Methods and tools for GDPR Compliance through Privacy and Data Protection 4 Engineering

Methods for data protection model-driven design

Project: PDP4E
Project Number: 787034
Deliverable: D5.4
Title: Methods for data protection model-driven design protection and privacy
Version: v1.0
Date: 29/07/2019
Confidentiality: Public
Author(s): Gabriel Pedroza (CEA),
Patrick Tessier (CEA),
Julien Signoles (CEA),
Thibaud Antignac (CEA),
Victor Muntes (Beawre),
Jacek Dominiak (Beawre),
Elena González (Beawre),
David Sanchez (Trialog),
Yod Samuel Martin (UPM)

Funded by

Table of Contents

Document History

Version	Status	Date
V0.1	Initial Table of Contents	24/04/2019
V0.2	First draft of Section 3 including overall method and phases descriptions.	20/06/2019
V0.3	Contribution from Trialog. First description of Section 2.2 on standards and regulation.	20/06/2019
V0.4	Contributions from BeAwre. First descriptions of Sections 2.2.2 on LINDDUN method and 3.2 on personal data identification.	26/06/2019
V0.5	Integration and harmonization of contributions.	27/06/2019
V0.6	Executive summary, overall summary, bibliography added and harmonized.	28/06/2019
V0.7	Integration of missing references.	01/07/2019
V0.8	Contribution about method for code validation: CEA-LSL	09/07/2019
V0.9	Addressing remarks from Tecnalia	17/07/2019
V1.0	Addressing remarks from UDE	24/07/2019
V1.0	Inputs from UPM. Final remarks on new sections.	29/07/2019

Approval		
	Name	**Date**
Prepared	Gabriel Pedroza (CEA)	24/04/2019
Reviewed	Jabier Martinez (Tecnalia)	16/07/2019
Reviewed	Nicolas E. Diaz Ferreyra (UDE)	22/07/2019
Authorised	Antonio Kung (Trialog)	31/07/2019

Circulation	
Recipient	**Date of submission**
Project partners	29/07/2019
European Commission	31/07/2019

List of Figures

List of Tables

Abbreviations and Definitions

Abbreviation	Definition
AID	Available Information Diagram
ASR	Architectural Significant Requirements
AST	Abstract Syntax Tree
BPMN	Business Processing Model and Notation
CAPRIV	Computer Assisted Privacy Engineering
CMU	Carnegie Mellon University
DFD	Data Flow Diagrams
DPIA	Data Protection Impact Assessment
DSIFD	Detailed Stakeholder Information Flow Diagram
DSL	Domain Specific Language
GDPR	General Data Protection Regulation
ICT	Information and Communication Technologies
IETF	Internet Engineering Task Force
IoT	Internet of Things
LGPL	Lesser General Public License
LINDDUN	Linkability, Identifiability, Non-repudiation, Dectectability, information Disclosure, content Unawareness, and policy and consent Non-compliance
MDE	Model Driven Engineering
OEM	Original Equipment Manufacturers

PbD-SE	Privacy by Design Documentation for Software Engineers
PDP	Privacy and Data Protection
PDPbD	Privacy and Data Protection by Design
PDP4E	Privacy and Data Protection 4 Engineering
PDP4E-Req	Tool resulted from WP4 to management GDPR and privacy requirements
PET	Privacy-enhancing Technologies
PRIPARE	PReparing Industry to Privacy-by-design by supporting its Application in REsearch
ProPAn	Problem-based Privacy Analysis
PID	Personal Information Diagram
PII	Personal Identifiable Information
PSCS	Precise Semantics of UML Composite Structures
ReqIF	Requirements Interchange Format
RFC	Request For Comments
SDLC	Systems and Software Development Life Cycle
SIPOC	Suppliers, Inputs, Process, Outputs, Customers
SQL	Structured Query Language
SysML	Systems Modeling Language
TFEU	Treaty on the Functioning of the European Union
UML	Unified Modelling Language
UML4PF	UML 4 Problem Frames
UDEPF	University Duisburg-Essen Problem Frames
V&V	Validation and Verification
WP	Work Package
WP29	Article 29 Data Protection Working Party

Executive Summary

This document describes a method to support engineers in the goal of achieving Privacy and Data Protection by Design (PDPbD) in systems and software projects. Such method takes into account the legal obligations introduced by the EU General Data Protection Regulation (GDPR) and seeks to incorporate them into the project at early stages. The method is composed by several phases which are also described. The method addresses several concerns related to privacy and data protection at different levels of design. In particular, it covers aspects like the identification of personal data and their linkability, the representation of processes and architectures conveying data at high level, and the validation of privacy-related properties via different strategies and techniques including validation at code level. When achieved, referred validation provides evidence of requirements fulfilment and increases certainty on the properties the system under design should have.

Overall, the core contributions of this deliverable are:

- A short state of the art including standards, methods and techniques selected as the background of the PDPbD method
- A first draft of the PDPbD method that aims to provide guidance to non-savvy privacy engineers in order to achieve compliance with privacy regulations and in particular with GDPR
- Identified stakes to achieve PDPbD along different phases of the method and the envisaged techniques and tool support to tackle them

1 Introduction

1.1 Objective of the document

This document provides a first overall description of the method proposed to achieve privacy and data protection by design (PDPbD). The method aims to provide guidance for the design activities usually conducted by an engineer to design a target system. The method is composed by several phases which are detailed in this document.

1.2 Structure of the document

To achieve the main objective, the document is structured as follows. In Section 2, selected references are considered as a context and basis to achieve PDPbD. The Section includes known standards, methods and techniques which have been previously proposed for privacy by design. In Section 3, the method proposed in PDP4E for PDPbD is introduced. The phases of the method are illustrated all along the different subsections. A summary of the document and some perspectives for method evolution are finally given in Section 4.

1.3 Relation with other deliverables

The method proposed in this deliverable aims to provide guidance to engineers so as to achieve PDPbD. The PDPbD framework developed to support the method is specified in deliverable D5.1. Both, the method and tool for PDPbD are developed within WP5. The design activities are interdependent and often guided by other elements like requirements, elicited to comply with regulations (e.g., GDPR), and also privacy countermeasures (e.g., PETs) designed to manage and reduce unacceptable risks. Consequently, the method specified in this document should be interfaced and harmonized with the methods and tools developed in WP3 (Privacy Risks) and WP4 (Privacy Requirements). In addition, it is foreseen that some of the design activities shall provide the elements necessary for the privacy assurance process which is specified and implemented in WP6.

2 PDP4E background to achieve PDP by Design

2.1 Reference standards and regulations

2.1.1 General Data Protection Regulation

The GDPR introduces the notion of '*data protection by design and by default*' in the legal framework of the European Union, suggesting that the protection of personal data should be considered from the conception of data processing systems. *Data protection by design and by default* is mostly described in Article 25, which details the obligation to integrate the necessary safeguards into data processing activities in order to meet the requirements of the regulation taking into account the state of the art, purposes of processing and risks of varying likelihood. This effort shall be conducted both at a design stage ("*time of the determination of the means for processing*") and at operationalization of the data processing system ("*and at the time of the processing itself*").

This article highlights the diversity of constraints on the architecture and design of data processing systems:

1. Some constraints shall be derived from high-level privacy goals which are depicted on Article 5. In particular, the regulation suggests organizations to: be transparent with respect to the data processing; limit secondary uses of data; collect data only if it is necessary for the processing purpose; keep data accurate and updated; remove data as soon as it is not necessary; and ensure confidentiality of the data subject.
2. High-level functional and non-functional requirements described by the GDPR and other domain-specific regulations, directives and code of conduct. When it comes to GDPR, all articles related to data subject's rights (Articles 15 to 22) describe both functional requirements of data processing systems (e.g. the user shall be able to revoke consent) and non-functional requirements (e.g. data controller has 30 days to remove all personal data). In deliverables D4.1 [3] and D4.4 [4], more details are provided on how to concretize such high-level requirements.
3. Finally, the result of a risk management process (see Article 35) may suggest modifications to the architecture and/or design of the data processing system. We refer the reader to deliverables D3.1 [1] and D3.4 [2] for more information on the definition of privacy controls based on a risk management process.

Besides those design constraints set by the *data protection by design and by default*, a couple of articles also ask organizations to be able to demonstrate that this approach has been considered: Article 5 (2) asks for accountability with respect to the high-level privacy goals; Article 30 forces organizations to have a detailed description of the data processing activities; and Article 35 regulates the obligation to validate the results of a risk management process with external third parties (e.g. supervisory authorities).

Nonetheless, the regulation does not detail how to systematically apply all these constraints on system design processes nor the level of details that are necessary to demonstrate the accountability principles described in Articles 5, 30 and 35.

2.1.2 Technical international standards

2.1.2.1 ISO 29100 - Privacy framework

The ISO/IEC 29100 [14] provides a privacy framework which specifies a common privacy terminology, actors and stakeholders to make future privacy-related standards consistent. The

document is freely available on the Information Technology Task Force web site[1] since 2011. Hence, ISO/IEC 29100 has been an important background for the creation of privacy and data protection regulations such as GDPR and one may consider this document as a reference to better understand such legal documents. More interestingly, the standard describes a set of privacy principles (see Table 1) and provides high-level best practices to achieve those objectives.

- **Consent and choice.** Presenting to data subjects the choice of whether or not allow the processing of their personal data. Data subjects' choice must be given freely, specific and on a knowledgeable basis.
- **Purpose legitimacy and specification.** Ensuring that data processing purposes comply with applicable laws and have been communicated to the data subject prior to the processing.
- **Collection limitation.** Limiting the collection of personal data to which is within the bounds of strictly necessary for the specified purpose.
- **Data minimization.** Minimizing the personal data which is processed and the number of stakeholders that can access it.
- **Use, retention and disclosure limitation.** Limiting the use, retention and disclosure (including transfer) of personal data to that which is necessary in order to fulfil the processing purpose.
- **Accuracy and quality.** Ensuring that personal data is accurate, complete and up-to-date, adequate and relevant for the processing purpose.
- **Openness, transparency and notice.** Providing data subjects with clear and easily accessible information about privacy policies, including purpose of data processing and stakeholders with access to the data.
- **Individual participation and access.** Giving data subjects the ability to access, correct and remove their personal data.
- **Accountability.** Documenting and communicating all privacy-related policies, procedures and practices.
- **Information Security.** Protecting personal data with appropriate controls to ensure the integrity, confidentiality and availability of the data.
- **Privacy Compliance.** Verifying and demonstrating that the processing meets privacy and data protections requirements and/or regulations.

Table 1. Privacy principles described in ISO 29100

2.1.2.2 ISO 27550 - Privacy engineering

The 27550 ISO standard [13] is a direct answer to doubts posed by ENISA on the applicability of the data protection by design approach [15], providing best practices on integrating data protection activities into different engineering disciplines. As shown in Figure 1, this integration includes risk management, requirements specification system architecture and design.

[1] https://standards.iso.org/ittf/PubliclyAvailableStandards/index.html

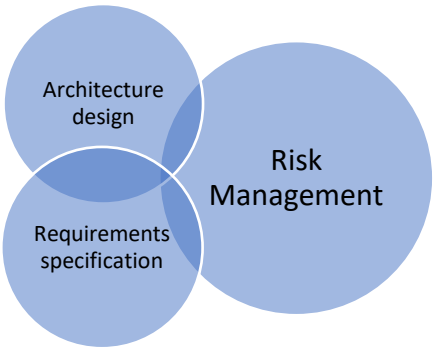

Figure 1. Activities related to data protection by design as proposed in ISO 27550

When it comes to data protection by design, this standard reflects the complexity described by the general data protection regulation. In particular, this standard explicitly suggests having an integrated risk management process within the system specification, architecture definition and design processes.

