JOT
JOURNAL OF
OBJECT TECHNOLOGY

Journal of Object Technology | **RESEARCH ARTICLE**

Support for Model-Based Data Sovereignty Analysis

Sanjeev Sun Shakya[*], **Qusai Ramadan**[†], **Alexander Peikert**[‡§], **and Julian Flake**[‡]
[*]Zenjob SE, Berlin, Germany
[†]University of Southern Denmark, Denmark
[‡]University of Koblenz, Germany
[§]University of Vienna, Austria

ABSTRACT The rapid growth of data as a commodity has led to the rise of data space ecosystems, emphasizing secure and sovereign data exchange across company borders. The International Data Spaces Reference Architecture Model (IDS-RAM) and Dataspace Protocol represent frameworks designed to enable secure and standardized data sharing across organizations and sectors. As with any traditional system, to ensure compliance with data security and sovereignty requirements, adherence to IDS-RAM and data space protocols must be integrated from the design phase of data space systems. For this, our contributions in this paper are threefold: (i) a mapping from requirements of IDS-RAM and Dataspace Protocol specifications to model-based analysis checks of UMLsec and existing extensions, (ii) two data sovereignty-oriented checks as extensions to UMLsec, namely Usage Control and Transfer Process Protocol, and (iii) an applicability evaluation based on a case study of the European Health Data Space (EHDS).

KEYWORDS Data Sovereignty, Data Spaces, Reference Architecture, Model-based Analysis, Requirements, UMLsec, CARiSMA

1. Introduction

Data has transformed from a simple source of information to a valuable commodity that can be traded. Recently, many businesses and corporations have started to see data as a roduct that can be shared and monetized. The growing need to share and monetize data has created data spaces (Abbas et al. 2021; Travizano et al. 2018). The data exchange in data spaces increases not only security concerns but also data sovereignty ones (Ahmadian et al. 2018; Ahmadian 2020). According to (Lauf et al. 2021; INNOPAY & Sitra 2020), data sovereignty is the ability to create self-defined data usage rules, influence, and track data flows, and freely share and migrate data as desired. It refers to an data provider's ability to control access and usage for their own data. However, as information in data spaces moves beyond organizational borders, there is the risk of violating data sovereignty requirements.

For instance, in the Schrems II case, the Court of Justice of the European Union (CJEU) invalidated the EU-U.S. Privacy Shield framework, which was designed to facilitate the transfer of personal data between the EU and the U.S. This ruling highlighted the critical importance of data sovereignty, as it invalidated a key mechanism for international data exchange due to concerns about the lack of adequate protection of EU citizens' data rights in the U.S (Mildebrath 2020).

To address data security and sovereignty concerns in data spaces, the International Data Spaces Association (IDSA) introduced the International Data Space Reference Architecture Model (IDS-RAM) (IDSA 2024b) and the Dataspace Protocol (IDSA 2024a). IDS-RAM incorporates several elements, roles, and interactions that make up an infrastructure for the exchange of data (IDSA 2024b), while the Dataspace Protocol uses the principles and guidelines outlined in the IDS-RAM by providing technical specifications and mechanisms for standardized data exchange (IDSA 2024a).

To mitigate challenges in detecting violations of data security and sovereignty, it is crucial to incorporate the IDS-RAM and Dataspace Protocol specifications from the initial stages of data space design. Software modeling languages, such as the Unified Modeling Language (UML (Object Management Group, OMG 2017)), facilitate the design of data space system architectures

JOT reference format:
Sanjeev Sun Shakya, Qusai Ramadan, Alexander Peikert, and Julian Flake. *Support for Model-Based Data Sovereignty Analysis.* Journal of Object Technology. Vol. 24, No. 2, 2025. http://dx.doi.org/10.5381/jot.2025.24.2.a9

at a high level of abstraction, enabling the analysis of system components and their interactions. Moreover, UML provides comprehensive system documentation, which is essential for achieving the required certifications.

Research challenge. While UML-based security analysis techniques, such as UMLsec (Jürjens 2005), have been extensively applied in traditional information systems (Jürjens 2002), their applicability to the domain of data spaces remains insufficiently explored. Therefore, currently, there is a lack of understanding of the extent to which existing UML-based security extensions can be leveraged to address the security specifications of IDS-RAM and Dataspace Protocol. In addition, existing UML-based methods are primarily security- and privacy-oriented, without emphasis on data sovereignty aspects. Therefore, the capability to reason about data sovereignty and ensure compliance with IDS-RAM remains unaddressed.

Contribution. In this paper, we *partially* address the aforementioned challenges by threefold contribution: First, we map the data security, privacy, and sovereignty aspects of IDS-RAM and Dataspace Protocol to an existing UML-based analysis method called UMLsec. The goal of this mapping is to identify which IDS-RAM and Dataspace Protocol specifications currently have at least partial reasoning support during the design phase. We choose UMLsec as the basis for our mapping due to its continuous development since its introduction in 2005, providing support for various security and privacy aspects across different levels of abstraction.

Second, we propose two *static checks* as an extension for UMLsec to enable reasoning on two *data sovereignty* specifications at a high abstraction level. Specifically, we propose: (i) *Usage Control Check* which validates whether all data exchanges between parties within a data space are initiated through IDS connectors. The IDS connectors facilitate secure communication, ensuring that preconditions and usage control policies are verified prior to any actual data transfer. (ii) *Transfer Process Protocol Check* validates the sequence of actions and messages between a data provider and consumer within data spaces. The goal of this check is to ensure compliance with the expected flow of data transfer instructions.

Third, we study the applicability of our proposed checks in addition to existing UMLsec checks using a *case study* based on the European Health Data Space (EHDS) (European Commission 2022) that describes four use cases namely, cross-organizational care, second opinion consultation, tracking vitals, and data sharing for research.

Our paper is structured as follows: Section 2 provides the necessary background. Section 3 presents our methodology. Section 4 provides our mappings for the specifications of IDS-RAM and Dataspace Protocol to UMLsec checks. Section 5 presents our new extension, *Extension4IDS*, to UMLsec. Section 6 provides the evaluation based on a case study. Section 7 and Section 8 provide related work and conclusion, respectively.

2. Background

In this section, we describe necessary foundations for our work focusing on the UMLsec, IDS-RAM and Dataspace protocol.

UMLsec is a UML *profile* that can be used to enrich UML diagrams for the purpose of secure systems development (Jürjens 2005) . A profile is a generic extension mechanism that permits refining the meta-model of UML to be tailored for specific domains or platforms. UMLsec extends UML with security- and privacy-specific *«stereotypes»* and *{tags}*, that permit checking whether the architectural and behavioral aspects of an annotated UML model preserve specific security or privacy policies. Verifying UMLsec diagrams can be done automatically by tool support called CARiSMA[1], which is a publicly available open-source project (Ahmadian, Peldszus, et al. 2017).

