Original paper was presented at 2016 CPDP conference.
Original paper appeared in:
Data Protection and Privacy: (In)visibilities and Infrastructures

https://link.springer.com/chapter/10.1007%2F978-3-319-50796-5_7

A Privacy Engineering Framework for the Internet of Things

Antonio Kung[1], Frank Kargl[2], Santiago Suppan[3], Jorge Cuellar[4], Henrich C. Pöhls[5], Adam Kapovits[6], Nicolas Notario[7], Yod Samuel Martin[8]

[1] Trialog, Paris, France
antonio.kung@trialog.com
[2] Ulm University, Ulm, Germany
frank.kargl@uni-ulm.de
[3,4] Siemens, Munich, Germany
[3] santiago.suppan.ext@siemens.com
[4] jorge.cuellar@siemens.com
[5] University of Passau, Passau, Germany
hp@sec.uni-passau.de
[6] Eurescom, Heidelberg, Germany
kapovits@eurescom.eu
[7] Atos, Madrid, Spain
nicolas.notario@atos.net
[8] Universidad Politécnica de Madrid, Madrid, Spain
samuelm@dit.upm.es

Abstract. This paper describes a privacy engineering framework for the Internet of Things (IoT). It shows how existing work and research on IoT privacy and on privacy engineering can be integrated into a set of foundational concepts that will help practice privacy engineering in the IoT. These concepts include privacy engineering objectives, privacy protection properties, privacy engineering principles, elicitation of requirements for privacy and design of associated features. The resulting framework makes the key difference between privacy engineering for IoT systems targeting data controllers, data processors and associated integrators, and privacy engineering for IoT subsystems, targeting suppliers.

Keywords: Privacy-by-design, Internet of things, IoT system, IoT subsystem, Integrator, Supplier.

1 Introduction

1.1 The Internet of Things

The Internet of Things (IoT) refers to smart devices, sensors, and actuators that are embedded in the physical world, connected to each other and to further computing resources, allowing applications and intelligent services to understand, track, and

control almost anything in the physical world through standard communication networks.

"Things" in the IoT can be machines controlling the production in a factory, electrocardiography sensors in clothing, temperature sensors or light bulbs in homes and buildings, moisture sensors in the garden, and persons and animals providing (via IoT devices) personal data to location-based services or to comfort control systems that adapt the environment to their preferences or context. The data can be linked together using semantic methods enhancing the information interoperability in heterogeneous systems and thus enabling automated services composition. The data can be analyzed with statistical methods, business intelligence, predictive analytics, and machine learning. As we interact with our world and explore the collected data, the benefits will increase. The resulting "reality mining" applications offer increasingly extensive information about our lives, both individually and collectively and transform our understanding of ourselves, our organizations, and our society. MIT's Technology Review has identified reality mining as one of the "10 Emerging Technologies That Will Change the World", see [1].

1.2 Privacy, a Transversal Problem

The IoT vision entails the tacit assumption that data can first be collected and then later the analysis shows for which concrete purposes it can be used. Large amounts of seemingly non-personal data (temperature, motion, ambient noise, etc.) may be linked together and may later be used to identify individuals and their activities. Technologies such as reality mining will be able to reveal "hidden" information and relationships that were never suspected in the first place, when the data was collected. This contradicts the main privacy principles: "Specification of purpose is an essential first step in applying data protection laws and designing data protection safeguards for any processing operation and a pre-requisite for applying other data quality requirements" (see [2]).

IoT is also a major trend driving growth in Big Data. Already today, data is an asset with an enormous economic value. Over the last years, the economic value of data followed by a respective industry has grown exponentially and the impact on other sectors (healthcare, transportation, ecommerce, etc.) has been equally increasing. A common understanding is that data is an asset that belongs to the parties generating the data, who are free to decide what they want to do with that asset to achieve their business goals. This again contradicts fundamental privacy principles: it is the "data subject", i.e. the person about whom the data is gathered and not simply the "data generator" that should determine how the data can be used and by whom.

Some authors believe that IoT and Big Data are fundamentally in contradiction with privacy (for a snapshot of the controversy, see [3]). Indeed, to reconcile them is clearly difficult, and there are many problems that are hard to solve: even innocently looking data, such as noise levels or room temperatures in the different rooms of a building can reveal the activities of a person and thus become privacy relevant. At the same time data gathered in public spaces relating to many different data subjects creates the challenge how to inform the data subjects about the purpose and obtain their consent. The transparency of the data provenance and integrity is difficult to

guarantee in a scenario where subjects are continuously being monitored and tracked by a large number of devices. Furthermore, the use of (resource) constrained devices in IoT makes it hard to implement, configure and use complex security and privacy mechanisms. Since each IoT instantiation will collect and use other data it may appear impossible to create privacy building blocks for IoT in general. Finally, even if all the rather technical problems can be solved, it needs to be considered that the business model of some companies is based on extracting value from personal data and tolerate data protection related risks or even financial penalties to continue their endeavours.

1.3 IoT Ecosystems

An IoT supply chain can involve many organisations as shown in **Fig. 1**. It will involve data controllers (defined in [4] as "*the natural or legal person, public authority, agency or any other body which alone or jointly with others determines the purposes and means of the processing of personal data*"). It might further involve data processors (defined in [4] as "*a natural or legal person, public authority, agency or any other body which processes personal data on behalf of the controller*"). It will involve integrators who are in charge of providing turnkey systems by bringing together component subsystems into a whole and ensure that those subsystems function together. The supply chain will finally include suppliers. Suppliers provide component subsystems which are then integrated. They bring the needed flexibility in an effective IoT ecosystem.

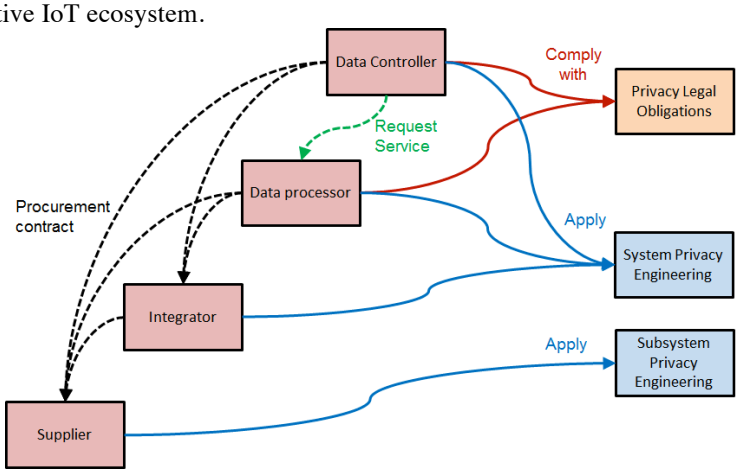

Fig. 1. The IoT Supply Chain

Data controllers and data processors must comply with existing privacy-related legal obligations, while integrators and suppliers do not, at least explicitly. This raises the issue that neither integrators nor suppliers take properly into account the privacy-related legal obligations of the data controllers and data processors they are working with. One can argue that data controllers and data processors will integrate privacy-related legal obligations in supply and procurement contracts. We therefore argue that both integrators and suppliers must include privacy engineering in their practice.

There is another point that we would like to emphasize: privacy engineering for suppliers of subsystems has to be approached differently as the engineering of such subsystems generally takes place independently from the design of the system in which they will finally be integrated. Suppliers therefore cannot be aware of the data collection intentions of the stakeholders that buy their systems. Here are a few examples: a biometric subsystem that is purchased by a bank office; a video camera subsystem that is used by a security operator; a smart phone operating system and middleware that is used by an application developer.

Therefore while data controllers, data processors and integrator practice privacy engineering should know the initial purpose for data collection, developers of subsystems on the other hand may have no precise clue on the purpose for which their subsystems will be integrated. One could argue that the issue will be fixed anyway since suppliers that do not provide suitable privacy management features will end up having less business. We believe this does not work well for the following reasons: firstly, the relationship between supply chain stakeholders can be unbalanced. The data controller could be much less influential than the supplier (consider the developer of a smart phone application in a small company and its relation with the smart phone operating system managed by a major company). A take it or leave it position would be created by powerful suppliers, yielding a situation where proper privacy engineering is only applied in minor parts of the overall system; further, data controller could take advantage of this situation to escape ethical obligations while still meeting legal obligations. Secondly it is not obvious that suppliers today are aware that they should be concerned. Current policy makers and regulations do not refer to them. Directive 95/46/EC [4] defines for instance the term third party but only uses this term to refer to stakeholders to whom data can be disclosed. The General Data Protection Regulation (GDPR) [5] published in May 2016 to replace the directive defines third parties as stakeholders that are authorised to process personal data; In addition it is interesting to note that none of the referenced work presented in sections 3 and 4 below covers suppliers. Recently however, the European Commission issued a mandate for standardization focusing on privacy management of security products and related services [6]. While not explicitly mentioning suppliers, the mandate mentions “manufacturers and service providers” for security technologies. We believe that many such manufacturers or service providers will not play the role of data controllers, data processors or integrators. They will just be suppliers.

Fig. 2 shows a typical IoT architecture diagram. It highlights the following points: there are three layers, the application layer, the capability layer and the network layer[1]; these layers are present in every IoT nodes e.g. a sensor, a smart phone, or a cloud system. An IoT system consists of a number of IoT nodes. An IoT subsystem can have different forms. It could be a node, e.g. a sensor, it could be a subsystem within a node e.g. the operating system of a smart phone, it could be a subsystem within several nodes e.g. a service-client capability involving two nodes, it could be a cloud system. What is striking in the provided examples is that those IoT subsystems

[1] The structure into three layers is inspired from architecture discussions held within AIOTI working group 4 (http://www.aioti.eu/)

are typical mainstream ICT subsystems, which are potentially developed by major stakeholders independently of any privacy constraint.

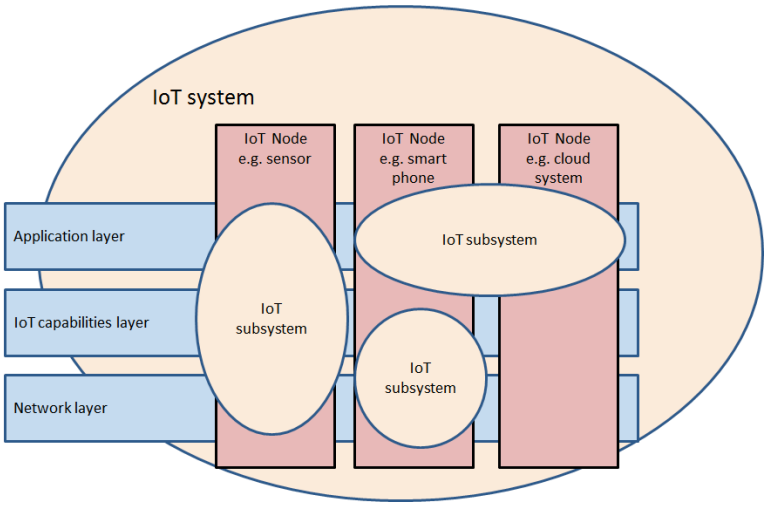

Fig. 2. IoT Architecture, Nodes and Subsystems.