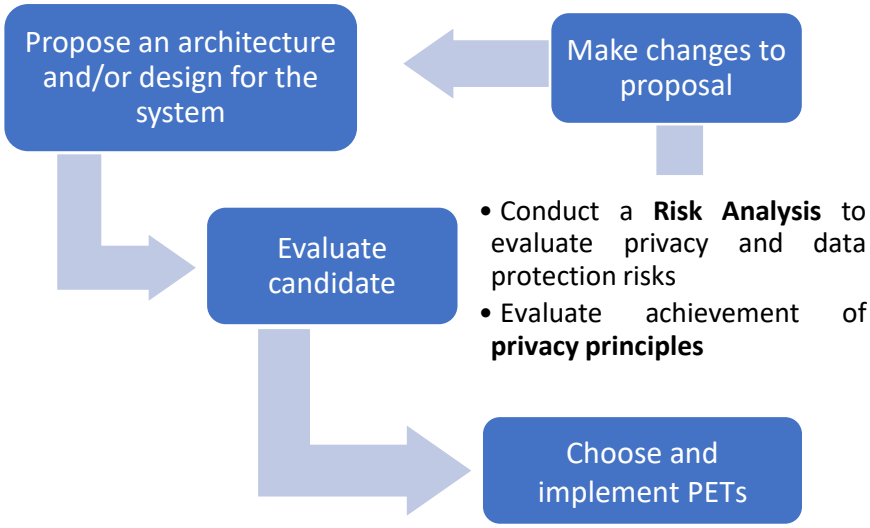

Figure 2. Integration of risks management into architecture and design processes as proposed in ISO 27550.

As depicted in Figure 2, the objective of integrating risk management in both architecture and design processes is to evaluate the a candidate. The result of such risk analysis might suggest changes to the architecture and/or design, creating a new proposal that needs to be reassessed. Criteria for accepting a candidate needs to be defined by the organization and may depend on the purpose of the processing and the sensitivity of data. Article 35 states that, in some cases, such evaluation needs to be consulted with third parties (e.g. supervisory authorities). Finally, when the level of privacy and data protection is acceptable, the development team may start the implementation of the different Privacy-Enhancing Techniques (PETs) outlined in the architecture.

The ISO 27550 standard does not only evaluate candidates based on the results of a risk analysis, but it is also suggested that one needs to take into consideration achievement of privacy requirements and/or principles. When it comes to the privacy principles stated by Article 5 of the GDPR, the literature of privacy engineering has extensively discussed the usage of the high-level strategies depicted in Table 2.

Data-oriented strategies	
Minimize	*The amount of personal data that is processed should be restricted to the minimal amount possible*
Separate	*Personal data should be processed in a distributed fashion, in separate compartments whenever possible*
Abstract	*Personal data should be processed at the highest level of aggregation (abstraction) and with the least possible detail in which it is still useful*
Hide	*Any personal data, and their interrelationships, should be hidden from plain view*
Process-oriented strategies	
Inform	*Data subjects should be adequately informed whenever personal data is processed*
Control	*Data subjects should be provided agency over the processing of their personal data*
Enforce	*A privacy policy compatible with legal requirements should be in place and should be enforces*
Demonstrate	*Be able to demonstrate compliance with the privacy policy and any applicable legal requirements*

Table 2. A collection of privacy patterns as presented in [65]

2.2 Engineering methods for PDP by design

2.2.1 PRIPARE: for iterative design

PRIPARE's methodology (PReparing Industry to Privacy-by-design by supporting its Application in REsearch) [16] defines a series of processes that address several privacy engineering practices related to different software and systems development disciplines, together with a conceptual reference model, description of roles involved, examples, application guidelines, etc. These processes are organized around the disciplines of the Systems Development Life Cycle (SDLC) [17] so as to ensure that the methodology can be easily integrated into mainstream development practice. In particular, PRIPARE methodology encompasses 7 phases, viz. environment and infrastructure, analysis, design, implementation, verification, release, maintenance and decommissioning. In the context of PDP4E, and in particular in the scope of WP5, the most relevant PRIPARE processes (within the Analysis and Design phases[2]) are detailed next, together with our analysis of their relevance for GDPR.

Detailed Privacy Analysis creates an inventory of all the privacy-relevant elements of a system or service, its environment and its constraints: stakeholders, sub-systems, personal data, etc. In particular, it does so by leveraging Data Flow Diagrams (DFD), a visual modelling notation which captures how data flows from users (modelled as 'entities' in DFD parlance), through the different processes it goes under within a system, to and from data stores, and to external stakeholders (modelled as 'entities' as well). In the case of PRIPARE, these DFDs are enriched with further concepts, relevant to privacy and personal data protection: stakeholders, (sub-)systems, domains, roles and responsibilities, touch points, and privacy constraints. This DFD model is just an option that is not directly required by GDPR, but it helps comply with the

[2] op. cit. sections 6.2.4 , 6.2.5, 6.2.6, 6.3.1, 6.3.2.

obligations it establishes to record the data processing activities and the measures and safeguards taken. Section 3.5 will present an example of such DFD model enriched with privacy information.

Once that privacy-enriched DFD has been created, PRIPARE's Detailed Privacy Analysis process also involves identifying personal data in each privacy domains and system, and specifying privacy and security controls required, associated with personal data.

Operationalization of Privacy Principles maps high-level, abstract, legal privacy principles onto specific system technical requirements, with the support of a catalogue of project-independent, privacy (meta-)requirements, which are organized along two dimensions: a tree of successively refined requirements and a prioritized structure [18].

1. The first step in this process consists in specifying *which* definition of privacy we are using, according to the legal, regulatory and privacy theory which the project at hand is aiming to stick to (as introduced in 2.1 above) In the case of PDP4E, such definition is encompassed in GDPR by the principles for data processing established in Art. 5 (lawfulness, fairness and transparency, purpose limitation, data minimisation, accuracy, storage limitation, integrity and confidentiality, and accountability) and detailed throughout the rest of its Chapter 2; together with the rights of the data subject in Chapter 3, and the obligations for controllers and processors in Chapter 4. Likewise, standards such as ISO 29100 are also organized around a set of principles (which may be different but are still structured as a set of privacy conceptual units).
2. The second step consists in selecting the specific privacy requirements that apply to a given project, depending on the applicability of the conformance criteria to the specific project at hand (and the level of compliance targeted by the project, if such concept is relevant in the given framework). In the case of GDPR (or ISO 29100, for what it's worth) we do encounter that there are some applicability constraints that act like switches with respect to requirements, i.e. they may activate or deactivate the need to comply with some specific requirements (e.g. Art. 30 exempts small organizations from keep records of processing operations). These generic, selectable and instantiable requirements can be assimilated to the meta-requirements defined in D4.4 [4].
3. The next step, even if not made explicit by PRIPARE, consists in instantiating the requirements, by particularizing them to the specifics of the project at hand, binding their open parameters to the corresponding project elements.

Risk Management aims at identifying the privacy risks associated to a system, plus the measures or treatments required to address those risks. This process follows the traditional risk management steps; thus we'll not go into details, as the related activities in PDP4E are addressed by WP3 and will are detailed in D3.1 [1] and D3.4 [2]. Nonetheless, we shall emphasize that PRIPARE's Risk Management process comes to complement the Operationalization of Privacy Principles by providing a risk-oriented approach that acts as the respective counterpart of the goal-driven requirements elicitation provided through the operationalization. Both transform high-level privacy principles into something that can be actually incorporated in a development process, and both allow eliciting a set of privacy controls required, which act as a bridge between the problem domain (requirements and risks) and the solution domain (design).

Privacy enhancing architecture design produces a high-level system decomposition which responds to functional and business requirements as well as to privacy and security requirements. PRIPARE recommends (but not imposes) dealing first with the detailed privacy design and then the architectural design (even if the intuition might dictate proceeding otherwise). The architecture would be documented according to the OASIS PbD-SE specification [20],[19], including a Services Layer, an Integration Layer, and a layer of Composite Privacy Processes.

PRIPARE describes three alternative approaches to design the privacy enhancing architectures:

- ***Iterative approach*** [21], based on CMU's Architectural Significant Requirements or ASRs [22].
- ***Top-down formal approach*****,** based on CAPRIV [23], where requirements and architectures are expressed according to a formal mathematic language called pi-calculus.
- ***Bottom-up approach*****,** similar to the previous, but departing from an already defined architecture which has been created by a designer, possibly attending other constraints, and then analysed against the set of requirements.

All in all, they represent offer different modelling approaches on top of which automated analysis and/or transforms can be carried out to ensure compliance with predefined privacy and data protection requirements —PDP4E will choose among such approaches to implement some of the strategies defined in 3.4 and 3.5.

Privacy-Enhancing Detailed Design produces detailed descriptions of system components, interfaces, relations, data and data flows by "*reus[ing] design solutions that guide the design with proven recipes based on previous experience and knowledge, which reduce uncertainty and cost in the design*" [16]. These solutions are called "techniques" in PRIPARE methodology parlance, but they fit the concept which is usually called "privacy patterns" elsewhere. (Other related terms are 'heuristics', 'mechanisms' and even, in some catalogues 'controls'.) This process in PRIPARE can be considered a continuation of the *Privacy Requirements Operationalization*, and it consists of similar steps, but applied to the Design discipline: 1) choose the catalogue of patterns (implicit in PRIPARE), 2) select the most suitable patterns, depending on the specifics of the system in relation to the context of application of the pattern and taking into account any other constraints (expertise, cost, relation to other requirements, etc.), and 3) instantiate the patterns.

A pattern is a well-established solution to a recurrent problem in a given context which can be applied in different projects, which implies that: it is not prescriptive but descriptive (other solutions may exist), it is not universal but context-dependent (it cannot be applied just anywhere), and it is not bound to a given technology but abstract (their implementations need tailoring and instantiation). This pattern-based approach has been much applied to privacy and data protection beyond PRIPARE. In particular, an international initiative called privacypatterns.org [24] is compiling a system of privacy patterns to foster its usage. Some of the privacypatterns.org contributors are trying to advance the status of the repository to that of a privacy pattern language (a cohesive and exhaustive set of interrelated patterns addressing the domain of privacy and to be used in conjunction), which can be taken advantage for the systematic selection of appropriate combination of patterns in a given project. Some other participants in the same initiative have also created external tools for patterns selection depending on precisely defined pattern characteristics and their relation to normative frameworks (e.g. GDPR) or standards (e.g. ISO 29100). A more interesting avenue also under consideration would consist in defining some privacy patterns as model fragments (in terms of an existing metamodel e.g. UML), so that pattern users could easily introduce the patterns into their models (in a similar way as how they may be used to introducing patterns in Object-Oriented Design).

2.2.2 LINDDUN: design guided by risks

LINDDUN[3] is a privacy threat analysis methodology that integrates 7 main privacy threat categories [73]: Linkability, Identifiability, Non-repudiation, Detectability, Disclosure of

[3] LINDDUN privacy threats modelling methodology, Available at: https://linddun.org/linddun.php# Last accessed on 17 April 2019.

information, Unawareness, Non-compliance. As illustrated in Figure 3, LINDDUN methodology steps are divided in problem space steps (step 1-3), which aim at describing privacy threats, and in solution space steps (step 4-6) necessary for the elicitation of mitigation measures and solutions corresponding to the threats identified.

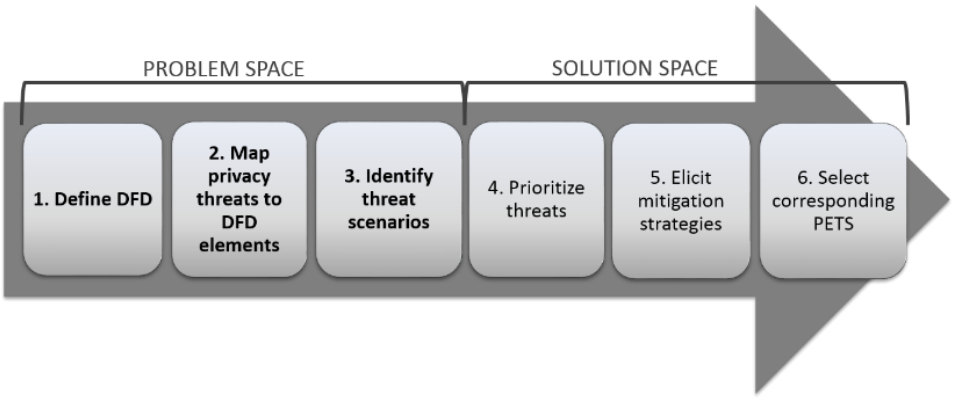

Figure 3. The LINDDUN methodology steps

The PDP4E Risk Management tool will take LINDDUN as the starting point for risk analysis, as well as STRIDE [63] to cover for those risks related to security that may affect privacy also. STRIDE is a security-oriented framework that classified security threats in 6 categories: Spoofing identity, Tampering, Repudiation, Information disclosure, Denial of service and Elevation of privilege. In fact, LINDDUN establishes a link between privacy threats and the security threads defined in STRIDE. The methodology used for risk analysis is inspired by ISO 31000 [7], Coras [64]] (being part of methodology used in the MUSA Risk Assessment tool) and ISO 29134 [8].