Since its introduction in 2005, UMLsec has shown its usefulness in several industrial applications (Jürjens 2001; Jürjens & Wimmel 2001; Schneider et al. 2012) and it has been extended multiple times with various sub-profiles to support security, privacy, and data protection aspects across different levels of abstraction. For instance, the authors in (Ahmadian, Strüber, et al. 2017) introduced a UMLsec extension to enable reasoning on privacy-specific aspects like purpose control and data retention. Additionally, the author in (Peikert 2023) proposed UMLsec4IDS, a profile designed to enable reasoning about security-specific concepts introduced in IDS-RAM 3.0, focusing on security-specific aspects such as identity management and access control (Otto et al. 2019). Table 1 lists the most relevant security checks supported by UMLsec and its extensions. The first column shows the name of the check, and the second column provides a brief description of the restrictions imposed by the check.

IDS-RAM and Dataspace Protocol. The International Data Spaces Reference Architecture Model (IDS-RAM) is a framework developed by the International Data Spaces Association (IDSA). IDS-RAM provides an abstract architectural model that defines the individual components of the International Data Spaces (Connector, Metadata Broker, App Store, etc) (IDSA 2024b). IDS-RAM has evolved through several versions. In this research, we focus on IDS-RAM 4.0 (IDSA 2024b), which enhances data sovereignty through dynamic data usage control for real-time policy enforcement.

IDS-RAM 4.0 uses a five-layer structure to address the concerns and viewpoints of various stakeholders at different levels of granularity, ranging from the Business Layer, which defines participant roles and their interactions, to the System Layer, which focuses on the integration, configuration, and extensibility of logical software components. In addition, IDS-RAM 4.0 includes various requirements that need to be implemented across its layers. These requirements cover several key aspects, including security (such as secure data transmission and identity management), trust management, usage control and data provenance.

In addition to the reference architecture model, IDSA has also published a specification called *Dataspace Protocol*. It describes three protocols that complement the principles and guidelines outlined in the IDS-RAM by providing technical specifications and mechanisms for standardized data exchange.

[1] https://github.com/CARiSMA-Tool/carisma-tool/blob/master/documentation/development.md

Table 1 The Relevant UMLsec Checks

Check	Description
Data Security (Jürjens 2002)	It checks basic data security requirements such as secrecy, integrity, and freshness for the contained objects or subsystems.
Secure Dependency (Jürjens 2002)	It ensures that dependencies between objects or subsystems respect the security requirements, such as secrecy, integrity, and high-sensitivity data.
Secure Links (Jürjens 2002)	It checks whether security requirements on the communication links are met by the physical layer.
No down-flow (Jürjens 2002)	It prevents or restricts the flow of information from high-security objects or actions to low-security ones, enforced by guard objects or actions.
No up-flow (Jürjens 2002)	It prevents or restricts the flow of information from low-security objects or actions to high-security ones, enforced by guard objects or actions.
Fair Exchange (Jürjens 2002)	It ensures that a fair exchange protocol is followed between two parties, ensuring both parties either receive the goods or services or can prove misbehavior.
Provable (Jürjens 2002)	It ensures that certain actions are provable (non-repudiable) by using certificates or signatures with specified actions.
RBAC (Jürjens 2002)	It implements Role-Based Access Control, controlling access to resources based on roles and permissions.
Trusted Platform (Peikert 2023)	It ensures that connectors have trustworthy or certified software stacks.
Identity Management (Peikert 2023)	It ensures that connectors have valid X.509 certificates for authentication and encryption, with expiration checks.
Trust Management (Peikert 2023)	It ensures connectors have different levels of trust based on their security properties.
Data Access Control (Peikert 2023)	It allows data providers to define access policies for their data based on attributes and actions of data consumers.
Data Usage Control (Peikert 2023)	It ensures data consumers can enforce usage policies for the data they receive from data providers.
Provenance Tracking (Peikert 2023)	It ensures that data consumers can verify the origin and history of the data they receive.
Retention Check (Ahmadian, Strüber, et al. 2017)	It verifies if appropriate operations exist to restrict or delete personal data after its intended use.
Granularity Check (Ahmadian, Strüber, et al. 2017)	It verifies that the granularity level is respected by data transmissions.
Purpose Check (Ahmadian, Strüber, et al. 2017)	It analyzes system operations that process personal data and ensures they align with intended objectives.
Visibility Check (Ahmadian, Strüber, et al. 2017)	It identifies all data recipients and verifies that they are authorized to process personal data.

The protocols outline the structures and procedures necessary for entities to publish data, negotiate agreements, and access data as components of a data space (IDSA 2024a). This ensures consistent interactions among components, promoting interoperability and maintaining data sovereignty within data spaces (Turkmayali & Gras 2024).

3. Methodology

Ensuring compliance with IDS-RAM and Dataspace Protocol requires dedication and support from the early phases of data space system development. This is a difficult task because of two reasons. First, the IDS requirements are not explicitly listed, but rather hidden and distributed throughout a long document.

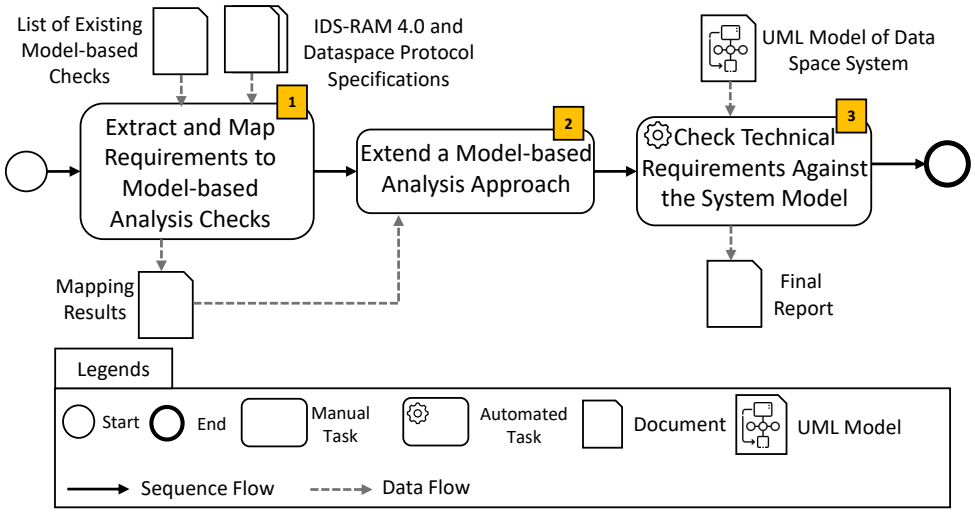

Figure 1 Methodology to achieve our research goals.

