# An Architectural View for Data Protection by Design

Laurens Sion,[*] Pierre Dewitte,[†] Dimitri Van Landuyt,[*] Kim Wuyts,[*] Ivo Emanuilov,[†] Peggy Valcke,[†] Wouter Joosen[*]
[*]*imec-DistriNet, KU Leuven*
[†]*imec-CiTiP, KU Leuven*
{firstname.lastname}@kuleuven.be

***Abstract*—Data Protection by Design (DPbD) is a truly interdisciplinary effort that involves many stakeholders such as legal experts, requirements engineers, software architects, developers, and system operators. Building software-intensive systems that respect the fundamental rights to privacy and data protection is the result of intensive dialogue and careful trade-off decisions.**

**In practice however, there is a dichotomy between the legal reasoning which is conducted in Data Protection Impact Assessments (DPIA) and software engineering approaches, such as threat modeling, aimed at identifying privacy requirements and privacy risks. These activities are commonly performed in total isolation, which negatively impacts (i) the compliance exercise, (ii) the ability to evolve the system over time, and (iii) the architectural trade-offs made during system design.**

**In this article, we present an architectural viewpoint for describing software architectures from a legal, data protection perspective whose core modeling abstractions are based on an in-depth legal analysis of the EU General Data Protection Regulation. This viewpoint is tied to Data Flow Diagrams –commonly used in threat modeling– through correspondence rules. The proposed viewpoint supports the automation of a number of data protection impact assessment steps through (i) meta-model constraints, (ii) model analysis, and (iii) interaction with the involved stakeholders. This enables a streamlined compliance exercise, reconciling legal privacy and data protection notions with architecture-driven software engineering practices. We validate our approach in the context of a realistic e-health application for a number of complementary development scenarios.**

***Index Terms*—privacy by design, data protection, architectural viewpoint, GDPR, data protection by design, data protection impact assessment, accountability**

## I. Introduction

Addressing data protection issues at the software design stage—rather than adding a clunky layer of legal compliance to a near-final system—is increasingly recognized as the right approach to privacy engineering. This has recently been acknowledged by the EU General Data Protection Regulation (GDPR) [1] which obliges controllers to adopt a proactive stance both when sketching (*by Design*) and setting up (*by Default*) their processing operations. Art. $24(1)$ of the GDPR indeed requires controllers to '*implement appropriate technical and organizational measures to ensure and demonstrate compliance with the Regulation*'. Furthermore, Art. $25(1)$ requires the adoption of those measures '*both at the time of the determination of the means for processing and at the time of the processing itself*', taking into account the state of the art, implementation costs, and the nature, scope, context and purposes of the processing operations.

The above provisions have two consequences. First, controllers can tailor the extent of their compliance duty to the actual risk posed by their processing activities to the data subjects' rights and freedoms, which calls for a risk assessment to determine appropriate mitigations. Second, controllers must embed privacy-conscious features in their systems at the design phase, as well as throughout the whole processing life cycle (Data Protection by Design (DPbD)).

An efficient DPbD approach requires a close collaboration between data protection experts and software engineers. The legal expertise of the former is essential to orient the compliance exercise, often involving a Data Protection Impact Assessment (DPIA). This usually consists of: (i) describing and mapping processing operations, (ii) identifying and documenting data protection threats, (iii) implementing appropriate mitigations, and (iv) ensuring accountability by documenting this process. In that sense, DPbD and DPIA essentially share the same approach, and always start with the description of the system. Most of the time, these exercises are performed manually, which requires tremendous efforts, can lead to human errors, and is highly sensitive to changes in the system. It is therefore not a sustainable solution in the long term. Complementary to that, software engineering expertise is required for substantiating the above in concrete software systems. Several tools and methodologies are available to facilitate the development of data protection-conscious software systems. Firstly, privacy threat modeling methodologies, such as LINDDUN [2], can be applied to systematically elicit privacy threats in a system by using a Data Flow Diagram-based abstraction [3] of the system. Subsequently, data protection countermeasures (e.g., privacy-enhancing technologies [4], privacy patterns [5]) can be applied to mitigate the uncovered privacy issues.

Although individual tool support exists to perform threat modeling and analysis [6], [7], and DPIAs [8], these activities are usually performed by different stakeholders in complete isolation [9]. As a result, each exercise is executed following its own methodology, using a dedicated, unharmonized lexicon as well as disparate concepts and abstractions. This negatively affects: (i) the ability to assess and demonstrate compliance to regulation, (ii) the accuracy of the system description, which does not evolve at the same pace as the system itself, and

(iii) the quality of architectural trade-offs made in this context.

In this paper, we first define a systematic model-driven approach for data protection by design that is structured according to these steps. Subsequently, we accomplish technical integration of the ensuring data protection architectural viewpoint with a DFD-based viewpoint which is common in privacy and security threat modeling. We argue how tight integration between both types of activities as part of software architecture design allows for improved risk assessment, and enables architecture-level design activities such as change impact analysis and trade-off decision making to take aspects of regulatory data protection by design into account. We validate the presented view in the context of a real-world IoT-based e-health application for monitoring cardiovascular disease patients.

In this paper, Section II first presents the necessary background concepts in terms of privacy engineering and data protection law and motivates the paper. Next, Section III presents the new data protection viewpoint for describing a system in terms of legal concepts and abstractions and aligning legal perspectives with software engineering practices. Subsequently, Section IV validates the viewpoint, with several usage scenarios, in the context of the e-health application. Section V discusses the benefits of tighter integration between both types of views, whereas Section VI discusses related work. Finally, Section VII concludes the paper.

## II. Background

This section elaborates on both the technical and legal data protection approaches and concludes with a motivation for aligning these activities.

### A. Privacy Threat Modeling

Security- and privacy-driven software engineering starts in the early stages of the software development life cycle [10] with activities of threat elicitation, modeling [11], assessment [12], [13] and mitigation.

Specifically for privacy threats, the LINDDUN [2] privacy threat modeling framework supports the systematic elicitation and mitigation of privacy threats. Conducting such threat analyses starts from an architectural model of the system, more specifically a Data Flow Diagram (DFD) [3], modeling how data flows through the system, is commonly used in this context [2], [10], [11], [14], [15].

As depicted in Figure 1, the DFD notation is based upon 5 distinct element types: (i) an *external entity* (i.e., users or third party services that are external to the system), (ii) a *process* (i.e., a unit of computation), (iii) a *data store* (i.e., a passive container of information), (iv) a *data flow* (detailing how the data is propagated through the system), and (v) a *trust boundary* (i.e., a logical or physical boundary of the system).

Due to the relative simplicity of this modeling paradigm, the main benefit of the DFD notation is its broad applicability. This however inhibits more advanced analyses, as the model lacks much of the necessary information.