As we describe in deliverable D3.4 [2], LINDDUN was proposed before GDPR and it may not fully cover all the aspects considered in the regulation. Because of this, we may need to extend the definition of LINDDUN. This is ongoing work in WP3.

However, with respect to the design phase, LINDDUN, and in general the methodology used by the WP3 Risk Management tool, presents several aspects to consider:

- Risk analysis should be initiated at early stages of the design process, in parallel or replacing requirements definition, depending on the internal culture of the organization using PDP4E tools, i.e. if the organization follows a goal-oriented approach or a risk-oriented approach for designing the application.
- Our risk analysis tool takes as input models created previously by other tools in PDP4E. Namely, it needs to consume at least:
    - *Data stores information*, including the outcomes of the Personal data detector developed in WP5.
    - *Data Flow Diagrams* (DFD), as defined in WP5.
- The risks analysis tool will generate a set of mitigation actions or controls. There may be different types of controls, but in particular we will focus on those that affect engineers and impact the design process. For instance, the implementation of a particular PET may be one of these controls. These mitigation actions may also be consumed by the tool for requirements definition.

In Figure 4, we show an example of the type of mitigation strategies proposed in LINDDUN. In particular, we show an example related to the protection of IDs, which is related to the identifiability and linkability of entities in a system.

Mitigation Strategy		Privacy Enhancing Techniques (PETs)
Protect ID	Pseudonyms	Privacy enhancing identity management system [66], User-controlled identity management system [67]
	Attributes	Privacy preserving biometrics [68], Private authentication [69][70]
	Properties	Anonymous credentials (single show [71], multishow [72])

Figure 4. Example of mitigation actions proposed in LINDDUN.

There is an important aspect to consider: mitigation actions are materialized as pointers to PETs. This should be taken into consideration at design time and broken down into features for development by an architect of the application. Note also that, in their current definition in LINDDUN, this is just described through academic papers, but not through other pointers that may help engineers to implement these PETs (e.g. existing open-source implementations, practical recommendations to embed these PETs in the architecture or a system, etc.). Besides, some of the PETs proposed, may only apply to specific systems with specific scenarios, but may not be easy to generalize to any system. Finally, LINDDUN authors claim the list of mitigation actions not to be complete and to be only illustrative as an example. While enabling the use of these PETs is not explicitly in the scope of PDP4E, we are now discussing how to approach this.

WP3 Risk Management tool will allow users to generate controls that can be directly translated into features in the software development process. The risk management process that we consider in WP3 is continuous. This means that iterations over the risk management process may be frequent. In fact, changes in the risk plan should be reviewed at the application design level, and changes in the design would require new risk assessments. At this stage, we envision that there should exist a mechanism to notify changes in the design or in the risk evaluation plan, so that this can be taken into consideration by other related tools or the Risk Management tool.

2.3 Minimisation-related techniques to meet PDP constraints

Data minimization, which can be defined as the collection of as little personal data as strictly necessary for a given purpose, is deemed a keystone which can be regarded as both a principle and a strategy to ensure privacy and data protection.[4] However, as pointed out by Gürses and Troncoso [25], such concept is interpreted in practice with quite diverse meanings, which are sometimes more and sometimes less aligned to the contents of the definition we have just sketched. Likewise, different perspectives of data minimization are sometimes rendered or subsumed under other concepts, such as data protection by default, collection limitation, storage limitation, use, retention and disclosure limitation, purpose limitation, purpose specification, user data protection, or even pseudonimity, anonymity, unlinkability, unobservability, undetectability, etc. Inspired by the said work by Gürses and Troncoso, we have

[4] Some authors distinguish “data protection” from “data minimization” in that the former would focus on protecting already collected personal data from improper access, while the later would directly avoid the existence of such data whatsoever since the beginning, so that it would not merit to be protected anymore. We do not make that distinction, as we consider ‘data protection’ as the subject matter of GDPR, which explicitly includes several ways of minimization.

analysed in depth some influential sources which deal with concepts related to data minimization. In particular, we have considered the following sources, due to their relevance in the field (which can be ascertained by their high number of citations and their eventual progress to standardization):

- the said paper by Gürses and Troncoso [25] —even if not highly cited yet due to their novelty, nonetheless their authors are authorities in the field—, as it explicitly addresses several data minimization perspectives, providing the analysis from which we departed for this classification;
- Hoepman's privacy strategies and tactics as described in different versions in [26][27][28], which are now also included in ISO 27550 [13], employed in section 3 of this document to delineate the design analysis and transformation approaches;
- Pfitzmann and Hansen's white paper on minimization-related terminology [29], later presented to IETF for standardization [30], which settled the conceptual foundations in this area, clarifying the scope of many terms which used to be confused with one another;
- RFC 6973 [31] which was eventually created by the synthesis of the document above with other proposals, and which expands to more terms;
- ISO 29100 privacy principles [7], which provided a modern view of operational privacy criteria, also used in the context of PDP4E WP4;
- Common Criteria version 3.1 [32] as a live evolution of ISO 15408, in what regards to privacy and data protection, since this standard is the measuring rod for security certification (even though it is not so used in the PDP realm, it still covers some privacy and minimisation aspects);
- the GDPR itself, as the reference legal framework for data protection in the EU;
- and Cavoukian's seminal definition of Privacy by Design principles [33] which made the concept of PbD known, and their operationalization [18].

We have analysed the scope of such concepts as presented in those sources, and synthesized different dimensions along which personal data can be 'minimized' into a bunch of questions, all which target different aspects of data minimization, and each of which can be answered independently. Here we introduce those dimensions, and mention some of the terms under which they have been addressed, with special emphasis on the privacy strategies and tactics (see section 3) that address each dimension.

- **HOW MANY? - *Minimal amount of personal data collected, stored, used or processed.*** Probably the most widespread definition of data minimization is the one which focuses on collecting the minimum possible amount of personal data necessary to fulfil a given purpose. When it focuses on the collection stage, it is usually rendered as 'Collection Limitation'. This dimension is addressed by the *Select* and *Exclude* tactics of the Minimise strategy. Although this is often presented as an atomic concept, even it can be decomposed into different subdimensions:
    - **WHAT? - *Minimal quantity of personal data attributes*** (collected, stored, used or processed)
    - **WHICH? - *Minimal number of data records or samples*** for a given attribute.
    - **WHEN? - *Minimal number of transactions*** that require the collection of a given attribute.
    - **HOW OFTEN? - *Minimal frequency of data capture*** or collection of a given attribute.
    - **WHOSE? - *Minimal number of individuals*** about whom data is collected, used or processed.

- **HOW LONG? - *Minimal amount of time that data is retained.*** When this dimension is tackled in isolation, it is usually presented as 'retention limitation' or 'storage limitation'. It is addressed by the *Strip* and *Destroy* tactics of the *Minimise* strategy.
- **HOW? - *Minimal processing activities to which the data is subject.*** This dimension is seldom considered explicitly, except by legal frameworks which deal with processing considerations.
- **WHERE? - *Minimal number of entities where data is stored and processed.*** Sometimes named as 'Minimize Replication'. This dimension, as well as the next two, are addressed by the *Isolate* and *Distribute* tactics of the *Separate* strategy (encompassing different approaches such link avoidance, table splitting, local storage or processing, etc.)
- **HOW TOGETHER? - *Minimal quantity of data stored in a single entity.*** Also called 'Minimize Centralization', it is the counterpart of the previous dimension, as it's not enough to ensure that a data item is available in as little entities as possible, but it's also necessary that a given entity doesn't amass by itself an uncontrolled amount of data.
- **TO WHOM? - *Minimal amount of data flowing or transferred to minimum number of entities.*** This dimension encompasses the access by unauthorized agents, the transfer to authorized third parties, and the access by authorized individuals (e.g. organization staff). Hence it's related to many privacy functions and attributes, which range from traditional information and communications security functions (e.g. confidentiality, access control, physical protection, etc.), to data- or process-oriented controls (pseudonymization, convertibility, undetectability, unobservability), to policy enforcement (use limitation, flow control, data transfer limitation, etc.). This dimension is addressed by the *Restrict* and *Obfuscate* tactics of the *Hide* strategy (together as the already mentioned *Isolate* and *Distribute*).
- **WHAT ELSE? - *Minimal amount of potential derived or inferred data.*** This is related to the potential creation of new personal data by derivation or inference from the original data. It's related to properties which difficult such derivation or inference (pseudonymity, partial identities), or make it virtually impossible (anonymity, unlinkability), or even introduce noise which would make new personal data be flawed (misinformation, disinformation). This dimension is addressed by the *Mix* and *Dissociate* tactics of the *Hide* strategy, and the *Summarize*, *Group* and *Perturb* tactics of the *Abstract*/*Aggregate* strategy.
- **WHAT FOR? - *Minimum quantity of purposes for which data is used.*** This is often presented by legal frameworks and standards as 'Purpose Limitation' or 'Purpose specification'. Neither this nor any of the following dimensions is addressed by any privacy strategy/tactic.
- **HOW SENSITIVE? - *Minimum sensitivity of the data collected, stored or processed.*** Between two approaches which process the same amount of data, the choice should be that which processes the least sensitive one.
- ***WHETHER* - *Minimum possibility of data collection* -** The strictest interpretation of data minimization would consist in not only limiting the collection of personal data, but also preventing beforehand the very possibility of such collection.

All these are indeed proxies for two other facets, which capture all of them and where the ultimate motivation for the different data minimization dimensions:

- **HOW BAD - *Minimum risk likelihood and impact to data subjects.*** - Understood as reducing the unexpected, unwanted, negative consequences for the data subjects.
- **HOW GOOD - *Minimum trust that needs to be placed into an entity to have a service provided*** - Understood as having guarantees that no harms to data subjects rights and freedoms may happen, even if there are no guarantees that the entities processing personal data are themselves trustworthy.

It should be noted that those dimensions can be sometimes related, but they are also often independent from one another (e.g. we can achieve a minimum number of entities holding personal data, but this doesn't say nothing about e.g. the frequency when data is collected). Thus, each dimension may require the application of different techniques, as already hinted in the privacy strategies and tactics): as we will explain in section 3, PDP4E aims to apply several strategies and tactics to address different dimensions, respectively. However, not all of them are always appropriate (or even the simultaneous application of some of them can render a contradiction). Thus, a choice will be made on the implementation of a subset of them, prioritized depending on the needs posed by the demonstration scenarios and the feasibility of their implementation.

3 PDP4E method for PDP by design

According to the PDP4E work plan, we have logically separated the design activities from others like those related to requirements management and risks analysis. However, we are aware that within a typical engineering process, like in the Systems Development Life Cycle (SDLC), those activities are indeed interdependent and iterations between design, requirements and risks engineering tasks usually occur. Moreover, nowadays, it is generally observed that the systems design can be oriented either by goals or by risks [74].

The design guided by goals typically targets the fulfilment of requirements which are elicited to achieve functional objectives/needs and also constraints. The design guided by risks typically targets the elicitation of costly-acceptable and technically-effective countermeasures which reduce impact of risks to acceptable levels. In a first approach, the method proposed in PDP4E for PDPbD harmonizes both perspectives by defining a framework that considers and integrates the elements related to requirements engineering and risks analyses –as developed in the scope of PDP4E. However, in this first iteration, the integration shall be partial and considering a minimal subset of interfaces. The definition and implementation of the PDP4E architecture also follow an iterative cycle which is consistent with the work plan.

The resulting methodological support is aligned with an iterative design process.

