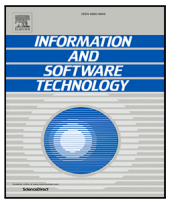
Information and Software Technology 195 (2026) 108121

Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof



Enabling a model-driven workflow for ongoing interdisciplinary collaboration in legal threat modeling☆

Nicolas Boltz *, Leonie Sterz, Oliver Raabe, Christopher Gerking

KASTEL – Institute of Information Security and Dependability, Karlsruhe Institute of Technology (KIT), Am Fasanengarten 5, 76131, Karlsruhe, Germany

ARTICLE INFO

Keywords:
Interdisciplinary collaboration
Threat modeling
Legal assessments
Legal subsumption
Legal interpretation
Data flow analysis
Data protection

ABSTRACT

Context: Software systems often provide critical functionality or process personal data, requiring compliance with applicable legal regulations. Ensuring legal conformity demands close collaboration between legal and technical experts, but differences in terminology and methodology make this challenging.
Objective: In this article, we aim to address the challenges in legal interdisciplinary collaboration by proposing a model-based workflow for ongoing and collaborative legal assessments within the context of threat modeling.
Method: The central aspects of the workflow are based on model-driven engineering techniques and were developed through active collaboration between researchers in software engineering and legal informatics/data protection at the KASTEL Security Research Labs. The goal of the collaboration was to integrate the methodologies of both domains into the workflow equally.
Result: The proposed workflow centers on maintaining consistency between a legal viewpoint and data flow diagrams, addressing legal subsumption, allowing each discipline to work from its own perspective while providing automated support in threat identification through an extended existing data flow analysis framework that considers legal interpretation. We evaluate the workflow and its modeling artifacts by applying it in the domain of the GDPR, discussing feasibility and applicability, and measuring the accuracy and scalability of the extended data flow analysis.
Conclusion: By combining discipline-specific viewpoints with automated consistency and threat identification, the workflow supports collaboration and enables iterative assessments. Our findings suggest that the presented workflow is suitable and operationalizable, but identify potential challenges in practical application or transfer to other legal domains.

1. Introduction

Software systems increasingly affect more aspects of modern life, from smart homes and mobility solutions to healthcare applications. This transformation introduces challenges such as privacy, security, resilience, discrimination, bias, transparency, and accountability in automated processes. Legal frameworks address these challenges by setting principles and requirements that mandate the development and operation of trustworthy and rights-preserving systems.

Ensuring legal compliance, however, is not purely a technical challenge. It requires close collaboration between legal and technical disciplines, which is often hindered by disparities in terminology, methodologies, and abstraction levels [1,2]. Technical experts often lack legal knowledge [3–5], while legal experts lack technical familiarity, resulting in communication barriers and effort during compliance assessment. This challenge is amplified by the dynamic nature of software systems and the interpretative character of law, which introduces variability through undefined legal terms and evolving societal contexts. A further challenge is that recent legal norms, such as the GDPR, the AI Act, and the Cyber Resilience Act, increasingly adopt a system-centric perspective and specify more technical details, increasing the likelihood that interdisciplinary collaboration is necessary to achieve compliance. Consequently, ensuring legal compliance cannot be a one-time activity. Rather, there is a continuous need for ongoing communication between the legal and technical disciplines, not only during design or when obvious changes are made, but throughout the software development lifecycle [6–8].

Existing approaches in privacy and data protection engineering — whether at design time [2,9–12] or in business process modeling [13–16] — primarily focus on technical solutions and specific legal aspects.

☆ This article is part of a Special issue entitled: 'RegCompliance in SE' published in Information and Software Technology.

* Corresponding author.
E-mail addresses: nicolas.boltz@kit.edu (N. Boltz), leonie.sterz@kit.edu (L. Sterz), oliver.raabe@kit.edu (O. Raabe), christopher.gerking@kit.edu (C. Gerking).

https://doi.org/10.1016/j.infsof.2026.108121
Received 9 September 2025; Received in revised form 5 March 2026; Accepted 11 March 2026
Available online 12 March 2026

They rarely address the interdisciplinary gap that makes collaboration costly and error-prone. To overcome this, there is a need for methods that bridge legal and technical perspectives, enable overcoming conceptual barriers, and support system-centric compliance assessments in evolving systems.

1.1. Contributions

To address the shortcomings regarding interdisciplinary collaboration and supported legal assessments, we propose a workflow that enables ongoing collaboration between legal experts and software engineers. This workflow aims to support joint legal threat modeling. It facilitates the identification of potential legal threats by combining a legal system viewpoint with data flow diagrams (DFDs), which are kept consistent and used for automated analysis. For our proposed workflow, we primarily focus on DFDs as they are an established representation of software architecture [17] and are widely used for threat modeling and other types of information security analyses [11,12,18–20].

Our contributions are:

- A legal assessment facts (LAF) reference metamodel as a foundation for creating domain-specific legal metamodels that capture the factual elements of a specific legal domain.
- A bidirectional incremental transformation that ensures that instances of metamodels based on the LAF reference metamodel and DFDs can be derived from each other and remain consistent.
- An extension of the LAF reference metamodel that enables the definition and annotation of undefined legal terms to elements of metamodel instances.
- An explicit pre-processing step to handle uncertainty introduced by undefined legal terms, as an extension of an existing data flow analysis.

We evaluate the proposed workflow and modeling artifacts by discussing their feasibility and applicability, and by measuring the accuracy and scalability of the extended data flow analysis. For this purpose, we use the GDPR as the exemplary legal domain, providing a concrete legal context in which to assess the applicability of our workflow and modeling artifacts.

1.2. Interdisciplinary collaboration

The contributions presented in this article are the product of long-term ongoing interdisciplinary collaboration between researchers in software engineering and formally trained researchers in legal informatics/data protection law. This is reflected in the list of authors, which comprises an equal number of scientists from each discipline. The collaboration is part of the KASTEL Security Research Labs, a German national Competence Center for IT Security, which attaches great importance to interdisciplinarity.

Regular collaborative meetings followed a rotating format in which each discipline presented relevant topics, followed by in-depth discussion. Additional experts from both disciplines were involved when needed. The goal was to initially transfer and exchange methodological knowledge between the two disciplines, thereby fostering a deeper mutual understanding of the legal methodology and background of data protection law, as well as general software engineering practices and model-driven software development.

The presented contributions were developed in close collaboration of the authors, in which all perspectives and suggestions were discussed and considered equally. By ensuring that both disciplines are given equal consideration, we aim to produce results that comprehensively reflect and support their respective interests and methodological approaches. We believe that this approach is a particularly strong way of fostering meaningful contributions at the interface between such inherently dissimilar disciplines.

1.3. Running example

To help illustrate our contributions, we define a simple system of systems as a running example and apply it in the domain of the GDPR. The example system is a *mobility provider* (e.g., rail, bus, bikes, scooters). The *mobility provider* offers its services to *customers*. To take advantage of the provided services, *customers* must first create an account in the system. The provider stores customer data for contract performance and invoicing. This customer data includes the customer's name and address, as well as information about previously used mobility services, such as a list of past trips and information about prior system usage, to individualize the website/booking platform. At fixed intervals, the mobility provider forwards customer data to a third-party *statistics provider* to get insights into improving services and internal processes. Before this, however, the customer data is pseudonymized by the mobility provider, e.g., by removing various directly identifying information such as name and aggregating address information. The calculated statistics and the pseudonymized customer data are subsequently transferred to a *marketing agency*, which uses them to provide the mobility provider with suggestions for better customer acquisition, support to increase sales, and targeted advertisements.

When collecting the customer's data, the mobility provider states three primary purposes: *creating the account* in their data management system, *calculating statistics* about their customer base, and *creating advertisements*. In the legal sense, the creation of the customer account and all successive processing of this data for the regular business operation of the mobility provider is done based on the *performance of the contract* that the customer and the mobility provider concluded when the customer account was created. For the other two purposes, the customer also has to *give consent* when creating the account.

2. Foundations

In this section, we describe our foundations regarding threat modeling, data flow diagrams, and data flow analysis. We also provide insights into legal methodology, providing descriptions of the two core concepts for legal assessments, subsumption and legal interpretation.

2.1. Threat modeling

Threat modeling is a structured approach for identifying, analyzing, and mitigating security and privacy risks during system design and development. The Threat Modeling Manifesto [21] describes an iterative process for understanding systems, identifying threats, and validating mitigations. Established methodologies such as STRIDE [20] for security threats and LINDDUN [12] for privacy threats provide structured threat classifications and mitigation guidance. Threat modeling is widely applied in industry, for example, in the Microsoft Security Development Lifecycle through the Microsoft Threat Modeling Tool, which operationalizes STRIDE. Most system-centric methodologies, including STRIDE and LINDDUN, rely on data flow diagrams (DFDs) as the primary system representation for systematic threat identification.

2.2. Data flow diagram analysis

For automated analysis of DFDs, we build on our well-validated and actively developed data flow diagram analysis *xDECAF*,[1] which was developed in previous work [22]. A central concept is the extended DFD syntax [19,22]. Notably, the extension adds labels that represent additional semantic (meta-)information and can either be determined as a characteristic of a node or of data flowing between nodes, such as specifying user roles or the sensitivity of data. Accordingly, these labels are called *node labels* and *data labels* respectively. Fig. 1, shows the

[1] https://dataflowanalysis.org/.

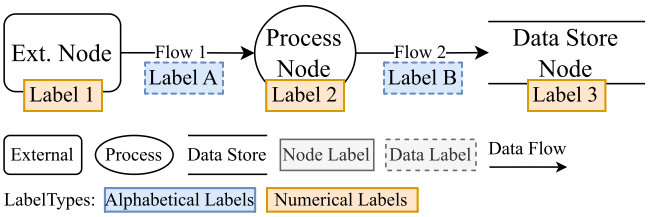


Fig. 1. Exemplary DFD in the extended syntax used by xDECAF [22], containing all syntactical elements.

graphical syntax of all relevant DFD elements. Labels are semantically grouped into *label types*. For example, in Fig. 1, labels are grouped by *alphabetic* or *numerical* values, but could also be used to indicate different levels of encryption grouped in an *Encryption* label type. All illustrations of DFDs in this article follow this syntax, and we do not include the basic elements shown in the legend.

Another core concept of xDECAF is the propagation of data labels along the flow of data. How data labels are propagated is described as the behavior of nodes. The analysis iterates over all DFD nodes, evaluates their propagation behavior, and propagates data labels accordingly. Once the propagation is finished, the analysis reduces the ambiguity of data flows represented in the DFD by extracting the set of *transpose flow graphs (TFGs)* — the transpose of a rooted directed graph, where the root is a single data sink, that each represents *one unambiguous flow of data from a one or multiple data sources to one data sink* [22]. To distinguish between DFDs and the internal TFGs, we refer to processing steps in a DFD as *nodes*, while in TFGs, we refer to them as *vertices*.