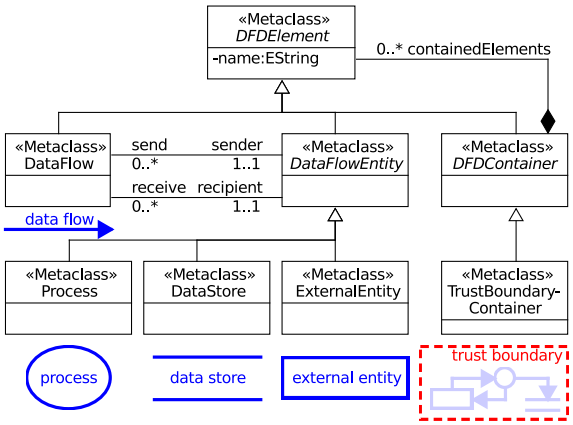


Fig. 1. The DFD meta-model (UML class diagram).
*Below the meta-model classes, the visualization of their instances in a Data Flow Diagram (DFD) is shown in color.*

### B. Data Protection Law

Data protection law requires *controllers*—and to a lesser extent *processors*—to comply with the numerous provisions of the GDPR when it comes to the *processing* of *personal data*. These crucial notions are strictly defined by the Regulation and circumscribe its personal and material scope of application.

*a) Scope of application:* The very first step of a traditional data protection compliance exercise consists of identifying the actors involved as well as the relevant processing activities to delineate the areas where conformity with data protection rules is required. Art. 4(7) defines the *controller* as '*the natural or legal person which, alone or jointly with others, determines the purposes and the means of the processing of personal data*', while Art. 4(8), defines the *processor* as '*the natural or legal person which processed personal data on behalf of the controller*'. Most of the rules contained in the GDPR mainly impact the former, while the latter only faces limited obligations with regard to security, liability, breach notification, and its interactions with other actors.

Under the GDPR, *processing* is to be understood as '*any operation or set of operations which is performed on personal data or on sets of personal data*' (Art. 4(2)). While a non-exhaustive list of activities is provided, one usually assumes it encompasses everything that can be done with personal data, from their original collection to their definitive erasure. Personal data is defined by Art. 4(1) as '*any information relating to an identified or identifiable natural person*'.

*b) GDPR Principles:* Every *controller* and *processor* involved in the processing of *personal data* must comply with the general principles listed by the GDPR [1, Art. 5], namely: (a) lawfulness, fairness and transparency, (b) purpose limitation, (c) data minimization, (d) accuracy, (e) storage limitation, (f) integrity and confidentiality, and (g) accountability.

Most importantly, the collection of personal data must be paired with a specific, explicit and legitimate purpose and be based on one of the legal bases listed in the Regulation [1, Art. 6(1)]. Every subsequent processing of the said data must,

in turn, be compatible with the purposes for which they were initially collected.

*c) Data Protection Impact Assessment (DPIA):* In order to achieve compliance with all the above, controllers usually rely on DPIAs. Not only is this exercise mandatory in case of processing activities that are likely to result in high risks to data subjects' rights and freedoms [1, Art. 35],[1] but it also lays the groundwork for a sound risk-based approach. The GDPR indeed encourages the adoption of 'appropriate' measures that are proportional to the likelihood and severity of the risks for data subjects' rights and freedoms posed by the processing. Quantifying that risk allows controllers to tailor the scope of their compliance duty [1, Art. $24(1)$, $25(1)$, and 32], implement data protection by design [1, Art. $25(1)$], and guarantee the security of their activities [1, Art. 32]. Since conducting such an assessment is far from trivial, templates, guidelines, recommendations, and methodologies to assist controllers have been released at various levels [16]–[21].

### C. Motivation

In practice, the methods and techniques discussed above are executed in total isolation, by different stakeholders, with different knowledge of the system under design. This disconnect leads to issues in terms of consistency and *architectural erosion*: the system as described in the DPIA is fundamentally different from the actual architecture, which might render the whole DPIA obsolete, outdated, or even incorrect. Contemporary software development practices, such a agile development, continuous integration and deployment, are based on frequent iterations, which exacerbates the negative consequences of such a disconnect.

Examples of such issues might include: 1) In the course of software evolution, the software architect decides to extend the system by, for instance, introducing functionality that involves automated decision-making. In that case, additional checks might be required. 2) The architecture might include data protection countermeasures where none are mandated by law, and vice-versa. For example, for data retention policies.

## III. Architectural viewpoint for data protection

In this section, we present: (i) support for an explicit data protection viewpoint for modeling a system's data processing activities from a legal perspective; (ii) the data protection checks leveraging this model.

### A. The Data Protection Viewpoint

As discussed, a compliance assessment starts with a detailed mapping of the system [16]. It is usually followed, with slight variations, by: (i) the identification and documentation of data protection threats, (ii) the implementation of appropriate technical and organizational measures, (iii) the documentation of the process to ensure controller accountability, and (iv) a periodic monitoring and review phase. As underlined above, there are strong incentives to align the way legal experts and software engineers describe a system. In turn, a technically accurate data protection view on a system will facilitate efficient, in-depth compliance with data protection rules.

*1) Construction of the data protection view:* In order to build a meta-model, it is necessary to determine the terminology to be used and the elements to be incorporated in the model.

*a) Terminology:* The meta-model should rely on GDPR concepts rather than on the abstractions that are commonly used in architectural views such as Data Flow Diagrams (DFD). This way, the meta-model will provide unambiguous references to notions that are clearly defined by the Regulation. The qualification of each actor following the GDPR lexicon will drastically ease the allocation of: (i) responsibility for compliance, (ii) accountability for the measures implemented, and (iii) liability in case of non-compliance. Additionally, using the legal lexicon will vastly improve the legibility of the system description—and therefore of the whole compliance exercise—in case of administrative or judicial procedures. This is also in line with the GDPR [1, Art. $24(1)$ and $25(1)$] which requires controllers to be able to demonstrate that they took all the necessary steps to comply its provisions.

Figure 2 depicts the meta-model for the data protection viewpoint. It introduces the main concepts and abstractions necessary for constructing data protection views and is based on an extensive inter-disciplinary study of the GDPR [1].

*b) Core abstractions:* The viewpoint is constructed to include all the necessary information to serve as an input for a comprehensive DPbD/DPIA endeavor. This way, the inclusion of concepts and attributes depends on whether they provide crucial information for the checks that have to be performed according to the Regulation. For instance, the obligation to pair a personal data collection with a purpose through a `ProcessingPurpose` element is reflective of the emphasis on clearly documenting the purposes of these processing activities. The explicit inclusion of this information facilitates subsequent compliance assessments with GDPR principles, such as purpose limitation [1, Art. $5(1)b$], as explained further on in Legal Requirement 1. In the same vein, the general prohibition to process special categories of personal data [1, Art. 9] has lead to a separate explicit `SpecialCategoryExemptionsToProhibition` data type which requires one of the exemptions listed in Art. 9. The proposed viewpoint uses the following concepts:

**Actor:** An `Actor` is an entity that plays a specific `LegalRole` in the processing of personal data. The GDPR distinguishes between controllers [1, Art. $4(7)$], processors [1, Art. $4(8)$], recipients [1, Art. $4(9)$], third parties [1, Art. $4(10)$] and, when controllers or processors are not established in the EU, representatives [1, Art. $4(17)$]. For each actor, details are provided about their private or public nature, legal or natural personality and establishment. The qualification of each entity involved in the data processing is an essential prerequisite in allocating their respective duties under the Regulation.