To achieve integration, we have selected three views found in the MDE ecosystem[5] considering that privacy-related concerns need to be modelled and analysed. The first view aims to capture data structures at different levels of abstractions (data-oriented model). The second view aims to capture the processes in which data and personal/sensitive data are involved (process-oriented model). The third view is related to a functional architecture supporting both the data processes and data structures (architecture model). More details on referred views are respectively found in subsections 3.4, 3.5, and 3.8. Of course, the traceability and consistency between the three views are main design needs and shall be ensured. The usage and adoption of Model Driven Engineering (MDE) approaches and languages is meant to facilitate that objective. The Figure 5 shows an overview of the PDPbD method including the information flows between phases. The method is intended to provide guidance to an engineer during the development of the three design views. The method phases are further explained in this Section 3: each subsection corresponds to a phase.

[5] The Object Management Group. In https://www.omg.org/

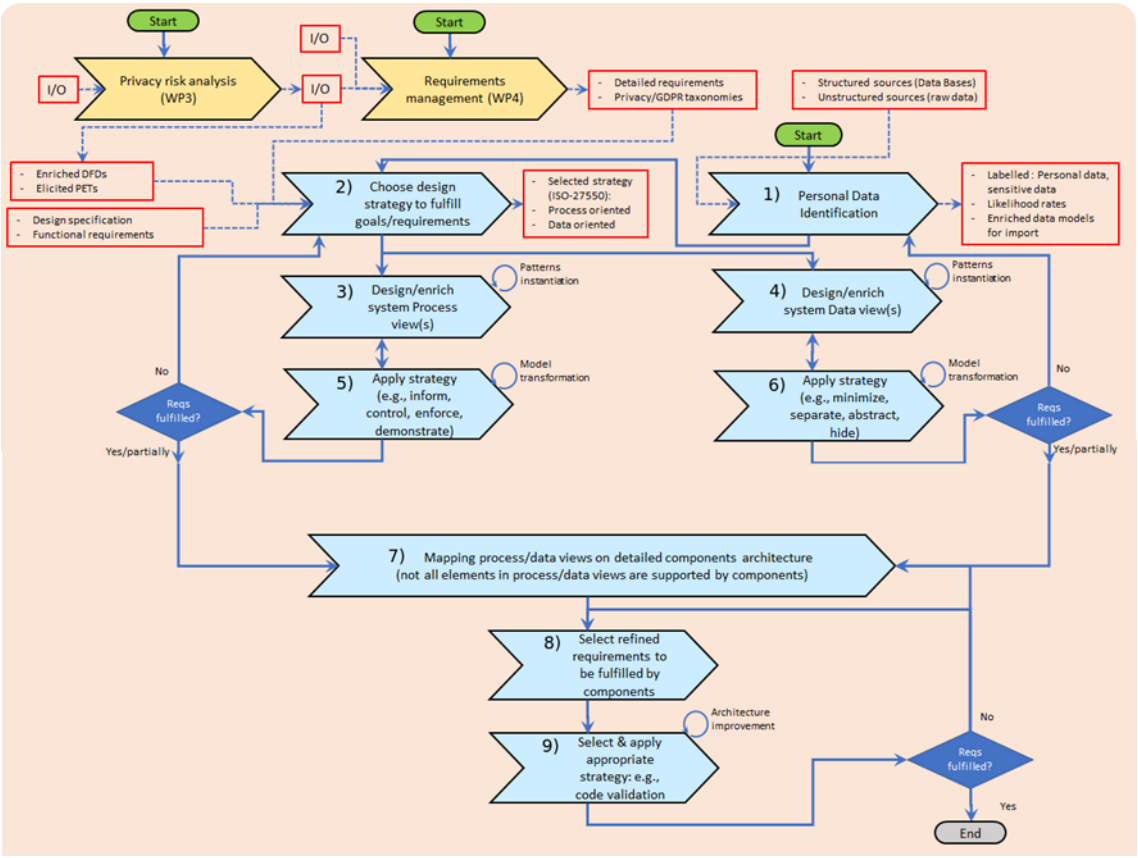

Figure 5. Method proposed in PDP4E for Privacy and Data Protection by Design

The design of each view roughly follows the following pattern. A first view is proposed including the set of requirements to be fulfilled (phases 1, 2 and 8). Some privacy-related strategies and techniques are selected in order to fulfil the requirements (phases 2 and 9). The view is then enriched by the design engineer in the aim of meeting requirements (phases 3, 4 and 7). The privacy-related strategies and techniques are then applied (phases 5, 6 and 9). If, after validation, the requirements are satisfied then design models can be refined (phase 7). The pattern is repeated for each fine-grained or detailed model.

3.1 Overall design assumptions

The following items introduce some hypotheses (H) to be considered for the PDPbD method to be applied:

H1. A data protection risks management process can be conducted in preparation and prior to the design process. State-of-the-art methods like LINDDUN [9] can be used for that purpose. Outcomes from the privacy risks analysis in WP3, such as assets, risks assessments as well as the elicited countermeasures to manage risks, are potential elements to consider during the design process.

H2. To consider the specificities introduced by regulations like GPDR [10] as well as privacy oriented methods in the state of the art like ProPan [11], [12], a requirements engineering process oriented to data protection can be conducted prior to the design process. The elicited requirements are potential outcomes to guide the design activities.

H3. The data protection design approach proposed in PDP4E is guided by Model Driven Engineering (MDE) techniques [34], [35], [36], [37]. The main goal is to leverage MDE techniques so as to support non-savvy engineers in addressing privacy and GDPR related concerns. An engineer can start a first design model and analysis targeting data protection with no prior engineering process.

H4. The validation of requirements fulfilment is assumed part of the design process. Thus, it is foreseen that design elements and outcomes can be considered as evidence to be used in the assurance process where data protection is finally ratified.

H5. According to previous assumptions, the method for PDPbD can be iterative, entangled with other engineering activities, methods and tools and in particular with those conducted and developed in WP3, WP4 and WP6. However, the method can be also applied standalone and the design tool and modules will adopt the same principle.

In the following subsections, the different phases of the PDPbD method are described (one phase per subsection). The reader is invited to consider that some phases extend existing approaches or propose novelties according to the state of the art. Since this is the first iteration of this work, the specification often includes perspectives to address identified concerns and issues. Moreover, the method for PDPbD will be further detailed according to the evolution of other methods (and tools) developed in the scope of PDP4E. In particular, the consistency and interoperability with risks analysis (WP3) and requirements engineering (WP4) methods and tools are to be ensured.

3.2 Personal data identification

With the pervasive use of the Internet and a growing number of computers and many other types of devices, it has become difficult for organizations to locate and effectively manage personal data. IT professionals must understand the need for personal data discovery to protect themselves and their company from the civil, legal and financial liabilities caused by violating data subjects' rights due to the inadequate protection of personal data.
The main objective of the Personal data identification tool is to identify and elicit categories of personal information that are stored by data controllers (or data processors) in their data warehouses. The tool will be composed of two different components: a personal data detector that will use techniques to scan and identify personal data and a support module to help users reflect on issues related to the potential linkability between data sources in the system and other external data sources that may put data subject privacy in risk.
While the first part has been already developed in different scientific papers and commercial tools, as we will show in the state of the art subsection, the second part is novel in the sense that it is the first time a tool will try to use open-data knowledge graphs such as WikiData or DBpedia to help users in the reflection of external entities or data that may be linked to the data in the system under development in a way that it may violate data subject rights. In the following subsection, we present a brief summary of the state of the art.

3.2.1 State of the art in personal data identification

Determining a sanitization strategy which guarantees that the data provided preserve confidentiality is a difficult task. Some initial work is focused in the sanitization of free text, mainly in the medical domain [49], [54], [57]. The challenge tackled in this previous work consists in general in identifying sensitive words based on a specialized domain semantics. They do not consider any links between terms except potentially synonymy. An exception to this, for Health information, is presented in [51], proposing a prototype for extracting information and identifying entities.

Geng et al [52] address the problem of predicting the presence of private information in e-mail using data mining and text mining methods. Korba et al [53] focus on automatically identifying private data in semi-structured and unstructured (free text) documents. In particular, the first part of the process involves identifying Personal Identifiable Information (PII) via named entity recognition.
Du Mouza et al. in [50] propose a technique that automates the detection of sensitive attributes. Their motivation is the increasing need for outsourcing application testing with realistic data. They propose a rule-based approach implemented on top of an expert system architecture. Their technique relies on two functionalities: (1) Automatic detection of the values to be scrambled; (2) Automatic propagation to other semantically linked values. Aura et al [48] also propose a system to scan electronic documents for PII, using regular expressions and other techniques.
The use of Knowledge Graphs to support concerns about trust, privacy and transparency, is proposed in the form of a framework in [56]. In [55], authors study how semantic web vocabularies can be used to express the provenance information required for managing GDPR compliance. However, they do not use graph-based databases such as WikiData or DBpedia to find additional data sources beyond those in the system.
There are also some products in the market that leverage two decades of research work and commercialize solutions to scan PII. Examples of these are CA Data Content Discovery[6] or CA Test Data Manager[7]. There are also several patents in this field. For instance, "Personally identifiable information detection" (USPTO application number: US8561185B1)[8].
In general, previous work and existing products focus on detecting PII information in structured or unstructured databases. However, to our knowledge, none of these pieces of work seeks to facilitate the evaluation of the potential risks of certain data to be linked to external sources, i.e external sources with data about individuals that can be linked to the data in our system, jeopardizing data subject rights.

3.2.2 PDP4E personal data identification approach

As stated at the beginning of this Section 3.2, the PDP4E Personal data identification tool is created to identify categories of personal information that are stored by controllers or processors in their data warehouses. This tool may be important in different contexts, including:

- Companies using legacy software that may not control personal data stored in the related data stores.
- Companies outsourcing software testing with actual data or other activities that require sharing data that may contain personal data.
- In general, controllers that require a greater control over data to be compliant with GDPR and other regulations or best practices.

In particular, GDPR requires a strict control on personal data. One of the big challenges when it comes to control personal data is to identify these data. According to Art. 41 of GDPR, personal data comprehends any information relating to an identified or identifiable natural person ('data subject'); an identifiable natural person is one who can be identified, directly or indirectly, in particular by reference to an identifier such as a name, an identification number, location data, an online identifier or to one or more factors specific to the physical, physiological, genetic, mental, economic, cultural or social identity of that natural person.

[6] https://www.ca.com/us/products/ca-data-content-discovery.html

[7] https://docops.ca.com/ca-test-data-manager/4-7/en/create-a-data-model-and-audit-pii-data/the-data-model-in-ca-tdm-portal/scan-data-model-for-pii

8 https://patents.google.com/patent/US8561185B1/en

Our tool will support the detection of critical data items that should be classified as Personal Identifiable Information. We will not only provide already known mechanisms for PII detection, but we will also support the user to reflect on how data in the system that may not be classified as PII initially could be linked to external data that may jeopardize the preservation of data subject rights.

Figure 6 shows a general overview of the steps followed by the Personal data identification tool. The tool consists of three main steps:

- **Automatic detection of sensitive information**: this step includes the basic functionality of the tool. This includes methods to detect PII in the structure data stores of the system under analysis. These methods will automatically scan actual SQL databases to detect potential personal data. In order to implement this step, we will use algorithms similar to those used in [50]. Input data may be imported from an existing model generated by other tools in the context of PDP4E or it may be automatically obtained from a live connection to the actual database. Actual information scanned as the input for the tool is described in D5.1, where we describe the architecture of the component.

- **Enrichment of analysis via open data sources**: one of the main challenges, when labelling the data in a system as personal data is that data might be misclassified as not being personal data. In some cases, the technology and/or external data necessary to link data sources to individuals is so difficult to obtain that organizations underestimate the risk of disclosing such data. For instance, the address of an office may not be personal data, except if someone is able to identify that an individual works there. Another example, that may be relevant for the PDP4E automotive use case, is the position and direction of a vehicle. Privacy-unsavvy engineers would misclassify this information as anonymous data, as there is no direct link to the driver identity. However, there are plausible chances that an attacker is able to link the vehicle with the driver or the passengers travelling in it. In such case, the position and the direction become personal data that may affect not only privacy but even the safety of the people in that vehicle. Our tool will be able to extract keywords from the database elements (table names, attributes names, tags/keywords from the user) and explore potential individuals that can be some related to these keywords or entities in the database. For this, we plan to explore open data sources, e.g. Wikidata, to find relations between these keywords and entities related to human roles, whose privacy may be somehow affected by the treatment of the data of the entities in the database.