Based on the fully propagated set of extracted TFGs, constraints can be checked to analyze the modeled data flow in the DFD. To that end, xDECAF defines a dedicated constraints language.

2.3. Legal methodology

This section introduces some fundamental concepts of legal methodology relevant for understanding the reasoning behind the content of the following sections. The focus lies on the structure and application of norms, presented in a simplified form to improve accessibility for readers with a non-legal background. In this article, we refer to the legal terminology and methodology of the prevailing legal system in continental Europe, the civil law. This term refers to the codified legal system of continental Europe and should not be confused with the civil law branch in common law jurisdictions. In civil law systems, statutory texts constitute the primary source of law and hold the highest authority in legal reasoning. These texts follow a fundamental structure: they define *definitional elements of a rule*[2] as abstract preconditions for a legal consequence. The legal consequences are traditionally categorized into rights, obligations, prohibitions, and permissions. During a legal assessment, a legal expert applies a structured methodology to examine the *matter of fact*,[3] in our case, the description of the system under consideration, and captures the *relevant legal assessment facts*[4] (LAFs) in relation to a specific legal norm. This process of aligning the concrete facts with the abstract elements of a norm is called subsumption in the narrower sense[5] in continental European law [23]. The expert captures the relevant facts by examining whether the factual situation aligns with the abstract definitional elements of the legal norm. If all definitional elements of an article are met by the LAFs, the prescribed legal consequence applies.

[2] Translated from the German legal term “Tatbestandsmerkmal”.
[3] Translated from the German legal term “Tatsache”.
[4] Translated from the German legal term “Sachverhalt”.
[5] Translated from the German legal term “Subsumtion im engeren Sinne”.

However, legal norms are often intentionally formulated in abstract terms to ensure broad applicability and long-term adaptability. This is achieved through the use of *undefined legal terms (ULTs)*, i.e., concepts not explicitly defined within the statutory text, that cannot be resolved through subsumption alone. ULTs lack a fixed meaning and must therefore be interpreted in a specific scope of application. Their interpretation follows established jurisprudential methodology, which examines the wording of the norm, its systematic position within the legal framework, the legislative intent, and the underlying purpose of the regulation [24]. This interpretative process enables legal norms to remain flexible while being applied consistently and in accordance with overarching legal principles.

3. Collaborative system-centric workflow

We propose an ongoing collaboration workflow that enables interdisciplinary threat modeling between legal experts and software engineers. The workflow integrates legal expertise into iterative threat modeling by defining fixed interaction points and supports (semi-)automation to maintain continuity. At its core, it combines two discipline-specific system representations — a legal viewpoint and a data flow diagram (DFD) — and ensures their consistency through bidirectional incremental transformations. This consistency allows each discipline to work from its own perspective while enabling automated threat identification via an extended data flow analysis.

Fig. 2 illustrates the main steps and iteration cycles of the workflow. It begins with creating a system model as a DFD, which serves as the technical representation of the software system. The DFD is transformed into a legal view that reflects the system in relation to a legal norm or domain. Legal experts then enrich this view through interpretation and subsumption, including handling undefined legal terms (ULTs). From here, the workflow iterates between updating the DFD and the legal view, resolving uncertainties, and performing automated data flow analysis to identify potential legal threats. Depending on the analysis results, mitigation strategies are applied either within the technical system or documented as external measures. In parallel, legal experts can conduct additional manual assessments beyond system boundaries, triggering further iterations when necessary.

The following sections detail key aspects of this workflow and the steps they enable. First, we introduce the Legal Assessment Facts (LAF) viewpoint reference metamodel (Section 4.1), which provides the structural foundation for modeling legal viewpoints and enables steps **2–5** and **9–11**. Next, we describe how consistency between legal viewpoints and DFDs is maintained through bidirectional incremental transformations (Section 4.2), supporting steps **2** and **4**. We then address the challenge of modeling undefined legal terms by extending the reference metamodel with scope-dependent assessment facts (Section 4.3), enabling steps **3**, **9**, and **10**. Finally, we explain how automated threat identification is achieved by extending data flow analysis to incorporate legal interpretation and resolve uncertainties (Section 5), which enables step **5**. Each section illustrates the presented concepts using a running example in the context of the GDPR.

4. Modeling legal concepts

In this section, we describe our abstract basis for legal viewpoints, how we keep them consistent with DFDs and how we propose to extend the legal viewpoints to integrate the concept of undefined legal terms (ULTs) in the context of our proposed collaborative workflow. We distinguish between legal viewpoints and their structural definitions. A legal viewpoint captures legally relevant aspects of a software system, while its structure is formalized by a legal viewpoint metamodel. Concrete legal views used in the workflow are instances of such metamodels.

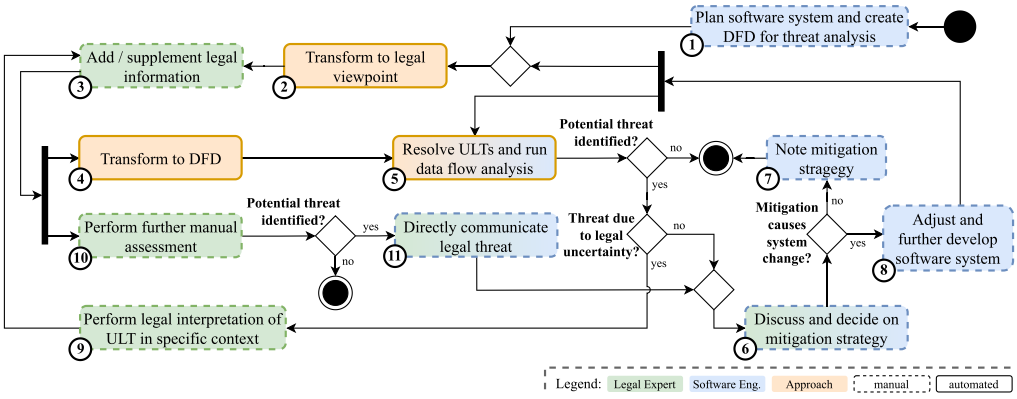


Fig. 2. Activity diagram representation of proposed workflow.

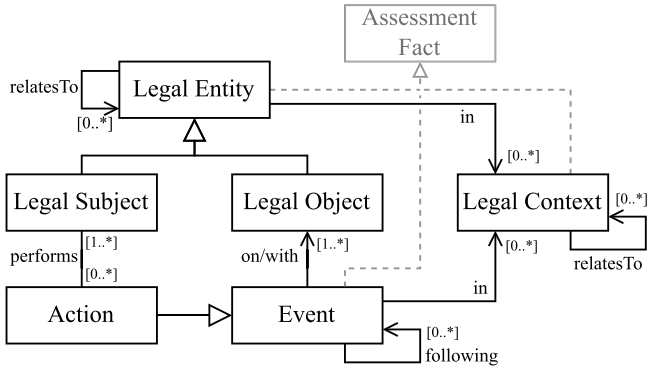


Fig. 3. Proposed legal assessment facts (LAF) reference metamodel for legal viewpoints.

4.1. Legal assessment facts viewpoint reference metamodel

To enable steps **2–5** and **9–11**, we define an abstract structural basis for modeling legal viewpoints for our proposed workflow. To this end, we derive a reference metamodel called the Legal Assessment Facts (LAF) metamodel. Its purpose is to define a reference structure (types and relations) that specifies how definitional elements (see Section 2.3) are represented in an abstract form. Metamodels created *in conformance* with our LAF reference metamodel (using types as basis and only having conforming references) can serve as domain-specific legal viewpoints in our workflow (see Section 3). These legal viewpoint metamodels specify the concrete definitional elements relevant to the domain of a particular legal norm or area. We call those metamodels *LAF-based domain-specific legal metamodels*.

When a LAF-based domain-specific legal metamodel is in turn instantiated, it produces a concrete view of the system under observation. This process of instantiation conceptionally reflects the legal methodology of *subsumption* (see Section 2.3): Based on the system under consideration (*matter of fact*), the relevant *legal assessment facts* are derived by determining which aspects of the system correspond to the abstract *definitional elements* in the viewpoint. In this analogy, the definitional elements act as types, and the legal assessment facts act as instances of these types. Thus, creating an instance of the domain-specific legal viewpoint metamodel is conceptually similar to performing subsumption by linking the abstract legal structure to the concrete characteristics of the observed system.

Fig. 3 shows our LAF reference metamodel. We grounded the structure of the LAF reference metamodel in established legal ontology research. The resulting model integrates semantical concepts from several legal and foundational ontologies. The elements *Legal Entity* and *Assessment Fact* represent abstract parent classes that combine semantical information to remove redundancy in the representation.

In contrast to the legal ontology frameworks, it does not aim to provide a comprehensive domain ontology of legal concepts or a standardized representation of legislative texts. While its elements are derived from legal ontologies, their role is as an explicit meta-level abstraction that serves as a structural foundation for defining domain-specific legal viewpoints that explicitly focus on definitional elements/legal assessment facts. Mainly, the reference metamodel provides a higher-level structure that enables the definition of transformations and metamodel extensions independently of the specific legal domain.

Exemplary LAF-based GDPR metamodel:

As an exemplary domain, we propose a LAF-based GDPR metamodel that is based on our LAF reference metamodel (see Section 4.1). To avoid a technically biased representation of the legal norm, domain elements were chosen in interdisciplinary collaboration of the authors (see Section 1.2).

For exemplary LAF-based GDPR metamodel we focused on definitions in Art. 4 GDPR as well as substantive lawfulness according to Art. 6 GDPR and the principles relating to the processing of personal data (Art. 5). The main elements include *Processing* (Art. 4(2)), *Roles* (Art. 4(7–10)), *Data*, *Legal Basis* (Art. 6(1)) and *Purpose* (Art. 4(7), Art. 5). Fig. 4 shows the metamodel, in gray, we denote the conforming LAF reference metamodel elements or relations. Processing has subclasses that correspond to *Collecting*, *Usage*, *Transferring*, or *Storage* of data (Art. 4(2)). Role has the exemplary subclasses of *Controller* (Art. 4(7)) and *Third Party* (Art. 4(10)). In conformance with the LAF reference metamodel, each *Processing* can refer to one or more following *Processings* and define *Data* that is processed. Also, each *Processing* must serve a specific *Purpose* and be done based on a valid *Legal Basis*. As we want to represent the whole system, we specify that *Data* might be any kind of data processed by the system. The only other explicitly modeled kind of data is *Personal Data* (Art. 4(1)), which references a *Natural Person*. A *Legal Basis* can either be the given *Consent* from a *Natural Person*, the performance of a *Contract*, or to comply with a legal *Obligation*. A *Legal Basis* must always be defined for at least one specific *Purposes*.

Other principles of the GDPR, like other types of roles and special categories of personal data (Art. 9), for example, are not represented as first-class entities. We deliberately chose this pragmatic approach, as this is intended as an illustrative example, and many principles of the GDPR aim to cover corner cases that do not apply to most systems. However, legal experts could already use instances of our exemplary metamodel as a basis to consider other, more detailed legal matters by consulting additional information or focusing on other legal norms related to the system context.

