OpenStreetMap*. URL: https://www.openstreetmap.org/ (visited on 10/28/2019).

[120] Organisation for Economic Co-operation and Development. *Skills Matter: Further Results from the Survey of Adult Skills*. Paris: OECD, 2016.

[121] Mark Otto. *Bootstrap*. URL: https://getbootstrap.com/ (visited on 08/22/2019).

[122] Primal Pappachan, Martin Degeling, Roberto Yus, Anupam Das, Sruti Bhagavatula, William Melicher, Pardis Emami Naeini, Shikun Zhang, Lujo Bauer, Alfred Kobsa, et al. "Towards Privacy-Aware Smart Buildings: Capturing, Communicating, and Enforcing Privacy Policies and Preferences." In: *Distributed Computing Systems Workshops (ICDCSW), 2017 IEEE 37th International Conference On*. IEEE, 2017, pp. 193–198. URL: http:

//ieeexplore.ieee.org/abstract/document/7979816/ (visited on 08/31/2017).

[123] Raúl Pardo and Daniel Le Métayer. “Analysis of Privacy Policies to Enhance Informed Consent.” In: *Data and Applications Security and Privacy XXXIII*. Ed. by Simon N. Foley. Vol. 11559. Cham: Springer International Publishing, 2019, pp. 177–198. ISBN: 978-3-030-22478-3 978-3-030-22479-0. DOI: 10.1007/978-3-030-22479-0_10. URL: http://link.springer.com/10.1007/978-3-030-22479-0_10 (visited on 08/22/2019).

[124] Raúl Pardo and Daniel Le Métayer. “Analysis of Privacy Policies to Enhance Informed Consent (Extended Version).” In: (), p. 26.

[125] *Performing Common Central Role Tasks*. URL: https://developer.apple.com/library/archive/documentation/NetworkingInternetWeb/Conceptual/CoreBluetooth_concepts/PerformingCommonCentralRoleTasks/PerformingCommonCentralRoleTasks.html#//apple_ref/doc/uid/TP40013257-CH3-SW1 (visited on 12/03/2019).

[126] *Privacy Concerns Regarding Google*. In: *Wikipedia*. URL: https://en.wikipedia.org/w/index.php?title=Privacy_concerns_regarding_Google&oldid=919932316 (visited on 10/17/2019).

[127] *Protocol Buffers*. URL: https://developers.google.com/protocol-buffers/ (visited on 08/22/2019).

[128] Joel R. Reidenberg, Jaspreet Bhatia, Travis D. Breaux, and Thomas B. Norton. “Ambiguity in Privacy Policies and the Impact of Regulation.” In: *The Journal of Legal Studies* 45.S2 (June 2016), S163–S190. ISSN: 0047-2530, 1537-5366. DOI: 10/gdcdzm. URL: https://www.journals.uchicago.edu/doi/10.1086/688669 (visited on 10/10/2018).

[129] Jonathan Rosenberg <jdrosen@cisco.com>. *The Extensible Markup Language (XML) Configuration Access Protocol (XCAP)*. URL: https://tools.ietf.org/html/rfc4825 (visited on 08/22/2019).

[130] *ScanCallback*. URL: https://developer.android.com/reference/android/bluetooth/le/ScanCallback (visited on 07/31/2019).

[131] Bruce Schneier. *Blockchain and Trust - Schneier on Security*. URL: https://www.schneier.com/blog/archives/2019/02/blockchain_and_.html (visited on 10/21/2019).

[132] Bruce Schneier and John Kelsey. “Cryptographic Support for Secure Logs on Untrusted Machines.” In: (), p. 11.

[133] Meera Sivanathan. *What Is Legal Design? | Q&A with Meera Sivanathan (Legal Designer) | The Legal Forecast*. URL: https://thelegalforecast.com/what-is-legal-design-qa-with-meera-sivanathan-legal-designer/ (visited on 03/26/2019).

[134] *Smart Places*. URL: https://smart-places.org/.

[135] Daniel J Solove. “Privacy Self-Management and the Consent Dilemma.” In: (), p. 25.

[136] *Sous-Surveillance.Net*. URL: https://www.sous-surveillance.net/ (visited on 12/13/2019).

[137] Special Privacy. *Deliverable D1.3 Policy, Transparency and Compliance Guidelines V1*. Jan. 9, 2017.

[138] S. Spiekermann and L.F. Cranor. “Engineering Privacy.” In: *IEEE Transactions on Software Engineering* 35.1 (Jan. 2009), pp. 67–82. ISSN: 0098-5589. DOI: 10.1109/TSE.2008.88. URL: http://ieeexplore.ieee.org/document/4657365/ (visited on 02/20/2017).

[139] Cass R. Sunstein. “Choosing Not to Choose.” In: *SSRN Electronic Journal* (2014). ISSN: 1556-5068. DOI: 10/gftmr3. URL: http://www.ssrn.com/abstract=2377364 (visited on 01/21/2019).

[140] Kiana Tehrani and Michael Andrew. *Wearable Technology and Wearable Devices: Everything You Need to Know | Wearable Devices*. Mar. 2014. URL: http://www.wearabledevices.com/what-is-a-wearable-device/ (visited on 10/02/2019).

[141] Michael Toth, Nataliia Bielova, Cristiana Santos, Vincent Roca, and Célestin Matte. *Contribution to the Public Consultation on the CNIL's Draft Recommendation on "Cookies and Other Trackers"*. Feb. 25, 2020. URL: https://hal.inria.fr/hal-02490531 (visited on 03/17/2020).