Second, IDS-RAM operates at a higher level of abstraction than common architecture models for concrete software solutions (IDSA 2024b). Consequently, extracting the requirements is a challenging task and it is not possible to automatically specify all the architectural details of the system and verify them against IDS requirements based solely on the documents of IDS-RAM and Dataspace Protocol. In addition, a one-to-one mapping from IDS-RAM and protocols requirements to model-based analysis checks is not always possible. Some requirements can be mapped to many checks as it might be needed to verify them at different design views such as structural and behavior views. Therefore, our methodology proposes the semi-automated process illustrated in Figure 1. The process of our methodology consists of three phases. In the following, we provide an overview of these phases, including their inputs and outputs. A detailed description of the phases is provided in the sections from Section 4 to Section 6.

Phase 1. Extract and Map Requirements to Model-based Analysis Checks. This phase takes the IDS-RAM 4.0 and Dataspace Protocol specification documents and a list of relevant model-based checks as inputs (see Table 1). First, we analyze the IDS-RAM 4.0 and Dataspace Protocol documents, and then extract security-, privacy-, and sovereignty-relevant requirements. Second, we map the extracted requirements to model-based analysis checks. The mapping task involves a detailed analysis of the objectives described by the extracted requirements and matching them with appropriate UMLsec checks that at least provide a *partial analysis support for them*. However, not for all requirements a mapping to a UMLsec check was identified. The output of this phase is a list of requirements and a document that describes the mapping results. Section 4 gives a detailed description of our mappings.

Phase 2. Extend a Model-based Analysis Approach. This phase takes the mapping results as an input. Based on the mapping results, one can determine which of the requirements are partially supported by existing UMLsec analysis checks and which are not supported at all. Therefore, in this phase, software developers can refine existing UML-based analysis checks and/or develop new ones to better align with the extracted requirements. Since IDS-RAM and Dataspace Protocol operate at a high level of abstraction, developers may need to consult additional resources that outline technical specifications and mechanisms to support the IDS-RAM and Dataspace Protocol. For instance, a data sovereignty requirement in the document of Dataspace Protocol states that "*Participants must adhere to the Transfer Process Protocol (TPP) during data transfer operations*". However, the technical details of this protocol are provided in a separate document as part of the technical specifications of Dataspace Protocol (IDSA 2024a; Turkmayali & Gras 2024).

In this paper, we extend UMLsec by two new data sovereignty checks as an incremental contribution and a first step toward improving alignment with IDS requirements. Sec. 5 gives a detailed description of our mappings.

Phase 3. Check Technical Requirements Against the System Model. This phase provides the evaluation based on a case study that describes the European Health Data Space (EHDS) (European Commission 2022). It takes as input a UML model enriched with security, privacy, and sovereignty requirements. The model can then be automatically verified against these requirements. In our work, we use the UMLsec profile to annotate the UML model with the requirements and CARiSMA for verifying the models against them. The output, the *Final Report*, contains the analysis results of the checks for each verified requirement. If the architectural model does not satisfy all the requirements, corrections are necessary. If all requirements are correctly enforced, developers have evidence that the data space model meets the security and data sovereignty requirements specified in IDS-RAM and Dataspace Protocol. Consequently, the model can then be used as a basis for implementation. Sec. 6 gives a detailed description of our evaluation based on the case study.

4. Extracting and Mapping IDS Requirements to UMLsec Checks

In this section, we first outline the process and the results of extracting the requirements from IDS-RAM 4.0 (IDSA 2024b) and Dataspace Protocol (IDSA 2024a) documents. Then we outline the process and the results of mapping the requirements into existings UMLsec checks.

Process of Requirements Extraction. We performed a systematic and structured process for extracting the requirements. Our process consists of the following steps:

- Scanning: We read the IDS-RAM 4.0 (IDSA 2024b) and Dataspace Protocol (IDSA 2024a) documents. The objective of this scanning was to identify and highlight all statements that define mandatory requirements.
- As-is Extraction: All highlighted statements from the scanning step are extracted exactly as stated in the documents, without any rephrasing or modification. The requirements are then stored in an excel sheet.
- Refine and Review: The extracted statements from the previous step follow different syntaxes depending on their occurrence in the documents and do not conform to a standard requirements engineering template. To improve documentation, we refine their syntax to match the structure in Listing 1. This syntax is a simplified version of the well-known MASTER template (Mustergültige Anforderungen – die SOPHIST Templates für Requirements)(SOPHIST 2016a). We chose the MASTER template for its comprehensive documentation and widespread use in real-world applications (SOPHIST 2016b). The requirements are then reviewed to ensure they are well-structured. In total we have extracted 36 security-, privacy-, and sovereignty-specific requirements.
- Categorization: In this step, we categorize the requirements into groups based on their main objectives and the document's structure from where the requirements are extracted. As a result, we distribute the 36 requirements into six groups (G1–G6), as provided in Tables 2 to 6.

```
<system> {SHALL | SHOULD | WILL}
[PROVIDE <actor> WITH THE ABILITY TO |
BE ABLE TO] <process verb> <object>.
```

Listing 1 Requirement Specification Syntax

In the following we provide an explanation of syntax elements in Listing 1:

- `<system>` (Mandatory): Refers to the system or component being specified.
- `SHALL | SHOULD | WILL` (Mandatory): Defines the requirement's level of obligation.
- `[PROVIDE <actor> WITH THE ABILITY TO | BE ABLE TO]` (Optional): Specifies that the system enables an actor (e.g., user, another system) to perform an action.
- `<process verb>` (Mandatory): The action or process the system must perform.
- `<object>` (Mandatory): The target or result of the action.

Process of Mappings to UMLsec Checks. We performed a systematic and structured process for mapping the requirements to UMLsec checks. Our process consists of the following steps:

- Listing the most relevant UMLsec checks: UMLsec consists of several checks designed to analyze different aspects, including security, privacy, resilience, and software fairness. In our work, we focused on security- and privacy-relevant checks and excluded others that were not applicable . The outcome of this step is a set of UMLsec checks provided in Table 1.
- Conceptual mapping: We performed the mapping at the *conceptual level* based on identified textual requirements and the corresponding textual descriptions of existing UMLsec checks, as provided in Table 1. This step results in mapping each requirement to zero or multiple UMLsec checks:

$$f : R \rightarrow \mathcal{P}(C), \tag{1}$$

  where R represents the set of requirements, C represents the set of UMLsec checks, and $\mathcal{P}(C)$ denotes the power set of C, indicating that each requirement may be mapped to multiple UMLsec checks or none.
- Review through modeling: We examined each mapped requirement to determine whether it could be formally modeled and verified using the corresponding UMLsec check. For this, we designed several test use cases to validate the applicability of the UMLsec checks in verifying the corresponding requirements across different system design views.