**Processing:** A processing activity is any operation performed on personal data by the listed actors. It always starts with the initial `Collection` of personal data —either

[1]Even for cases that do not formally require a full-fledged DPIA, the compliance checks required to determine the necessity of a DPIA are based on comprehensive mapping of the different data processing activities.

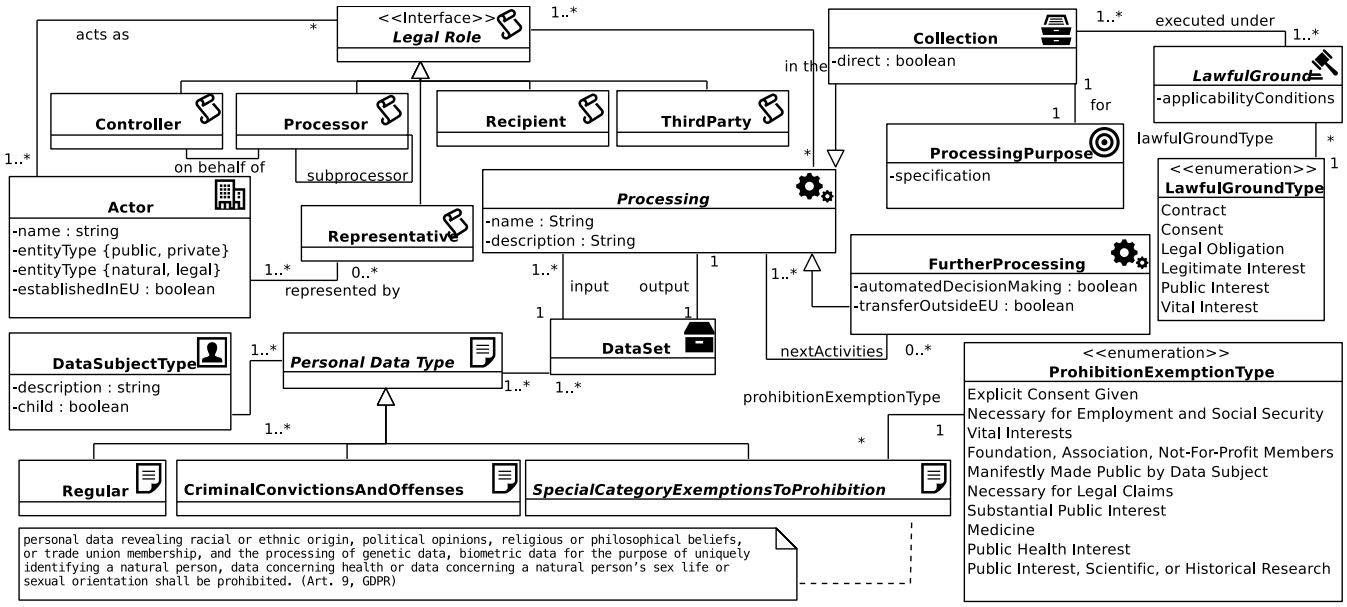


Fig. 2. The meta-model for the data protection architectural viewpoint (*UML class diagram*)
*The corner of each class includes the icon used in the instance model to more easily make a distinction between the different types. Besides the specified multiplicities, there are some additional constrains imposed, which are listed in Section III-A.*

directly from the data subject or through another source—and encompasses every subsequent use of the said data (FurtherProcessing). According to the lawfulness [1, Art. $5(1)a$] and purpose limitation [1, Art. $5(1)b$] principles, the Collection must be based on one of the LawfulGrounds listed in Art. $6(1)$ and paired with a specified, explicit and legitimate ProcessingPurpose. Every FurtherProcessing, on the other hand, must be compatible with the purposes for which the data were initially collected. Finally, some FurtherProcessings trigger additional, specific requirements, namely: automated decision-making and transfers outside the EU.

**Data Subject Type:** The natural person whose data are processed is referred to as the 'data subject'. His/her qualification as a child—a notion whose scope is partially left up to Member States [1, Art. $8$]—influences the compliance assessment at various levels. A direct reference is included to the types of data that are processed.

**Personal Data Type:** The PersonalDataTypes are used to specify the different categories of personal data that are being processed. The GDPR prescribes additional safeguards for some DataTypes, namely: (i) special categories of personal data [1, Art. $9$] and (ii) criminal convictions and offenses [1, Art. $10$]. More specifically, collecting and processing the former is subject to a general prohibition which can be lifted when one of the ProhibitionExemptionTypes listed in Art. $9(2)$ is applicable.

### *B. DPbD/DPIA Support*

The Article 29 Working Party (WP29) has published tangible criteria for a suitable system description in the context of a DPIA [16, Annex 2]. Below, we discuss to which extent the data protection viewpoint outlined above fulfills the WP29 and GDPR [1] requirements when it comes to describing a system under design. We confront our viewpoint with a subset of the criteria extracted from the WP29 and the GDPR to illustrate how the viewpoint supports their evaluation.[2] The following four categories of criteria are considered: (1) documenting the processing operations, (2) soundness of the model, (3) legal requirements that are either fully automatic or trigger a manual assessment by a stakeholder, and (4) risk management.

*1) Documentation Criteria:* This section discusses, for a number of the WP29 documentation criteria, how the viewpoint supports the documentation of the required information.

DOCUMENTATION CRITERION 1. ***Nature, scope, context, and purposes***
*"[The] nature, scope, context and purposes of the processing are taken into account" [16, Annex 2]*

The proposed viewpoint supports the modeling of every Collection and FurtherProcessing, including, for each of these, the ProcessingPurpose, the LawfulGround, the involved Actors and their LegalRoles, and the DataSets and DataTypes involved.

DOCUMENTATION CRITERION 2. ***Processing description***
*"[A] functional description of the processing operation is provided" [16, Annex 2]*

Each Processing element has a name and description. The activities also link to subsequent and previous activities.

The systematic mapping of the data processing activities facilitates the compliance with transparency requirements [1, Art. $5(1)a$, 12, 13 and 14] since suitable documentation can

[2]This is a limited set of criteria to illustrate their assessment via the data protection viewpoint. The viewpoint supports the evaluation of much broader set of criteria. See the WP29 DPIA requirements [16, Annex 2].

be automatically generated from the models that are part of the data protection view.