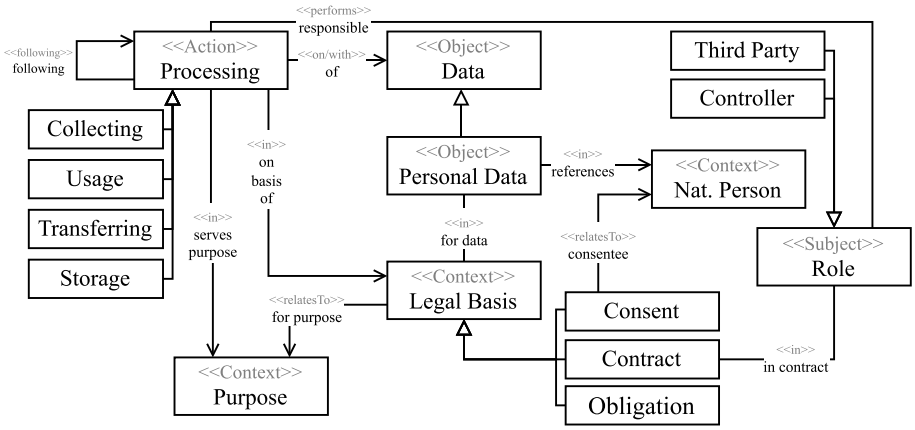


Fig. 4. Exemplary GDPR LAF metamodel, with the conforming counterparts in the LAF reference metamodel in annotated in gray.

4.2. Ensuring consistency

To ensure consistency between domain-specific legal metamodels and DFDs, we define a bidirectional incremental transformation at the meta-level that is compatible with all domain-specific metamodels that conform to our LAF reference metamodel. This enables steps **2** and **4** in our workflow (see Fig. 2). We identify elements within each metamodel that are semantically equivalent to their counterparts in the other metamodel.

DFD *Nodes* represent processing steps, which directly correspond to the *Event* and *Action* in the LAF reference metamodel. As there might exist domain-specific subclasses of events and actions, ambiguity exists when transforming from DFDs to a LAF-based domain-specific metamodel. To enable a bidirectional transformation and resolve ambiguity, we utilize DFD *Labels* (see Section 2.2). For each subclass of event or action in the LAF-based domain-specific metamodel, we create a dedicated label that is annotated to the corresponding DFD nodes. DFD *Flows* lack a direct representation in our LAF reference metamodel. However, they are effectively represented through the *following* attribute of *Events*. A flow from *Node A* to *Node B* corresponds to a *Event/Action A*, which refers to *Event/Action B* as its following Event. If an Event/Action is performed on/with a *Legal Object*, the Object itself corresponds to the data flow variable flowing in or out of the node that corresponds to the Event/Action.

To represent the remaining LAF reference metamodel elements, we propose to transform them to labels, annotated to the DFD nodes or flows. We handle *Legal Subjects* and *Legal Contexts* the same way as subclasses of Event/Action: For each subclass in the LAF-based domain-specific metamodel, we create a label type that combines all labels that correspond to instances of the subclass. Labels for Legal Subjects are annotated to the nodes that correspond to the Action that is performed by the specific Legal Subject, while labels for Legal Contexts are annotated either on nodes (if the element that is *in* the context is a Legal Subject or an Event) or on flows (if the element that is *in* the context is a Legal Object).

To enable incremental transformations, we create a trace T which induces the *bijective* mapping function τ for each transformation execution.

Application to running example:

Fig. 5 shows an excerpt of the running example (see Section 1.3) as an instance of the LAF-based GDPR Metamodel. The excerpt focuses on the data processing activities of the mobility provider and the interactions with the statistics provider. The mobility provider acts as the *Controller*, responsible for collecting and subsequently processing the *Customer data*. Purposes and legal bases are also represented as instances of the corresponding type. All processing instances, described in Section 1.3, are connected with the *following* relation and have input and output relations to their corresponding data.

Fig. 6 shows the result after transforming the GDPR LAF instance to a DFD. As shown, the following processes, like the account creation and saving of the customer data, are represented as flows between nodes. Nodes and flows have annotated labels, highlighted by color, that describe the different legal information associated with them.

4.3. Modeling undefined legal terms

As described in Section 2.3, the legal value of undefined legal terms (ULTs) depends on the observed scope of a system or situation during a legal assessment. ULTs can be used in every part of a legal norm, including in the definitional elements. In a LAF-based domain-specific legal metamodel instance (see Section 4.1), however, the actual expression of the ULT depends on the scope under consideration. To represent such information in the context of our approach, we create a *Scope-dependent Assessment Fact (SAF)* metamodel as an extension of our LAF reference metamodel. For this extension, we assume that only those ULTs are suitable for modeling whose possible expressions in the system context can be represented as a closed set of discrete values. Each *Expression* of a SAF reflects one such value that the ULT may assume and is contained in an *SAF* representing the general term of the ULT. ULTs whose meaning is inherently continuous, temporal, or highly vague, so that they cannot be meaningfully abstracted into a discrete set, cannot be modeled. For example, the ULT `reasonable time` cannot be represented, since DFDs are stateless and do not capture runtime behavior or temporal aspects. A ULT such as `adequate security` may, depending on the chosen granularity of consideration, be abstracted into a simplified closed set containing its expressions, such as {true, false}. The focus of this extension is therefore not on covering all possible ULTs or their expressions, but on those ULTs that can be modeled in the combined context of the system, the legal domain, and the legal matter under consideration.

Fig. 7 shows the structure of our proposed SAF metamodel. *SAF Annotation* serves as the extension mechanism for the *Assessment Facts* superclass of all elements in our LAF reference metamodel (see Section 4.1). This allows SAFs to be annotated without actively changing the LAF reference metamodel or LAF-based domain-specific legal metamodels. *Scope-dep. Assessment Fact* represents a modelable ULT, and the associated *Expressions* represent the closed set of possible meaningful expressions of the ULT. The *SAF Annotation* annotates its associated *Scope-dep. Assessement Fact* to the referenced element of a LAF-based domain-specific legal metamodel instance. However, as described in Section 2.3, the expression of an ULT and as a consequence of a *Scope-dep. Assessment Fact* depends on the specific *Scope* in which the

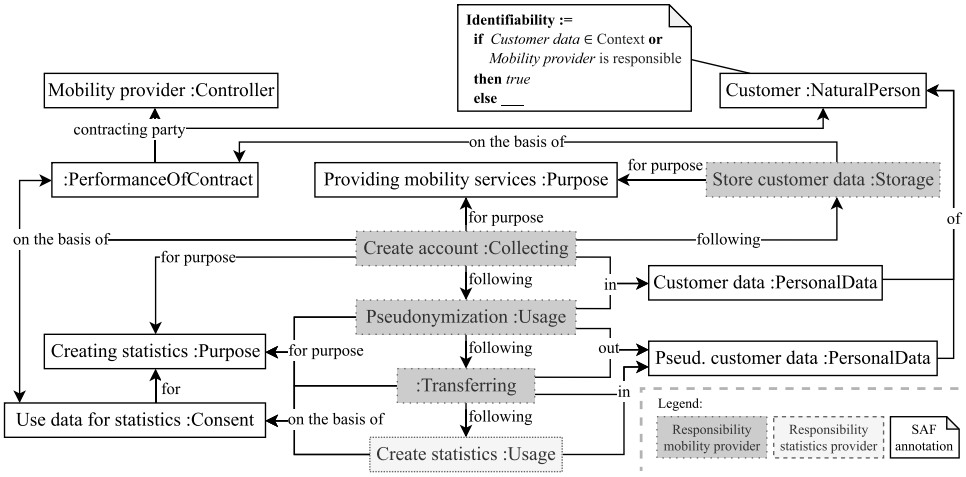


Fig. 5. Excerpt of LAF-based GDPR Metamodel instance of the running example system.

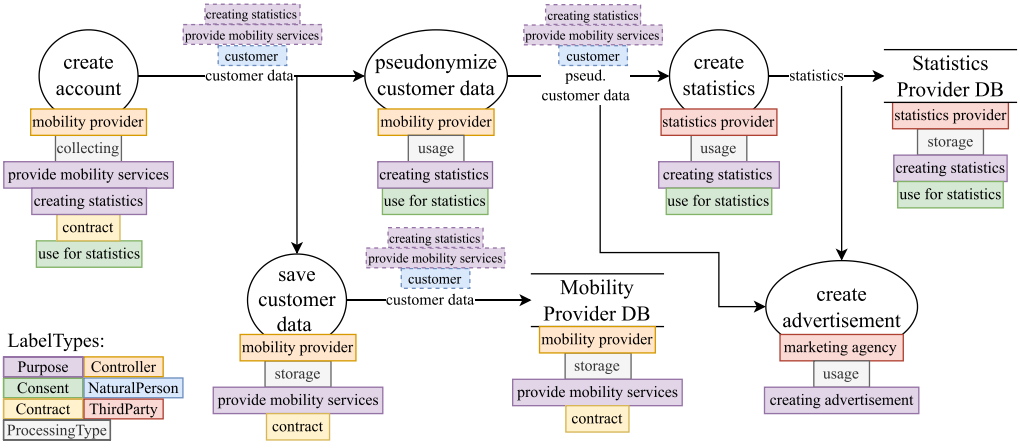


Fig. 6. Running example as DFD, transformed from GDPR LAF instance shown in Fig. 5.

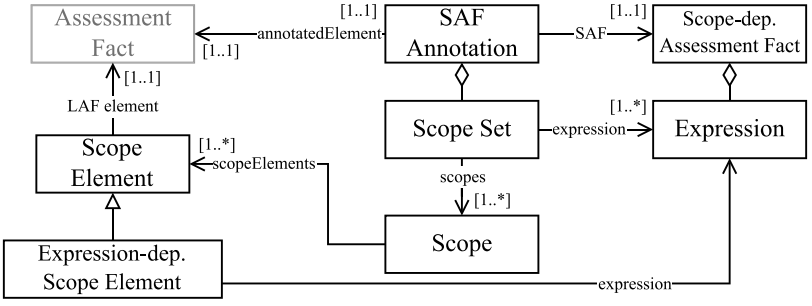


Fig. 7. Proposed SAF metamodel extension of our LAF reference metamodel.

assessment fact is considered. To accommodate this, *SAF Annotations* contain *Scope Sets*. Each *Scope Set* represents the *Scopes* in which the annotated SAF has the referenced *Expressions*. A *Scope* is a subset of the instance of a LAF-based domain-specific legal metamodel (via the references of *LAF Scope Elements*), combined with other already existing SAF Annotations, on whose Expression the current SAF Annotation might depend (via the references of *Expression-dep. Scope Elements*). Therefore, *Scopes* represent a focused, narrow view of the system.