Tables 2 to 6 list the extracted requirements along with our final mappings to the corresponding UMLsec checks. The label `N/A` in the tables indicates that no UMLsec check was identified as sufficient for reasoning about the corresponding requirement. Moreover, mapping a requirement to a UMLsec check indicates *at least partial support* rather than full support. This is because each UMLsec check is designed to reasoning about a single aspect at a specific system design view (e.g., structural or behavioral). For example, requirement $G1.1$ in Table 2 states that *"The system should ensure confidentiality and authenticity of data transfer."* We mapped this requirement to the `Secure Links` check. However, this mapping indicates only partial support, as the `Secure Links` check is designed to analyze confidentiality and integrity during data transmission between two physical nodes. Specifically, it verifies whether data are transmitted over an encrypted communication path, thereby preventing unauthorized access or modification during transmission. However, this check does not verify whether for example the data remain encrypted when stored at the receiving node or whether only authorize people can access it after transmission. Therefore, our mapping in this case represents partial support.

In the following, we briefly describe each group of the extracted requirements. For each group, we provide a table listing the requirements in that group along with their final mappings to UMLsec checks.

G1. Secure Communication. Data transfer within the IDS must be secured by safeguarding communication between its

Table 2 (G1) Secure Communication

ID	Requirement	UMLsec Check
G1.1	The system should ensure confidentiality and authenticity of data transfer.	Secure Links
G1.2	The system should utilize point-to-point encryption between connectors.	Trusted Platform
G1.3	The system should implement end-to-end authorization between connectors.	Data Usage Control
G1.4	The system should transmit data over the Internet or VPN.	Secure Links
G1.5	The system should employ IDS communication protocol (IDSCP).	N/A
G1.6	The system should establish high-level protocol via WebSocket Secure (WSS).	N/A
G1.7	The system should support mutual remote attestation.	N/A
G1.8	The connectors should communicate via an encrypted tunnel (TLS).	Secure Links
G1.9	The participants should follow a five-step process for establishing a secure communication channel: identity validation, integrity validation, validation of up-to-dateness, and dynamic attributes.	N/A

components. This includes ensuring identification, authentication, and authorization of the components, as well as providing confidentiality and integrity protection for the data being transferred (IDSA 2024b). Table 2 lists the IDS-RAM requirements that need to be realized to establish a secure communication channel between IDS components. Also the table shows the result of our mappings to UMLsec checks.

G2. Secure Platform. A secure data space platform is essential to ensure the isolated and secure execution of its deployed applications. It must provide the necessary security mechanisms, enabling all applications deployed on the connector to meet their security requirements (IDSA 2024b). Table 3 lists the key IDS-RAM requirements that must be fulfilled to establish a secure data space platform, along with our corresponding mappings to UMLsec checks.

G3. Data Provenance Tracking. In IDS-RAM 4.0, data provenance tracking refers to the ability to trace the origin, history, and lifecycle of data within a data space. It ensures transparency by providing detailed information about the data's source, how it has been processed, and who has accessed or modified it (IDSA 2024b). The key IDS-RAM requirement that falls under this concept is listed in Table 4. Due to the lack of technical details within the data provenance section in IDS-RAM 4.0 document, only one requirement has been extracted. However, this high-level requirement encapsulates the core objective of data provenance tracking. We believe that this requirement can be refined into additional, more specific technical requirements. However, refining the extracted requirements is beyond the scope of this paper.

G4. Identity and Trust Management. Data spaces facilitate cross-company data exchange. In many cases, participants lack prior knowledge about other companies and their utilized components, making it difficult to fully assess the implications of such exchanges. To address this, IDS-RAM provides support for identity management, ensuring that all participants in a data space are authenticated and authorized, and that trust relationships between them are established and maintained (IDSA 2024b). Table 5 lists the key IDS-RAM requirements that must be fulfilled to establish a secure data space platform, along with our corresponding mappings to UMLsec checks.

G5. Communication Protocols. The Dataspace Protocol specification is designed to enable seamless and secure data sharing

Table 3 (G2) Secure Platform

ID	Requirement	UMLsec Check
G2.1	The applications deployed in the connector should authenticate and authorize external interfaces.	N/A
G2.2	The applications deployed in the connector should ensure the integrity and confidentiality of all communication channels and services.	Secure Links

Table 4 (G3) Data Provenance Tracking

ID	Requirement	UMLsec Check
G3.1	The system should implement data provenance tracking for transparency and accountability.	Data Provenance Tracking

Table 5 (G4) Identity and Trust Management

ID	Requirement	UMLsec Check
G4.1	All the participating entities should have certificates from Certificate Authority.	Identity Management
G4.2	The connectors should use certificates for authentication and encryption.	Identity Management
G4.3	Each connector should have a dedicated authorization service.	N/A
G4.4	Each company should provide verified company description and software manifest for operation environment certificate and component certificate.	Trusted Platform
G4.5	The system should utilize cryptographic for trust management like Public Key Infrastructure.	Trusted Platform
G4.6	Each connector should have a security profile/trust level: Base Free, Base, Trust, Trust +.	Trusted Management
G4.7	The connectors should mutually verify each other's security profiles.	Trusted Platform, Secure Dependency
G4.8	The system should provide strong isolation of components and processes.	Trusted Platform
G4.9	The system should utilize a Trusted Platform Module (TPM), Hardware Security Module (HSM), or Confidential Computing.	N/A
G4.10	The platform of the IDS connector should check the integrity and authenticity of the software stack.	Trusted Platform, Secure Dependency
G4.11	The platform of the IDS connector should protect the integrity and confidentiality of data at rest.	Secure Dependency
G4.12	The platform of the IDS connector should protect the connector from internal attackers.	RBAC, Purpose

Table 6 (G5) Communication Protocols

ID	Requirement	UMLsec Check
G5.1	Participants should adhere to the Contract Negotiation Protocol during contract negotiations.	N/A
G5.2	Participants should adhere to the Transfer Process Protocol during data transfer operations.	N/A
G5.3	Participants should adhere to the Catalog Protocol during advertisement and during discovering of data offerings.	N/A

between participants through more specific protocol specifications. They establish the necessary schemas for publishing data, negotiating agreements, accessing data, and ensuring compliance with security and sovereignty requirements (IDSA 2024a). Table 6 lists the sub protocols of the specification that need to be addressed by each data space participant.

G6. Data Usage Control. In IDS-RAM, data usage control refers to the mechanisms and policies that govern how data can be used after it has been accessed or shared. It ensures that data usage adheres to predefined rules, such as access restrictions, privacy constraints, or limitations on redistribution (IDSA 2024b). This allows data providers to maintain oversight and enforce conditions on how their data is utilized within the data space. Table 7 lists the key IDS-RAM requirements that must be fulfilled to ensure data usage control, along with our corresponding mappings to UMLsec checks.

An overview of our mapping results. Our mapping results show that many requirements have partial support, while other currently lack any model-based analysis support. This indicates a gap at the model-based development level that may require further development to enable a comprehensive analysis for the requirements of IDS-RAM and Dataspace Protocol.