Two recommendations are identified for privacy in the context of the Internet of Things. The first is to make a clear difference between IoT system privacy engineering and IoT subsystem privacy engineering. The second is to build a wealth of IoT subsystem privacy engineering practice.

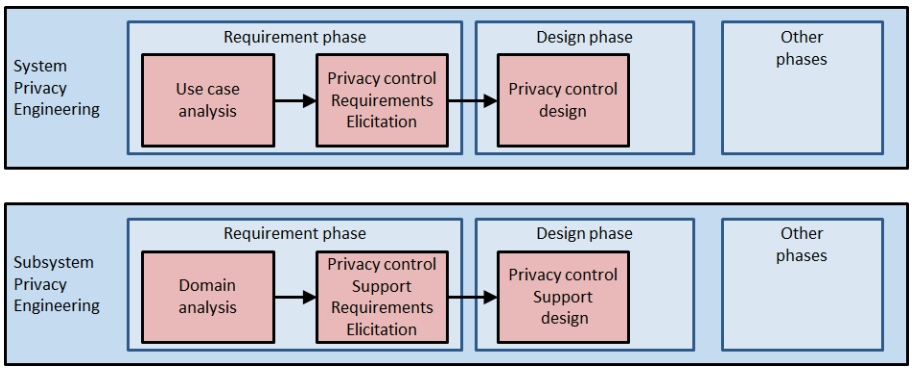

Fig. 3. System Privacy Engineering versus Subsystem Privacy Engineering

Fig. 3 illustrates the difference between system privacy engineering (e.g. the design of an e-health monitoring system integrating an information system, a smart phone and sensors) and subsystem privacy engineering (e.g. the design of a body sensor). In the engineering of a system, the requirement phase focuses on analysing a precise use case where data flows can be identified so that concrete operational

requirements for privacy controls can be elicited. These requirements are the input to the next phase where the design of privacy controls is carried out. In the engineering of a subsystem, the requirement phase cannot focus on a precise use case but rather on a domain specific set of use cases. This more global analysis of a domain must then lead to the elicitation of operational requirements for privacy control support. Examples of requirements of privacy control support are provided in **Table 1**. The requirements are then input to the next phase where the design of privacy control support is carried out.

But the needed genericity for IoT subsystems privacy engineering could deter suppliers from providing privacy control support if no clear business incentives are provided. It is therefore important to create a wealth of successful privacy engineering practices. Those practices should actually involve concertation with the demand side, i.e., IoT system designers.

1.4 The need for Privacy Engineering Guidelines

Privacy-by-design is a term that has been a buzzword since the very moment when it was coined by Ann Cavoukian [7]. With the advent of the General Data Protection Regulation (GDPR) which explicitly refers to *data protection by design* and *data protection by default* [5], and with a projection that the Internet of Things will consist of 50 billion connected devices by 2020 [8], this very term will have to become a reality. In other words privacy engineering in the ICT ecosystem where such devices are produced will have to be a reality.

This paper defines a privacy engineering framework for the IoT that will help going towards that direction. This framework is the result of a cooperation of the two European FP7 projects RERUM[2] and PRIPARE[3].

RERUM's objective is to develop a framework enabling dependable, reliable and innovative Smart City applications[4]. The framework follows the concept of "security and privacy by design", addressing the most critical factors for the success of Smart City applications. RERUM therefore has developed an architectural framework with the addition of hardware products that allow one to adjust the level of privacy, security and trust from the earliest possible stage of data acquisition up to a point next to data processing by service providers, covering most of the data lifecycle in IoT.

PRIPARE's objective is to define a privacy-by-design methodology and support its uptake by the ICT research community in order to prepare for industry practice, and to produce a set of educational material. PRIPARE has therefore developed a methodology which integrates the various contributions made in the area of privacy engineering. The results of PRIPARE have led to the creation of a new work item in ISO JTC 1/SC 27/WG 5[5] on privacy engineering.

The rest of the paper is structured into three main sections. Section 2 explains the impact at the architecture level of security and privacy in an Internet of Things

[2] https://ict-rerum.eu/
[3] http://prepareproject.eu/
[4] For an overview, the reader is referred to [9].
[5] http://www.din.de/en/meta/jtc1sc27/structure

system. Section 3 provides a rationale for the elaboration of a privacy engineering framework, as well as an analysis of recent contributions to privacy engineering. Section 4 finally describes the proposed privacy engineering framework for IoT. The framework includes four categories of information, concept, stakeholders, process and organisation.

2 Privacy in the Internet of Things

Data is an asset of immense business value that involves the interest of companies in keeping them secure as well as the trust of customers and users. Security and privacy breaches endanger both, the economic value of the companies' assets and the trust that customers have placed in them. In the previous section we have pointed out how data can be collected and later be processed in such a way that it can be referred to a subject. In the following sections, we explain how privacy protection can be embedded into a system's architecture and how it can help to prevent privacy problems incurred after data collection. This section in particular presents how an IoT architecture could look like based on proposed reference models and how privacy engineering can be applied to the architecture using privacy controls.

2.1 Internet of Things Architecture

In order to successfully apply privacy engineering to IoT, the underlying architecture has to be designed such that it can support privacy enhancements. There are several proposals for an architecture reference model for IoT, the most prominent are:

- The IoT-A: Internet of Things - Architectural Reference Model (ARM), see [11], is the most influential proposal. This reference model was developed as part of the European Internet of Things Research Cluster, it analyses use cases and business needs, eliciting common requirements and proposing a reference that unifies building blocks that fulfil those requirements. The proposal has served as the ground for numerous IoT projects, including RERUM.
- The AIOTI: Alliance for Internet of Things Innovation High Level Architecture, see [12], is working group under the IERC as well that takes into account the work of several projects such as IoT-A, RERUM, BUTLER and others. The proposal focuses on Large Scale Pilot deployments and points out lessons learned from respective partner projects.
- The ISO/IEC Working Draft 30141, see [13], specifies a layered structure identifying IoT standardization areas and key components of IoT Systems. The draft includes definitions of conceptual models by generic IoT domains, an IoT Reference Architecture (IoT RA), IoT specific requirements and terminology. The draft either adopts from existing proposals, modifies existing ones, or develops new definitions if required. This reference architecture focuses on systems developers, but conceptually stays close to e.g. the IoT-A model. In this respect it is important to note that ISO JTC 1 SWG 5 pays special attention to following requirements: “...IoT systems cannot be used for malicious purposes by unauthorized entities” and that “...personal and business information is kept

confidential"[6]. In order to minimize future effort RERUM provided comments during the standardization process to the ISO/IEC 30141 working group[7].

The RERUM project has used the IoT-A model as it can defines the terminology, the basic building blocks and the entity relations of IoT systems that can be found or mapped in most of the other proposals and IoT projects. The reader is referred to the IoT-A reference model in [11] and the RERUM architecture that expands the IoT-A ARM with privacy relevant concepts in [14].

2.2 RERUM Architecture and Privacy Controls for the IoT

The RERUM project has focused on the identification and development of a set of privacy control features for the IoT, taking into account work from past projects or ISO 29151, see [10] for the methodology. These controls are described in this section.

There are several generic IoT architectures, for instance the IoT-A ARM as described in the previous subsection. The first questions to ask are: Which components of the given architecture have been defined with privacy in mind? What further privacy-related extensions are necessary in the architecture?

PMRM [16] recommends to base architectural components and modifications on privacy requirements and their technical solutions. That implies that privacy affects the design of the architecture. This can be achieved by codifying them into **user-defined privacy policies**. These should be linked in the architecture in such a way that they are available when needed.

In IoT systems, smart devices monitor or control physical entities (which could be users or particular aspects of users). In IoT-A physical entities are represented in the virtual space as software artefacts called "virtual entities". A natural way of binding the privacy policies to the object is to link them to the entity that represents the object in the virtual space. In this way the policy can be consequently consulted each time that the virtual object is being accessed. In order to follow a strict opt-in approach and guarantee data minimization a number of privacy related extensions are still necessary. All together these extensions must enable the user to describe his policies, to have fine grained control on the data collection, and to understand which data is being collected for what purpose. The components that RERUM envisioned are as follows.

The **Consent Manager** supports also the goal of transparency and intervenability; it is the component of the architecture that allows the data subject to review which applications are requesting his personal information and the purpose of the request, and may give or deny his consent to them, or withdraw a previously granted one.

The **Deactivator/Activator** of Data Collection allows the data subject to intervene and de-activate the collection of data from any devices when he/she wishes to protect his/her privacy and to re-activate the collection later on, when he/she decides to do so.

The **Privacy Dashboard** empowers the user to manage its policies for private information. It derives from the goal of availability (the user shall be able to access all

[6] See the report published in http://www.iso.org/iso/internet_of_things_report-jtc1.pdf
[7] See https://ict-rerum.eu/765-2/.

data about him/her at any time), intervenability and transparency. Generally, it cannot be expected and assumed that users (data subjects) that use an IoT application have the technical knowledge to be able to express their privacy policies in a conventional policy language. The role of the Privacy Dashboard is to support the user with a graphical user interface, which visualizes a device's behaviour and allows setting a specific behaviour according to users' preferences. The user preferences are then translated automatically to detailed machine-readable policies. To sum it up, a privacy dashboard answers the common users' question "What does the system know (track) about me?" and provides a graphical interface that the user can understand and take appropriate action, if necessary. Additionally, the Privacy Dashboard allows tracking how many Physical Entities are connected an IoT system and which kind of data they are exposing. Without the visibility of the actual data collected data subjects may not fully understand the abstract description of what types of data are collected; simultaneously, data subjects may be overwhelmed by access to raw data without knowing what that data means and what are the implications.

A **lightweight and efficient pseudonym system** is able to hide the identities of the users from the applications that do not necessarily require them, but also from any attacker or intruder that is able to exploit vulnerabilities of the system or weaknesses in humans to get access to the central databases, preventing any of those to track down individuals through their identities and thus serves the goal of unlinkability.

Important secure pseudonym exchanging concepts can be categorized in spatial concepts, time-related concepts and user-oriented concepts (see [17]). Spatial concepts are best based on mix-zones, where pseudonyms are exchanged when system participants meet physically. Time-related mechanisms propose to change pseudonyms after a certain time, where a secure pseudonym exchange is only possible when the changing participant is not participating in the system any more. One possible solution is a so called silent period. This means that a system participant stops his/her participation for a short time until his/her pseudonym is changed successfully. User-oriented concepts allow the user to decide when he/she wants to change his/her current identity. The decision can hereby be completely subjective, allowing defining own policies and thresholds for the pseudonym change.