The influencing factors that determine the actual expression of a ULT can be numerous and fundamentally open-ended. These factors are not limited to the boundaries of the system model or the responsible organization but can include external circumstances, societal conditions, or domain-specific considerations. This openness is precisely the reason why undefined legal terms are used in legislation: they allow norms to remain adaptable to unforeseen future developments and evolving contexts without requiring constant amendments to the legal text (see Section 2.3). Consequently, an exhaustive and explicit modeling of all such factors is neither practical nor desirable. Instead, the goal of our SAF extension is to enable legal experts, within the proposed collaborative workflow, to incorporate *the results* of their legal interpretation and

subsequent subsumption into instances of a LAF-based domain-specific legal metamodel (steps **3**, **9**, and **10** in Fig. 2).

Identifiability in running example:

In our running example, we take a simplified look at the ULT of customer *identifiability*: In the factual case, a natural person is either identifiable or not, so we define the set of expressions as *true* and *false*. Regarding the *identifiability* of the customer in our running example, we create a corresponding SAF annotation that references the customer. In our example, the mobility provider pseudonymizes the customer data before sending it to the statistics provider. The resulting pseudonymized customer data, however, remains of type *Personal Data*, as pseudonymization is not enough to fully anonymize data according to GDPR Rec. 26. This is why, for the purpose of creating statistics, the consent of the customer is needed. However, it is not clear if the customer is actually *identifiable* by the statistics provider using the provided pseudonymized data. Recital 26 GDPR states that data can only be considered anonymous — anonymity being the opposite of identifiability — if, taking all objective factors into account, it is unlikely that a person can be identified based on the data. In our example, however, this depends on external factors, such as the economic resources and technical capabilities of the statistics provider, as well as on the data itself, the type of pseudonymization applied, and the overall system architecture. As shown in Fig. 5, for the SAF Annotation of *identifiability* of the customer in our running example, we assume that a legal expert has defined that the customer is identifiable in scopes when the non-pseudonymized data is present or when the Mobility Provider is responsible for the processing. For every other scope, it is not defined whether the customer is identifiable or not.

5. Enabling automated threat identification using data flow analysis

To enable step **5** in our collaborative workflow (see Fig. 2), we extend the data flow analysis of xDECAF (see Section 2.2) for legal threat identification. In this section, we describe how we pre-process the TFGs of xDECAF prior to analysis by creating legally relevant scopes and resolving the SAF annotations, thereby dealing with their inherent uncertainty.

5.1. Creating legally relevant scopes

To answer legal inquiries, legal experts often focus only on a partial view of the system that is centered around certain events or the actions performed by individual legal subjects. Reducing the investigated scope allows for more concrete subsumption and legal interpretation of undefined legal terms, which in turn enables them to make more detailed statements. Following this practice, we pre-process the TFGs that are extracted by xDECAF and create legally relevant scopes before analysis. These scopes center around events and legal subjects, and are represented by the *connected subject-induced subgraphs* of all TFGs, over the power set of all subjects. As this is done on DFDs that where transformed from a LAF-based domain-specific legal metamodel instance, we use all labels that correspond to legal subjects. We define connected subject-induced subgraphs as:

Let S be the set of all legal subject labels in a TFG. For each vertex $v \in V$ of a TFG

$$\sigma(v) \subseteq S$$

denotes the subset of subject labels annotated to the vertex, e.g., the legal subjects associated with the corresponding event. For any set $S \in \mathcal{P}(S)$, we define a subject-induced subgraph as

$$G[S] = G\big[\{v \in V \mid \sigma(v) \subseteq S\}\big].$$

Since the subject-induced subgraph may not be connected, we then extract its individual connected components. Doing this while iterating over all $S \in \mathcal{P}(S)$, we obtain all subject-induced TFGs.

When looking at the DFD of our running example shown in Fig. 6, xDECAF extracts three distinct TFGs: **(a)** *create account* to *Mobility Provider DB*; **(b)** *create account* to *Statistics Provider DB*; **(c)** *create account* to *create advertisement*. In the example, $S = mobility\ provider, statistics\ provider, marketing\ agency$. For the TFG **(c)**, as there are no vertices without association to a subject (i.e., transformed from an event), we extract *six* individual subject-induced TFGs.

5.2. Defining the contextual signature of elements

To resolve SAFs with respect to TFGs, we must first identify the TFG vertices affected by a SAF Annotation. To that end, we utilize the mapping function τ that is induced when transforming between the LAF-based domain-specific legal metamodel and DFD (see Section 4.2) and map an annotated *Assessment Fact* to its corresponding DFD element. How and which vertices are affected depends on the concrete type of annotated *Assessment Fact*: Annotated *Events* directly correspond to a single vertex. For annotated *Legal Objects*, we look up all *Events* in the LAF-based domain-specific instance that refer to the annotated *Legal Object* and map each of the *Events* to their corresponding vertex. During the transformation between LAF-based domain-specific metamodel instance and DFD, *Legal Subjects* and *Legal Contexts* are transformed into labels. Consequently, vertices with corresponding labels are affected by the SAF, which applies to *Assessment Fact* of those three types. To use xDECAF's analysis functionality, we additionally need to represent all SAFs and their discrete *Expressions* as label types and labels, in order to annotate them and address them in the analysis. To this end, we add a label type with labels that correspond to each *SAF* and *Expression* of the *SAF Annotations*.

We resolve the ambiguity of a SAF by checking if any *Scope Sets* of the SAF are *satisfied* by vertices in a TFG. For this, the vertex of the TFG currently under consideration must be *in* one of the *Scopes* specified in the *Scope Set*. For this, we do not only look at a TFG vertex itself, but create the *contextual signature* $Signature_{tfg}$ of a vertex, within a TFG. We define this contextual signature of a vertex as the combination of the vertex v itself, all vertices u with $dist(u, v) = 1$ in the TFG, the set of all flows in or out of v, and all labels on v that are either defined on the vertex or propagated along the data flow. For a *vertex v* to be *in* a *Scope Y* of a *Scope Set Z*, and thereby satisfy it, we use the inverse of τ to map $Signature_{tfg}(v)$ back to elements of the LAF-based domain-specific metamodel instance and compare it with Y:

$$v\ isIn\ Y \Leftrightarrow Y \subseteq \tau^{-1}(Signature_{tfg}(v))$$
$$v\ satisfies\ Z \Leftrightarrow \exists Y \in Z : v\ isIn\ Y$$

5.3. Resolving SAF annotations

To allow the legal expert more flexibility in expressing SAF Annotations in LAF-based domain-specific metamodel instances, our SAF model (see Section 4.3) does not require SAFs to cover every possible *Expression* or possible *Scope*. However, when trying to resolve SAFs for analysis, this creates uncertainty about the value of a SAF in specific scopes. To handle this uncertainty, we first distinguish between *fully resolvable SAF Annotations* and *uncertain SAF Annotations*. For *fully resolvable SAF Annotations*, each affected vertex in a TFG can satisfy at least one *Scope Set*, consequently resolving the SAF Annotation to at least one distinct value. An *uncertain SAF Annotation* either does not define scopes for each distinct *Expression*, or there is at least one affected vertex in the TFG that cannot satisfy any *Scope Set*.

To handle the introduced uncertainty, our approach for resolving *uncertain SAF Annotations* follows principles of design space exploration on software architecture models. If a SAF Annotation cannot be resolved to at least one distinct Expression for a vertex, we consider all

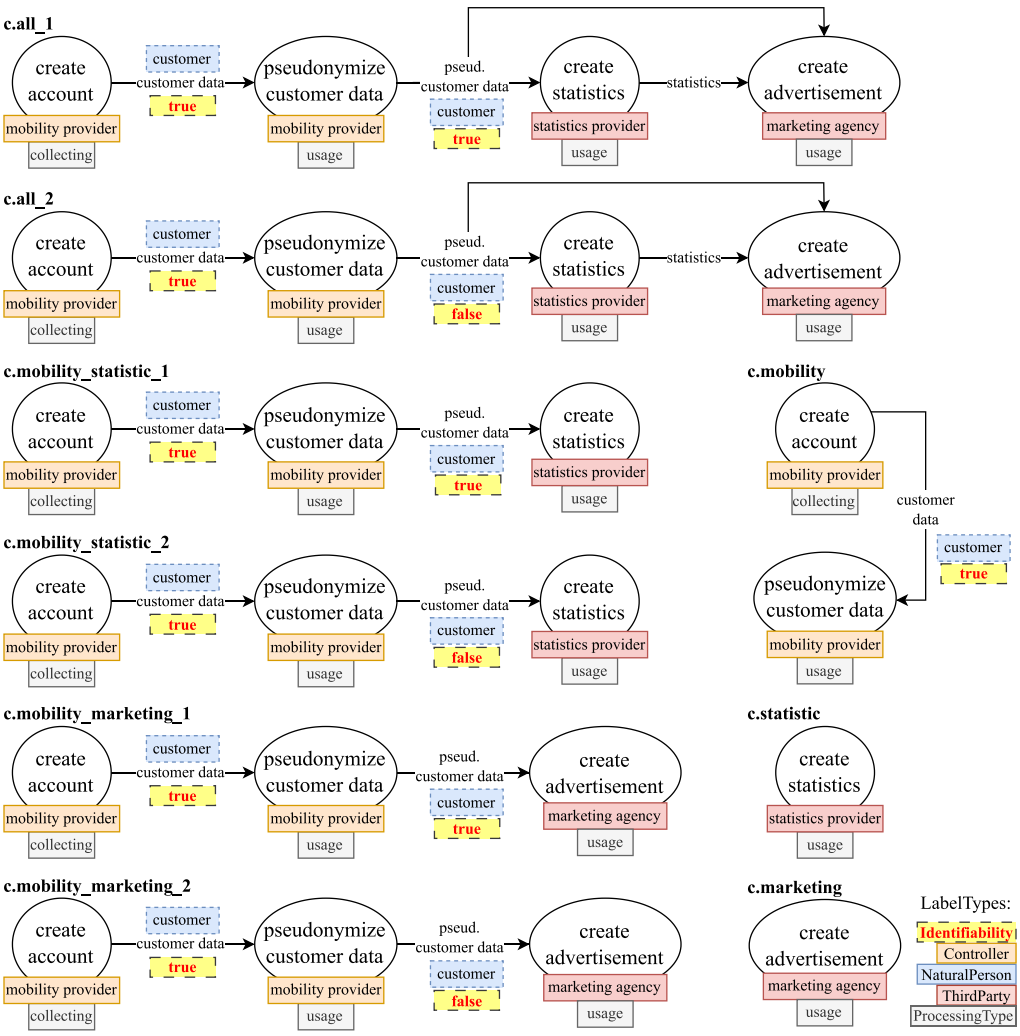


Fig. 8. Resulting nine TFGs of the running example after resolving SAF Annotations.

Expressions of the SAF and continue as if the affected vertex could satisfy all Scope Sets for each Expression.