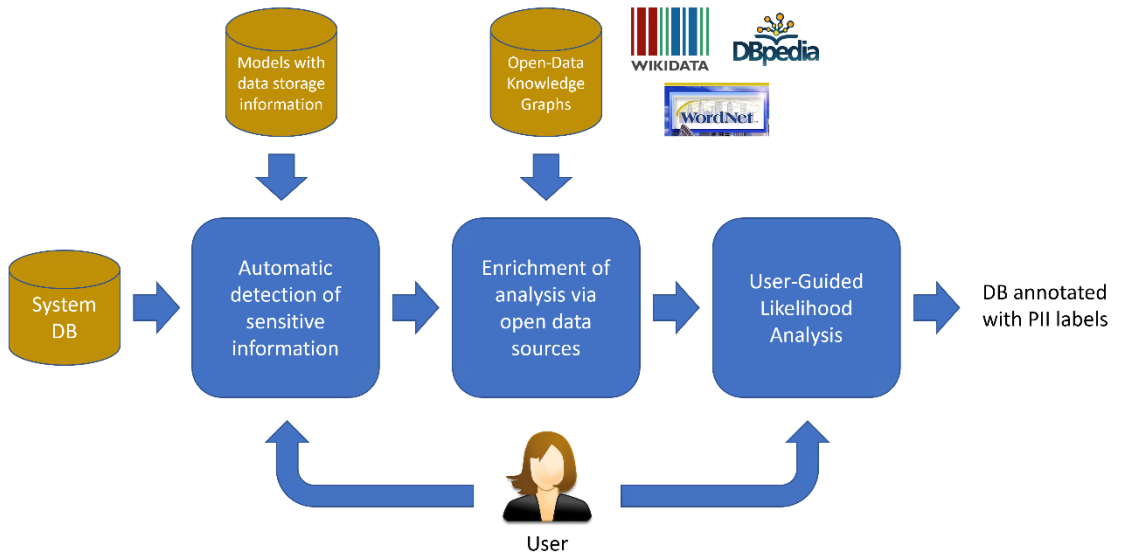

Figure 6. General overview of the methodology used in the Personal data identification tool.

- **User-guided Likelihood Analysis**: With the data retrieved by our tool, users will be presented with results and guided to reflect on the relationship between actual entities stored in the system database and external human entities whose data subject's rights may be affected by the system data treatment. Users may be required to estimate the likelihood of particular data entity to be linked with external entities. This information can also be fed to the risk management tool developed in WP3.

Open Data Sources

We plan to test the tool using open data. Our initial plan would be to test the tool consuming data from WikiData, DBpedia and other open data sources to find data instances representing identifiable individuals that can be somehow related to existing concepts in the existing database. We will explore ontologies related to persons[9] and we will look for people related to different ontologies derived from keywords in the system.

3.3 Select design strategy to fulfil goals and requirements

As mentioned in subsection 3.1, a set of requirements can be already identified and selected to guide the design process. More specifically, the referred requirements can be the outcome of the data protection requirement engineering as specified in deliverable D4.4 [4]. These requirements include in particular the specificities introduced by regulations like GDPR as well as other standards or privacy related methods to operationalize requirements (e.g., ProPan [11], [12]). The PDPbD framework shall include the elements necessary to manage referred specificities and the means to ensure and provide evidence of requirements fulfilment. To achieve this crucial goal, a design strategy, at least, needs to be selected and applied. We are aware that the notion of "design strategy" is generic and not unique. In PDP4E, our goal is to show how MDE and other techniques can be leveraged to support design engineers, once such notions haven been adopted. To do so, we find suitable to search and align our approach with a framework that (1) addresses privacy design concerns and (2) has been published and gained some international acceptance. A good candidate fulfilling referred criteria is the ISO 27550 standard [13]. The Table 3 is taken from ISO 27550 and shows a list of strategies associated to

[9] Such as http://dbpedia.org/ontology/Person or https://schema.org/Person

instances of privacy controls. To support engineers in the task of selecting a design strategy, the PDPbD method rely upon the following considerations:

a) Data and process views/models need to be first defined. To facilitate the selection of a design strategy, requirements can be assigned either to the data or to the process views. Once a first set of requirements is elicited (e.g., following WP4 methods and tools), their granularity (level of detail, specificity) must be reviewed in order to validate the pertinence and effectiveness of such separation. An adequate level of requirement specification shall ease their association to data or process oriented models.

b) To fulfil a specific requirement, more than one strategy may need to be applied. For instance, to achieve data protection of a web-data-based application compliant with GDPR principles, minimization may be required. In addition, to prevent any risk related to data leak or disclosure to non-intended parties, data hiding can be demanded and implemented via encryption mechanisms.

c) Once the requirements are operationalized and refined, their syntax may already include references to the specific strategies for the requirement to be fulfilled.

d) It is expected that as long as high-level requirements (e.g., those directly derived from GDPR) are broken down and refined, they will be expressed in terms of more fundamental properties which may ease the selection of a candidate design strategy.

Table 3. Design strategies as proposed and structured in ISO 27550. The image is borrowed from [13]

Design strategy		Description	Privacy control examples
Data oriented strategies	Minimize	Limit as much as possible the processing of PII	Selection before collection Anonymization
	Separate	Distribute or isolate personal data as much as possible, to prevent correlation	Logical or physical separation Peer-to-peer arrangement Endpoint processing
	Abstract	Limit as much as possible the detail in which personal data is processed, while still being useful	Aggregation over time (used in smart grids) Dynamic location granularity (used in location based services) k-anonymity
	Hide	Prevent PII from becoming public or known.	Encryption Mixing Perturbation (e.g. differential privacy, statistical disclosure control) Unlinking (e.g. through pseudonymisation) Attribute based credentials
Process oriented strategies	Inform	Inform PII principals about the processing of PII	Privacy icons Layered privacy policies Data breach notification
	Control	Provide PII principals control over the processing of their PII.	Privacy dashboard Consent (including withdrawal)
	Enforce	Commit to PII processing in a privacy friendly way, and enforce this	Sticky policies and privacy rights management Privacy management system Commitment of resources Assignment of responsibilities
	Demonstrate	Demonstrate that PII is processed in a privacy friendly way.	Logging and auditing Privacy impact assessment Design decisions documentation

3.3.1 Summary of properties targeted in PDP4E

Requirements are usually (and for most of their parts) specified in natural language. The expressiveness of requirements leads nonetheless to a huge variability which may finally limit or even impede a systematic association between requirements and design strategies for their fulfilment. Conduct research for exploring the space of associations possible is out of the scope of PDP4E. However, it is expected that once they are broken down, the detailed requirements can be specified in terms of more fundamental properties. Some instances of the candidate properties to be targeted by the PDPbD method are listed in line:

- Unlinkability
- Anonymity
- Confidentiality
- Pseudonimity *(this list is to be completed according to method and work plan evolutions)*

3.4 Design and enrichment of system data-oriented models

System data-oriented models are developed in order to capture and represent the data structures under study in preparation for further analyses. Data-oriented models can contain meta-data, in particular, the outcomes from the personal data identification phase (as described in Section 3.2). Following a MDE perspective, data-oriented models are meant to contain high-level representations of data instances still amenable to apply techniques as suggested in the existing data protection strategies: Minimize, Separate, Abstract, Hide. It is recalled that these strategies can be selected by the engineer in order to fulfil a set of requirements. A more detailed description of the support for data-oriented models is provided in deliverable D5.1 [5]. Once a first data-oriented model is developed, it can be enriched according to the following design activities:

a) ***Data completion:*** when information related to data are missing and need to be added, the design engineer shall be able to complete the model. Since we follow a MDE approach, the representation of data is based upon stereotypes defined via attributes and associations which may need to be filled/defined by the designer. For instance, in a model of a Relational Data Base, the relations between SQL tables, or between tables and users/context may need to be manually modelled.

b) ***Context annotations:*** the analysis of a data-related model may require introducing elements related to the context (e.g., meta-data, stakeholders, etc.). For instance, the application of a data-hiding strategy presupposes the existence of intended and non-intended parties (e.g., stakeholders and attackers). Depending upon the specific property under analysis (e.g., unlinkability), a logical border defining public and private zones may need to be settled. In that case, the contextual annotations help to determine whether data supposedly private truly remain hidden and are not reachable from parties within the public zone. Following a MDE perspective, the annotations to be supported are implemented within the meta-model and profile for PDPbD.

c) ***Analysis and results:*** the outcomes after applying a design strategy can be stored within the model itself. For instance, after applying a data Separation strategy, some logical or physical borders shall appear as part of the privacy solution[10]. Being outcomes of the referred strategy, the separation borders can be created and become part of the model. In the case of Abstraction of data, e.g., via k-anonymity, a model attribute can be defined to store the percentage of data loss per instance or item. More specifically, as explained in Section 3.2, after conducting the identification of personal data among structured/unstructured sources, the outcomes can include the likelihood of data being personal, linkable, etc. Those likelihoods shall be properly associated and/or included within the data instances in the design model.

d) ***Data model transformation:*** for some strategies like data Minimization, the readable/accessible data should be reduced to a minimal level according to certain metrics settled in advance. For this kind of strategies, their application via an algorithm or transformation can generate an optimized data model which is indeed an enhancement of the original model.

[10] Even if some of the design strategies can be automated, for now, we mainly focus on the support provided to the engineer for implementing them.

3.5 Design and enrichment of data-process-oriented models

Data-process-oriented models are developed to capture flows involving data. The data-process-oriented models adopt the form of a directed graph where edges represent exchanged data and graph vertexes represent data sources/consumers, storages or processing units. A well-known structure corresponding to that pattern is the Data Flow Diagrams (DFD). From a conceptual point of view, a DFD provides a high-level and synthetic view of a process which mostly keep a data-centric perspective. From a MDE perspective, a DFD can be seen as a specialization of a modelling language like BPMN [34] or UML Activities [35]. A more detailed specification of the framework to develop data-process-oriented models is provided in deliverable D5.1 [5]. To our knowledge, there is no standard, commonly accepted definition for DFDs. Definitions found in the state of the art usually come with different modelling rules but less often with explanations to justify statements or to validate the consistency between them. For instance, the Figure 7 shows a particular DFD example used in the PRIPARE project [40]. To provide support for achieving PDPbD, we will adopt a DFD definition commonly agreed with other WPs and amenable for importing/exporting models (or subparts of them) between PDP4E tools. Candidate DFD definitions are specified in [38], [39]. A relevant feature of DFDs is the definition of, at least, 3 levels of abstraction [38]:

- **0-Level DFD:** also known as context diagrams, this view represents the whole system as a single process including the exchanges with external entities as inputs/outputs via directed edges.

- **1-Level DFD:** in this level, the whole system process is broken down into sub-processes which represent the main functions supported by the system. This level mostly provides a functional view of the system including details of concerned data sources/consumers, and storages.

- **2-Level DFD:** this level allows the designer to decompose main functions into their parts (also represented as sub-processes). The view provides details about the functions' operation including specific inputs/outputs and concerned data sources/consumers, and storages.

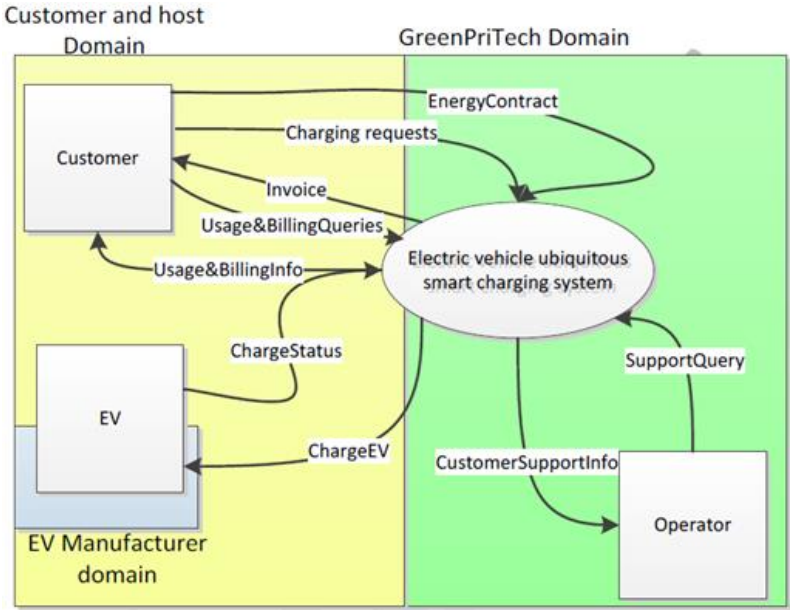

Figure 7. Instance of DFD as proposed in the PRIPARE project. The figure is borrowed from [40]

DFD levels are amenable to conduct relevant design tasks like system refinements. It is recalled that refinements of a design model allow in particular the identification of candidate architectures and the exploration of the design space [41], [42] (more details are given in Section 3.8). Along with traceability, keeping the consistency between DFD models at different levels is a main stake. In addition, the traceability and consistency w.r.t. data-oriented models (introduced in Section 3.4) should also be preserved. The adoption of MDE languages is meant to facilitate those goals. In particular, the inherited UML/SysML mechanisms [35], [36] for modelling extension, specialization, traceability, and decomposition are to be exploited.