Geo-Location PETs enable the system to send the minimal amount of information to, say, traffic monitoring or other location-based applications. In general, two methods exist: pseudonym exchange technologies for vehicular ad-hoc networks (see [18]), and data based technologies for floating car observation such as the one suggested in RERUM (see [19]). Through the combination of both, data obfuscation, pseudonym systems and methods for replacing regularly the pseudonyms, the association of users to location can be entirely obfuscated, depending on circumstances. This is very important, as the tracking of location information discloses a large amount of information about the habits, activities, preferences of people.

Sticky Policies are privacy policies that travel with the data as they are transmitted all the way in the system, supporting user's intervenability by promoting awareness of allowed actions and consent obligations for them. A sticky policy mechanism includes a secure way of binding the data to the privacy policy, for example, as seen in [20].

Malleable and Group Signatures (see [21], [22]) allow balancing the requirements for integrity and origin authentication for the gathered data with the need to remove information to protect privacy. Namely, Malleable Signatures allow the data origin to authorize selected subsequent changes to signed data, e.g. to blacken-out a privacy-violating data field. Group Signatures allow hiding the specific creator of data for enhancing the data privacy, e.g. instead of the exact origin only a group of potential creators can be identified as the source. Both mechanisms decrease the data quality only gradually, but enhance intervenability, confidentiality and integrity.

Reconfiguration facility of security and privacy mechanisms allows updating or exchanging the mechanisms to enforce security and privacy. In order to be able to adjust to a changing landscape (to circumvent new vulnerabilities found in security and privacy mechanisms as well as in applications) devices need to be re-programmable. This can be achieved by making the device firmware updatable by secure remote over-the air programming (OAP), or related methods.

A secure **credential bootstrapping mechanism** is necessary to place the necessary cryptographic keys into the devices, in order to enable them to communicate securely with the rest of the system. Of course, lightweight and efficient privacy preserving authentication and authorization protocols are necessary. They must support constrained nodes, in terms of computing power, energy consumption and bandwidth.

The experiences of RERUM show that the generic and specific PET-extensions to IoT-A are valuable in four quite different application scenarios (Smart Transportation, Environmental Monitoring, Domestic and Building Energy Efficiency and Indoor Comfort Quality Management). However, the IoT Privacy Architecture must be flexible to support different scenarios as requirements will change over time.

The table below summarises the resulting IoT privacy control support features for privacy.

Table 1. IoT Support Features for Privacy

Privacy control support	**Protection objective**
User-defined privacy policies	Intervenability
Consent manager	Transparency, Intervenability
Deactivator/Activator	Unlinkability, Intervenability
Privacy dashboard	Transparency, Intervenability, Availability
Lightweight pseudonym system	Unlinkability
Geo-Location PET	Unlinkability
Sticky policies	Intervenability
Malleable and group signatures	Integrity, Authentication, Data minimization
Reconfiguration facility of security and privacy mechanisms	Intervenability
Credential bootstrapping mechanism	Integrity, Authentication

2.3 Comparison with other solutions

In this subsection, an overview of proposed privacy controls in RERUM and other privacy related projects will be given. This overview shall help the reader to understand the different focus of each of the IERC privacy related projects[8].

Table 2. Comparison of Privacy Related IoT Projects

	IoT Specific	Confi-dentia-lity	Integrity	Interve-nability	Unlink-ability	Data Minimi-zation	Authen-tication
RERUM	X	X	X	X	X	X	X
IoT-A	X	X			X		
PEARS feasibility		X	X				X
Prime		X	X	X	X	X	X
Prime Life		X	X	X	X	X	X

RERUM has integrated and developed privacy enhanced technologies to address all relevant privacy controls in IoT systems.

IoT-A has covered conceptual aspects of confidentiality protection (by introducing the idea of pseudonym into the IoT architecture) but has not proposed details on engineering privacy-by-design, data minimization technologies and other privacy controls.

PEARS feasibility has introduced privacy enhanced identification tags to the area of RFID, allowing authentication, confidentiality and integrity protection. Neither privacy-by-design, nor data minimization nor IoT use cases have been specifically addressed.

Prime and Prime Life have addressed many controls to allow privacy-by-design and proposed several privacy enhanced technologies such as the prominent Idemix anonymous credential system, see [23]. Prime and Prime life have not focused their efforts on the constraints and lossy environment of IoT systems, but rather on the user of IT systems. The Prime and RERUM projects can therefore be complimentary: RERUM enhances privacy on IoT devices and the architectural part of the system, while Prime and Prime Life support privacy controls of the user.

2.4 Recommendations for Privacy Engineering in IoT

The privacy controls described in the previous section do not replace the privacy engineering effort. In each concrete IoT implementation or deployment the initial steps are to define the functional and operational requirements (purpose), the personal data that should be collected for the purpose and start deciding how to minimize the collection and analysis and which PETs could be used.

[8] A comprehensive list of the IERC projects can be found in http://www.internet-of-things-research.eu/partners.htm

It is important to use an IoT framework that easily allows for or already incorporates a privacy-by-design approach, and provides several PETs, in particular to create, hold, and use privacy policies. The chosen PET components and the generic or default privacy policies should be instantiated to the specific application and domain. The privacy policies need to be installed as close to the data collection points, i.e. the devices, as possible, or at the first place where a data subject is identified.

It is recommended to use hardware components that are able to apply the necessary PETs as early as possible, i.e. on data collection points or their gateways. Simply put there is a need for "intelligence" in the data collecting subsystems such that privacy policies can be retrieved, understood, policies can be stuck to data sets and data minimization and aggregation can be applied already near the data collection points.

The hardware components, especially for the smart devices or the "things", should be able to run security and privacy protocols and the software should be securely managed remotely. This allows software to be adapted to changes over time, e.g. to fix flaws, or to add new more advanced mechanisms. Also it allows to re-purpose hardware for other applications in the future, reducing costs for physical re-deployments.

It should be noted, that the application of security and privacy mechanisms does not correlate with increased hardware costs. Efficient PETs should be identified and implemented during privacy engineering using state of the art technologies, in particular for constrained environment, as elliptic curve cryptography, see [24], delegation of heavy computation such as the generation and verification of policies, see [25], etc.. Advances in security and privacy research yield mechanisms with continuous efficiency improvements, which can be updated with proper hardware (again, flexibility is key).

Furthermore, the deployed hardware itself should be updatable. This can be achieved by making it interoperable, e.g. through offering common APIs, and adhering to agreed standards. This allows hardware flaws to be fixed, thus reducing the risk of a hardware vendor lock in and the security risks specific to "monoculture" deployments.

3 Understanding Privacy Engineering

3.1 Privacy Engineering in Organisations

Fig. 4 provides a model of the essential elements of privacy support in organisations. The left part shows important objectives for privacy-by-design: integrating the concept of privacy in an organisation; integrating the concept of privacy in the engineering of systems.

The centre part shows two important viewpoints: the management viewpoint, which focuses on elements (processes, practices, concepts) that are important to managers in their activities; the engineering viewpoint, which focuses on elements (engineering requirements, design, implantation, verification, maintenance) that are important to engineers in their activities.

The right part shows important process concerns to both managers and engineers: risk assessment, system development, and system compliance: risk assessment focuses on quantifying privacy risks in systems dealing with personal data and mitigating them by reducing their likelihood or their consequences; system development focuses on specifying and implementing technical solutions for privacy control in systems dealing with personal data. System development can involve decisions to integrate sub-systems supplied by third parties; system compliance focuses on ensuring that an organisation is doing what is expected and that developed systems developed have the expected behavior. System compliance involves challenging processes such as privacy protection assurance, evaluation and verification. System compliance allows external stakeholders (e.g. consumers, policy makers, procurers) to assess whether they can trust the organisation and/or the systems.

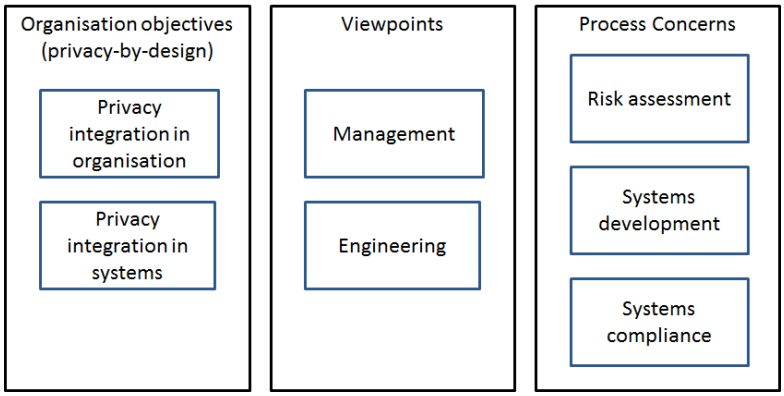

Fig. 4. Organisation Privacy Support Model.

The difference between a management viewpoint and an engineering viewpoint is the following: management focuses on what system is developed and on checking that it is developed properly while engineering focuses on how a system is developed and on testing it properly. We observe that managers and engineers do not work in isolation, because they build the system together. They must interact in such a way that sufficiently precise information is exchanged. Interactions can also be iterative, for instance when using the Deming cycle[9]. Examples of interactions that will take place are discussions on legal and ethical aspects which are treated at management level. Managers are concerned that the resulting undertakings comply with regulations and also meet ethical principles. They must interact with engineers so that these latter understand the requirements they must meet.

The model can be used to illustrate specific process concerns and the current state of practice. **Fig. 5** shows examples of management oriented process concerns.

Privacy impact assessment (PIA) is a risk assessment process concern. For instance, ISO 29134 [26] is a privacy impact assessment impact standard under

[9] also called Plan-Do-Check-Act cycle

development. The CNIL PIA [27] and the data protection impact assessment template specified by the smart grid task force [28] are other examples.

Privacy analysis is a system development process concern. The OASIS privacy management reference model and methodology (PMRM) [16] is a standard focusing on privacy requirement analysis enabling the identification of a set of privacy management functions. Using a code of practice is another system development process concern. ISO 29151 [29] is a code of practice for personally identifiable information standard under development that will provide recommendations on the practice and use of privacy controls.

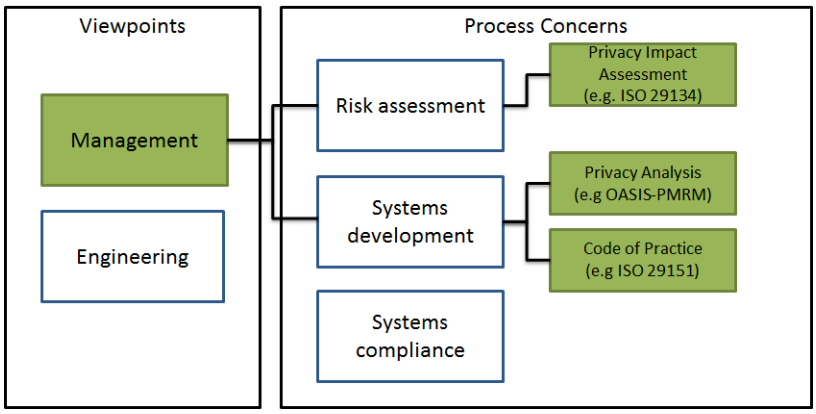

Fig. 5. Management Viewpoint of Process Concerns.