*2) Soundness Criteria:* A benefit of adopting a systematic modeling approach is that it allows making explicit the completeness and soundness criteria for verifying whether a model is correct. Below, we provide a set of examples for the data protection viewpoint:

SOUNDNESS CRITERION 1. ***Collection***
*Every chain of* `Processings` *needs to start with a* `Collection`*.*

Every `FurtherProcessing` requires a previous activity except the `Collection`. This way, every activity can be evaluated in light of the original purpose for which the data was collected.

SOUNDNESS CRITERION 2. ***Input and output consistency***
*Every input* `DataSet` *to a* `Processing` *needs to be the output* `DataSet` *of a* `Processing` *that is located earlier in the chain of* `Processings`*.*

For every input `DataSet` to a `Processing`, the chain of previous `Processings` needs to be traversed to verify that the `DataSet` is an output of a previous `Processing`.

With the appropriate tool support, these rules can be implemented and checked when the model is constructed, providing the modeler with direct feedback (see Section IV-A).

*3) Legal requirements:* Finally, a sound model supports evaluating the legal requirements imposed by the GDPR [1]. A subset of them is used to illustrate the assessment:

LEGAL REQUIREMENT 1. ***Purpose limitation***
*The principle of purpose limitation [1, Art.* $5(1)b$*] dictates that personal data must be collected for* 'specified, explicit and legitimate purposes'*, and any further processing must be compatible with the purpose for which it was originally collected.*
*"In such a case, no legal basis separate from that which allowed the collection of the personal data is required." [1, Rec. 50]*

The data protection view provides support for this assessment not only by explicitly modeling the required information (Documentation Criteria 1 and 2), but also by supporting systematically traversing a chain of previous `FurtherProcessings` to verify compatibility with the `ProcessingPurpose` of the initial `Collection`.

LEGAL REQUIREMENT 2. ***Data minimization***
*Personal data shall be limited to what is necessary in relation to the purposes for which they are processed [1, Art.* $5(1)c$*].*

The meta-model facilitates this assessment by having a broad overview of all the personal data collected and processed, as well as a clearly specified purpose. This streamlines the assessment that must be conducted (i.e. are all `DataSets` really necessary to achieve the specified `ProcessingPurpose`?).

LEGAL REQUIREMENT 3. ***Automated decision making on special categories of data***
*Automated decisions based on special categories of personal information is prohibited, unless the exceptions of Art. 9(2)a or g apply [1, Art. 9] and suitable measures have been implemented to safeguard rights and freedoms and legitimate interests.*

Check for every `FurtherProcessing` that involves automated decisions (`automatedDecisionMaking`) and `SpecialCategoryExemptionsToProhibition` data, that there is a valid exemption [1, Art. $9(2)a$ or $g$].

Other legal requirements the data protection viewpoint supports in assessing are: storage limitation [1, Art. $5(1)e$], exemption on the prohibition on processing special categories of personal data [1, Art. 9], exemption on the prohibition on processing personal data relating to criminal convictions and offenses [1, Art. 10], automated decision making [1, Art. $13(2)f$, $14(2)g$, $15(1)h$, $22(1)$, $22(3)$, and Rec. 71)], transfer to a third country [1, Art. 45, 46, 47, and 49], EU representative for non-EU controllers [1, Art. 27], joint controllership [1, Art. 26], consent [1, Art. $7(1)$], etc.

*4) Risk Management:* The two final criteria imposed by the GDPR [1] and WP29 [16, Annex 2] requirements are those involving risk assessment and the determination of appropriate countermeasures in light of the estimated risks.

RISK MANAGEMENT CRITERION 1. ***Risk assessment***
*"[O]rigin, nature, particularity and severity of the risks are appreciated (cf. [1, Rec. 84]) or, more specifically, for each risk (illegitimate access, undesired modification, and disappearance of data) from the perspective of data subjects:*

- *risk sources are taken into account [1, Rec. 90];*
- *potential impacts to the rights and freedoms of data subjects are identified in case of events including illegitimate access, undesired modification and disappearance of data;*
- *threats that could lead to illegitimate access, undesired modification and disappearance of data are identified;*
- *likelihood and severity are estimated [1, Rec.* $90$*];" [16, Annex 2]*

*The risk assessment should in particular take into account the risks presented by the processing. [1, Art.* $32(2)$*]*

The data protection viewpoint provides only limited support for conducting a risk assessment. This assessment can be performed by evaluating for each `Processing` activity whether it involves sensitive `DataTypes` or data from `DataSubjectTypes` such as children. This assessment is limited to a legal risk assessment because of the nature of this viewpoint.

RISK MANAGEMENT CRITERION 2. ***Measures***
*"[M]easures envisaged to treat those risks are determined [1, Art.* $35(7)(d)$ *and Rec.* $90$*]" [16, Annex 2] The measures should* "ensure a level of security appropriate to the risk" *[1, Art.* $32(1)$*]*

The data protection viewpoint does not have a first-class representation of measures. Any measures to treat the identified risks are limited to making changes to the model itself (e.g., limiting the `DataTypes` that are collected/processed).

It is for the above two risk management criteria that the integration with complementary viewpoints in a traditional software architecture will be especially useful, as that information is required for properly assessing these risk management criteria. The integration of these two views, and leveraging the integration for the assessment is discussed in Section V-B.

## IV. VALIDATION

We have validated our contributions in the context of a *Patient Monitoring System (PMS)*, an e-health system for the treatment of cardiovascular diseases. The primary goal of the PMS is to support extra-mural, continuous and remote monitoring, timely decision making, and prediction of malignant events. This is done by fitting patients with wearables