5. UMLsec Extension

In this section, we propose two static analysis checks as extensions to UMLsec: the *Usage Control Check* and the *Transfer*

Table 7 (G6) Data Usage Control

ID	Requirement	UMLsec Check
G6.1	The system should regulate access to resources through usage policies.	Data Usage Control, Purpose
G6.2	Classified data should not be forwarded to nodes which do not have the respective clearance.	N/A
G6.3	Critical data should not be modified by untrusted nodes, as otherwise its integrity cannot be guaranteed anymore.	Secure Dependency, RBAC
G6.4	Data should be deleted from storage after a certain period of time.	Retention
G6.5	Personal data should be used only in an aggregated form by untrusted parties to prevent deanonymization of individual records.	Granularity
G6.6	Data allowing personal identification (e.g., faces in video files) should be replaced by an adequate substitute (e.g., pixelized) to guarantee that individuals cannot be deanonymized.	Granularity
G6.7	Two datasets from competitive entities (e.g., two automotive OEMs) should never be aggregated or processed by the same service.	N/A
G6.8	Data should only serve as input for data pipes within the Connector; it should never leave the Connector or be sent to an external endpoint.	N/A
G6.9	Data Owners should define Attribute-Based Access Control policies for their endpoints.	ABAC, Data Access Control

Process Protocol (TPP) Check. The Usage Control Check analyses a deployment diagram annotated with stereotypes specific to usage control and provides support for analyzing requirement G6.8 (see Table 7), while the TPP Check analyses a sequence diagram and offers reasoning support for requirement G5.2 (see Table 6).

The UMLsec extension described here is called Extension4IDS and uses the UML profile mechanism, a lightweight extension mechanism of the UML 2.5 metamodel (Object Management Group, OMG 2017). We extended metaclasses in the metamodel through the use of `«stereotypes»`, that can additionally be described by properties, referred to as `{tags}`. This profile was constructed using Papyrus (Eclipse 2024), an Eclipse-based UML editing tool that is freely available as an open-source Eclipse plugin. Figure 2 shows Extension4IDS profile. It shows the extended UML metaclasses, `«stereotypes»` and `{tags}`. The stereotypes and their tags are explained in Sections 5.1 and 5.2.

5.1. Data Usage Control

Requirement G6.8 in Table 7 states that *"Data should only serve as input for data pipes within the Connector; it should never leave the Connector or be sent to an external endpoint."* The goal of this requirement is to ensure that data is transferred between nodes through IDS connectors. It ensures that data shared between entities follows predefined policies. Data usage control is essential for maintaining data sovereignty, where data providers can define conditions on how Data Consumers use their data. On the consumer side an IDS connector has the task to enforce those conditions defined by the Data Provider IDS connectors play a crucial role in enabling and enforcing usage control. These connectors serve as secure gateways that facilitate data exchange while measuring the enforced usage policies (IDSA 2024b). In the following, we provide a detailed description of our presented UMLsec profile extension and the constraint of the Usage Control check that is proposed to support reasoning about requirement G6.8.

Profile Details. As shown in Figure 2, the Extension4IDS profile introduces the following two stereotypes:

- `«UsageControl»`: It extends the Dependency metaclass. It can be used to annotate dependency links between artifacts deployed on two communicated UML nodes to indicate that a Usage Control requirement must be satisfied.
- `«IDSconnector»`: It extends the Artifact metaclass. It can be used to annotate artifacts deployed on UML nodes to indicate that the nodes are equipped with IDS connectors.

Constraint of UsageControl Check. For any communication path between two nodes, A and B, we say that data usage control requirement is fulfilled if, for any `«UsageControl»`-annotated dependency between two artifacts, C and D, deployed

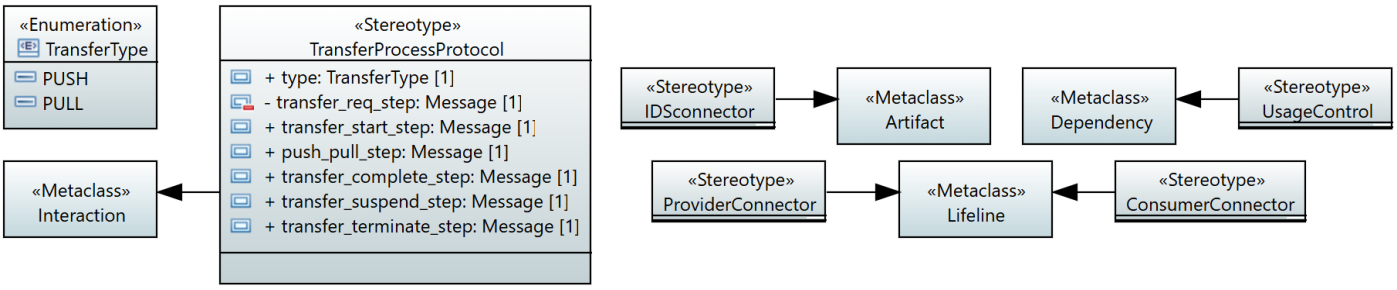

Figure 2 Extension4IDS Profile

on A and B respectively, both C and D are annotated with the «IDSconnector» stereotype.

Example. The deployment diagram presented in Figure 3 illustrates that the dependency link between the artifacts deployed in the "Router A" and "Router B" has «UsageControl» as the requirement, but only the artifact deployed in "Router A" is annotated with «IDSconnector» stereotype, suggesting that "Router B" doesn't use IDS connector for data exchange. This indicates that the participants involved in the data exchange cannot guarantee usage control, or a malicious participant can simply ignore the usage control policies associated with the exchanged data set. So, if the Usage Control Check is executed in this model, the check fails. To fix the model in Figure 3, the software artifact deployed on "Router B" should be annotated with «IDSconnector» stereotype .

5.2. Transfer Process Protocol (TPP)

Requirement G5.2 in Table 7 states that *"Participants should adhere to the Transfer Process Protocol during data transfer operations."* The goal of this requirement is to ensure that data is transferred between data provider and consumer following the standard protocol TPP. This protocol is designed to facilitate secure and efficient data exchange between a data provider and a consumer within the IDS ecosystem (IDSA 2024a). The protocol also provides information on various states involved in the transfer process and the type of messages. Figure 4 presents the state machine diagram of the various states and their order in the TPP. Each state represents a phase in the transfer process, and each transition occurs due to specific messages exchanged between the Provider (P) and Consumer (C). In the following, we present the lables of the transitions:

- **P** = Provider initiates the transition.
- **C** = Consumer initiates the transition.
- **P/C** = Either the provider or the consumer can trigger the transition.