Fig. 6 shows examples of how privacy could be supported from an engineering viewpoint.

Threat analysis is a risk assessment process concern. For instance, LINDDUN [30] is a methodology that can be used by engineers to identify threats and design mitigation solutions.

Architecture is a system development process concern. For instance PEARS [31] explains how to specify an architecture which improves privacy using architecture concepts such as quality attributes and architecture tactics. It is based on Carnegie-Mellon work on software architecture [32]. It provides a list of architecture strategies (*minimisation*, *enforcement*, *transparency*, *modifiability*).

Design strategy is another system development concern. Hoepman [33] describes how a system can be designed following a number or strategies and how to identify and describe reusable solutions called privacy patterns.

The integration of privacy engineering into development process methodologies is a system development concern. For instance an organization could use the agile development methodology [34], a design methodology which focuses on flexible and evolutionary development, allowing engineers to develop prototypes that can iteratively evolve into improved versions[10].

[10] Note that the integration of privacy engineering into Agile methodologies is a challenge because of the lack of a clear design phase.

Software documentation is another system development process concern. OASIS Privacy by Design Documentation for Software Engineers (PbD-SE) [35] is an example of standard under development.

Examples in **Fig. 5** and **Fig. 6** are not meant to be definitive categorization of viewpoints (management vs engineering). Standards such as ISO 29134 [26], OASIS-PMRM [16], ISO 29151 [29] are also useful from an engineering viewpoint. In particular OASIS-PRMM explains to engineers how to apply an iterative process to identify privacy management functions and associated requirements.

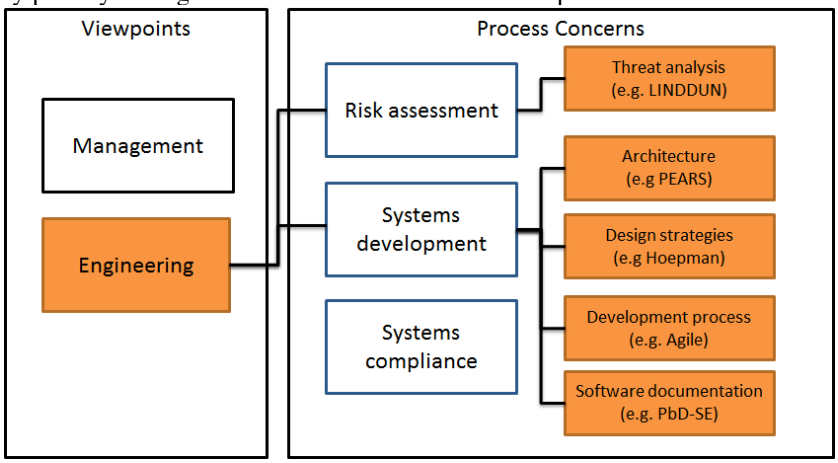

Fig. 6. Engineering Viewpoint of Process Concerns.

3.2 The Need for a Privacy Engineering Framework

The previous section showed the many aspects of privacy engineering and the complexity of integrating them in organisations. As we have shown earlier, there is an ample range of privacy controls that could be applied in IoT systems but a structured engineering approach is required for appropriate selection and integration. To this end, an integrated vision is necessary. We suggest that this can be achieved by defining a privacy engineering framework.

It may be useful to define the meaning of the term *framework*. The Cambridge online dictionary defines a framework as a "*system of rules, ideas, or beliefs that is used to plan or decide something*", e.g. a legal framework for resolving disputes. The Oxford online dictionary defines framework as "*a basic structure underlying a system, concept, or text*", e.g. the theoretical framework of political sociology. Finally, the free dictionary[11] defines a framework as "*a set of assumptions, concepts, values, and practices that constitutes a way of viewing reality*". In the rest of this section we will use this latter definition.

[11] www.thefreedictionary.com

Interestingly, there is already a standard, ISO 29100 [36] which defines a privacy framework. As stated in the standard "*it specifies a common privacy terminology; it defines the actors and their roles in processing personally identifiable information (PII); it describes privacy safeguarding considerations; and it provides references to known privacy principles for information technology*". ISO 29100 provides therefore a set of assumptions, concepts, values and practices for privacy in organisations dealing with personal data. A detailed look at ISO 29100 shows that it contains two parts: a set of concepts (*actors and roles, interactions, recognizing Personally Identifiable Information, privacy safeguarding requirements, privacy policies; and privacy controls*) and a set of privacy principles (c*onsent and choice, purpose legitimacy and specification, collection limitation, data minimization, use, retention and disclosure limitation, accuracy and quality, openness, transparency and notice, individual participation and access, accountability, information security, privacy compliance*).

From a privacy engineering viewpoint we believe that a number of concepts and principles should be added. Paraphrasing ISO 29100, a privacy engineering framework is needed which specifies a common privacy engineering terminology; which defines the actors and their roles in the engineering of systems integrating privacy management; which describes considerations on engineering privacy safeguards; and which provides references to known privacy engineering principles.

There are a number of reasons to define a privacy engineering framework. First we need a convergence of terms. A number of concepts and principles for privacy engineering have been debated in the last years, for instance privacy-by-design principles (*privacy-by-policy, privacy-by-architecture* [37], *minimization* [38], *enforcement, transparency* [39]), privacy protection objectives (*predictability, manageability, disassociability* [40]) or privacy engineering objectives (*unlinkability, transparency, intervenability* [41]).

Secondly we also need some guidance on how to have an integrated view of the today complex maze of initiatives and how to extract concerns that are important from an engineering viewpoint. Most guideline documents today are management oriented and not engineering oriented making them difficult to use.

Finally, we must pave the way to future privacy engineering standards. The advent of a privacy engineering practice will depend on the availability of a number of standards. We believe that the definition of such standard will be facilitated by the existence of a privacy engineering framework, constructed as an extension of a privacy framework. **Fig. 7** shows the relationship between a privacy framework such is ISO29100, a privacy engineering framework and privacy standards. The *privacy framework* box is extended by a *privacy engineering framework* box acting as a placeholder for future standards, e.g. a *privacy engineering methodology*, or a *privacy risk analysis*.

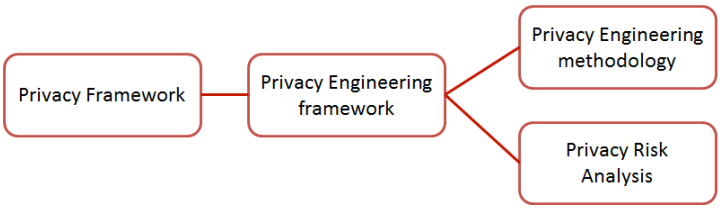

Fig. 7. Privacy Engineering Framework Placeholder for Future Standards

The call for a privacy engineering framework has been made in a technical paper published by MITRE in July 2014 [42]. It highlights the need to address privacy from both an organisational and an engineering viewpoint. The organisational viewpoint integrates elements such as a privacy program management, a compliance-focused risk assessment approach, a strategy and planning, and policies. The engineering viewpoint integrates elements such as privacy testing, privacy-sensitive design decisions, privacy requirements and control selection, and system focused risk assessment. The paper argues that the latter elements are not well taken into account.

3.3 Analysis of Privacy Engineering

This section provides an analysis of what is meant by privacy engineering, relying on research contributions made in the area, and showing that privacy engineering goes beyond security engineering. It will first cover definitions suggested concerning privacy engineering, privacy engineering objectives and privacy protection properties: it will then explain the advances made concerning the understanding of a resulting engineering lifecycle, covering the specific capabilities needed for privacy engineering, i.e. the operationalisation of privacy principles, the application of design strategies, and the integration of risk management.

Privacy Engineering

The Privacy Engineer's Manifesto [43] states that it uses the term privacy engineering *in recognition that the techniques used to design and build other types of purposefully architected systems can and should be deployed to build or repair systems that manage data related to human beings*. It then explains that the book discusses *how to develop good functionalized privacy policies and shows recognized methodologies and modeling approaches adapted to solve privacy problems*. NIST published in May 2015 a report focusing on privacy risk management [40] where it states that *for the purposes of the publication, privacy engineering is a collection of methods to support the mitigation of risks to individuals arising from the processing of their personal information within information systems*.

In the work carried out by PRIPARE to define a privacy engineering methodology [44, 45], it was realized that two viewpoints must be taken, the goal oriented approach in design and the risk oriented approach concepts. We therefore define privacy engineering *as a collection of methods to design and build privacy capabilities while*

integrating the treatment of risks to individuals arising from the processing of their personal information.

Privacy Engineering Objectives

In the development of systems, concern is on agreeing and achieving system or product qualities. The ISO standard 25010 [46] which focuses on systems and software quality requirements states the following: "*this can be achieved by defining the necessary and desired quality characteristics associated with the stakeholders' goals and objectives for the system"*. The standard also makes a difference between *quality in use of a system* which characterizes the impact that the product (system or software product) has on stakeholders, and *product quality*. Examples of qualities in use are effectiveness, efficiency, freedom of risk. The standard further provides examples of product quality categories such as *functional suitability, performance efficiency, compatibility, security,* or *portability*, and for each category examples of qualities. For instance *confidentiality, integrity*, and *availability* are well-known qualities in the security category.

The NIST report [43] defines three privacy engineering objectives, *predictability, manageability,* and *disassociability*. Predictability is the enabling of reliable assumptions by individuals, owners, and operators about personal information and its processing by an information system. Manageability is providing the capability for granular administration of personal information including alteration, deletion, and selective disclosure. Disassociability is enabling the processing of personal information or events without association to individuals or devices beyond the operational requirements of the system.

Compared to ISO 25010, we conclude that privacy engineering integrate new concerns and qualities. While Predictability and manageability are well known categories, there is a need to express the specific capabilities that are related to personal information processing. Disassociability can be considered as a specific engineering objective for privacy.

Properties for Privacy Protection

ULD[12] presented in May 2015 a paper on privacy protection goals for privacy engineering [41]. The paper extends security protection goals (i.e. *confidentiality, integrity, availability*) with privacy protection goals (i.e. *unlinkability, transparency, and intervenability*). It further defines three axes which can be considered degrees of freedom, the confidentiality – availability axis, the integrity – intervenability axis, and the unlinkability – transparency axis. For instance, user empowerment capability is covered by the transparency goal.