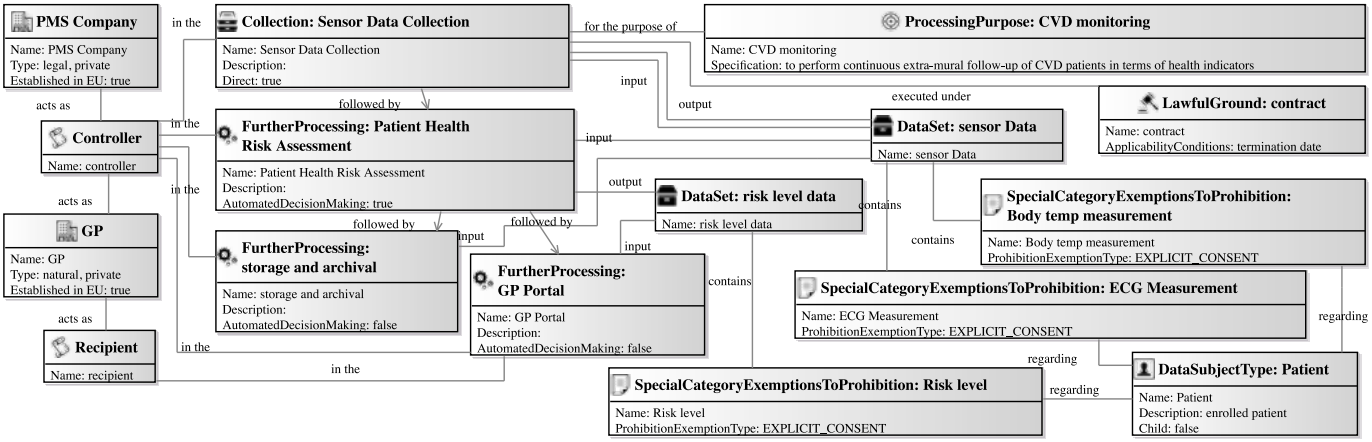


Fig. 3. Data Protection View of the Patient Monitoring System (PMS).
*This view is constructed with the concepts from the meta-model depicted in Figure 2. It shows which actors are involved in which data processing activities. Furthermore, the diagram also shows the order of processing, the purpose and legal basis, and which data sets are the inputs and outputs of those processing activities. The data sets themselves again contain the data types, which link back to the data subject type they belong to.*

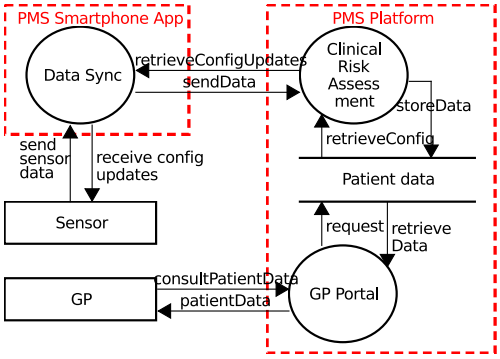


Fig. 4. Data Flow Diagram (DFD) of the Patient Monitoring System
*This is a simplified version of the system, which makes abstraction of additional server-side functionality (e.g., interaction with hospital doctors, nurses, the hospital information system).*

to measure health parameters such as body temperature, electrocardiograph (ECG), etc.

In this section, we first briefly discuss the prototype implementation of the data protection viewpoint. Next, we illustrate the different architectural views of the PMS: (i) the Data Flow Diagram (DFD), and (ii) the corresponding data protection view. Afterwards, Section V-A discusses the integration of these two views, including the application on the PMS case.

### A. Implementation

A prototype implementation was created to validate the proposed viewpoint. The meta-model itself was implemented in the Eclipse Modeling Framework. The graphical visualization of the viewpoint was implemented in a Sirius viewpoint specification. This viewpoint specification supports the enforcement of model soundness checks in the Acceleo Query Language.

The implementation of the compliance assessment rules is currently work in progress. These rules are being implemented in VIATRA's graph-based pattern language, after which the VIATRA query engine can be used to query the concrete data protection models for the applicability of the patterns.

Future implementation efforts involve the enforcement of the correspondence rules with the technical architecture.

### B. System Modeling

*a) Data flow diagram (DFD):* Figure 4 depicts a (simplified) Data Flow Diagram-based view on the architecture of the PMS. It depicts the `sensor`, which communicates with a smartphone app (`Data Sync`) for the synchronization of the data. The data is sent from the smartphone app to the PMS platform, where a risk assessment is performed on the sensor data (`Clinical Risk Assessment`). Both the received sensor data and the risk assessment are stored in the `Patient Data` data store to make them available for later retrieval. This retrieval is performed by general practitioners (GPs), who use the `GP Portal` to access both sensor data and risk levels.

*b) Data Protection View:* Figure 3 shows the data protection view that corresponds with the DFD presented above. This view is constructed in accordance with the meta-model proposed in Section II-B. In constructing this view, the modeler is forced to explicitly define and document (i) the distinct data processing activities, (ii) the involved actors and their (legal) roles, (iii) the purpose and legal basis for the processing, (iv) the data sets and types of personal data that are processed, and (v) the types of data subjects whose data is involved. Figure 3 shows all these elements in a diagram, including their relations, which indicates the sequence of processing activities and which data sets are the inputs and outputs of the different data processing activities.

## V. Integration in Architectural Frameworks

Apart from the purpose of systematically documenting the system under analysis in terms of appropriate legal system abstractions, the data protection viewpoint presented in the previous section is further aligned with different system abstractions used in other architectural viewpoints. Architectural

frameworks [22] define architectural viewpoints to document different perspectives on a system, tailored to address specific stakeholder concerns. Examples of technical viewpoints are Kruchten's $4+1$ model [23] and the client-server, module, and allocation viewpoints by Bass et al. [24]. The main advantage of a model according to different viewpoints is that *correspondence rules* ensure these views remain consistent. A modification in one view affects the others in accordance with these rules. As such, modeling the data protection view of a system enables tighter integration with other architectural views. In the first instance, this section discusses and motivates the technical integration with DFD system abstractions, commonly used in privacy threat modeling. In-depth investigation and further alignment between the data protection view and other architectural viewpoint and relevant artifacts (e.g. access control or data protection policies) is considered future work.

Section V-A first accomplishes this integration technically through defining correspondence rules [22] between both views, followed by an application on the e-health case. The combination fo both views allows for a tight integration of threat analysis and DPbD/DPIA, as discussed in Section V-B, which revisits the DPbD/DPIA criteria. Sections V-C and V-D respectively discuss the benefits in terms of architectural change impact analysis and trade-off decision making.

### A. Integration with DFDs through Correspondence Rules

In this section, we discuss integration of the data protection viewpoint presented in the previous section with Data Flow Diagram (DFDs).

The correspondence rules between data processing elements and DFD elements (and vice versa) are summarized in Table I. As the multiplicities in the table illustrate, there is no trivial, straight-forward correspondence between data protection and DFD elements, which is indicative of the distinct nature of both viewpoints. One view may be modeled at a higher level of abstraction or granularity and concepts at that abstraction level may correspond with a larger set of elements in the other view which may be documented at a more fine-grained level.

Furthermore, not all elements of one view will necessarily correspond with elements of the other view. For example, the data controller will typically not interact directly with the system (as in the example in Fig. 4), and hence, will not appear as an entity in the DFD. Additionally, data flows that do not involve any personal data (e.g., control flows) will not have any corresponding elements in the data protection view.

Finally, although the data subject is at the core of the data protection analysis, its role in the system itself is often less prominent. In many systems, the data subject itself does not have an active role and is therefore not represented in the DFD (e.g., a system to share medical patient records among practitioners). In other systems, a data subject type can have a 1-to-1 or even a 1-to-$n$ correspondence with its external entity counterpart in the DFD. For example, external devices can also interact with the system on behalf of a data subject. These devices (e.g., a body sensor that collects measurements) which are modeled as external entities in the DFD also correspond with a data subject in the data protection view.