If multiple *Expressions* of the same SAF are applicable to a vertex, we create a variant of the current TFG by cloning it and adding the label to the vertex in each variant that corresponds to the applicable *Expression*. This approach creates multiple copies of the same TFG, each with a distinct *Expression* of the SAF annotated to the vertex.

How the label is annotated again depends on the type of annotated GDPR LAF element: If it is of type *Legal Context* or *Legal Object*, we annotate the label as a data label, either flowing to or from the vertex in the variant. For all other annotated elements, the label that corresponds to the *Expression* is added as a label to the vertex directly. Once a variant TFG has been processed, we add the TFG to the list of all TFGs to further check if other SAF Annotations still need to be resolved.

Resolving SAF of identifiability in running example:

xDECAF extracts three individual data flows as TFGs (see Section 5.1). For the TFG **(c)**, we create six subject-induced TFGs. The SAF Annotation of *identifiability* of the customer does not explicitly define contexts where the identifiability is *false*, which creates uncertainty in the *create statistics* vertex. As a result, we create variants for each of the possible expressions, true and false. Fig. 8 shows the resulting nine TFGs that are used for the analysis in xDECAF.

5.4. Static data protection assessment constraints

As described in Section 5.3, we build upon and extend our data flow analysis framework xDECAF for the identification of potential legal threats. The analysis applies data flow constraints to the extracted TFGs to detect data flow violations (see Section 2.2), which in our case represent potential legal threats.

Constraints are defined using the constraints language of xDECAF. As examples with regard to our running example in the context of the GDPR, we created reusable constraints that address the applicability of the GDPR (Art. 2), purpose limitation (Art. 5(b)), subjective lawfulness (Art. 6(1)), and consent for all purposes (Recital 32.5). The authors, with a background in legal informatics (see Section 1.2), selected these aspects with particular emphasis on *purpose*, which is central to many GDPR-related legal questions. A short description of their concrete meaning and legal background is provided in our dataset [25]. These constraints do not depend on the instance of our exemplary LAF-based GDPR Metamodel but rather can be reused for all instances of the LAF-based GDPR Metamodel, e.g., for different systems. To consider SAFs in the constraint definition, the labels that correspond to the SAF Expressions need to be referenced in the constraint. However, as Scopes and Expression definitions might differ between legal experts, constraints that take SAFs into account cannot universally be reused.

As an example, we define a constraint that includes the identifiability of the customer. The constraint shown in Listing 1 defines that data

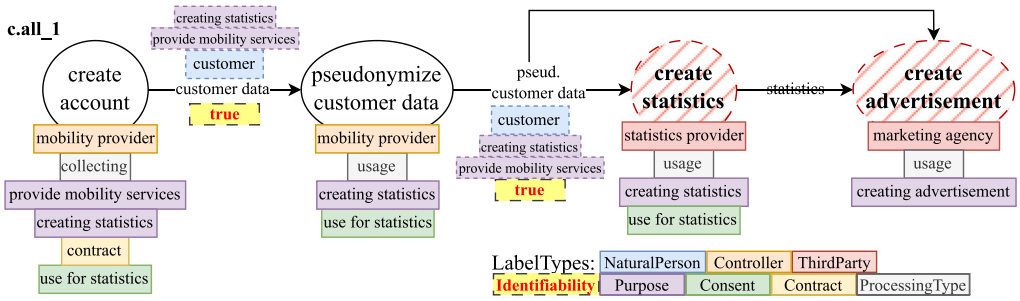


Fig. 9. TFG c.all_1 of our running example with highlighted violations that result from the constraint defined in Listing 1.

```
1 Data
2    with NaturalPerson.customer and
3    with identifiability.true
4 never flows to
5 Vertex
6    without Controller.mobility provider or
7    with Consent.$consent
8 where $consent.isEmpty
```

Listing 1: Analysis constraint for identifiability of running example.

of the customer, which also identifies the customer, shall not flow out of the responsibility of the mobility provider if the customer has given no fitting consent for further processing.

Fig. 9 shows the resulting violation in TFG c.all_1 of our running example. The *create statistics* and *create advertisement* vertices result in a violation, as they are not within the responsibility of the mobility provider and/or do not have the fitting consent of the customer, but have data flowing into them with which the customer can be identified.

In the case of our workflow (see Section 3), it is clear that the identified threat is due to legal uncertainty regarding the identifiability of the customer. This would prompt the legal experts to perform additional legal interpretation regarding the identifiability in the specific scopes where the threats are identified (step **9**). In the example, legal experts should further investigate and interpret identifiability in the TFGs that produce a violation, i.e., whether the customer is actually identifiable by the statistics provider and marketing agency in these data flows, extending the SAF Annotation with their results (step **3**). This way, if the uncertainty of an ULT can result in a potential legal threat, legal experts are directly pointed to the exact reduced scope in order to try to resolve the uncertainty.

6. Evaluation

The evaluation of our work aims to achieve two goals. First, we investigate the proposed collaboration workflow. To this end, we examine the generalizability of the LAF reference metamodel by grounding its elements in established legal and foundational ontology research. We further demonstrate how the reference metamodel supports the definition of domain-specific legal viewpoints. For the overall collaboration workflow, we provide a discussion, drawing on related work and perspectives from both the legal informatics and software engineering authors, that reflects on our findings beyond the exemplary case shown throughout this paper. We refer to this as evaluating the generalizability and applicability of the workflow. Second, we evaluate the extension of the xDECAF analysis. This includes investigating the correctness of the extension in handling ULTs and examining the scalability of the analysis.

Based on these goals, we define the following evaluation questions:

EQ1 How generalizable and applicable is the proposed collaboration workflow?

EQ2 Does the pre-processing of TFGs correctly handle and resolve SAFs?

EQ3 How does the execution time of the extended analysis scale regarding different aspects?

We further include a preliminary evaluation of the completeness of our exemplary LAF-based GDPR metamodel and the accuracy and scalability of the bidirectional transformation between DFDs and LAF-based GDPR metamodel instances that are derived from related work in our dataset [25]. We do not assess the usability of our overall approach. Usability is typically evaluated through user studies and is centered around tool support, including a concrete syntax used for modeling. Neither of them is a direct contribution of this work.

6.1. Evaluation design

For **EQ1**, we rely on qualitative discussions that are based on findings of or during the interdisciplinary collaboration that is at the center of the contributions of this article and draws on the perspectives of both legal informatics and software engineering authors. As it is central to enabling our workflow, we assess the generalizability of our LAF reference metamodel. To this end, we first detail the semantical grounding of each element in the metamodel. We further show its applicability and generalizability by applying our LAF reference metamodel structure to five approaches of related work [2,13,26–28], which provide an explicit GDPR representation on the same abstraction level as our LAF reference metamodel. To this end, we check if the representations *conform* with our LAF reference metamodel. A representation conforms to our LAF reference metamodel if each of its elements either semantically matches (conforms) to one or more types in our reference metamodel or does not have semantics that actively conflict with our LAF reference metamodel. If an element semantically conforms or actively conflicts with our LAF reference metamodel, was assessed in interdisciplinary discourse between the authors. Conforming metamodels could be used as a legal viewpoint in our proposed workflow, similar to our exemplary LAF-based GDPR metamodel. To further provide an initial showcase of its applicability to other legal norms, we also include a preliminary LAF-based Cyber Resilience Act metamodel in our dataset [25].

We complement this demonstration of generalizability with a discussion drawing on the perspectives of both legal informatics and software engineering authors. This discussion also addresses which parts of the workflow generalize to other legal domains, which steps require adaptation, and which considerations arise when integrating legal and software perspectives.

We evaluate the pre-processing of TFGs prior to analysis, by checking whether the expected subject-induced TFGs are produced for given evaluation scenarios and whether SAF Annotations are correctly handled by annotating corresponding labels (**EQ2**). We check the correct handling of SAF Annotations by comparing analysis results to a manually created expected gold standard. To quantify correctness, we use the standard metrics of precision, recall, and F1 score:

$$p = \frac{t_p}{t_p + f_p}, \quad r = \frac{t_p}{t_p + f_n}, \quad F_1 = 2\frac{p \cdot r}{p + r},$$

where t_p denotes true positives, f_p false positives, and f_n false negatives. True positives correspond to TFGs that are correctly created or violations correctly identified, false positives to TFGs or violations incorrectly created or identified, and false negatives to TFGs or violations that should have been created or identified but were missed. The evaluation scenarios use our exemplary LAF-based GDPR Metamodel. They are based on variations of our running example and selected related case studies, where violations and SAFs for ULTs such as *identifiability*, *transparency*, or *necessity* (Art. 5(3) GDPR) are introduced. The full set of scenarios, variations, and added violations is provided in our dataset [25].

We evaluate the scalability of the extended data flow analysis of xDECAF (**EQ3**) in two ways: We first conduct a worst-case time complexity analysis on the algorithm for pre-processing of TFGs (deriving subject-induced TFGs and resolving SAF Annotations, see Section 5). Second, we check the actual execution time of the extended analysis, including pre-processing, while increasing the size of individual aspects of the input models:

- The number of following Events/Action elements, as they impact the length of the data flow.
- The number of Legal Object elements as they affect the propagation effort along the data flow.
- The number of Legal Subject, as they impact the pre-processing subject-induced TFGs.
- The number of Legal Contexts and the number of Scope Sets with Expressions defined in a SAF Annotation as they play a key role in the calculation of whether a vertex satisfies a Scope Set.
- The number of resolvable and uncertain SAF Annotations. As both types affect the resolving of SAF Annotations differently.

For each of the resulting scenarios, we define an analysis constraint that does not identify any violations but requires the analysis to check every TFG and every vertex.

To better distinguish the effects of different features of the LAF reference metamodel and SAF extension on scalability, we generate individual minimal metamodel instances with an increasing number of respective instances of the metamodel element. For each run, we increase the model feature under consideration by a power of ten, starting at 10^0 and ending at 10^5. We consider 10^5 elements in a LAF-based domain-specific legal metamodel instance, far beyond typical sizes of model instances, as creating and extending the instances is mostly manual work. In fact, in the preliminary evaluation of transformation accuracy based on our exemplary LAF-based GDPR metamodel (see dataset [25]), no evaluation scenario from related work ever exceeded a number of 10^3 overall elements. We run each case 10 times and take the median execution time to exclude outliers or measurement anomalies. We deliberately omit the time needed to load the model instances, as for some of the larger model instances, this exceeds the execution time and thus distorts the results. We executed the analyses on a dedicated VM that has 4 AMD Opteron 8435 cores, 97 GB RAM, and runs Debian 11 with OpenJDK 17.

6.2. Results and discussion

In this section, we split the presentation and discussion of the results of our evaluation according to our evaluation goals.