The enrichment of DFD models can be conducted according to the following activities:

a) ***DFD annotating and completion:*** the annotation of DFDs mostly depends upon the specificities addressed by the requirements elicited from regulations like GDPR and also from those derived from the risks assessment. For instance, to validate whether a DFD model is in conformity with a GDPR principle (e.g., *the right of a data subject to give limited consent to the controller for his/her data to be processed targeting a specific purpose*), the fundamental notions introduced by the regulation need to be integrated within the PDPbD framework. To do so, a meta-model and profile need to be developed. To ease the design phase, we plan to reuse and adopt the meta-model that shall be developed in WP4 (D4.4 [4]) since the alignment is conceptually and technically ensured by construction. The requirements and design frameworks are both inspired by MDE and supported by the same background tool: Papyrus. Wherever needed, the DFDs shall be annotated relying upon the notions captured within the meta-model and profile. Moreover, it is foreseen that DFDs will be specialized by introducing new tangible elements (able to be materialized) like for instance *data subject, controller, processor*, etc. Non-tangible elements (mostly abstract and not necessarily materialized) like *right, consent, purpose,* etc. can also be supported by the PDPbD framework as long as they are needed for the design strategies to be applied.

b) ***DFD analysis and results:*** the strategies suggested to guide the design of data-process-oriented models are rather generic, diverse and in some cases not easily implementable. For instance, as it is shown in Table 3, the *Inform* and *Control* strategies are related to specific control mechanisms like *privacy icons* and *dashboards* to be part of a user

interface to collect data, whereas the *Demonstration* and *Enforcement* strategies can be achieved through more complex techniques like *rights management* and *privacy impact assessment*. In addition to that, the GDPR introduces abstract elements like *right, consent, purpose,* etc. along with properties like *fairness, friendly, lawful*, etc. which are mostly inherited from the legal arena. Our design framework should properly address both genericity and diversity of design strategies and the legal notions/properties traceable to the system design. Referred aspects need to be addressed as part of the evidence for requirements fulfilment. To do so, two generic design patterns are proposed. It is expected that these patterns shall be manually applied by the user over a DFD thus enriching it:

I. ***Provider – receiver proof:*** three modelling elements within the DFD respectively play the roles of provider, receiver and privilege item(s). The provider is supposed to ensure that (1) the receiver acknowledges the reception of the privilege item(s) and (2) the privilege items truly give the receiver the intended privileges. The design pattern should help to prove the reception and, if necessary, to ensure no-repudiation of acknowledgement. This design pattern seems adequate to deploy *Inform* and *Control* strategies. To ensure proof correctness, it may be necessary to differentiate between collaborative and hostile environments in which the provider and the receiver interact.

II. ***Proof of endorsement:*** three modelling elements within the DFD respectively play the role of endorser, recipient and qualifier or property. The endorser is supposed to hold methods, information, qualifications, etc. so as to evaluate qualities or properties of the recipient and endorse or disapprove it. This design pattern seems adequate to apply *Enforce* and *Demonstrate* strategies. In particular, processes, sub-processes and functions represented within a DFD may require to be endorsed with properties by stakeholders as demanded by the GDPR.

c) DFD evolutions: the application of data-process oriented strategies should lead to the evolution of DFD models. In general, the modifications are expected to be manually conducted by the designer. For now, no generic algorithms for transformation have been defined. However, in approaches like [43], design patterns are proposed to achieve a so called “privacy aware design”. For instance, in Figure 8 we show a design pattern that is applied to a 0-level DFD and leads to a more detailed DFD satisfying a privacy constraint. The application and implementation of such patterns need to be evaluated with regards to DFD variability and in particular w.r.t. the existing DFD levels.

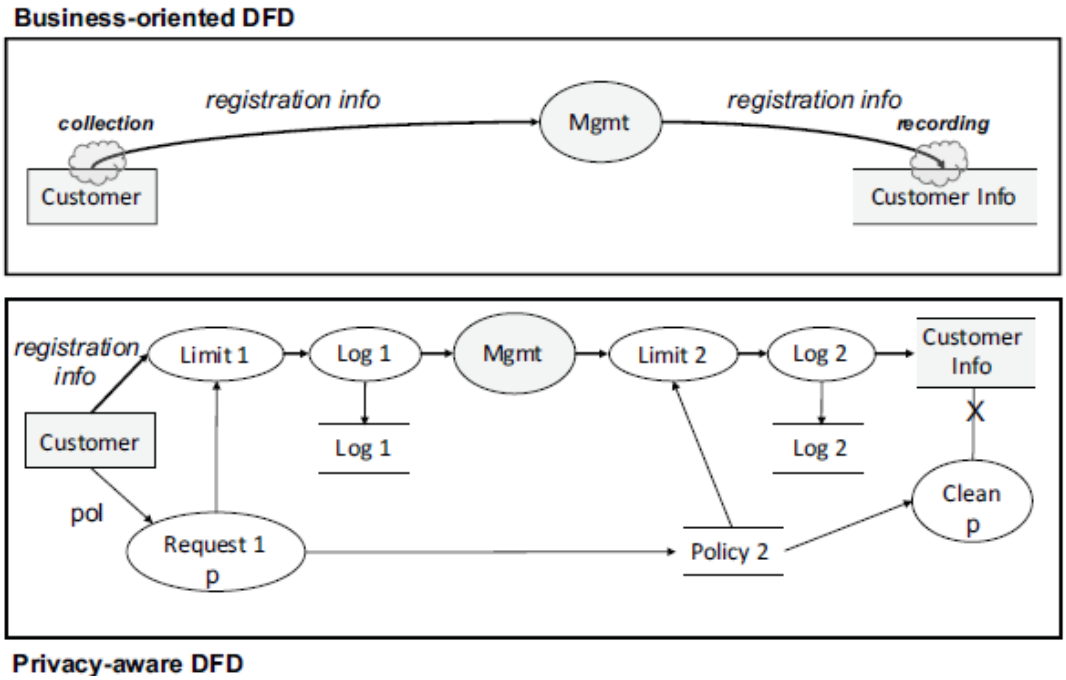

Figure 8. Instance of a Privacy aware DFD; privacy is ensured by design. The figure is borrowed from [43]

3.6 Apply strategy on data-oriented models

To achieve PDPbD, the design framework may support the application of different strategies to data-oriented models. The design strategies in the following subsections are proposed in ISO 27550 [13].

3.6.1 Minimize

Minimize data aims to reduce as much as possible the size of interpretable information which are exposed, available, transmitted or gathered by a system, software or process. A certain variability has been identified among cases where data minimization is required. For instance, in a data base system, the information within tables may require to be suppressed, bucketed or sliced [44], [45] so as to limit access to non-intended parties. In other use cases like vehicle-to-vehicle communications and in particular the Cooperative Awareness Messages (CAM), the broadcast headers need to be anonymized in order to prevent unnecessary (or even illegal) vehicle tracking. The techniques to achieve data minimization can vary among categories of use cases. Generic support can be provided by the PDPbD framework in the form of lists including existing techniques for minimization and related documentation. The specific support shall be primarily developed in terms of targeted pilots developed in WP7.

3.6.2 Separate

Separate data means to rely on logical or physical borders to distribute or even isolate data/information within a system. The goal is to impose restrictions for the distributed data to be correlated or to control/impede the access to isolated data. Along with a list of techniques for data separation, the PDPbD framework can support the application of this strategy by introducing modelling elements that represent physical and logical borders as well as allocation mechanisms to explore/evaluate possible data distributions. The allocation may be supported by settling allocation associations between data-oriented and architecture models. The implementation of this technique can support the design of Data Breach management mechanisms, for instance, mechanisms for breach incidents and system response.

3.6.3 Abstract

Data abstraction consists in increasing the level of granularity of data within frames, structures, storages in order to prevent (or make more complex) inferring information related to individuals. Several techniques exist to achieve data abstraction but they depend upon specific use cases. For instance, in the Smart Grid domain, the energy consumption data is represented as time series with very frequent measures enabling persons' profiling among other privacy-unfriendly practices [61]. This data is usually aggregated by reducing the resolution of the data. This means that a time interval of seconds might not be needed for several purposes where half-an-hour or hour intervals might be enough. In the case of location based services [46], location data are gathered from personal devices to provide a service, e.g., search of nearby places. The granularity of location data can be adapted to the purpose of the service, e.g., weather forecast service only requires prefix digits of postal code. The k-anonymity technique [47] can be applied on a table structure (*n* attribute columns x *m* data rows) by suppression or generalization of values (replacement of specific values by categories of values). The k-anonymity property is ensured w.r.t. selected attribute columns if for any row in the table, there exist at least another k-1 rows containing the same values within the selected attribute columns. The PDPbD framework can support the application of this strategy by listing and documenting possible techniques and by implementing generic algorithms, for instance to validate k-anonymity.

3.6.4 Hide

Hiding data means to render their interpretation difficult or non-feasible at all. Techniques like perturbation, encryption, and pseudo-anonymization can be applied to implement this strategy [62]. The application of referred techniques over data sets is usually conducted at implementation level. Along with lists of hiding techniques, the PDPbD framework may include, when appropriate, dedicated attributes to store scores related to quality/performance/confidence of techniques on specific data instances.

3.7 Apply strategy on process-oriented models

3.7.1 Inform

The Inform strategy is meant to provide data subjects with sufficient information about the aspects related to data processing. The PDPbD framework can support this strategy relying upon the *provider-receiver* design pattern introduced in Section 3.5.

3.7.2 Control

The Control strategy should ensure the data subject has enough control on the different aspects and phases of data processing (e.g., access privileges, consent). As in the previous category, the PDPbD framework can support this strategy relying upon the *provider-receiver* design pattern introduced in Section 3.5.

3.7.3 Enforce

Enforcing is intended to ensure the application of a policy or rule which in turns ensure a desired property, e.g., related to privacy or GDPR specificities. The PDPbD framework can support this strategy relying upon the *endorsement* design pattern introduced in Section 3.5. Concrete policies and rules to be enforced need to be analysed in order to determine feasible support.

3.7.4 Demonstrate

The Demonstration strategy is meant to provide evidence that justify or prove that a data process exhibits a desired property, e.g., related to privacy or GDPR specificities. The PDPbD

framework can support this strategy relying upon the *endorsement* design pattern as it was the case for the enforce strategy (introduced in Section 3.5).

3.8 Mapping data and process-oriented models over an architecture

Following the MDE approach, the typical design of an architecture unfolds as follows. A first model is obtained capturing the functional decomposition of a system. Then, a candidate architecture to support the functions is proposed. Afterwards, the allocation of functions to architecture components can be finally carried out. The referred allocation is part of the so called design space exploration problem [41], [42]: the design space is indeed generated by the possible distributions of functions among architectural components (NP-hard problem). As shown in Figure 9, the PDPbD framework is integrated by data, process and architecture models. Since the process-oriented models (i.e., the DFDs) contain sufficient information about the data and functions involved, the allocation shall be finally conducted from DFDs towards the architecture model. Notice that each process can be decomposed in one or more functions, and each function allocated to one or more components of the architecture. The distributions between processes-functions and architecture components define indeed the design space. Along with traceability and consistency between data and process models, other candidate features to be supported by the PDPbD framework are described in the following subsections 3.8.1, 3.8.2.