[12] Unabhängiges Landeszentrum für Datenschutz Schleswig-Holstein (www.datenschutzzentrum.de). Data protection authority in the federal state of Schleswig-Holstein, Germany

Table 3. Three Privacy Engineering Axes

Confidentiality	Availability
No access to data	Full access to data
No access to services	Full access to services
Authorised entities only	Everybody

Integrity	Intervenability
No changes to data	All types of changes
No access to process	Full process flexibility
Defined by processor	Defined by individual

Unlinkability	Transparency
No linkable data	Full linkability of data
No disclosure of process	Full disclosure of process
Need-to-know	Want-to-know

Privacy Engineering Lifecycle

PRIPARE's privacy engineering methodology [44, 45] shows that privacy concerns have to be dealt with in all phases of a development lifecycle. As shown in **Table 4**, a classical model is used, including the following phases: *analysis*, *design*, *implementation*, *verification*; *release* (delivery to customer), *maintenance*, and *decommission*. An additional component has been added: *environment & infrastructure,* the objective of which is to put in place an appropriate organisational structure for privacy management, as well as in in-house awareness program for privacy. ISO 27034 [47] is a standard which specifies a similar organisational structure for security called organisation normative framework or ONF. An ONF is defined as a repository where all application security best practices recognized by the organization are stored, derived and refined. This additional component can therefore be considered as an extension of ISO 27034 to privacy. Here is a brief description of the other proposed phases for privacy engineering.

The *analysis* phase includes a high level functional analysis (e.g. identifying the types of data that have to be collected), an operational requirements elicitation process, the objective of which is transform high-level privacy principle requirements into operational engineering requirements. For instance the accountability principle can lead to a requirement on protection enforcement capabilities. It also includes a legal compliance process which will involve privacy impact assessment activities.

The *design* phase will cover the specification and design of privacy controls or PETs. For instance an access log capability could be a privacy control covering the accountability requirement. The design phase must also include an evaluation of the impact on architecture. For instance the decision to keep data in a device instead of transferring it in the cloud has a strong architectural impact.

The *implementation* phase deals with transforming a design into a built system, For the sake of simplicity, we have assumed that this phase also includes all integration activities. For instance, the resulting implementation could contain several privacy controls that need to be integrated.

The *verification* phase ensures that the system meets privacy operational requirements. The verification that privacy controls are correctly implemented could be done through different approaches including static and dynamic verifications. For instance [48] elaborates on how privacy can be verified through ontologies. An accountability capability check has to be included in order to ensure that all the needed measures for protection (e.g. enforcing confidential access to some data) and for proof of protection (e.g. provable log of access to data) have been implemented correctly.

The *release* phase focuses on publishing the first complete privacy impact assessment report after the final privacy review, and creating two important plans: the incident response plan which focuses on all the measures that are anticipated in case of the discovery of a privacy breach, and the system decommissioning plan which focuses on all the measures related to obligations to remove data.

The *maintenance* phase focuses on reacting to privacy breach incidents, i.e. the execution of the incident response plan, on preventive maintenance, and on updating the privacy impact assessment report.

The *decommission* phase focuses on executing the system decommissioning plan and on correctly dismantling the systems according to current legislation and policies.

Table 4. Life Cycle Processes

	Processes
Environment & Infrastructure	Organisational privacy architecture Promote privacy awareness
Analysis	Preliminary, functional description and high-level privacy analysis, privacy requirements operationalisation, legal compliance
Design	Privacy control design, Architecture impact evaluation, Privacy control detailed Design
Implementation	Privacy control implementation
Verification	Privacy control verification (static analysis, dynamic Analysis), Accountability,
Release	Create incident response plan, Create system decommissioning plan, Final privacy review, Publish privacy impact assessment report
Maintenance	Execute incident response plan, Privacy verifications, Updating privacy impact assessment report
Decommissioning	Execute decommissioning plan

Operationalisation of Privacy Principles

Privacy principles as defined by Ann Cavoukian [49][13] or in ISO 29100 [36][14] are the starting points for a privacy engineering practice. The operationalization of privacy

[13] i.e. proactive not reactive; preventative not remedial, privacy as the default setting, privacy embedded into design, full functionality, end-to-end security, visibility and transparency, respect for user privacy.

[14] i.e. consent and choice, purpose legitimacy and specification, collection limitation, data minimization, use retention and disclosure limitation, accuracy and quality, openness,

principles is a process that leads to the definition of services necessary to support privacy management,

The privacy management reference model and methodology or PMRM [16] defines a comprehensive methodology for operationalization. The methodology is iterative and based on a use case specification approach. It leads to the identification of a set of operational services (*agreement, usage, validation, certification, enforcement, security, interaction, access*). PMRM methodology includes on important concept: *touch points*. Touch points are defined as *intersection of data flows with privacy domains or systems within privacy domains*. From an engineering and management viewpoint they represent important interfaces that may lead to contractual and/or legal obligations (e.g. a touch point could be an interface with a business partner). Here is an example taken from the PMRM specification, related to electric vehicle charging: when a customer plugs into the charging station, the electric vehicle on-board system embeds communication functionality to send its identification and charge requirements to the customer communication portal. This functionality corresponds to a touch point.

Application of Design Strategies

Once privacy principles have been operationalized, i.e. once services to support privacy management have been defined, we have to focus on the design phase, i.e. on how these services will be implemented. Hoepman [33] defines two important engineering concepts: design strategies, and privacy patterns. Design strategies allow for the selection of privacy controls. Four data oriented strategies (minimize, hide, separate, aggregate) and four process oriented strategies (inform, control, enforce demonstrate) are identified, Privacy patterns are high level engineering representations of privacy controls [50, 51, 52]. They can be therefore documented as best practice solutions in an organization normative framework as suggested by ISO 27034 [47]. **Table 5** provides an overview of patterns that can be used for each design strategy.

Table 5. Design Strategies and Examples of Patterns

Strategy	Description	Examples
Minimize	Amount of processed personal data restricted to the minimal amount possible	Select before you collect Anonymisation / Pseudonyms
Hide	Personal data, and their interrelationships, hidden from plain view	Storage and transit encryption of data Mix networks Hide traffic patterns Attribute based credentials Anonymisation / Pseudonyms
Separate	Personal data processed in a	Peer-to-peer arrangement

transparency and notice, individual participation and access, accountability, information security, privacy compliance.

	distributed fashion, in separate compartments whenever possible	Isolation and virtualization
Aggregate	Personal data processed at highest level of aggregation and with least possible detail in which it is (still) useful	Aggregation over time (used in smart metering) Dynamic location granularity (used in location based services) k-anonymity Differential privacy
Inform	Transparency	Platform for privacy preferences Data breach notification
Control	Data subjects provided agency over the processing of their personal data	User centric identity management End-to-end encryption support control
Enforce	Privacy policy compatible with legal requirements to be enforced	Access control Sticky policies and privacy rights management
Demonstrate	Demonstrate compliance with privacy policy and any applicable legal requirements	Privacy management systems Use of logging and auditing

A design phase in a development lifecycle includes decisions on the architecture of the system being developed. Privacy engineering can there therefore lead to architecture changes. For instance applying the minimize design strategy can lead to the decision that data is kept locally in a smart phone rather that globally on the cloud. Such architecture decisions are called PEARS for Privacy Enhancing Architectures [31]. **Fig. 8** shows the relationship between operational requirements, privacy controles, PETS and PEARs, Privacy controls are specified as the results of an operationalization. PETs are specified as the result of a privacy control design process, PETs can involve architecture change decisions or PEARs.

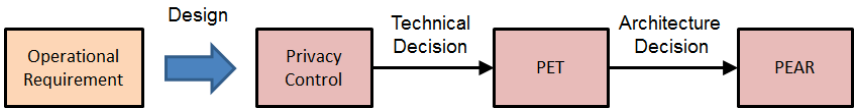

Fig. 8. Architecture decisions (PEAR) vs Technical decisions (PET)

Integration of Risk Management

As pointed out previously, risk assessment is a major concern of privacy management. One major aspect of privacy engineering is the proper integration of risk management activities. Risk management is a well-known domain. ISO 31000 [53] is the overarching standard, providing generic guidelines for the design, implementation and maintenance of risk management processes. Security risk is also quite well covered: for instance ISO/IEC 27005 [54] is a standard on security risk management; TVRA [55] is a security risk analysis methodology published by ETSI; STRIDE [56] is an approach which focuses on the threats associated with desirable security properties. Spoofing, tampering, repudiation, information disclosure, denial of service and elevation of privileges threats (hence STRIDE threats) will prevent

authentication, integrity, non-repudiation, confidentiality, availability, and authorization properties respectively.

Concerning privacy risks, references such as ISO/IEC 29134 [26], the CNIL PIA [27] or the smart grid data protection impact assessment template [28] focus on risks to the privacy of citizens. The NIST report published in May 2015 [43] provides a risk management approach for privacy management in information systems which further includes risks on business operations (e.g. a privacy breach could have a reputation impact which could then jeopardize the operations of an organization). LINDDUN [30] extends STRIDE by focusing on threats that would prevent desirable privacy properties, i.e. unlinkability, anonymity, plausible deniability, undetectability and unobservability, confidentiality, content awareness, policy and consent compliance. **Table 6** shows the resulting categories of threats, i.e. linkability, identifiability, non-repudiation, detectability, disclosure of information, unawareness, and non-compliance. In LINDDUN data flow diagrams describe the system, and threat tree patterns are used to guide the analysis.

Table 6. LINDDUN Categories of Privacy Threats

Property	Description	Threat
Unlinkability	Hiding the link between two or more actions, identities, and pieces of information.	Linkability
Anonymity	Hiding the link between an identity and an action or a piece of information	Identifiability
Plausible deniability	Ability to deny having performed an action that other parties can neither confirm nor contradict	Non-repudiation
Undetectability and unobservability	Hiding the user's activities	Detectability
Confidentiality	Hiding the data content or controlled release of data content	Disclosure of information
Content awareness	User's consciousness regarding his own data	Unawareness
Policy and consent compliance	Data controller to inform the data subject about the system's privacy policy, or allow the data subject to specify consents in compliance with legislation	Non compliance

While risk management is an important component of privacy engineering, it often conveys a negative mindset. As a matter of fact the engineering of a system generally follows a positive mindset based on goal-orientation [57]: engineers understand and build systems in terms of the goals they are intended to meet. While it can be understood that privacy must first be approached from a risk assessment viewpoint, this is not satisfactory from an engineering viewpoint. PRIPARE [45] has therefore defined an approach that combines a goal-oriented and a risk-based approach to discover and identify operational privacy requirements. This approach is depicted in **Fig. 9**.

The requirement analysis phase takes place in conjunction with risk management analysis. Risk management focuses on identifying the assets to protect in the system under development and the *threats* that might compromise the accomplishment of the *privacy principles* on these assets. Then a *treatment* is proposed to address the risk

associated with a threat. This treatment may range from doing nothing (accept the risk) to including requirements that may avoid or reduce the *risk*.

Requirement analysis is goal oriented: each *principle* is considered as a high level goal that the system must fulfil. Each goal is then refined into a set of lower-level *guidelines* required to meet the goal. Then a *success criterion* is proposed to address a guideline. The set of *treatments* and *success criteria* are jointly referred as *operational requirements* for privacy controls.