6.2.1. EQ1 - Workflow generalizability and applicability

Continuous compliance has been recognized as a critical aspect of modern software engineering. Fitzgerald and Stol [8] emphasize that software engineering should be a continuous process encompassing compliance, security, and evolution. Other works in the field of privacy engineering highlight the necessity of ongoing legal assessments, which inherently require continuous interaction between legal and technical stakeholders [6,7], and point out the challenges of such collaboration. During our own interdisciplinary collaboration (see Section 1.2), we also found that divergent terminology, methodologies, and conceptual frameworks introduce significant risks of miscommunication, which further aligns with the findings of related work [2]. Explicitly defined interfaces for communication between the two disciplines and automated support for these interactions represent a valid way of improving collaboration. This makes a structured process, such as our proposed workflow, desirable because it reflects a need that is already recognized in practice.

Grounding of LAF reference metamodel:. The elements of the LAF reference metamodel are grounded in concepts that recur across several established legal ontologies, or ontology-like approaches. In interdisciplinary discourse of the authors, we analyzed LKIF-Core [29–31], LRI-Core [32], ODRL-RCP [33], and FOLaw [34], identified concepts that refer to definitional elements, and consolidated the findings. Each element corresponds to ontology constructs with equivalent semantics in multiple sources:

Legal Subject represents an entity capable of performing or initiating an action. This aligns with the explicit modeling of agents in several legal ontologies. LKIF-Core defines actors who bear roles in legal situations, and LRI-Core models agents as participants in normative and factual structures. ODRL-RCP expresses parties/actors (e.g., Assigner/Assignee) that perform or are assigned actions on assets; these map directly to LegalSubject as the actor role in action expressions. FOLaw models agents/actors in legal frames and treats them as participants in factual patterns.

Legal Object captures an entity to which something happens or upon which something is performed. Similar distinctions exist in LKIF-Core, where legal objects are explicitly distinguished as the passive or affected participants in legal relations and events. LRI-Core also models objects of actions and events as first-class participants in factual patterns. The ODRL-RCP Asset is the object of policy actions (the thing being used, transferred, etc.). FOLaw explicitly models "object of action" and "object of rights" as first-class categories; this aligns with LegalObject as the affected participant.

Event represents something that happens and involves one or more Legal Objects. LKIF-Core models events as factual occurrences that may have legal relevance, and LRI-Core similarly treats events as core elements in constructing factual patterns underlying legal reasoning. FOLaw models occurrences that ground legal reasoning.

Action is modeled as a specialized form of Event that is performed by a Legal Subject. This distinction appears across legal ontologies: LKIF-Core differentiates agentive actions from general events through participation roles, while LRI-Core explicitly represents actions as agent-caused events within normative reasoning structures. ODRL models actions (use, transfer, create, etc.) as first-class constructs linking parties and assets. FOLaw explicitly distinguishes actions (agent-caused events) from occurrences, providing a factual/actional pattern.

Legal Context captures descriptive circumstances or situational qualifiers that relate to a Subject, Object, Event/Action, or another Context. LRI-Core includes contextual information as part of its description of legal reasoning patterns. LKIF-Core incorporates situations and conditions to qualify legal facts. FOLaw models circumstances and situational qualifiers as part of legal facts.

Application to related work and other legal domains. Of the five investigated GDPR representations from related work, we could only identify semantic conflicts in one approach. In their GDPR representation, Matulevičius et al. [13] define artifact elements and a *PrincipleOfProcessing* element, that, while they are part of the GDPR domain, do not fit the abstraction level of our LAF reference metamodel. Excluding those elements, the remaining representation conforms with our LAF reference metamodel. Three [26–28] out of the five approaches contained elements that represent legal consequences, like obligations or rights, which we do not include in our LAF reference metamodel, as we focus exclusively on the definitional elements of legal norms (see Section 4.1). Explicitly, the approaches of Torre et al. [26] and Palmirani et al. [27] both define packages of GDPR domain elements that contain large parts that focus on the representation of legal consequences of the GDPR. In these three approaches, however, these elements do not represent a semantic conflict, but rather additions in the legal domain model and could, excluding those elements, still be used in our bidirectional transformation, SAF extension, and extended analysis of the workflow. Of the five approaches, the representation of Sion et al. [2] conforms the best: Including no additional elements and each element conforming to exactly one type in our LAF reference metamodel.

To show the applicability to other legal norms, we created an initial LAF-based Cyber Resilience Act (CRA) metamodel. The norm was chosen because it is also technology-oriented and centered on software-based systems. The metamodel represents our initial effort and has not been used in the context of the overall workflow. All GDPR representations from related work, with indications of the conforming LAF reference metamodel types, and the initial LAF-based CRA metamodel are available in our dataset [25].

Discussion on results:. The system-centric and action-oriented character of the GDPR has lent itself to our LAF reference metamodel, but this may not hold for other legal norms. Since our workflow is intended to support the identification of legal threats directly from system design or architecture, its applicability is more likely in legal domains that are technology-oriented. Adapting the workflow to a completely new legal domain requires considerable effort, especially in defining appropriate viewpoints and analysis rules based on those domains. However, our experience when creating our exemplary GDPR Metamodel and preliminary CRA Metamodel as part of our interdisciplinary collaboration showed that ontologies in general also play a role in jurisprudence. During our collaboration, the authors with a legal informatics background had no problems identifying and clustering domain elements of a legal norm and defining relations between them. While our assessment of generalizability and applicability is not based on empirical user studies, it builds on the workflow's grounding in established legal ontologies and its demonstrated applicability across five GDPR representations, providing initial evidence of conceptual feasibility.

We hypothesize that the workflow is applicable for other legal domains and showcase our initial preliminary application to the CRA, however, systematic empirical validation of this claim remains future work.

6.2.2. EQ2 - Analysis extension accuracy

Our analysis was able to correctly identify all introduced violations that result from issues pertaining to central aspects of the GDPR and issues resulting from CDAs while creating the expected TFGs in the process. Regarding TFG creation, for both considered scenarios, our analysis created the expected TFGs ($t_p = 40$) without false positives $f_p = 0$ or false negatives $f_n = 0$. This results in a precision $p = \frac{40}{40+0} = 1.0$, recall $r = \frac{40}{40+0} = 1.0$ and $F_1 = 2\frac{1*1}{1+1} = 1.0$ Regarding the correct identification of violations, our analysis identified all introduced data protection violations ($t_p = 20$) without false positives f_p or false negatives f_n. This results in a precision $p = \frac{20}{20+0} = 1.0$, recall $r = \frac{20}{20+0} = 1.0$ and $F_1 = 2\frac{1*1}{1+1} = 1.0$, indicating that our analysis is accurate. These are typical results of such analyses. Since our analysis does not use heuristics, the models analyzed have only limited expressive power, and our analysis can only check a closed set of clearly defined specifications; checking accuracy is similar to checking functional correctness.

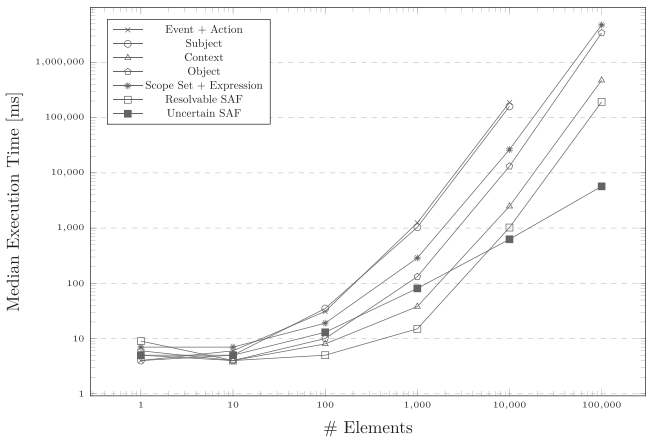


Fig. 10. Findings on analysis scalability with increased data flow or context size.

6.2.3. EQ3 - Scalability

In the following, we present the results of our scalability evaluation, including a comparison of the theoretical worst-case time complexity to the measured execution times across varying model sizes.

Algorithm 1 Subject-Induced Subgraph Construction and SAF Resolution

```
for each TFG                                          // O(|TFG|)
  Create connected subject-induced subgraphs          // O(2^|Subj| · |V|)
for each subject-induced TFG                          //O(|TFG| · 2^|Subj| · |V|)
  for each vertex v ∈ V                               // O(|V|)
    Check if v has SAF annotation                     // O(1)
    for each SAF annotation at v                      // O(|SAF.Ann|)
      for each ScopeSet                               // O(|ScopeSet|)
        Check if v isIn Scope                         // O(|Scope|)
        Clone TFG with additional expressions         // O(|V|)
```

$|ScopeSet|$ is upper bounded by $|Expr.|$, and $|Scope|$ has a logical upper bound of $2^{|V|}$. After applying these substitutions and reducing the term to its dominant factors, the Big-O worst-case time complexity is:

$$O\left(2^{|Subj.|} \cdot 2^{|V|} \cdot |V|^2 \cdot |TFG| \cdot |SAF.Ann| \cdot |Expr.|\right)$$

Looking at the results of our execution time measurements in Fig. 10, we exhibit a similar exponential growth in execution times for most model aspects. The results for increasing numbers of Events/Action and Legal Subject elements confirm their high impact on time complexity, as we were even unable to run the analysis for more than 10^4 elements. The high impact on the number of created TFGs during the analysis also manifests in a high demand for RAM, as all instances are held in storage. Even though the machine is equipped with 97 Gb of RAM, it is not configured with memory paging, resulting in memory overflowing in these cases. Given the growth of the graphs up to this point, we can assume further exponential growth in execution times. However, as already stated in Section 6.1, looking at the evaluation scenarios from related work (see dataset [25]), we can see that they rarely exceed 100 elements, for which the worst-case median execution time over all metamodel features is below 35 ms.

Looking at the worst-case scenario, for 100,000 Scope Sets and Expressions in a SAF Annotation, we observe the highest median execution time of just under 1.5 h. While these execution times are not feasible for run-time analyses, during design time, longer analysis times are common and considered feasible [19,35].

Regarding the scalability with an increasing number of SAF Annotations, we can see that scaling the number of uncertain SAF annotations shows a lower execution time for larger numbers compared to resolvable SAF annotations. This seems counterintuitive at first since uncertain SAF annotations require consideration of all possible Expressions of the SAF, while resolvable SAF annotations may require the consideration of fewer, but at most all possible Expressions. The observed discrepancy can be explained with optimizations in the early identification of uncertain SAF annotations, which are not explained in Section 5 for the sake of clarity. Most effort in the calculation of all discrete Expressions of the SAF annotation for each vertex lies in the calculation of whether a vertex satisfies a Scope Set ($O(|ScopeSet| \cdot |Scope|)$). The optimizations make it possible to skip this calculation, as all Expressions have to be considered anyway. However, an increasing number of resolvable SAF annotations exponentially increases the effort.

Comparing the impact of elements of the LAF reference metamodel to the impact of elements of the SAF extension (worst-case time complexity and measured execution times), it becomes clear that the complexity of the analysis is dominated by the model elements that have an effect on the length and number of TFGs rather than the number or size of SAFs.