Profile Details. To model the TPP, we use a UML *Sequence Diagram*. As shown in Figure 2, the Extension4IDS profile introduces the following new stereotypes and tags to support reasoning on requirement G5.2, which requires that participants in a data transfer should adhere to the TPP specification.

- «ProviderConnector» : Annotates a Lifeline that represents the Provider's connector

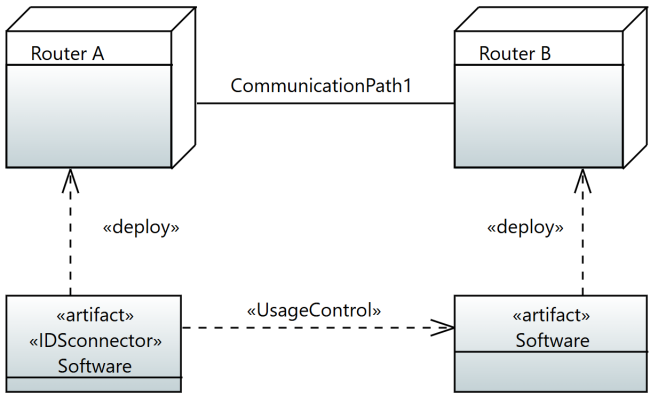

Figure 3 Deployment diagram with only one «IDSconnector» stereotype

- «ConsumerConnector» : Annotates a Lifeline that represents the Consumer's connector
- «TransferProcessProtocol»: Annotate a sequence diagram to indicate that the interaction between the provider and consumer connectors should adhere to the correct order of messages specified by TPP. This stereotype has the following tags:
  - transfer_start_step: Allows specifying the name of the start message.
  - type: allows specifying the type of data transfer; it can either be PUSH or PULL
  - transfer_req_step: Allows specifying the name of the request message
  - push_pull_step: A message that represents either the Push or Pull Step
  - transfer_complete_step: Allows specifying the name of the transfer complete message
  - transfer_suspend_step: Allows specifying the name of the message of transfer suspend
  - transfer_terminate_step: Allows specifying the name of the transfer terminate message

Constraint of TransferProcessProtocol Check. For any «TransferProcessProtocol»-annotated sequence diagram annotated with and any two lifelines, A and B, in the sequence diagram that are annotated with «ProviderConnector» and «ConsumerConnector», respectively, we say the sequence diagram preserves the specification of the TPP if the sequence

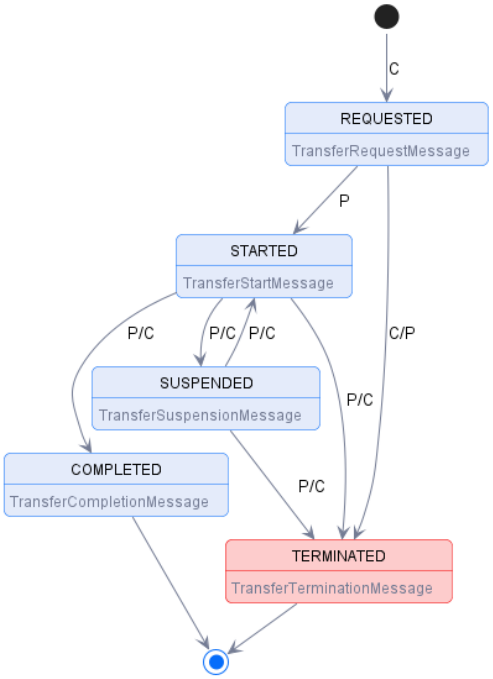

Figure 4 TPP State Machine Diagram (IDSA 2024a)

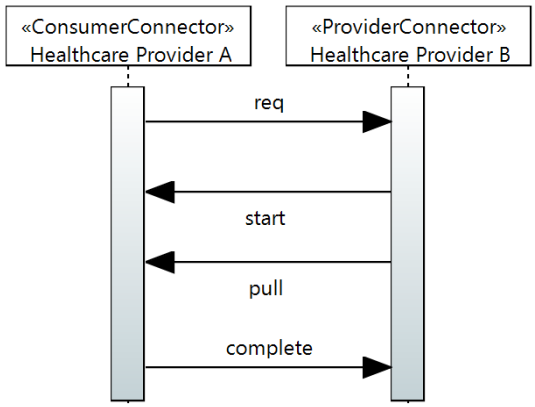

Figure 5 Wrong sender of Pull message

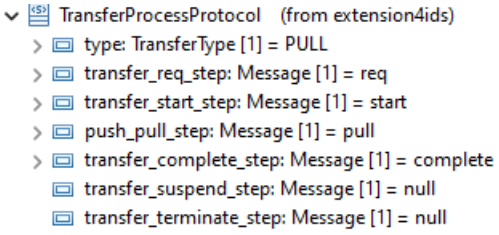

Figure 6 Tag values of «TransferProcessProtocol»

of messages between A and B follows the following order, as depicted in Figure 4:

1. The data transfer process must begin with a *"transfer request"* message, which is sent by the Consumer (B, annotated as «ConsumerConnector») to the Provider (A, annotated as «ProviderConnector»).
2. The Provider (A) must respond with a *"transfer start"* message.
3. Depending on the type of transfer:
   - For a **push transfer**, the next message must be sent by the Provider (A) to the Consumer (B).
   - For a **pull transfer**, the next message must be sent by the Consumer (B) to the Provider (A).
4. After a successful transfer:
   - For a **push transfer**, the Provider (A) must send a *"transfer complete"* message.
   - For a **pull transfer**, the *"transfer complete"* message can be sent by either the Provider (A) or the Consumer (B).
5. If *"transfer suspend"* message is sent, then there should be either *"transfer start"* or *"transfer terminate"* message.
6. **Mutual Exclusion**: If a *"transfer complete"* message is sent, neither a *"terminate"* nor a *"suspend"* message can be sent afterward, and vice versa.

Example. Figure 5 depicts a sequence diagram annotated with the TPP check related stereotypes. In the figure, it can be seen that "Healthcare Provider A" is annotated with «ConsumerConnector» and "Healthcare Provider B" is annotated with «ProviderConnector», and the whole interaction is also annotated with «TransferProcessProtocol». The tag values can also be seen in Figure 6. According to the TPP constraint, when the transfer is of type "PULL", the Pull message should be sent from the Consumer to the Provider (see Step 3). However, in Figure 5 the Pull message is sent by the Provider to the Consumer, which violates the order of messages. The check result is shown in Figure 7.

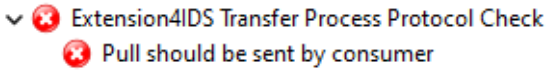

Figure 7 TPP check fail result

6. Case Study

In this section, we study the applicability of UMLsec, UMLsec4IDS, and our new checks based on a case study that represent European Health Data Space (EHDS) (European Commission 2022). We aimed at answering the following research question:

RQ1. *How applicable are the mapped UMLsec checks in the tables from Table 2 to 7 and our new proposed checks in this paper in modeling IDS-specific use cases and reasoning about their relevant IDS-RAM and Dataspace Protocol requirements?*

Since finding a single use case that encompasses all IDS-RAM and dataspace protocol requirements is challenging, we decided to model critical use cases from EHDS using UML and apply the relevant UMLsec checks to them. In the following, we describe the setup of our applicability evaluation.