The design phase has the objective to design the *privacy controls*. They are realised by *measures* designed to meet the *success criteria* and by *countermeasures* designed to meet the *treatments*.

There is a correspondence between the concepts of threats-treatments and of guidelines-criteria: threat is the counterpart of guideline; treatment is the counterpart of success criterion. Both are operational requirements; countermeasure is the counterpart of measure. Both are privacy controls.

It is not expected to have a one-to-one mapping between threats and guidelines (or between treatments and criteria). It is rather expected that different operational requirements will be elicited by applying both the goal-oriented and risk based approaches. The tables below show two examples of threats where the data minimization and proportionality and transparency principles are applied.

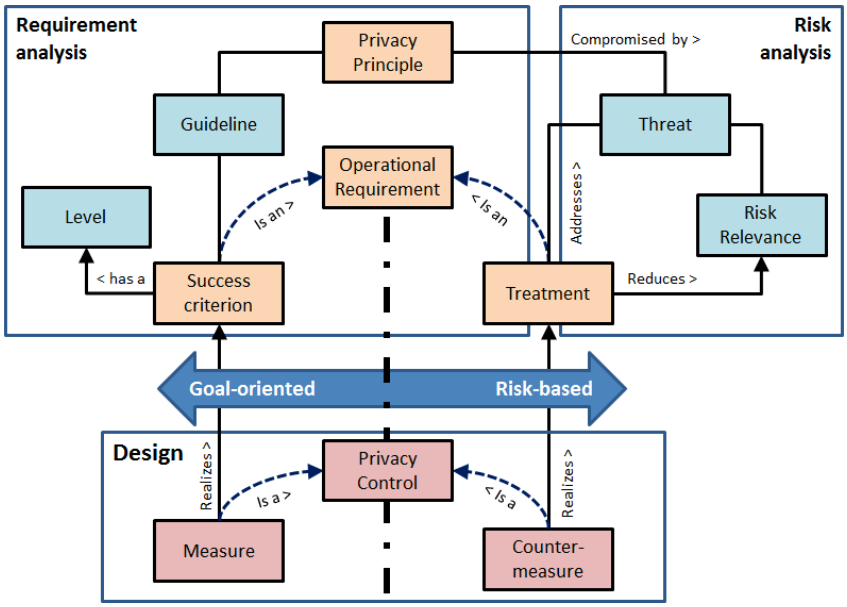

Fig. 9. PRIPARE Goal-oriented and Risk-based Requirement Elicitation

Table 7. Operationalisation Example (Data Minimization and Proportionality)

Concept	Example on data minimization and proportionality
Principle	Data minimization and proportionality

Threat	Accidental Data Leak
Risk relevance[15]	Significant
Guideline	Avoid or minimise the use of personal data along its whole lifecycle
Goal relevance[16]	Relevant
Operational requirement (Treatment or Criteria)	Keep data from different services or different parties separated, and avoid combining them When some personal data is no longer needed for the specified purpose, delete or anonymise all the back-up data corresponding to that personal data
Privacy control	Architecture change to keep personal data in smart phone Anonymisation and attribute based credentials
Test	Conformance testing of architecture (personal data kept in smart phone) Conformance testing of anonymisation

Table 8. Operationalisation Example (Transparency)

Concept	**Example on transparency**
Principle	Transparency (ex post transparency)
Threat	A data leak occurred. Organisation does not know which operation caused the leak
Risk relevance[9]	Maximum
Guideline	Provide a public privacy notice to the data subject
Goal relevance[10]	Relevant and essential
Operational requirement (Treatment or Criteria)	Describe how the organisation processes personal data Describe the internal uses of personal data
Privacy control	Secure log of access and operations
Test	Battery of penetration tests

4 Privacy Engineering Framework for the IoT

The previous sections showed that an understanding has been reached on privacy in the Internet of Things and on privacy engineering. This section describes a privacy engineering framework for the IoT.

We first propose a structure containing four sections: a concept section, a stakeholder section, a process section and an organisation section. The concept section is domain independent, i.e. we believe it can be used for any IT domain. The other sections are specific to IoT. They further make the difference between privacy engineering of IoT systems and privacy engineering of IoT subsystems. The rationale to make a distinction between IoT systems and IoT subsystems is to address the profound difference between integrators (who know the use case they have to cover) and suppliers (who only know the domain they have to cover).

[15] negligible, limited, significant, maximum.

[16] less relevant, relevant, relevant and essential.

4.1 Structure of a Privacy Engineering Framework

We propose the following structure for a privacy engineering framework in the table below.

The framework includes the following categories of artefacts: the concepts, the stakeholders, the processes to build a system, and the organisations building the system.

The following sections focus on each category of artefacts. In order to allow for easy use of the framework, all the important definitions in the framework are listed in tables, while the rest of the text focuses on rationale and explanations.

Table 9. Privacy Engineering Framework

Category of artefacts	Item in Framework
Concept	Privacy Engineering and Privacy-by-Design Privacy Engineering Objectives Privacy Protection Properties Privacy Engineering Principles
Stakeholder	Organisations' Role
Process	Privacy Control Requirements Privacy Control Design
Organisation	Environment and Infrastructure Lifecycle Approach

4.2 Concepts in the Framework

This section is common to all domains, i.e. it is not specific to the IoT. It covers the definition of the following concepts: privacy-by-design, privacy engineering, privacy engineering objectives and privacy engineering principles.

Privacy Engineering and Privacy-by-Design

The privacy engineering framework starts with the definition of two terms: privacy engineering and privacy-by-design.

Table 10. Core Definitions

Concepts	Definition
Privacy-by-design	Institutionalisation of the concepts of privacy and security in organisations and integration of these concepts in the engineering of systems.
Privacy engineering	Collection of methods to design and build privacy capabilities while integrating the treatment of risks to individuals arising from the processing of their personal information.

The privacy-by-design definition is taken from a blog entry contributed by PRIPARE[17]. The blog entry also provides other definitions: (1) *an approach to protecting privacy by embedding it into the design specifications of technologies, business practices, and physical infrastructures* (this definition is inspired from Ann Cavoukian); (2) *an approach to system engineering which takes into account privacy and measures to protect ICT assets during the whole engineering process*; (3) *embedding privacy and security in the technology and system development from the early stages of conceptualisation and design and institutionalizing privacy and security considerations in organisations;* (4) *applying a set of principles from the design phase of ICT systems in order to mitigate security and privacy concerns guiding designers and implementers decisions throughout the development of the systems*.

Privacy Engineering Objectives

Privacy engineering objectives extends other engineering objectives as described in **Table 11**.

Table 11. Privacy Engineering Objectives

Objective	Description
Predictability	Enabling reliable assumptions by individuals, owners, and operators about personal information and its processing by a system.
Manageability	Providing capability for granular administration of personal information including alteration, deletion, and selective disclosure.
Disassociability	Enabling the processing of personal information or events without association to individuals or devices beyond the operational requirements of the system.

This section is taken from [43].

Privacy Protection Properties

Privacy protection properties extend other engineering properties (e.g. security properties) as described in **Table 12**.

Table 12. Privacy Protection Properties

Objective	Description
Unlinkability	Ensures that privacy-relevant data cannot be linked across privacy domains or used for a different purpose than originally intended.
Transparency	Ensures that all privacy-relevant data processing including the legal, technical and organizational setting can be understood and reconstructed.
Intervenability	Ensures that data subjects, operators and supervisory authorities can intervene in all privacy-relevant data processing.

[17] http://www.securityengineeringforum.org/blog/show/id/27

This section is taken from [41].

Privacy Engineering Principles

A number of privacy engineering principles are defined. These principles are added to the ISO 29100 privacy principles to further *guide the design, development, and implementation of privacy policies and privacy controls* from an engineering viewpoint.

Table 13. Privacy Engineering Principles

Principle	Description	Rationale
Integration of privacy engineering objectives	Privacy engineering activities extend other engineering objectives focusing on predictability, manageability of managed data and on disassociability	Specific engineering objectives for privacy management are needed
Integration of risk management	Privacy engineering activities must be carried out jointly with the risk management activities needed to ensure proper handling of privacy. ISO 29134 and associated practices can be used as reference.	While an engineering viewpoint must be taken, engineers must include a risk management perspective
Integration of compliance	Privacy engineering activities must be carried out jointly with the compliance checking (e.g. technical obligations, legal obligations).	While an engineering viewpoint must be taken, engineers must include a compliance perspective. This can involve impact assessment documents, assurance and conformance activities.
Integration or privacy protection properties	Privacy engineering activities must integrate specific protection properties, unlinkability, transparency, intervenability	Specific privacy protection properties extend other requirement properties
Integration of goal-orientation in requirement engineering	The identification of requirements in privacy engineering must include is goal orientation approach where engineers describe requirements in terms of the goals that must be met by systems.	Goal orientation is needed for engineering. It will complement requirements elicited through risk analysis.
Data oriented Design strategies	Privacy engineering includes data oriented design strategies. These strategies can help address the unlinkability objective. They often lead to architectural decisions (privacy enhancing architectures).	Data oriented design strategies will help meet unlinkability properties
Process oriented Design strategies	Privacy engineering includes process oriented design strategies. These strategies can help address the transparency and intervenability objectives	Process oriented design will help meet transparency and intervenability properties.
Lifecycle Support	Privacy engineering extends to the entire lifecycle	Privacy management extends over the entire

		lifecycle. Consequently privacy engineering must extend over the entire lifecycle.
Privacy engineering knowledge capitalisation	Privacy engineering relies on knowledge capitalisation. Privacy controls can be stored and reuse (e.g. through privacy patterns). Processes can also be stored in organisation libraries	Privacy-by-design must be institutionalised within organisations.

4.3 Stakeholders in the Framework

This section focuses on stakeholders in the framework. It identifies the roles of organizations from a privacy viewpoint in a supply chain.

Table 14. Organisations' Role

Stakeholder	Definition	Type	Example
Data controller	Stakeholder operating a service that involves personal data collection	System	An operator of a social care network to assist elderly people.
Data processor	Stakeholder processing personal data on behalf to a data controller	System	A cloud platform operator providing data storage and processing capability
Integrator	Stakeholder integrating supplier systems in order to build a service	System	The developer of turnkey social care system
Supplier	Stakeholder developing a subsystem that is subsequently integrated	Subsystem	The designer of a sensor that can be integrated in the turnkey social care system The designer of an smart phone operating system that is subsequently used to run social care network capability

Privacy engineering for subsystems is not the same as privacy engineering for systems, because suppliers of subsystems are generally not aware of the privacy requirements of the system in which the subsystem will be integrated.

4.4 Processes in the Framework

This section focuses on process consideration in IoT privacy engineering, covering two items: the process for eliciting privacy control requirements and the process for designing privacy control. The framework makes a distinction between processes to build an IoT system compared to processes to build an IoT subsystem.