6.3. Threats to validity

In this section, we discuss the external validity, internal validity, and reliability of our evaluation. Our main threats to external validity are the limited generalizability of the evaluation of the applicability of our LAF reference metamodel, as we only used five approaches of related work, all in the field of the GDPR, risking overfitting the chosen cases to our contributions. While we cannot fully mitigate this threat, the selection of approaches was guided by the availability of a clear and complete representation of their modeling concepts —– preferably illustrated with comprehensible figures — which enabled a reliable compliance check against the LAF reference metamodel. Furthermore, while the GDPR represents one of the most influential and prevalent technology-driven regulations, with widespread associated research activity, making it a good case for showcasing applicability, we also provide an initial LAF-based metamodel for the Cyber Resilience Act to demonstrate the potential applicability of our approach beyond the GDPR.

A potential threat to internal validity arises from our grounding of the LAF reference metamodel and the resolution of ULTs. For the grounding of our LAF reference metamodel in legal ontology research, there is a risk that relevant approaches in legal ontology research were missed. We try to partly mitigate this threat by grounding our LAF reference metamodel in multiple prominent approaches that are often used as a base in other works and exhibit high citation numbers. In addition, the modeling was also directly influenced by the jurisprudential expertise of the authors with a background in legal informatics. To mitigate the general risk of subjective interpretation, all contributions presented in this paper have been developed in close interdisciplinary collaboration between software engineering and legal informatics researchers (see Section 1.2). While we cannot entirely mitigate subjectivity, we also openly present direct legal references and reasoning wherever possible.

To mitigate threats to the reliability of our evaluation, we have published a dataset [25] that aids in reproducing our results. The dataset contains all code and model artifacts used for evaluation, annotated illustrations of related work regarding the applicability of our LAF reference metamodel, and further evaluation cases of the GDPR running example.

6.4. Limitations

This section outlines the limitations and assumptions of the presented approach and its evaluation. Our proposed workflow assumes that at least one or more legal viewpoints on a software system can be created, which in themselves provide a comprehensive legal picture of the system for legal experts to have real added value to work with and persist information in it. An additional limitation of the associated SAF metamodel extension is the assumption that undefined legal terms can assume a finite set of clearly distinguishable values that can be represented by the SAFs and Expressions of our SAF metamodel extension. We argue, however, that legal experts can aggregate large or infinite sets of possible values of an attribute to a reduced or summarized set of values tailored to the current legal situation.

Our evaluation is focused on the feasibility of the proposed workflow. We did not evaluate the usability of the workflow. Other related work has shown that users can model DFDs and use their analysis [36] and that DFDs are commonly used in industry to assess different aspects of information security [17]. However, with regard to our legal perspective, we plan to evaluate the usability and other quality attributes of our proposed workflow through user studies in future work.

7. Related work

In this section, we provide an overview of related work and the state of the art in the topics of legal knowledge representation, GDPR modeling approaches, and compliance checking techniques in business processes and system architectures.

Work on representing legal knowledge has produced several foundational ontologies and XML standards that enable knowledge representation and automated reasoning in the legal domain. LKIF Core [29–31] provides a foundational legal ontology that formalizes basic legal and commonsense concepts in OWL-DL to support knowledge exchange across legal reasoning systems. Its top levels describe context (location, time), then the intentional level regarding actions, agents, and roles, and lastly the legal level with legal agents, actions, rights, and powers. MetaLex XML [30] is used for the XML encoding of the structure and metadata of documents that function as a source of law. Similarly, LegalRuleML [37,38] provides an XML-based language for modeling normative rules that satisfy legal domain requirements. It captures obligations, permissions, prohibitions, temporal validity, defeasibility, and reparations, and introduces mechanisms to link rules to textual legal sources. Foundational approaches such as LRI-Core [32] and FO-Law [34] aim to cover the main concepts common to all legal domains and are explicitly grounded in common-sense conceptualizations rather than purely formal commitments. UFO-L [39] is a legal core ontology grounded in the Unified Foundational Ontology (UFO) [40,41] for representing legal relations and legal positions in conceptual models, making relational legal structures explicit through relators. CLO [42], grounded in DOLCE [43], supports the definition of domain ontologies, the definition of a juridical wordnet, and the design of legal decision support systems. Finally, ODRL-RCP [33] models legal requirements and business process permissions using concepts like permissions, prohibitions, dispensations, and obligations, and translates them via InstAL into Answer Set Programming for automated compliance checking. While these approaches focus on the formal representation of legal knowledge and reasoning, they do not explicitly address the integration of legally relevant concepts into software architecture models or support iterative collaboration between legal and technical stakeholders during system design, which is the focus of our work.

Building on the legal knowledge representations, a significant body of work has focused on formalizing the GDPR into machine-readable representations to enable compliance checking, reasoning, and integration into technical systems. GDPRtEXT [44] represents the regulation as linked data, assigning persistent URIs to all structural elements

and reusing existing vocabularies, thereby supporting precise referencing. Several ontologies aim to capture GDPR concepts for reasoning and compliance. PrOnto [27] provides a modular legal ontology for privacy, modeling actors, roles, workflows, and deontic rules, and aligns with LegalRuleML to enable automated normative reasoning. Similarly, GDPR-IS [28] focuses on linking GDPR obligations with information security concepts. While these approaches aim to make legal requirements operational for technical systems, they primarily focus on capturing the internal structure and semantics of the regulation itself. While not contradictory, our approach focuses on providing a norm-agnostic approach to modeling aspects of legal norms to facilitate a collaborative workflow. Beyond ontologies, Torre et al. [26] create UML class diagrams and OCL constraints that structure GDPR domain elements and describe how to use them for compliance rule extraction. Sion et al. [2] define a DFD-based GDPR compliance approach that introduces a GDPR-based architectural viewpoint. Their aim is to support Data Protection Impact Assessments and describe how the viewpoint could be mapped to DFDs. In contrast, our approach does not introduce a GDPR-specific architectural viewpoint, but instead provides regulation-agnostic reference metamodel, with which such norm-specific viewpoints could be created.

Compliance checking has also been addressed at the level of business processes, using BPMN-based approaches. Governatori et al. [14] use a formal contract language to encode obligations and compares BPMN execution paths against these rules under ideal semantics. Similarly, Awad et al. [15] combine BPMN-Q queries with temporal logic model checking to verify sequencing and dependency constraints. In the GDPR context, Matulevičius et al. [13] introduce an analysis method for a previously developed GDPR representation [45], by employing iterative refinement. Their analysis is designed to work on BPMN models by extracting an "As-Is compliance model", comparing that to a fully compliant GDPR model, with the goal of refining BPMN model until it is compliant. Similarly, Bartolini et al. [16] propose an ontology-based approach for integrating GDPR requirements into business process design, by annotating BPMN workflows with data protection concepts and supporting automated compliance checks during process modeling. The iterative refinement mechanisms proposed by Matulevičius et al. [13] and Bartolini et al. [16] are conceptually related to our workflow in that they support repeated compliance assessment during modeling. However, these approaches operate primarily at the level of business process models and focus on the GDPR, whereas our approach provides a less norm-specific modeling and aims to explicitly include legal experts in our workflow.

Several approaches also enrich DFDs with formal semantics and verification mechanisms to address broader privacy and security concerns. Antignac et al. [46,47] present two extended DFD syntaxes to model privacy requirements. The Privacy-Aware DFD [46] syntax extends the DFD syntax by defining data flow annotations that represent the different roles defined in Art. 4 GDPR. Antignac et al. [46] also present a model transformation between their business-oriented DFD syntax [47] and a different privacy-aware DFD syntax. Alshareef et al. [11,48] present a line of work focused on modeling and extending DFDs with privacy-related information. Their formal framework enables the annotation of DFDs with purpose labels and privacy signatures [11]. By leveraging these signatures, they automatically derive labels and check them for consistency. Ahmadian et al. [9] provide a methodology to support privacy impact assessment, using model-based privacy and security analyses to calculate the impact of the threats on the privacy targets derived from legal texts and recommendations. Complementary efforts integrate security solutions as first-class elements in DFDs [18]. Similar to the privacy-aware DFD extensions, our work also relies on DFD-based representations to reason about data processing and privacy. However, while these approaches extend DFDs with specific annotations and analysis mechanisms, our approach additionally treats DFDs and domain-specific legal metamodel instances as co-evolving artifacts and focuses on maintaining consistency.

Wright et al. [7] discuss the need for continuous Privacy Impact Assessments but do not provide a concept or solution to do so. They also say that assessments should begin at the earliest possible stages, which aligns with the aim of our proposed approach. Sion et al. [6] also discuss the need for continuous privacy impact assessments. They emphasize the advantages of design-level assessment for privacy in software-intensive systems but highlight the problem of predicting key operational aspects during design time. They envision an extension of the DevOps loop, called DevPrivOps, to tackle this problem.

8. Conclusion

In this paper, we presented a model-driven workflow that supports ongoing interdisciplinary collaboration in legal threat modeling by bridging legal and technical perspectives. Central to the approach is the Legal Assessment Facts (LAF) reference metamodel, which provides an abstract, legally grounded foundation for defining domain-specific legal viewpoints based on established legal and foundational ontologies. Bidirectional incremental transformations maintain consistency between legal views and data flow diagrams, enabling legal experts and software engineers to work with familiar representations while sharing a common system description. To showcase our contributions and evaluate the workflow, we used the GDPR as an exemplary legal domain and as a running example, allowing us to assess the applicability of our approach in a concrete, widely relevant legal context.

To address the interpretative nature of law, we introduced the Scope-dependent Assessment Fact (SAF) extension, which enables the modeling of undefined legal terms and captures scope-specific interpretation results. To manage the resulting uncertainty, we extended our xDECAF data flow analysis by generating system variants that reflect alternative interpretations and using them to identify potential legal threats within specific scopes.

Our evaluation demonstrates the generality of the LAF reference metamodel, its compatibility with existing GDPR representations, and its applicability beyond the GDPR through an initial Cyber Resilience Act metamodel. The results further show that the extended analysis correctly identifies legal threats for realistically sized design-time models.

In future work, we aim to evaluate the usability of the collaboration workflow through an extensive user study involving legal experts. We also aim to further our interdisciplinary collaboration by applying the LAF reference metamodel to additional legal norms, thereby demonstrating its broader applicability and establishing an initial set of legal viewpoints for the workflow.

CRediT authorship contribution statement

Nicolas Boltz: Writing – review & editing, Writing – original draft, Visualization, Software, Data curation, Conceptualization. **Leonie Sterz:** Writing – original draft, Methodology, Conceptualization. **Oliver Raabe:** Supervision, Methodology, Conceptualization. **Christopher Gerking:** Writing – review & editing, Supervision, Methodology, Conceptualization.