TABLE I
ALIGNMENT OF THE MAIN DATA PROTECTION ABSTRACTIONS AND DFD ELEMENT TYPES.

| Data Protection Element Type | Corresponds with | DFD Element Type |
|---|---|---|
| Processing | 1..∗ ≻ 1..∗ | Process |
| | 1..1 | Data Store |
| DataSet/DataType | 1..∗ ≻ 1..∗ | |
| | 0..∗ | Data Flow |
| Actor | 0..∗ — 0..∗ | External Entity |
| DataSubjectType | | |

*A multiplicity at a line indicates that either element type above or below can match. Multiple multiplicities indicate the differences between the originating element types. For example, a* Processing *can correspond to* 1..∗ Processes OR Data Stores*. While, in the other direction, a* Data Store *corresponds with* 1 Processing *and a* Process *can correspond with* 1..∗ Processings*.*

*Application on the PMS case:* Figure 5 depicts the correspondences defined between the two PMS views introduced before in Section IV: the DFD (Figure 4) and data protection view (Figure 3). It shows how the different (legal) elements from the data protection view correspond with concrete DFD elements, in accordance with the correspondence rules from Table I. The various processing activities correspond with the processes and data stores that represent the technical realizations of these processing activities. Furthermore, it highlights which data types are used in which data flows, and documents the involvement of the data subject and the GP (in the legal role of a recipient). The correspondences between these two views enable a wide variety of compliance checks and feedback mechanisms to ensure consistency and technical compliance with requirements imposed by data protection law.

### B. Integrated DPbD/DPIA Support

Section III introduced the data protection viewpoint and motivated it in terms of the ability to evaluate the documented system against principles of Data Protection by Design (DPbD). In this section, we discuss the benefits of bidirectionally integrating these views and activities from the points of view of (i) DPbD/DPIA principles, (ii) architecture-level threat modeling and (iii) risk management.

DOCUMENTATION CRITERION 1 **Nature, scope, context, and purposes**

In software architecture, aspects of scope, context or purposes are seldom documented—this is information added by the data protection viewpoint. Additionally, the description of the nature of the processing activities can be crosschecked or derived from other architectural viewpoints, e.g., steps involving automated decision making can be directly mapped onto the software components or processes responsible for big data analytics and machine learning-based classification.

DOCUMENTATION CRITERION 2 **Processing description**

Instead of textually documenting the processing description, a *functional description* can be instantiated at the basis of the information encoded in other architectural views (e.g., from logical views or process views such as DFDs). For example, the description attribute can refer directly to a chain of processes in the DFD that logically correspond to the Processing at hand. For example, a detailed decomposition of the Clinical-RiskAssessment Process in Figure 5 supports the automatic retrieval of all the processing steps involved in determining a patient's risk level. This is particularly helpful when the system evolves, as changes to these views will ripple to the functional system description, which in turn may trigger an assessment of whether aspects of data protection have to be revised. An example of such a change in Figure 5 is the inclusion of another External Entity that accesses the GP Portal, this would trigger update to the data protection view as a new Recipient will have to be added.

SOUNDNESS CRITERION 1 **Collection**

The identification of Collection activities and the distinction between direct acquisition or reuse (the direct boolean attribute in the meta-model) of the involved DataSets can be verified in other architectural views, for example in DFDs, by analyzing where the data originates from. For example, the SensorDataCollection in Figure 5 receives data directly from an ExternalEntity mapped to a DataSubjectType. This assists in ensuring consistency across multiple views.

SOUNDNESS CRITERION 2 **Input and output consistency**

Keeping track of Data Flows in DFDs, with particular attention on which DataTypes are used by which components, allows ensuring the correctness of the Processing chain documented in the data protection viewpoint with respect to the actual implementation of these activities and flows in other architectural views. Furthermore, the integration with the DFDs can also support in the identification of DataSets with DataTypes with conflicting ProcessingPurposes, as a technical implementation may group DataTypes together where DPbD principles may require a strict separation.

LEGAL REQUIREMENT 1 **Purpose limitation**

Existing architecture descriptions can assist in assessing purpose limitation. Traversing a chain or Processings can be complemented with an analogous traversal check in the software architecture. This would involve checking for each Processing the corresponding Process in the DFD to ensure its consistency with the data protection view and guarantee that there are no additional processing operations on the data, for which the purpose limitation assessment was not performed. This strengthens a purpose limitation compliance claim.

Additionally, synchronization in the opposite direction makes the ProcessingPurpose information available when considering changes to the architecture such as adding new Processes.

LEGAL REQUIREMENT 2 **Data Minimization**

Realizing the data minimization principle also benefits from the consideration of both complementary views. In the data protection viewpoint, data minimization imposes a reduction of the number of DataTypes being processed to

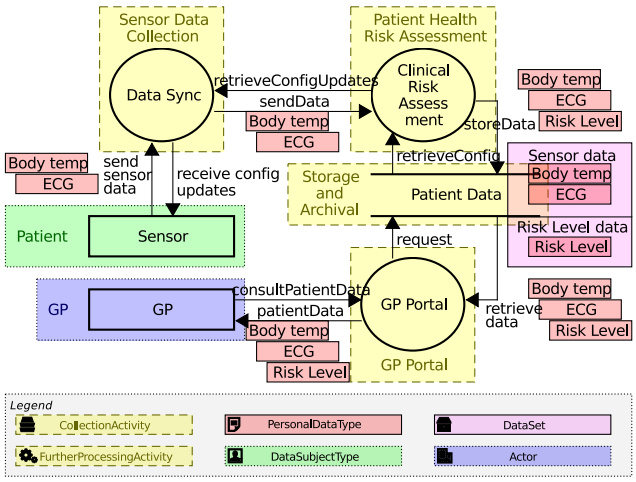


Fig. 5. Graphical representation of the correspondences in the example of the Patient Monitoring System DFD.
*This diagram shows which DFD elements correspond with which data protection elements. Not all data protection elements have a corresponding DFD element (e.g., the controller), or vice versa (e.g., control flows).*

those strictly necessary for the purpose. For example, if the body temperature in Figure 5 would no longer be necessary for the risk assessment, this change would need to be propagated to the Data Sync, Sensor, Patient Data, and GP Portal components. The correspondences assist in identifying them.

Starting from the technical viewpoint, countermeasures can be instantiated to reduce the amount of data being processed or to reduce the direct link to data subjects (e.g., through anonymization or pseudonymization). The application of such countermeasures requires a analogous update to the data protection view so that their effect can be taken into account.

LEGAL REQUIREMENT 3 **Automated decision making on special categories of data**

In case of automated decisions that are made based on special categories of data, the GDPR dictates that additional security and privacy measures have to be implemented. By integrating the data protection view with complementary architectural views in which the mitigations have been instantiated, keeping track of which mitigations are in place (for demonstrability and accountability) can be performed within the model.

Furthermore, correspondences between the DataTypes can be used to track and control how the data of these special categories is effectively used in the architecture. This can in turn be used as a confirmation to ensure that the data protection view does not overlook any automated decision processes.