IoT systems are systems that are under the responsibility of a data controller, a data processor or an integrator carrying out a turnkey development for a data controller or

a data processor. From a privacy point of view, the purpose for which personal data is collected is known in an IoT system. Here is an example: Let the IoT system consists of a set of body sensors monitoring specific health data, a smart phone managing the data collected by the sensors, and an information system facility at the cloud level managing social networking capabilities between the user of the system, carers, family members and friends.

IoT subsystems are systems that will be used by an integrator carrying out a development of an IoT system. From a privacy point of view, the purpose for which personal data can be collected in the IoT system in which the IoT subsystem will be integrate is not known beforehand. This has a strong impact on how privacy engineering is carried out. While it is possible to determine privacy control requirements at the IoT system level, it is only possible to determine generic requirements at the IoT subsystem level.

Privacy Control Requirements for IoT

Table 15 shows the important elements of the process to elicit privacy control requirements for IoT systems. Privacy principles (e.g. in ISO 29100) need to be applied in the requirement analysis of a system to identify privacy control requirements. The engineering requirement analysis involves both a risk and a goal viewpoint. The starting points for the analysis are the privacy principles. Then the assets to protect are identified. From a risk analysis viewpoint, threats on assets are then defined and features to address threats are identified. From a goal viewpoint, concerns on assets are defined and features to address concerns are identified. In the IoT system example, data collected by sensors would be the asset, the threat would be unwanted geo-localisation, and the privacy control requirement would be to protect against geo-localisation.

Table 15. Artefacts used in Operationalisation in IoT

Analysis Process	Risk viewpoint	Goal Viewpoint
Input: Principle	Privacy principles	Privacy Principles
Intermediate artefacts	Assets to protect	Assets to protect
	Threats on assets	Concerns on assets
Output: Privacy control requirements	Features to address threats	Features to address concerns

The specification of features necessitates a number of analysis steps as showed below (based from OASIS-PMRM [16]).

Table 16 and **Table 17** list analysis steps as well as organisation help which consists of best practices and inventory of design artefacts (e.g. inventory of threats). The approach for IoT subsystem is made difficult by the fact that the designer does not know in advance in which IoT system it is going to be integrated. For instance, installing a video camera in the premises of a company is not the same as installing

the same video camera in the street. Consequently, the analysis steps must focus on a range of use cases and on privacy control support requirements.

Table 16. Analysis Steps for IoT Systems

Step	Description	Organisational help
1	Definition of scope of use case involving system	Inventory of applications and services associated with systems
2	Detailed use case analysis	Inventory of stakeholders, data flows and touch points (i.e. data exchange with other stakeholders)
3	Identification of privacy control requirements by carrying out a risk and a goal oriented analysis	Inventory of threats Inventory of privacy controls to address threats Risk analysis practices Inventory of concerns Inventory of privacy controls to address concerns

Table 17. Analysis Steps for IoT Subsystems

Step	Description	Organisational help
1	Identification of use cases involving the subsystem	Inventory of use cases involving the subsystem. Each use case is associated with data flow diagrams including touch points
2	Identification of range of privacy control support requirements	Inventory of privacy control support requirements Each use case is associated with privacy control needs scenarios

Design of Privacy Controls for IoT

Once privacy control requirements are available, a design phase must be carried out that leads to the design of privacy controls. The table below describes the design steps as well as organizational help items. In the IoT example, the designer could select from the suppliers an IoT subsystem which supports geo-localisation PETS (see **Table 1**). Step 5 shows that the design of IoT systems could include the identification of privacy control support provided by IoT subsystems. For instance the underlying operating system in a smart phone device might provide security components that are handy to use.

Table 18. Design Steps for IoT Systems

Step	Description	Organisational help
1	Starting from privacy control requirements, global design of privacy control	Inventory of design strategies and privacy controls

2	If needed identification of architecture decisions (PEARs)	Inventory of architecture decisions associated with privacy controls Architecture design practices
3	If needed evaluate architecture	Architecture evaluation practices
4	Detailed design of privacy control If possible, identification of privacy patterns	Inventory of privacy patterns
5	Identification of resulting privacy control support in subsystems If possible, identification of subsystems composition rules	Inventory of supplier products with privacy control support features Inventory of IoT subsystems compositions schemes
6	Evaluation of privacy control effectiveness (e.g. privacy quantification)	Privacy control usage return on experience
7	Evaluation of compliance	

The design steps in the case of IoT subsystems are showed in the table below.

Table 19. Design Steps for IoT Subsystems

Step	Description	Organisational help
1	Design of privacy control features	Inventory of features
2	if needed identification of architecture decisions (PEARs)	Inventory of architecture decisions
5	Identification of resulting privacy control features in subsystems	Inventory of privacy control support features

4.5 Organisations in the Framework

This section focuses on organisation considerations in IoT privacy engineering, covering two items: the overall environment and infrastructure process and the lifecycle approach.

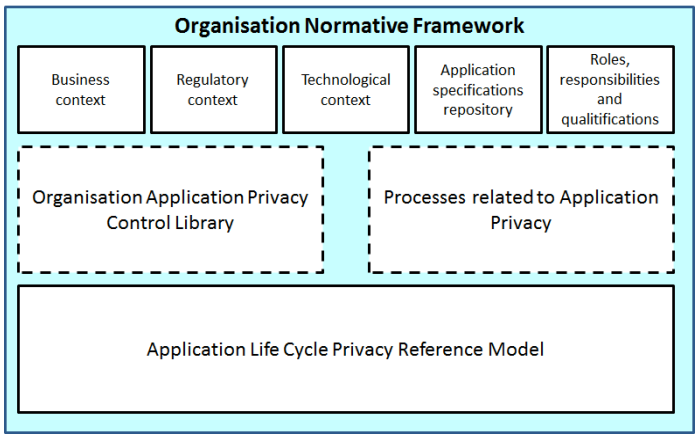

Fig. 10. Organisation Normative Framework for Privacy

Environment & Infrastructure for IoT

Following ISO 27034 [26], we suggest to define an organization normative framework (ONF) adapted to the need of privacy engineering for the IoT. **Fig. 10** shows a high-level view of the ONF contents. It highlights the following components: references on the business context, the regulatory context, the technological context, a repository of application specifications, roles, responsibilities and qualifications; a repository of privacy controls, a repository of processes related to application privacy, and an application privacy life cycle reference model. **Fig. 10** is valid for both IoT systems and subsystems. In the latter case, the application life cycle privacy reference model must be replaced with a subsystem life cycle privacy reference model.

Lifecycle Approach for IoT

Organisations integrating privacy in their engineering activities must take into account all phases of the lifecycle. In order to allow for easier integration of privacy engineering activities into existing methodologies (waterfall, agile, prototyping), it is advised to structure a privacy engineering methodology into phases and processes that can then be easily integrated in an organisation development methodology.

The table below describes the main phases of privacy engineering (based on PRIPARE handbook [44]), the associated activities for organisations building IoT systems, and the equivalent ISO 15288 system life cycle processes [58].

Table 20. Phases and activities for IoT systems.

Privacy Engineering Phase	**Privacy Engineering Activities adapted to IoT systems**	**System Life Cycle Processes (ISO 15288)**
Environment & Infrastructure	Organisational privacy architecture Promote privacy awareness	Infrastructure management process Project privacy portfolio management process
Analysis	Use case functional description Use case privacy analysis Use case privacy requirements Legal compliance	Stakeholder privacy requirements definition process Privacy requirements analysis process
Design	Privacy Control Design Privacy Control detailed design	Privacy architectural design process
Implementation	Privacy control implementation	Privacy implementation process
Verification	Privacy Control Verification, Accountability, Static analysis, Dynamic Analysis	Privacy Verification process
Release	Create Incident Response Plan, Create system decommissioning	Transition process

	plan, Final Privacy review, Publish PIA report	
Maintenance	Execute incident response plan, Privacy verifications	Maintenance process
Decommissioning	Execute decommissioning plan	Disposal process

The below table shows the privacy engineering activities for IoT subsystems.

Table 21. Phases and Activities

Privacy Engineering Phase	**Privacy Engineering Activities adapted to IoT subsystems**
Environment & Infrastructure	Organisational privacy architecture Promote privacy awareness
Analysis	Domain functional description Domain privacy analysis Privacy support requirements
Design	Privacy support design Privacy support detailed design
Implementation	Privacy implementation
Verification	Privacy support verification
Release	Create privacy support documentation
Maintenance	Subsystem maintenance
Decommissioning	-

5 Conclusions

As we have explained, privacy issues are of particular relevance in the Internet of Things. Besides the large amount of personal data amassed by such systems, traditional privacy measures, based on well-established principles such as transparency, purpose specification, legitimate use or consent, break down in the face of IoT features such as pervasive sensing, indiscriminate data collection, invisible interfaces, and deferred de-anonymization. However the relevance it may have, privacy is oftentimes dismissed or overlooked when developing IoT systems. One of the possible reasons is that privacy initiatives in the field are yet disparate, unorganized and unconnected. This lack of a systematic approach upholds the need for a framework that provides common grounds to methodically capture and address privacy issues in the IoT.

This paper represents a first step towards the description of such a privacy engineering framework for the Internet of Things. The framework draws on complementary perspectives. A conceptual framework covers privacy engineering, privacy objectives, principles, properties, life cycle, and strategies. These concepts pave the foundations of a series of privacy engineering processes or development activities involved in privacy engineering, which move from the requirement elicitation (including the operationalization of abstract privacy principles into concrete requirements) to the analysis (including the analysis of privacy threats), to the design (including strategies for architectural design), to the implementation (including a catalogue of useful controls for IoT) and the validation; besides others that take place already during the operation of a system (release, maintenance and

decommissioning). These processes synthesize different perspectives: managerial and engineering, organizational and systemic, risk-based and goal-oriented, environment and infrastructure, etc.

Special emphasis is given to the fact that IoT value chain includes a role such as the subsystem supplier, which is not usually considered by privacy regulations (focused only on data controllers and processor), yet it may have a decisive impact in the final properties of all the systems that include their products. The difference stems from the fact that the engineering of an IoT system requires the design of privacy controls while the engineering of an IoT subsystem requires the design of privacy control supports, or features that can be used to build privacy controls at integration time. This distinction also shows that unfortunately not much has been done in the area of privacy engineering for subsystems. We believe that future research is needed in this area.

The research leading to these results has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration, through the projects PRIPARE (PReparing Industry to Privacy-by-design by supporting its Application in Research) and RERUM (RERUM: REliable, Resilient and secUre IoT for sMart city applications) under grant agreements nº 610613 and 609094 respectively. We would like to acknowledge as well the contributions of all the partners of both projects.