Open science and data availability

We provide a dataset [25] containing code artifacts, metamodel instances, and a pre-configured IDE for replication. We further provide additional descriptions and reasoning for presented aspects of this work, including further evaluation results and artifacts.

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work, the authors used Overleaf Writefull and ChatGPT to improve grammar and readability. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is partially based on the research project SofDCar (19S21002), which was funded by the German Federal Ministry for Economic Affairs and Climate Action. This work was also supported by funding from the topic Engineering Secure Systems of the Helmholtz Association (HGF) and by KASTEL Security Research Labs, Karlsruhe. We want to thank Tom Hüller and Felix Schwickerath, as parts of this work build on their respective theses and on their continued contributions as research assistants. We also thank Manuel Córcoles for his valuable support during the revision.

References

[1] N. Boltz, L. Sterz, C. Gerking, O. Raabe, A model-based framework for simplified collaboration of legal and software experts in data protection assessments, Informatik (2022).
[2] L. Sion, P. Dewitte, D. Van Landuyt, K. Wuyts, I. Emanuilov, P. Valcke, W. Joosen, An architectural view for data protection by design, in: International Conference on Software Architecture, ICSA, IEEE, 2019, pp. 11–20.
[3] I. Hadar, T. Hasson, O. Ayalon, E. Toch, M. Birnhack, S. Sherman, A. Balissa, Privacy by designers: software developers' privacy mindset, Empir. Softw. Eng. (2018) 259–289.
[4] A. Alhazmi, N.A.G. Arachchilage, I'm all ears! listening to software developers on putting GDPR principles into software development practice, Pers. Ubiquitous Comput. 25 (5) (2021) 879–892.
[5] A. Senarath, N.A. Arachchilage, Why developers cannot embed privacy into software systems? An empirical investigation, in: International Conference on Evaluation and Assessment in Software Engineering, EASE, ACM, 2018, pp. 211–216.
[6] L. Sion, D.V. Landuyt, W. Joosen, The never-ending story: On the need for continuous privacy impact assessment, in: European Symposium on Security and Privacy Workshops (EuroS&PW), IEEE, 2020, pp. 314–317.
[7] D. Wright, R. Finn, R. Rodrigues, A comparative analysis of privacy impact assessment in six countries, J. Contemp. Eur. Res. 9 (1) (2013).
[8] B. Fitzgerald, K.-J. Stol, Continuous software engineering: A roadmap and agenda, J. Syst. Softw. 123 (2017) 176–189.
[9] A.S. Ahmadian, D. Strüber, V. Riediger, J. Jürjens, Supporting privacy impact assessment by model-based privacy analysis, in: Symposium on Applied Computing, SAC, ACM, 2018, pp. 1467–1474.
[10] L. Sion, D. Van Landuyt, K. Yskout, S. Verreydt, W. Joosen, CTAM: a tool for continuous threat analysis and management, in: CyberSecurity in a DevOps Environment: From Requirements To Monitoring, Springer Nature Switzerland, 2023, pp. 195–223.
[11] H. Alshareef, K. Tuma, S. Stucki, G. Schneider, R. Scandariato, Precise analysis of purpose limitation in data flow diagrams, in: International Conference on Availability, Reliability and Security, ARES, 2022.
[12] M. Deng, K. Wuyts, R. Scandariato, B. Preneel, W. Joosen, A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements, Requir. Eng. 16 (1) (2011) 3–32.
[13] R. Matulevičius, J. Tom, K. Kala, E. Sing, A method for managing GDPR compliance in business processes, in: Advanced Information Systems Engineering: CAiSE Forum, Springer, 2020, pp. 100–112.
[14] G. Governatori, Z. Milosevic, S. Sadiq, Compliance checking between business processes and business contracts, in: International Enterprise Distributed Object Computing Conference, EDOC, IEEE, 2006, pp. 221–232.
[15] A. Awad, G. Decker, M. Weske, Efficient compliance checking using BPMN-q and temporal logic, in: International Conference on Business Process Management, Springer, 2008, pp. 326–341.
[16] C. Bartolini, R. Muthuri, C. Santos, Using ontologies to model data protection requirements in workflows, in: JSAI International Symposium on Artificial Intelligence, Springer, 2015, pp. 233–248.
[17] K. Bernsmed, D.S. Cruzes, M.G. Jaatun, M. Iovan, Adopting threat modelling in agile software development projects, J. Syst. Softw. 183 (2022) 111090.
[18] L. Sion, K. Yskout, D. Van Landuyt, W. Joosen, Solution-aware data flow diagrams for security threat modeling, in: Symposium on Applied Computing, SAC, ACM, 2018, pp. 1425–1432.
[19] S. Seifermann, R. Heinrich, D. Werle, R. Reussner, Detecting violations of access control and information flow policies in data flow diagrams, J. Syst. Softw. 184 (2022) 111138.
[20] M. Howard, S. Lipner, The Security Development Lifecycle, Vol. 8, Microsoft Press Redmond, 2006.
[21] Z. Braiterman, A. Shostack, J. Marcil, S. de Vries, I. Michlin, K. Wuyts, R. Hurlbut, B.S. Schoenfield, F. Scott, M. Coles, C. Romeo, A. Miller, I. Tarandach, A. Douglen, M. French, Threat modeling manifesto, 2020, (Accessed 18 August 2025) https://www.threatmodelingmanifesto.org/.
[22] N. Boltz, S. Hahner, C. Gerking, R. Heinrich, An extensible framework for architecture-based data flow analysis for information security, in: European Conference on Software Architecture, ECSA, Springer, 2023, pp. 342–358.
[23] O. Raabe, R. Wacker, D. Oberle, C. Baumann, C. Funk, Subsumtion im engeren sinne, in: Recht ex machina: Formalisierung des Rechts im Internet der Dienste, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 263–274.
[24] R. Zippelius, Juristische Methodenlehre, Vol. 93, C.H.BECK, 2021,
[25] N. Boltz, L. Sterz, O. Raabe, C. Gerking, Dataset, 2025, http://dx.doi.org/10.5281/zenodo.17061194.
[26] D. Torre, M. Alferez, G. Soltana, M. Sabetzadeh, L. Briand, Modeling data protection and privacy: application and experience with GDPR, Softw. Syst. Model. 20 (6) (2021) 2071–2087.
[27] M. Palmirani, M. Martoni, A. Rossi, C. Bartolini, L. Robaldo, Pronto: Privacy ontology for legal reasoning, in: International Conference on Electronic Government and the Information Systems Perspective, Springer, 2018, pp. 139–152.
[28] M. Geko, S. Tjoa, An ontology capturing the interdependence of the general data protection regulation (GDPR) and information security, in: Proceedings of the Central European Cybersecurity Conference, 2018, pp. 1–6.
[29] R. Hoekstra, J. Breuker, M. Di Bello, A. Boer, et al., The lkif core ontology of basic legal concepts., LOAIT 321 (2007) 43–63.
[30] A. Boer, R. Winkels, F. Vitali, Metalex XML and the legal knowledge interchange format, in: Computable Models of the Law: Languages, Dialogues, Games, Ontologies, Springer, 2008, pp. 21–41.
[31] R. Hoekstra, J. Breuker, M. Di Bello, A. Boer, LKIF core: Principled ontology development for the legal domain, in: Law, Ontologies and the Semantic Web, IOS Press, 2009, pp. 21–52.
[32] J. Breuker, R. Hoekstra, et al., Core concepts of law: taking common-sense seriously, in: Proceedings of Formal Ontologies in Information Systems, FOIS, 2004, pp. 210–221.
[33] M. De Vos, S. Kirrane, J. Padget, K. Satoh, ODRL policy modelling and compliance checking, in: International Joint Conference on Rules and Reasoning, Springer, 2019, pp. 36–51.
[34] A. Valente, J. Breuker, et al., A functional ontology of law, Towar. A Glob. Expert. Syst. Law (1994) 112–136.
[35] S. Hahner, T. Bitschi, M. Walter, T. Bureš, P. Hnětynka, R. Heinrich, Model-based confidentiality analysis under uncertainty, in: International Conference on Software Architecture Companion, ICSA-C, IEEE, 2023, pp. 256–263.
[36] K. Tuma, L. Sion, R. Scandariato, K. Yskout, Automating the early detection of security design flaws, in: International Conference on Model Driven Engineering Languages and Systems, ACM/IEEE, 2020, pp. 332–342.
[37] T. Athan, H. Boley, G. Governatori, M. Palmirani, A. Paschke, A. Wyner, OASIS LegalRuleML, in: Fourteenth International Conference on Artificial Intelligence and Law, ACM, 2013, pp. 3–12.
[38] T. Athan, G. Governatori, M. Palmirani, A. Paschke, A. Wyner, LegalRuleML: Design principles and foundations, in: Reasoning Web International Summer School, Springer, 2015, pp. 151–188.
[39] C. Griffo, J.P.A. Almeida, G. Guizzardi, Conceptual modeling of legal relations, in: International Conference on Conceptual Modeling, Springer, 2018, pp. 169–183.
[40] G. Guizzardi, G. Wagner, J.P.A. Almeida, R.S. Guizzardi, Towards ontological foundations for conceptual modeling: The unified foundational ontology (UFO) story, Appl. Ontol. 10 (3–4) (2015) 259–271.
[41] G. Guizzardi, A. Botti Benevides, C.M. Fonseca, D. Porello, J.P.A. Almeida, T. Prince Sales, UFO: Unified foundational ontology, Appl. Ontol. 17 (1) (2022) 167–210.
[42] A. Gangemi, M.-T. Sagri, D. Tiscornia, A constructive framework for legal ontologies., Law Web 3369 (2003) 97–124.
[43] A. Gangemi, N. Guarino, C. Masolo, A. Oltramari, L. Schneider, Sweetening ontologies with DOLCE, in: International Conference on Knowledge Engineering and Knowledge Management, Springer, 2002, pp. 166–181.
[44] H.J. Pandit, K. Fatema, D. O'Sullivan, D. Lewis, Gdprtext-GDPR as a linked data resource, in: The Semantic Web: European Semantic Web Conference, Springer, 2018, pp. 481–495.
[45] J. Tom, E. Sing, R. Matulevičius, Conceptual representation of the GDPR: model and application directions, in: Perspectives in Business Informatics Research, BIR, Springer, 2018, pp. 18–28.
[46] T. Antignac, R. Scandariato, G. Schneider, A privacy-aware conceptual model for handling personal data, in: Leveraging Applications of Formal Methods, Verification and Validation: Foundational Techniques, ISoLA, 2016, pp. 942–957.
[47] T. Antignac, R. Scandariato, G. Schneider, Privacy compliance via model transformations, in: European Symposium on Security and Privacy Workshops , EuroS&PW, IEEE, 2018, pp. 120–126.
[48] H. Alshareef, S. Stucki, G. Schneider, Refining privacy-aware data flow diagrams, in: International Conference on Software Engineering and Formal Methods, SEFM, Springer, 2021, pp. 121–140.