Table 8 Summary of Requirements, Corresponding Checks, and UML Models

Use case	Relevant Requirement	Relevant Check	UML Diagram	Number of Elements
Use case 1	G1.1, G1.4, G1.8, G2.2, G4.1, G4.2, G6.8	Secure Links, Identity Management, Usage Control	Deployment diagram	35
	G5.2	Transfer Process Protocol	Sequence diagram	27
Use case 2	G1.2, G4.4, G4.5, G4.6, G4.7, G4.8, G4.10, G6.8	Trusted Platform, Trust Management, Usage Control	Deployment diagram	36
	G5.2	Transfer Process Protocol	Sequence diagram	25
Use case 3	G5.2	Transfer Process Protocol	Sequence diagram	27
Use case 4	G1.3, G3.1, G4.9, G6.1	Data Usage Control, Data Access Control, Data Provenance Tracking	Activity diagram	34

6.1. Setup

In the following, we describe the use cases. Then, we provide an overview of the analyzed UML models.

Use cases identification. The European Parliament and the European Council have released a proposal of how to initiate, govern and regulate a EHDS. The proposal has categorized medical data usage into primary and secondary use. The primary use of health data is more concerned with using a patient's data for the patient's health. In contrast, the secondary use of health data is more concerned with patient data for research, innovation, and policy planning. Based on the primary and secondary uses outlined in the EHDS proposal, we defined the following use cases:

- **Use Case 1**: Cross-Organizational Care - A patient currently treated by one healthcare provider receives informed care while visiting another healthcare provider. Patient information is exchanged between the healthcare providers.
- **Use Case 2**: Second Opinion Consultation - A patient shares their health records via a patient portal with a specialist in another EU country, enabling a detailed second opinion on a diagnosis or treatment plan without geographical barriers.
- **Use Case 3**: Tracking Vitals - Health Care Provider (HCP) sets tracking for a patient's heart rate for a certain time interval. The patient's wellness application (e.g., smartwatch) collects data between the defined interval and sends the data to the HCP to analyse the data and send a report back to the patient via the patient portal.
- **Use Case 4**: Data sharing for research - A research institute submits a data request to a health data access body to obtain health-specific data for research purposes, with no monetary agendas involved.

Use case modeling. Based on the description of the use cases, we identified the most relevant requirements for each use case. In the second step, using our mappings in the tables from Table 2 to 7, we determined the UMLsec checks to be applied for reasoning about requirements at the design level. Since each check can be applied to a specific UML view (i.e., diagram), we modeled the necessary UML diagrams accordingly. For example, applying the Secure Links check requires a deployment diagram, while the Transfer Process Protocol check requires a sequence diagram. In some cases, a single UML diagram was used to analyze multiple security requirements. For instance, in Use Case 1, the deployment diagram is used to reason about data security and identity management. In our paper, We use Papyrus to model the UML diagrams.

Per use case, Table 8 provides a summary of the mapped requirements, the relevant UMLsec checks, the types of UML diagrams modeled, and the number of UML elements in each diagram. In each UML diagram we modeled various components introduced by IDS, such as connectors, metadata brokers, identity providers, and clearing houses. For example, for Use Case 1 (Cross-Organizational Care), as shown in Table 8, the deployment diagram contains a total of 35 elements. This includes 8 nodes, 6 artifacts, 2 packages, 10 communication paths, 6 deployment links, and 3 dependency links. In addition to the deployment diagram, we created a sequence diagram to illustrate the behavioral aspects of Use Case 1. This model consists of 27 elements, including 9 different lifelines and 18 distinct messages exchanged between them. Due to space limitations, this section presents excerpts from our designed UML diagrams, while the full versions are available online [2].

[2] https://github.com/sanjeev55/EHDS-use-cases

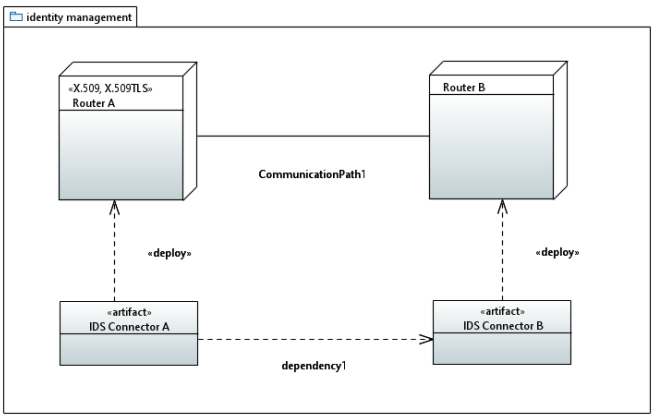

Figure 8 Incorrect Deployment Diagram

6.2. Apply UMLsec checks

In this step, we use a tool that supports UMLsec and its extension called CARiSMA[3] to: First, annotate the UML diagrams with UMLsec stereotypes and tags. Second, execute the corresponding checks. The result of executing each check indicate the following outcomes:

- Successful: The requirement is correctly specified in the UML diagram.
- Violation: The requirement is not properly specified in the UML diagram.
- Error: CRiSMA may fail to execute the check if the provided model is not consistent with the expected syntax of the input model. This can occur due to the variability of UML models.

In the latter two cases, we manually refactor the model to fix the source of the violation or error and then rerun the check. For example, Figure 8 presents an excerpt from the UML deployment diagram of Use Case 1 (Cross-Organizational Care). It illustrates two nodes, "Router A" and "Router B," which communicate with each other over a communication path using IDS connectors. According to IDS-RAM requirements G4.1 and G4.2, participating entities must use certificates for authentication and encryption. To verify whether "Router A" and "Router B" have the appropriate certificates, we apply the Identity Management Check. Upon executing the check, a violation is detected. This occurs because, as shown in Figure 8, only "Router A" is annotated with the «X.509» and «X.509TLS» stereotypes. This indicates that the Health Care Provider's connector, "Router B", lacks the necessary certification to participate in the data exchange, thereby violating G4.1 and G4.2. To resolve this issue, we refactor the model by adding the «X.509» and «X.509TLS» stereotypes to "Router B" with a valid expiration date for the certificates.