RISK MANAGEMENT CRITERION 1 **Risk assessment**

Risk assessment benefits considerably from the integration with technical viewpoints. Starting from a DFD-based [3] abstraction of the system, traditional approaches for security [11] and privacy [2] threat modeling can be applied to systematically elicit threats. Such analyses can be further enriched with the incorporation of the information from the data protection view: (i) data types and their sensitivity through DataStores and

DataFlows, (ii) data subject types through ExternalEntities or via the DataTypes, (iii) the nature of the processing activities through the Processes. After identifying the threats in the previous step, a risk estimation can be performed [25] for each threat. This estimation step can again be improved with the inclusion of the aforementioned data protection artifacts as resources in the risk estimation. Estimating the impact of a certain threat to a data subject's rights requires information on what type of data of that data subject is involved. For example, de-identification [26] is not a boolean process and the strength of the outcome depends on many factors such as (i) the technical feasibility and cost of re-identification, (ii) the likelihood of re-identification given the incentives and technological context of the involved parties, etc. [1, Rec. 26].

RISK MANAGEMENT CRITERION 2 **Measures**

Finally, after assessing the risk, threats can be prioritized and appropriate countermeasures can be determined and applied. As countermeasures can be both technical as legal in nature, this involves both views too. The technical view can support the instantiation of technical security and privacy countermeasures. These countermeasures will introduce changes to the software architecture of the system under consideration, and may have effects on the data protection view (e.g., anonymization influencing the data types being processed). Complementary to technical countermeasures are legal ones. Legal countermeasures can involve changes to data protection model such as: (i) the types of data being processed, (ii) the types of data subjects (e.g., no children), (iii) the nature of processing activities (e.g., automated decisions, transfers outside the EU), and so on. Analogously to the technical countermeasures, these changes can again influence the complementary architectural views and require updates there as well. The mutually influencing effects of these changes illustrate how mitigating risk to data subjects is a trade-off exercise that influences both views simultaneously.

### C. Architecture-level change impact analysis

Contemporary development practices such as agile development, continuous integration and deployment, frequent architecture-level change. This poses challenges with respect to the data protection efforts which are quickly based on outdated and *eroded* system models. The correspondences between views introduced in this paper allow changes within one view to be reflected in the other views to make the implications of these changes on the data protection aspects of the system visible and raises these checks again at the time of software evolution.

For example, as a consequence of architectural extension, the introduction of new processes in the DFD will trigger the assessment to verify whether the newly-introduced technical process is in line with DPbD constraints that apply to the data processing activity it contributes to, or whether a new data processing activity has to be specified. This in turn will trigger additional checks to verify that the processing activity is still compliant with the original purpose and lawful grounds for which the data was originally collected, and if required, whether the lawful grounds have to be extended accordingly.

### D. Architectural trade-off analysis

Architectural trade-offs [24] are decisions that impact a number of software qualities simultaneously. Trade-off analysis [27] involves identifying key trade-offs, qualifying the impact and business value of each decision and its impact on the software architecture. The availability of a data protection model greatly improves awareness of data protection implications whilst making such architectural trade-off decisions.

For example, the architectural decision to host customer data with a cloud provider has significant consequences not only on data availability, data query performance, security, but also on legal data protection aspects. In the case of a cloud provider, a legal agreement has to be established with the cloud provider (who is a data processor), and a new DPbD/DPIA exercise has to be executed over the modified data protection view, which in turn may require new legal or technical countermeasures.

## VI. RELATED WORK

Several approaches and techniques have been proposed in the literature to assist in the engineering of data protection regulation-compliant software systems. We discuss two categories of approaches and techniques: (i) starting from a legal perspective and (ii) starting from an engineering perspective.

*Legal perspective:* Starting from the legal perspective, Breaux et al. [28], [29] have looked into natural language processing to extract technical requirements from regulatory rights and obligations. For meeting such requirements, Compagna et al. [30] have proposed a framework to model security and privacy patterns. Oetzel and Spiekermann's [20] proposal also starts from legal requirements (privacy targets) for a step-by-step privacy impact assessment (PIA) to obtain a list of control recommendations. They recommend system, functional, data, and physicalviews to model the system. However, users are still needed to ensure consistency and provide the inputs.

*Software engineering approaches:* From a modeling perspective, Antignac et al. [31] have proposed PA-DFD, an extension of DFDs which integrates concepts from the GDPR and ISO 29100. It does not, however, cover all legal concepts. Other DFD extensions have been proposed in the literature [7], [25], [32], [33]. Examples are security or privacy solutions [7] to take existing countermeasures into account and enable the up-front elimination of inapplicable threats. Further extensions include adding risk assessment information such as asset values, countermeasures strengths, and explicit attacker models [25] to enable a full-fledged risk analysis of the resulting privacy and security threats. Not focused on DFDs, but also on model enrichment with privacy information, Ahmadian et al. [34], [35] have proposed a similar approach with stereotypes such as sensitiveData to attach to model elements. In contrast to these approaches, our proposal involves creating a separate, yet connected, data protection view on the system, instead of further complicating DFDs. This provides the benefit of triggering legal checks such as an initial collection, without needing elements in an engineering view to trigger this. Furthermore, not all of the required information, such as data subjects, lawful grounds, and purposes, can be unambiguously assigned to technical model

elements. Our approach does not impose the use of DFDs, as the correspondences with other complementary architectural views can be used as well, enabling other types of security analyses [36]–[38]. The data protection view can be beneficial to capture legally-relevant input for other frameworks such as of PRIAM [12] to support the systematic privacy risk assessment.

## VII. Conclusion

In the current state of the art, threat modeling and DPbD/DPIA exercises are distinct, isolated analysis activities that are nonetheless both conducted in the early stages of the development life cycle and should both be revisited iteratively as the system evolves. This disconnect causes several issues: (i) technical implementations that violate restrictions imposed by a DPIA, (ii) ineffective or insufficient countermeasures, (iii) inaccurate or outdated DPbD and DPIA exercises.

In this paper, we have presented an architectural viewpoint for data protection, modeled upon the concepts and requirements imposed by the GDPR [1]. This data protection view is complementary to the existing Data Flow Diagram (DFD) view and kept consistent through explicit correspondence rules.

As shown, by aligning both types of views, the resulting analysis activities can mutually reinforce each other and provide useful feedback for further improving both the technical architecture, as well as the DPbD/DPIA exercises. This helps to create and maintain consistent, up-to-date, and detailed documentation of the compliance process that can be presented when requested by data protection authorities.

Future extensions include (i) the automatic generation of the data protection checklist that has to be confirmed by a legal professional, (ii) automatic verification whether required privacy countermeasures triggered by the data protection view are presented in the technical architecture, (iii) dynamic data protection impact reassessment triggered by changes to the technical model, (iv) further exploring the symbiotic relationship between technical privacy threats and legal privacy threats with the integration of threat modeling and DPIAs.