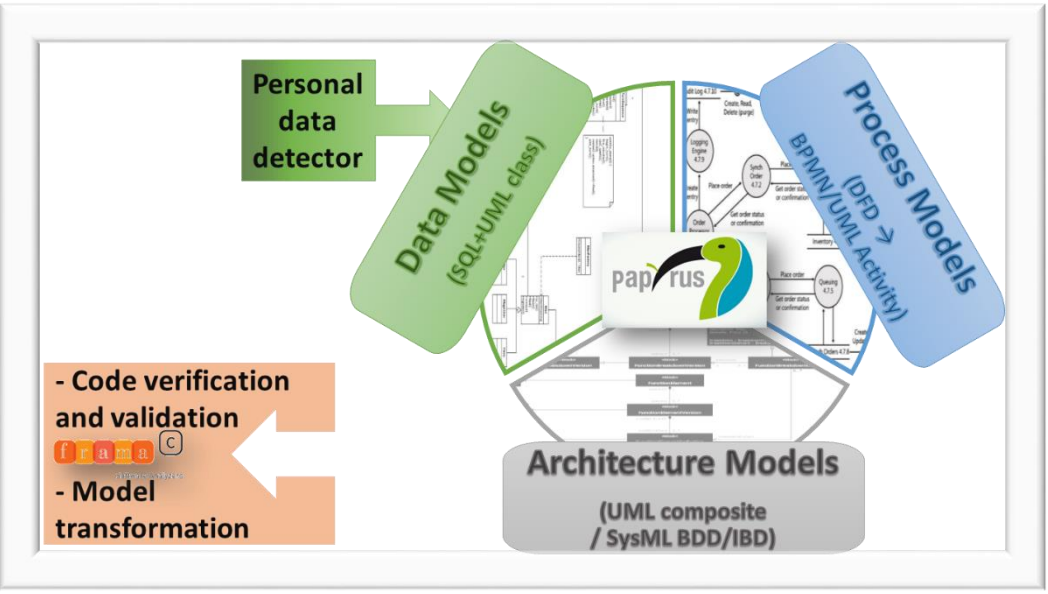

Figure 9. Overview of the PDPbD framework including data, process and architecture models, modules for personal data detection and code verification

3.8.1 Allocation mechanisms

The allocation mechanisms allow to associate elements and subparts within different model views. To keep consistency between data-oriented and process-oriented models, two options are identified:

a) ***Reusing modelling elements:*** the model element (e.g., a data instance) within a view is directly reused to model another view. This mechanism can be applied in particular whenever views or models are part of the same modelling project and no conflicts between languages appear. Reuse of modelling elements is particularly useful to keep consistency between data-oriented and process-oriented models.

b) ***Associating modelling elements:*** modelling elements can be correlated via associations. Associations are directed edges defined with a generic or specific type and a given semantics. Associations can be used to link instances within a data-oriented model to respective elements within the DFDs, e.g., data instances.

Once consistency between data and process-oriented models is ensured, they can be allocated to specific architecture components, for instance relying upon the UML *<<allocate>>* association. It is expected that after allocation, a first functional architecture is obtained. Subsequent refinements of the model shall lead to a fine-grained architecture, including more details of physical and logical components.

3.8.2 Architecture refinements

Architecture model refinements [42] can be carried out by introducing layers where components are decomposed thus including fine-grained specifications about their internal structure, e.g., details about ports, connectors, interfaces, data frames, allocated functions. The detailed description of an architecture, containing functional allocations, is a prior step to implementation and deployment phases. Of course, other design phases like SW/HW separation, SW distribution, etc., can be conducted prior to deployment (phases out of the scope of PDP4E). The PDPbD framework aims to introduce a layer where specific SW pieces or artefacts are referenced (e.g., via a library). This layer aims to identify potential privacy-related properties to be verified at code level.

3.9 Allocation of requirements to detailed architecture

A model including a detailed view of the architecture is amenable to be associated to requirements to fulfil. The following hypotheses are adopted for the referred associations to be settled:

H6. Some requirements have been broken down so as to be expressed in terms of more basic properties and constraints to be validated (instances of targeted properties are listed in Section 3.3.1).

H7. The refined requirements still include the inherited specificities from GDPR and privacy related concerns.

During the requirements breaking down task, the architecture can also be refined and accordingly detailed. Indeed, requirements can be further refined once a first detailed architecture is available and conversely. In general, refinements of requirements and architecture can be conducted in parallel following an iterative process. It is expected that after iterations, the requirements are finally allocated to the architecture. However, despite the methodological support, the allocation is a manual task that mostly relies upon the engineers' expertise.

3.10 Select and apply validation strategy

As expressed in the hypothesis H6, the fulfilment of detailed requirements can be accomplished by the validation or verification of more basic properties as the ones listed in Section 3.3.1. For now, the PDPbD framework is especially interested on properties that can be validated at code level. The validation strategies to be finally supported are under discussion.

In PDP4E, we aim at automatically generating code-level annotations and properties from a high-level formal description of the privacy and data protection requirements in order to integrate Frama-C/SecureFlow (and their possible extensions) in the global PDP4E validation methodology.

3.10.1 Code Verification

Validation and verification of privacy properties (e.g., unlinkability or confidentiality) at code level will be carried out through Frama-C, a code analysis framework for C programs [58]. Frama-C includes several analysers, but the one of primary importance for the target privacy properties is SecureFlow [59] which focuses on tracking information flows throughout the program.

Frama-C/SecureFlow follows a classical approach of information flow control tools which relies on annotating some program variables with their level of information, either *private* or *public*, meaning sensitive and non-sensitive, respectively. For instance, the annotation:

```
//@ private
```

must be added when declaring a variable in order to declare it as private (otherwise it is considered as public). SecureFlow also requires specifying verification points on which the tool checks information flow properties. For instance, the annotation:

```
/*@ assert security_status(output_value) == public; */
```

is written right before a statement:

```
return output_value;
```

The assertions above checks that the returned output value is public, meaning that its contents only depends on public pieces of information. Once these kinds of annotations have been inserted into the code, Frama-C/SecureFlow (in combination with another analyser of Frama-C) is able to check that no private data leaks on a public channel (here, `output_value`). Therefore, it is able to ensure a *non-interference* property that is at the basis of confidentiality: any private data *s* that interferes with a public one *p* leaks a piece of information that anyone reading *p* could deduce in order to break confidentiality.

While tracking information flow leakage for *security* properties - as Frama-C/SecureFlow does - has been already extensively studied [60], the challenge remains to be able to express a *privacy* property (e.g., unlinkability) as a set of such security annotations and properties.

4 Summary and perspectives

This document includes the specification of a method to guide engineers in seeking the goal of Privacy and Data Protection by Design (PDPbD). A set of standards, methods and techniques were selected and described as background. The PDPbD method is composed by 9 phases which are also explained. They cover several aspects related to privacy and data protection like personal data detection and data linkability, the modelling of structures, processes and architectures conveying data, and the validation of privacy-related requirements at different layers including code level. These aspects are addressed by encompassing (1) the requirements a system design may need to fulfil, (2) the strategies suggested to achieve privacy goals and (3) the respective techniques to implement the strategies. Along with ensuring consistency between phases (and supporting modules), salient features of systems design are also targeted like for instance traceability, consistency, refinements, and allocations. The use of MDE languages and techniques is meant to facilitate the realization of features. Several problematic design issues and stakes were highlighted and perspectives for solutions exposed.

As perspectives, the following can be mentioned. First, the PDPbD method is expected to evolve as long as its feasibility is evaluated and the final choices and techniques are adopted. In particular, referred evolutions should consider and integrate the algorithms and techniques used to implement risk-oriented and goal-oriented strategies (still to be selected). Since the PDPbD method is supposed to interoperate with other PDP4E frameworks, in particular risks analysis (WP3) and requirements engineering (WP4), more detailed specifications need to be achieved in respective work packages. More concretely, the inputs to be imported by the design framework need to be clearly specified, specially the inputs obtained after a first risks analysis and requirements engineering have been conducted. Last but not least, the usage of ISO 27552 [75] as background for the PDPbD method is foreseen. The standard contains privacy-related requirements that can be obtained as a result of applying both design strategies: risk-oriented and goal-oriented.

5 Bibliography

[1] The PDP4E consortium, Deliverable D3.1, *"Specification and design of risk management tool for data protection and privacy"*, Technical report of PDP4E, June 2019.

[2] The PDP4E consortium, Deliverable D3.4, *"Risk management methods for privacy and data protection"*, Technical report of PDP4E, June 2019.

[3] The PDP4E consortium, Deliverable D4.1, *"Specification and design of requirements engineering tool for privacy and data protection"*, Technical report of PDP4E, June 2019.

[4] The PDP4E consortium, Deliverable D4.4, *"Requirements engineering methods for privacy and data protection"*, Technical report of PDP4E, June 2019.

[5] The PDP4E consortium, Deliverable D5.1, *"Specification and design of model-driven design tool for privacy and data protection"*, Technical report of PDP4E, June 2019.

[6] The PDP4E consortium, Deliverable D5.4, *"Methods for data protection model-driven design"*, Technical report of PDP4E, June 2019.

[7] The International Organization for Standardization, *"ISO-31000 – Risk Management"*, Available in https://www.iso.org/iso-31000-risk-management.html, 2019.

[8] The International Organization for Standardization, *"ISO/IEC-29134 - Information technology -- Security techniques -- Guidelines for privacy impact assessment"*, Available in https://www.iso.org/iso-31000-risk-management.html, 2019.

[9] DistriNet Research Group, *"LINDDUN Privacy Threat Modeling and Method"*, Available in https://linddun.org/index.php, 2019.

[10] Council of the European Union, European Parliament, *"Regulation (EU) 2016/679 of the European Parliament and of the Council (General Data Protection Regulation)"*, Available in https://publications.europa.eu/en/publication-detail/-/publication/3e485e15-11bd-11e6-ba9a-01aa75ed71a1/language-en, 2019.

[11] Beckers K., Faßbender S., Heisel M., Meis R., *"A Problem-Based Approach for Computer-Aided Privacy Threat Identification"*. In: Preneel B., Ikonomou D. (eds) Privacy Technologies and Policy. APF 2012. Lecture Notes in Computer Science, vol 8319. Springer, Berlin, Heidelberg.

[12] R. Meis and M. Heisel, *"Computer-Aided Identification and Validation of Privacy Requirements"*, Information (Journal), vol. 7, no. 2, pp. 1-32, 2016.

[13] The International Organization for Standardization, *"ISO/IEC PRF TR 27550 - Information technology -- Security techniques -- Privacy engineering"*, Available in https://www.iso.org/standard/72024.html, 2019.

[14] The International Organization for Standardization, *"ISO/IEC 29100- Information technology -- Security techniques -- Privacy framework"*, Available in https://www.iso.org/standard/45123.html, 2019.

[15] European Union Agency for Network and Information Security ENISA, *"Privacy and Data Protection by Design – from policy to engineering"*, ENISA Report, December 2014.

[16] Alberto Crespo García, Nicolás Notario McDonnell, Carmela Troncoso, Daniel Le Métayer, Inga Kroener, David Wright, José María del Álamo and Yod Samuel Martín. Privacy- and Security-by-Design Methodology Handbook. PRIPARE Consortium. Available at http://pripareproject.eu/wp-content/uploads/2013/11/PRIPARE-Methodology-Handbook-Final-Feb-24-2016.pdf, 2016.

[17] ISO/IEC/IEEE 15288 First edition 2015-05-15 : ISO/IEC/IEEE International Standard - Systems and software engineering -- System life cycle processes. (2015). IEEE. https://doi.org/10.1109/IEEESTD.2015.7106435

[18] Ann Cavoukian. Operationalizing Privacy by Design: A Guide to Implementing Strong Privacy Practices. Information and Privacy Commisionner, Ontario, Canada, 2012. Available at http://www.ontla.on.ca/library/repository/mon/26012/320221.pdf

[19] Cavoukian, A., Jutla, D. N., Carter, F., Sabo, J., Dawson, F., Fieten, S., … Finneran, T. (2014). Annex Guide to Privacy by Design Documentation for Software Engineers Version 1.0. OASIS

Open. Retrieved from http://docs.oasis-open.org/pbd-se/pbd-se-annex/v1.0/cnd01/pbd-se-annex-v1.0-cnd01.html

[20]Ann Cavoukian, Fred Carter, Dawn Jutla, John Sabo, Frank Dawson, Jonathan Fox, Tom Finneran, and Sander Fieten (eds.) Privacy by Design Documentation for Software Engineers Version 1.0. 25 June 2014. Committee Specification Draft 01. http://docs.oasis-open.org/pbd-se/pbd-se/v1.0/csd01/pbd-se-v1.0-csd01.html. Latest version: http://docs.oasis-open.org/pbd-se/pbd-se/v1.0/pbd-se-v1.0.html.