References

1. David Sweeny, MIT Technology Review's New Issue Reveals Annual 10 Breakthrough Technologies. Digital Press Release. 2013. Available via: http://www.technologyreview.com/pressroom/pressrelease/20130423-10-breakthrough-technologies/, last visited on 21.06.2016.
2. Article 29 Data Protection Working Party: Opinion 03/2013 on purpose limitation adopted on 2 April 2013. http://ec.europa.eu/justice/data-protection/article-29/documentation/opinion-recommendation/files/2013/wp203_en.pdf;, last visited on 21.06.2016.
3. Lee Rainie and Janna Anderson, The Future of Privacy. Pew Research Center. December 18, 2014. Available via: http://www.pewinternet.org/2014/12/18/future-of-privacy/, last visited on 21.06.2016.
4. Directive 95/46/EC http://ec.europa.eu/justice/policies/privacy/docs/95-46-ce/dir1995-46_part1_en.pdf, last visited on 21.06.2016.
5. General Data Protection Regulation: http://ec.europa.eu/justice/data-protection/reform/files/regulation_oj_en.pdf, last visited on 21.06.2016.
6. Mandate M530 on privacy management of security projects and services. http://ec.europa.eu/growth/tools-databases/mandates/index.cfm?fuseaction=search.detail&id=548; last visited on 21.06.2016.
7. Privacy-by-Design. http://www.ipc.on.ca/english/Privacy/Introduction-to-PbD, last visited on 21.06.2016.
8. Dave Evans, The Internet of Things: How the Next Evolution of the Internet Is Changing Everything. April 2011. http://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf, last visited on 21.06.2016

9. Tragos, E. Z., Angelakis, V., Fragkiadakis, A., Gundlegard, D., Nechifor, C. S., Oikonomou, G. & Gavras, A. (2014, March). Enabling reliable and secure IoT-based smart city applications. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2014 IEEE International Conference on* (pp. 111-116). IEEE.
10. Pöhls, H. C., Angelakis, V., Suppan, S., Fischer, K., Oikonomou, G., Tragos, E. Z., ... & Mouroutis, T. (2014, April). RERUM: Building a reliable IoT upon privacy-and security-enabled smart objects. In *Wireless Communications and Networking Conference Workshops (WCNCW), 2014 IEEE* (pp. 122-127). IEEE.
11. Bassi, A., Bauer, M., Fiedler, M., Kramp, T., Van Kranenburg, R., Lange, S., & Meissner, S. (2013). Enabling things to talk. *Designing IoT Solutions With the IoT Architectural Reference Model*, 163-211.
12. AIOTI - High Level Architecture, 2015. https://docbox.etsi.org/smartM2M/Open/AIOTI/!!20151014Deliverables/AIOTI WG3 IoT High Level Architecture - Release_2_0-lines.pdf, last visited on 25.06.2016.
13. International Organization for Standardization (ISO), Internet of Things Reference Architecture (IoT RA), Under development.
14. Elias Tragos, et al., Deliverable D2.5 – Final System Architecture. RERUM Deliverable. 2014. Available via: https://bscw.ict-rerum.eu/pub/bscw.cgi/d31979/RERUM%20deliverable%20D2_5.pdf, last visited on 21.06.2016.
15. Jayavardhana Gubbi, et al., Internet of Things (IoT): A vision, architectural elements, and future directions. Future Generation Computer Systems 29.7 (2013): 1645-1660.
16. Organization for the Advancement of Structured Information Standards (OASIS) Privacy Management Reference Model and Methodology (PMRM), Version 1.0. July 2013. http://docs.oasis-open.org/pmrm/PMRM/v1.0/PMRM-v1.0.pdf, last visited on 21.06.2016.
17. Leonid Titkov, Poslad Stefan, and Jim Tan Juan, An integrated approach to user-centered privacy for mobile information services. Applied Artificial Intelligence 20.2-4 (2006): 159-178.
18. Florian Scheuer, Klaus Plößl and Hannes Federrath, Preventing profile generation in vehicular networks. Networking and Communications, 2008. WIMOB'08. IEEE International Conference on Wireless and Mobile Computing, IEEE, 2008.
19. Elias Tragos, et al., Deliverable D2.3 - System Architecture. RERUM Deliverable. 2014. Available via: https://bscw.ict-rerum.eu/pub/bscw.cgi/d18321/RERUM%20deliverable%20D2_3.pdf, last visited on 21.06.2016.
20. Siani Pearson and Marco Casassa Mont, Sticky policies: an approach for managing privacy across multiple parties. Computer 9 (2011): 60-68.
21. Denise Demirel et al., Deliverable D4.4 - Overview of Functional and Malleable Signature Schemes. Prisma Cloud Deliverable. 2015. Available via: https://online.tugraz.at/tug_online/voe_main2.getvolltext?pCurrPk=86456, last visited on 21.06.2016.
22. Mark Manulis, et al., Group Signatures: Authentication with Privacy. Federal Office for Information Security-Study, Cryptographic Protocols Group, Department of Computer Science, Technische Universität Darmstadt, Germany, 2012.
23. Camenisch, Jan, and Els Van Herreweghen. "Design and implementation of the idemix anonymous credential system." Proceedings of the 9th ACM conference on Computer and communications security. ACM, 2002.
24. Batina Lejla, et al., Low-cost elliptic curve cryptography for wireless sensor networks. Security and Privacy in Ad-Hoc and Sensor Networks (pp. 6-17). Springer Berlin Heidelberg, 2006.
25. Jorge Cuellar, Santiago Suppan, and Henrich Poehls. Privacy-Enhanced Tokens for Authorization in ACE. Internet Draft. 2015.

26. ISO/IEC 29134 Draft International Standard. Information technology — Security techniques — Privacy impact assessment — Guidelines
27. CNIL Privacy Impact Assessment. Methodology: https://www.cnil.fr/sites/default/files/typo/document/CNIL-PIA-1-Methodology.pdf Tool: https://www.cnil.fr/sites/default/files/typo/document/CNIL-PIA-2-Tools.pdf. Good practices: https://www.cnil.fr/sites/default/files/typo/document/CNIL-PIA-3-GoodPractices.pdf, last visited on 21.06.2016
28. EC Data Protection Impact Assessment Template for Smart Grid and Smart Metering Systems . https://ec.europa.eu/energy/sites/ener/files/documents/2014_dpia_smart_grids_forces.pdf, last visited on 21.06.2016
29. ISO/IEC 29151. Draft International Standard. Code of Practice for Personally identifiable information protection,
30. LINDDUN privacy threat analysis methodology, https://distrinet.cs.kuleuven.be/software/linddun/. last visited on 21.06.2016
31. Antonio Kung, PEARs: Privacy Enhancing Architectures. Annual Privacy Forum, May 21-22, 2014, Athens, Greece. Proceedings APF14 "Privacy Technologies and Policy". Lecture Notes in Computer Science Volume 8450, 2014, pp 18-29
32. Software Architecture in Practice (3rd Edition), Len Bass, Paul Clementz, Rick Kazman. Addison-Wesley, 2012
33. Japp Henk Hoepman, Privacy design strategies. ICT Systems Security and Privacy Protection - 29th IFIP TC 11 Int.Conf, SEC 2014, Marrakech, Morocco
34. Kent Beck et al., Manifesto for Agile Software Development. Agile Alliance. http://agilemanifesto.org/, last visited on 29.09.2015.
35. OASIS Privacy by Design Documentation for Software Engineers (PbD-SE) TC https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=pbd-se, last visited on 21.06.2016.
36. ISO/IEC 29100:2011. Information technology – Security techniques – Privacy framework,
37. Sarah Spiekermann and Lorrie Cranor, Privacy Engineering. IEEE Transactions on Software Engineering, Vol. 35, Nr. 1, January/February 2009, pp. 67-82.
38. Sesa Gürses, Carmela Troncoso, and Claudia Diaz, Engineering Privacy-by-Design. Computers, Privacy & Data Protection, 2011
39. Antonio Kung, Johan-Christoph Freytag, and Frank Kargl, “Privacy-by-design in ITS applications. 2nd IEEE International Workshop on Data Security and Privacy in wireless Networks, June 20, 2011, Lucca, Italy.
40. NISTIR 8062 (Draft). “Privacy Risk Management for Federal Information Systems”. May 2015. http://csrc.nist.gov/publications/drafts/nistir-8062/nistir_8062_draft.pdf, last visited on 21.06.2016.
41. Marit Hansen, Meiko Jensen, and Martin Rost, Protection Goals for Engineering Privacy. 2015 International Workshop on Privacy Engineering – IWPE'15.
42. MITRE Privacy engineering framework. July 2014. http://www.mitre.org/publications/technical-papers/privacy-engineering-framework, last visited on 21.06.2016.
43. The Privacy Engineer’s Manifesto. Getting from Policy to Code to QA to Value. Michelle Finnaran Dennedy, Jonathan Fox, Thomas Finneran. Apress. ISBN13: 978-1-4302-6355-5, January 2014.
44. PRIPARE methodology. Final version. http://prepareproject.eu/wp-content/uploads/2013/11/PRIPARE-Methodology-Handbook-Final-Feb-24-2016.pdf , last visited 21.06.2016.
45. Nicolás Notario et al., PRIPARE: Integrating Privacy Best Practices into a Privacy Engineering Methodology. 2015 International Workshop on Privacy Engineering – IWPE'15.

46. ISO/IEC 25010:2011 Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE)) — System and software quality models.
47. ISO/IEC 27034:2011 Information technology — Security techniques — Application security
48. Martin Kost, Johann-Christoph Freytag, Frank Kargl, Antonio Kung. Privacy Verification Using Ontologies. First International Workshop on Privacy by Design (PBD 2011), August 28, 2011, Vienna, Austria
49. Ann Cavoukian. Privacy-by-Design. The seven foundational principles. https://www.ipc.on.ca/images/resources/7foundationalprinciples.pdf, last visited on 21.06.2016.
50. Munawar Hafiz, A Collection of Privacy Design Patterns. Proceedings of the Pattern Language of Programs Conference, 2006.
51. Sasha Romanosky, et al., Privacy Patterns for Online Interactions. Proceedings of the Pattern Languages of Programs Conference, 2006
52. Nick Doty, Privacy Design Patterns and Anti-Patterns. Trustbusters Workshop at the Symposium on Usable Privacy and Security. July 2013.
53. ISO 31000:2009. Risk management
54. ISO/IEC 27005:2011 Information technology — Security techniques — Information security risk management
55. ETSI. Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); Methods and protocols; Part 1: Method and proforma for Threat, Risk, Vulnerability Analysis ETSI TS 102 165-1 V4.2.3 (2011-03)
56. J.D. Meier, Alex Mackman, Michael Dunner, Srinath Vasireddy, Ray Escamilla and Anandha Murukan. Improving Web Application Security: Threats and Countermeasures, Microsoft Corporation. Published: June 2003. Chapter 2 Threats and Countermeasures. https://msdn.microsoft.com/en-us/library/ff648641.aspx, last visited 21.06.2016.
57. A. van Lamsweerde, Goal-Oriented Requirements Engineering: A Guided Tour. 5th International Symposium on Requirements Engineering, IEEE Computer Society Press, 2001
58. ISO/IEC/IEEE 15288:2015 Systems and software engineering – System life cycle processes