The coherent mapping of all the processing activities facilitates the identification of threats and, therefore, the adoption of the appropriate mitigation strategies, be they technical or legal in nature. This, in turn, is a veritable stepping stone towards implementing a 'risk-based' approach (as the GDPR prescribes) that is not exclusively based on the legal notion of risk.

## Acknowledgements

This research is partially funded by the Research Fund KU Leuven and the PRiSE KU Leuven-C2 research project.

## References

[1] European Union, "Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016," *Official Journal of the EU*, 2016.
[2] M. Deng, K. Wuyts, R. Scandariato, B. Preneel, and W. Joosen, "A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements," *Requirements Engineering*, 2011.
[3] T. DeMarco, *Structured Analysis and System Specification*, 1979.
[4] D. L. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms," *Communications of the ACM*, vol. 24, no. 2, 1981.
[5] M. Hafiz, "A collection of privacy design patterns," in *Proceedings of the 2006 Conference on Pattern Languages of Programs*, 2006.
[6] Microsoft Corporation, "Microsoft Threat Modeling Tool 2016," 2016.
[7] L. Sion, K. Yskout, D. Van Landuyt, and W. Joosen, "Solution-aware data flow diagrams for security threat modeling," in *Proceedings of ACM SAC: Software Architecture: Theory, Technology, and Applications*, 2018.
[8] CNIL, "The open source pia software helps to carry out data protection impact assesment," 2018. [Online]. Available: https://www.cnil.fr/en/open-source-pia-software-helps-carry-out-data-protection-impact-assesment
[9] P. Dewitte, K. Wuyts, L. Sion, D. Van Landuyt, I. Emanuilov, P. Valcke, and W. Joosen, "A comparison of system description models for data protection by design," in *Proceedings of ACM SAC: PDP*, 2019.
[10] M. Howard and S. Lipner, *The Security Development Lifecycle*, 2006.
[11] A. Shostack, *Threat Modeling: Designing for Security*, 2014.
[12] S. Joyee De and D. Le Métayer, "PRIAM: A privacy risk analysis methodology," *Lecture Notes in Computer Science*, pp. 221–229, 2016.
[13] T. Brüggemann, J. Hansen, T. Dehling, and A. Sunyaev, "An information privacy risk index for mhealth apps," in *Annual Privacy Forum*, 2016.
[14] A. Shostack, "Experiences threat modeling at Microsoft," in *Modeling Security Workshop. Dept. of Computing, Lancaster University, UK*, 2008.
[15] D. Dhillon, "Developer-Driven Threat Modeling: Lessons Learned in the Trenches," *IEEE Security Privacy*, vol. 9, no. 4, pp. 41–47, jul 2011.
[16] Article 29 Working Party, "Guidelines on data protection impact assessment (DPIA) (WP248 rev.01)," 2017.
[17] Bitkom, "Risk assessment und datenschutz-folgenabschaetzung," 2017.
[18] F. Bieker, M. Friedewald, M. Hansen, H. Obersteller, and M. Rost, "A process for data protection impact assessment under the european general data protection regulation," in *Annual Privacy Forum*. Springer, 2016.
[19] D. Wright, M. Friedewald, and R. Gellert, "Developing and testing a surveillance impact assessment methodology," *International Data Privacy Law*, vol. 5, no. 1, pp. 40–53, 2014.
[20] M. C. Oetzel and S. Spiekermann, "A systematic methodology for privacy impact assessments: A design science approach," *European Journal of Information Systems*, vol. 23, no. 2, pp. 126–150, 2014.
[21] European Data Protection Supervisor, "Accountability on the ground part II: Data protection impact assessments & prior consultation," 2018.
[22] ISO/IEC/IEEE, "ISO/IEC/IEEE Systems and software engineering – Architecture description," *ISO/IEC/IEEE 42010:2011(E)*, 2011.
[23] P. Kruchten, "The 4+1 view model of architecture," *IEEE software*, 1995.
[24] L. Bass, P. Clements, and R. Kazman, *Software architecture in practice*. Addison-Wesley Professional, 2003.
[25] L. Sion, K. Yskout, D. Van Landuyt, and W. Joosen, "Risk-based design security analysis," in *Proceedings of the 1st International Workshop on Security Awareness from Design to Deployment*, 2018, pp. 11–18.
[26] S. L. Garfinkel, "De-identification of personal information," 2015.
[27] R. Kazman, M. Klein, and P. Clements, "Atam: Method for architecture evaluation," Carnegie-Mellon University, Tech. Rep., 2000.
[28] T. D. Breaux, M. W. Vail, and A. I. Anton, "Towards Regulatory Compliance: Extracting Rights and Obligations to Align Requirements with Regulations," in *14th Intl. Requirements Engineering Conf.*, 2006.
[29] T. Breaux and A. Antón, "Analyzing Regulatory Rules for Privacy and Security Requirements," *IEEE Trans. on Software Engineering*, 2008.
[30] L. Compagna, P. El Khoury, A. Krausová, F. Massacci, and N. Zannone, "How to integrate legal requirements into a requirements engineering methodology for the development of security and privacy patterns," *Artificial Intelligence and Law*, vol. 17, no. 1, pp. 1–30, mar 2009.
[31] T. Antignac, R. Scandariato, and G. Schneider, *A Privacy-Aware Conceptual Model for Handling Personal Data*. Cham: Springer, 2016.
[32] B. J. Berger, K. Sohr, and R. Koschke, "Automatically extracting threats from extended data flow diagrams," *LNCS*, vol. 9639, pp. 56–71, 2016.
[33] K. Tuma, R. Scandariato, M. Widman, and C. Sandberg, "Towards security threats that matter," in *3rd Workshop On The Security Of Industrial Control Systems & Of Cyber-Physical Systems*, 2017.
[34] A. S. Ahmadian, D. Strüber, V. Riediger, and J. Jürjens, "Supporting privacy impact assessment by model-based privacy analysis," in *Proceedings of ACM SAC 2018: Software Engineering*, 2018.
[35] A. S. Ahmadian, J. Jürjens, and D. Strüber, "Extending model-based privacy analysis for the industrial data space by exploiting privacy level agreements," in *Proceedings of ACM SAC 2018: PDP*, 2018.
[36] Q. Feng, R. Kazman, Y. Cai, R. Mo, and L. Xiao, "Towards an Architecture-Centric Approach to Security Analysis," in *2016 13th Working IEEE/IFIP Conference on Software Architecture (WICSA)*, 2016.
[37] S. Seifermann, "Architectural Data Flow Analysis," in *2016 13th Working IEEE/IFIP Conference on Software Architecture (WICSA)*, 2016.
[38] E. Taspolatoglu and R. Heinrich, "Context-Based Architectural Security Analysis," in *13th Working IEEE/IFIP Conf. on SA (WICSA)*, 2016.