[142] *Types of Beacons*. In: *Wikipedia*. URL: https://en.wikipedia.org/w/index.php?title=Types_of_beacons&oldid=896280489 (visited on 07/29/2019).

[143] *URME Surveillance*. URL: http://leoselvaggio.com/urmesurveillance (visited on 11/06/2019).

[144] “Unified Modeling Language, v2.5.1.” In: *Unified Modeling Language* (), p. 796.

[145] *Use Wi-Fi Direct (P2P) for Service Discovery | Android Developers*. URL: https://developer.android.com/training/connect-devices-wirelessly/nsd-wifi-direct (visited on 12/03/2019).

[146] Christine Utz, Martin Degeling, Sascha Fahl, Florian Schaub, and Thorsten Holz. “(Un)Informed Consent: Studying GDPR Consent Notices in the Field.” In: (), p. 18.

[147] Jordan Valinsky. *'Alexa, Order Me Room Service.' Amazon's Voice Assistant Checks in to Marriott Hotels*. URL: https://money.cnn.com/2018/06/19/technology/amazon-alexa-marriott/index.html (visited on 03/19/2019).

[148] Jasper van de Ven and Frank Dylla. "Qualitative Privacy Description Language." In: *Annual Privacy Forum*. Springer, 2016, pp. 171–189. URL: http://link.springer.com/chapter/10.1007/978-3-319-44760-5_11 (visited on 01/11/2017).

[149] Henk C.A. van Tilborg. *Encyclopedia Of Cryptography And Security*. In: Springer, 2005.

[150] Liesbet van Zoonen. "Privacy Concerns in Smart Cities." In: *Government Information Quarterly* 33.3 (July 2016), pp. 472–480. ISSN: 0740624X. DOI: 10.1016/j.giq.2016.06.004. URL: http://linkinghub.elsevier.com/retrieve/pii/S0740624X16300818 (visited on 08/22/2017).

[151] Ludovic Vievard. *La Ville Intelligente : Modèles et Finalités*. Oct. 2014.

[152] Ville de Paris. *Paris Data*. URL: https://opendata.paris.fr/pages/home/ (visited on 10/04/2019).

[153] WP29. "Opinion 8/2014 on the Recent Developments on the Internet of Things." In: 2014.

[154] WP29. *Guidelines on Consent under Regulation 2016/679*. Nov. 28, 2017.

[155] WP29. *Guidelines on Transparency under Regulation 2016/679*. Dec. 15, 2017.

[156] Mark Weiser. "The Computer for the 21st Century." In: *Scientific American* 265.3 (Sept. 1991), pp. 94–104. ISSN: 0036-8733. DOI: 10.1038/scientificamerican0991-94. URL: http://www.nature.com/doifinder/10.1038/scientificamerican0991-94 (visited on 02/13/2017).

[157] *What Is The Quantified Self?* URL: https://quantifiedself.com/blog/what-is-the-quantified-self/ (visited on 10/02/2019).

[158] "Wi-Fi P2P Technical Specification." In: (2016), p. 201.

[159] *Windows and Screens | Apple Developer Documentation*. URL: https://developer.apple.com/documentation/uikit/windows_and_screens (visited on 12/03/2019).

[160] Andrea Zanella, Nicola Bui, Angelo Castellani, Lorenzo Vangelista, and Michele Zorzi. "Internet of Things for Smart Cities." In: *IEEE Internet of Things Journal* 1.1 (Feb. 2014), pp. 22–32. ISSN: 2327-4662. DOI: 10.1109/JIOT.2014.2306328. URL: http://ieeexplore.ieee.org/document/6740844/ (visited on 11/14/2016).

[161] Sherali Zeadally, Ray Hunt, Yuh-Shyan Chen, Angela Irwin, and Aamir Hassan. "Vehicular Ad Hoc Networks (VANETS): Status, Results, and Challenges." In: *Telecommunication Systems* 50.4 (Aug. 2012), pp. 217–241. ISSN: 1018-4864, 1572-9451. DOI: 10/cn7hxj. URL: http://link.springer.com/10.1007/s11235-010-9400-5 (visited on 10/04/2019).

[162] Jan Henrik Ziegeldorf, Oscar Garcia Morchon, and Klaus Wehrle. "Privacy in the Internet of Things: Threats and Challenges." In: *Security and Communication Networks* 7.12 (Dec. 2014), pp. 2728–2742. ISSN: 19390114. DOI: 10.1002/sec.795. URL: http://doi.wiley.com/10.1002/sec.795 (visited on 11/14/2016).

[163] calimaq. *Richard Stallman, le RGPD et les deux faces du consentement*. URL: https://scinfolex.com/2018/04/05/richard-stallman-le-rgpd-et-les-deux-faces-du-consentement/ (visited on 04/05/2018).

[164] wsantos. *Which API Types and Architectural Styles Are Most Used?* Sun, 2017-11-26 15:10. URL: https://www.programmableweb.com/news/which-api-types-and-architectural-styles-are-most-used/research/2017/11/26 (visited on 08/22/2019).

[165] • *Mobile OS Market Share 2018 | Statista*. URL: https://www.statista.com/statistics/266136/global-market-share-held-by-smartphone-operating-systems/ (visited on 08/22/2019).