[21]Kung, A. (2014). PEARs: Privacy Enhancing ARchitectures. In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). https://doi.org/10.1007/978-3-319-06749-0_2

[22]Kazman, R., Klein, M., & Clements, P. (2000). ATAM : Method for Architecture Evaluation. Cmusei. https://doi.org/(CMU/SEI-2000-TR-004, ADA382629)

[23]Antignac, T., & Métayer, D. Le. (2015). Trust Driven Strategies for Privacy by Design (Long Version). Retrieved from https://hal.inria.fr/hal-01112856

[24] Privacy Patterns website Available at https://privacypatterns.org

[25] Gürses, Seda; Troncoso, Carmela Gradiant; Diaz, C. (2015). Engineering privacy by design reloaded. Amsterdam Privacy Conference. Available at http://carmelatroncoso.com/papers/Gurses-APC15.pdf

[26] Hoepman, J. H. (2012). Privacy design strategies eprint. arXiv preprint arXiv:1210.6621. Available at https://arxiv.org/abs/1210.6621

[27] Hoepman, J.-H. (2014). Privacy Design Strategies (pp. 446–459). https://doi.org/10.1007/978-3-642-55415-5_38

[28] Hoepman, J.-H. (2019). Privacy Design Strategies (The Little Blue Book). Retrieved from https://www.cs.ru.nl/~jhh/publications/pds-booklet.pdf

[29]Pfitzmann, A., & Hansen, M. (2010). A terminology for talking about privacy by data minimization: Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management. Technical University Dresden. https://doi.org/10.1.1.154.635

[30]Cooper, A., Hansen, M., Smith, R., & Tschofenig, H. (n.d.). Privacy Terminology and Concepts. Retrieved from https://tools.ietf.org/html/draft-iab-privacy-terminology-01

[31]Cooper, A., Tschofenig, H., Aboba, B., Peterson, J., Morris, J., Hansen, M., & Smith, R. (2014). Privacy Considerations for Internet Protocols. RFC6973. IETF https://dx.doi.org/10.17487/rfc6973 Available at https://tools.ietf.org/html/rfc6973

[32]Common Criteria for Information Technology Security Evaluation Part 2: Security functional components. (2017). Retrieved from https://www.commoncriteriaportal.org/files/ccfiles/CCPART2V3.1R5.pdf

[33]Cavoukian, A. (2009). Privacy by design: The 7 foundational principles. Information and Privacy Commissioner of Ontario, Canada. Available at http://www.ontla.on.ca/library/repository/mon/24005/301946.pdf

[34]The Object Management Group, *"Business Process Model And Notation"*. BPMN Specification 2.0, Available in https://www.omg.org/spec/BPMN/2.0/About-BPMN/, 2019.

[35] The Object Management Group, "Unified Modelling Language". UML Specification 2.5.1, Available in https://www.omg.org/spec/UML/About-UML/, 2019.

[36] The Object Management Group, *"System Modelling Language"*. SysML Specification 1.6, Available in https://www.omg.org/spec/SysML/About-SysML/, 2019.

[37] The Eclipse Foundation, *"Eclipse Papyrus Modelling Environment"*, Available in https://www.eclipse.org/papyrus/, 2019.

[38] Lucid Software Incorporated, *"What is Data Flow Diagram"*, Available in https://www.lucidchart.com/pages/data-flow-diagram, 2019.

[39]Kim Wuyts and Wouter Joosen, LINDDUN privacy threat modeling: a tutorial, Technical Report (CW Reports), volume CW685, Department of Computer Science, KU Leuven, July 2015, Available in https://linddun.org/downloads/LINDDUN_tutorial.pdf, 2019.

[40] The PRIPARE consortium, Deliverable D1.3, *"Updated Privacy and Security-by-design Methodology"*, Technical report of PRIPARE, September 2015, Available in http://prepareproject.eu/research/, 2019.

[41] S. Pandey, M. Glesner and M. Muhlhauser, *"Architecture level design space exploration and mapping of hardware,"* International Symposium on Signals, Circuits and Systems, 2005. ISSCS 2005., Lasi, Romania, 2005, pp. 553-556 Vol. 2.

[42] J. Peng, S. Abdi and D. Gajski, *"Automatic model refinement for fast architecture exploration [SoC design],"* Proceedings of ASP-DAC/VLSI Design 2002. 7th Asia and South Pacific Design Automation Conference and 15h International Conference on VLSI Design, Bangalore, India, 2002, pp. 332-337.

[43] T. Antignac, R. Scandariato and G. Schneider, *"Privacy Compliance Via Model Transformations,"* 2018 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), London, 2018, pp. 120-126.

[44] B. Santhosh Kumar, S. Karthik and V. P. Arunachalam, *"vA Slicing with Generalization Techniques Used for Privacy Preserving Data Publishing,"* 2018 International Conference on Soft-computing and Network Security (ICSNS), Coimbatore, 2018, pp. 1-7.

[45] M. Colesky, J. Hoepman and C. Hillen, *"A Critical Analysis of Privacy Design Strategies,"* 2016 IEEE Security and Privacy Workshops (SPW), San Jose, CA, 2016, pp. 33-40.

[46] B. N. Jagdale and J. W. Bakal, *"Survey and Review of Location Privacy Techniques in Location Based Services,"* 2013 6th International Conference on Emerging Trends in Engineering and Technology, Nagpur, 2013, pp. 136-137.

[47] P. Samarati, *"Protecting respondents identities in microdata release,"* in IEEE Transactions on Knowledge and Data Engineering, vol. 13, no. 6, pp. 1010-1027, Nov.-Dec. 2001.

[48] Aura, T., Kuhn, T. A., & Roe, M. (2006, October). Scanning electronic documents for personally identifiable information. In Proceedings of the 5th ACM workshop on Privacy in electronic society (pp. 41-50). ACM.

[49] B. A. Beckwith, R. Mahaadevan, U. J. Balis, and F. Kuo. *"Development and Evaluation of an Open Source Software Tool for Deidentification of Pathology Reports"*. BMC Medical Informatics and Decision Making, 6(12), 2006.

[50] Du Mouza, C., Métais, E., Lammari, N., Akoka, J., Aubonnet, T., Comyn-Wattiau, I., ... & Cherfi, S. S. S. (2010, April). *"Towards an automatic detection of sensitive information in a database"*. In 2010 Second International Conference on Advances in Databases, Knowledge, and Data Applications (pp. 247-252). IEEE.

[51] J. Gardner and L. Xiong. "*HIDE: An Integrated System for Health Information DEidentification".* In Proc. Intl. Symp. on Computer-Based Medical Systems (CBMS), pages 254–259, 2008.

[52] Geng, L., Korba, L., Wang, X., Wang, Y., Liu, H., & You, Y. (2008, October). "*Using data mining methods to predict personally identifiable information in emails"*. In International Conference on Advanced Data Mining and Applications (pp. 272-281). Springer, Berlin, Heidelberg.

[53] Korba, L., Wang, Y., Geng, L., Song, R., Yee, G., Patrick, A. S., ... & You, Y. (2008, September). *"Private data discovery for privacy compliance in collaborative environments"*. In International Conference on Cooperative Design, Visualization and Engineering (pp. 142-150). Springer, Berlin, Heidelberg.

[54] I. Neamatullah, M. M. Douglass, L. wei H Lehman, A. Reisner, M. Villarroel, W. J. Long, P. Szolovits, G. B. Moody, R. G. Mark, and G. D. Clifford. *"Automated De-Identification of Free-Text Medical Records"*. BMC Medical Informatics and Decision Making, 8(32), 2008.

[55] Pandit, H. J., & Lewis, D. (2017, October). *"Modelling Provenance for GDPR Compliance using Linked Open Data Vocabularies"*. In PrivOn@ ISWC.

[56] Schwabe, D., & Laufer, C. (2019). *"Trust and Privacy in Knowledge Graphs"*. arXiv preprint arXiv:1903.07673.

[57] L. Sweeney. *Datafly: "A system for providing anonymity in medical data"*. In Proc. Intl. Conf. on on Database Security(DBSec), pages 356–381, 1997.
[58] Florent Kirchner, Nikolai Kosmatov, Virgile Prevosto, Julien Signoles, and Boris Yakobowski. 2015. *"Frama-C: A software analysis perspective"*. Form. Asp. Comput. 27, 3 (May 2015), 573-609.
[59] Gergö Barany and Julien Signoles, *"Hybrid Information Flow Analysis for Real-World C Code"*, Tests and Proofs, 2017, Springer International Publishing, 23-40.
[60] A. Russo and A. Sabelfeld, *"Dynamic vs. Static Flow-Sensitive Security Analysis,"* 2010 23rd IEEE Computer Security Foundations Symposium, Edinburgh, 2010, pp. 186-199.
[61] J. Martinez, A. Ruiz, J. Puelles, I. Arechalde, Y. Miadzvetskaya, *"Smart Grid Challenges through the lens of the European General Data Protection Regulation"*, ISD 2019, Toulon, France, 28-30 August, 2019.
[62] Harshavardhan Kayarkar, Sugata Sanyal, "A Survey on Various Data Hiding Techniques and their Comparative Analysis", ACTA Technica Corviniensis, Vol. 5, Issue 3, July-September 2012, pp. 35-40.
[63] Shostack, Threat modeling: Designing for security. John Wiley & Sons, 2014.
[64] M. S. Lund, B. Solhaug, and K. Stølen, Model-driven risk analysis: the CORAS approach. Springer Science & Business Media, 2010.
[65] Hoepman, Jaap-Henk. "Privacy design strategies." IFIP International Information Security Conference. Springer, Berlin, Heidelberg, 2014.
[66] Hansen, M., Berlich, P., Camenisch, J., Clauß, S., Pfitzmann, A., & Waidner, M. (2004). Privacy-enhancing identity management. Information security technical report, 9(1), 35-44.
[67] Sebastian Clauß, Andreas Pfitzmann, Marit Hansen, and Els Van Herreweghen. Privacy-enhancing identity management. The IPTS Report 67, 8-16, September 2002.
[68] Koen Simoens, Pim Tuyls, and Bart Preneel. Privacy weaknesses in biometric sketches. In Proceedings of the 2009 30th IEEE Symposium on Security and Privacy, pages 188–203. IEEE Computer Society, 2009.
[69] Martín Abadia and Cédric Fournet. Private authentication. Theoretical Computer Science, 322(3):427–476, September 2004.
[70] William Aiello, Steven M. Bellovin, Matt Blaze, Ran Canetti, John Ioannidis, Angelos D. Keromytis, and Omer Reingold. Just fast keying: Key agreement in a hostile internet. ACM Transactions on Information and System Security (TISSEC), 7(2):242–273, 2004
[71] Stefan Brands and David Chaum. Distance-bounding protocols (extended abstract). In Advances in Cryptology (EUROCRYPT'93), volume 765 of LNCS, pages 344–359. Springer, 1993.
[72] Jan Camenisch and Anna Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In Advances in Cryptology (CRYPTO'04), volume 3152 of LNCS, pages 56–72. Springer, 2004.
[73]K. Wuyts, "Privacy threats in software architectures," 2015
[74] A. Cailliau and A. van Lamsweerde, "A probabilistic framework for goal-oriented risk analysis," *2012 20th IEEE International Requirements Engineering Conference (RE)*, Chicago, IL, 2012, pp. 201-210.
[75]The International Organization for Standardization, "ISO/IEC DIS 27552 - Security techniques - Extension to ISO/IEC 27001 and ISO/IEC 27002 for privacy information management - Requirements and guidelines", Available in https://www.boutique.afnor.org/, 2019.