6.3. Results and Discussion

Our systematic approach to extracting and mapping requirements to model-based checks facilitates the identification of requirements that lack support for model-based analysis. Despite the structured methodology for extracting and mapping requirements to ensure comprehensive coverage of IDS-RAM and data space protocol requirements, achieving completeness is not feasible. This is because IDS requirements are not explicitly listed but rather embedded and dispersed throughout an extensive document. Additionally, IDS-RAM operates at a higher level of abstraction than conventional architecture models for concrete software solutions, leading to variations in the granularity of the extracted requirements. Nevertheless, the aim of this paper is to provide an *initial profile* of requirements for engineering a trustworthy data space system at the system design level.

Concerning research question RQ1: How applicable are the mapped UMLsec checks in the tables from Table 2 to 7 and our new proposed checks in this paper in modeling IDS-specific use cases and reasoning about their relevant IDS-RAM and Dataspace Protocol requirements? Our case study demonstrates that UMLsec and its extension allows for modeling IDS requirements in the software design phase. Specifically, applied to the four use cases in our case study our results show that we successfully modeled 19 distinct IDS-RAM and dataspace protocol requirements and analyzed them using 9 UMLsec checks. Out of these 19 requirements 17 were modeled using existing UMLsec checks while 2 concerning data sovereignty namely G5.2 and G6.8 could not be analyzed without the proposed checks introduced in this paper. This result provides valuable insight into the applicability of UMLsec and its extensions for modeling IDS requirements and reasoning about them at the design phase.

Threats to validity. A threat to *external validity* is that our applicability evaluation was based on a single case study covering 19 relevant requirements out of 24 mapped to UMLsec checks. Hence, the obtained results cannot be generalized to other case studies. A comprehensive study with a greater variety of case studies is left for future work.

A threat to *internal validity* is the lack of a formal validation of the reliability of our proposed checks. Due to the large variability of possible UML models, our checks could still be affected by errors. However, to provide insights on the the reliability of our proposed checks we developed a test suite consisting of 38 test cases classified as follows: 5 test cases for the deployment diagram and 33 test cases for the sequence diagram. Each test case covers a distinct way of modeling the deployment or sequence diagram. We included more test cases for the sequence diagram because unlike the deployment diagram the sequence diagram captures behavioral aspects of the system and is therefore highly variable. After implementing the checks we executed them against the test cases to verify that they produced the expected outcomes without errors. On average our proposed checks achieved a code coverage of 85.2% as measured using EclEmma (Hoffmann et al. 2024).

Furthermore due to practical constraints related to the limited availability of experts in model-based analysis and data spaces the same group of participants was responsible for mapping requirements to UMLsec checks as well as performing

[3] https://github.com/CARiSMA-Tool/carisma-tool/blob/master/documentation/development.md

the modeling and evaluation tasks. Therefore a potential threat to internal validity arises in the form of confirmation bias, as the participants who defined the mappings were also involved in evaluating them. While we acknowledge this threat we believe that the insights gained from our experiment still provide valuable contributions to the field of model-based analysis.

7. Related Work

Despite the emergence of data space systems since their introduction in the mid-2010s as a means to enable sovereign and interoperable data exchange, no existing approach described in the literature has been found that explores the use of model-based analysis in addressing the requirements of IDS-RAM and data space protocols. Two other approaches are worth mentioning that have the same goal – the conformance of data spaces and their software components to existing specifications in order to improve interoperability – but in a different way: the IDS Testbed[4] and Eclipse Dataspace TCK[5].

The IDS Testbed is an open-source project, which is jointly developed by the IDSA open-source community, and which combines several software components. It aims to enable companies and organizations to develop, test, and certify components that adhere to the International Data Spaces (IDS) specifications and standards.

The Eclipse Dataspace TCK project is an initiative hosted by the Eclipse Foundation. Its goal is to enable verification of compliance and interoperability across different dataspace protocols and standards. Initially the project focuses on a verification method specifically for checking compliance of implementations with the Dataspace Protocols as described in Section 2.

These two projects complement our work by providing analysis support during the development and testing phases of the system development life cycle, while our work focuses on addressing security and data sovereignty during the design phase. By amending our work with these projects, a more structured and security-aware development approach can be established.

8. Conclusion

In this paper, we aimed to provide support for engineering trustworthy data space systems from the outset of system development. To achieve this, we explored how existing UML-based security extensions can be leveraged to support addressing the security specifications of IDS-RAM and the Dataspace Protocol. Our contributions are threefold(i) a mapping from the requirements of IDS-RAM 4.0 and Dataspace Protocol specifications to existing model-based analysis checks in UMLsec, (ii) two data sovereignty checks, namely UsageControl and TransferProcessProtocol, and (iii) an applicability evaluation through a case study based on the European Health Data Space (EHDS). Our evaluation provides a good insights about the applicability of existing and proposed checks in supporting secure data exchanges, especially in contexts where data sovereignty is critical.

More recently, a draft version of IDS RAM 5.0 is released[6]. The new version will provide more technical information on how to create an architecture for a data space and is planned to be in a final state by end of the year 2025. Therefore, as future work, we aim to design new checks to support reasoning about additional IDS-RAM and dataspace protocol requirements taking into consideration the specifications of IDS RAM 5.0.

Additionally, this paper considers a static check for reasoning about Dataspace Protocol requirements. In future work, we aim to develop dynamic checks to enhance reasoning about Dataspace Protocol specifications. We also plan to apply UMLsec and its extensions to more comprehensive use cases covering nearly all IDS-RAM requirements.

References

Abbas, A. E., Agahari, W., van de Ven, M., Zuiderwijk, A., & de Reuver, M. (2021). Business data sharing through data marketplaces: A systematic literature review. *Journal of Theoretical and Applied Electronic Commerce Research*, *16*(7), 3321–3339. Retrieved from https://www.mdpi.com/0718-1876/16/7/180 doi: 10.3390/jtaer16070180

Ahmadian, A. S. (2020). *Model-based privacy by design.* Germany. Retrieved from https://kola.opus.hbz-nrw.de/frontdoor/index/index/docId/2024 (Accessed: 2024-10-01)

Ahmadian, A. S., Peldszus, S., Ramadan, Q., & Jürjens, J. (2017). Model-Based Privacy and Security Analysis with CARiSMA. In *Proceedings of the 2017 11th joint meeting on foundations of software engineering* (pp. 989–993).

Ahmadian, A. S., Strüber, D., Riediger, V., & Jürjens, J. (2017). Model-based privacy analysis in industrial ecosystems. In *Modelling foundations and applications: 13th european conference, ecmfa 2017, held as part of staf 2017, marburg, germany, july 19-20, 2017, proceedings 13* (pp. 215–231).

Ahmadian, A. S., Strüber, D., Riediger, V., & Jürjens, J. (2018). Supporting privacy impact assessment by model-based privacy analysis. In H. M. Haddad, R. L. Wainwright, & R. Chbeir (Eds.), *Proceedings of the 33rd annual ACM symposium on applied computing, SAC 2018, pau, france, april 09-13, 2018* (pp. 1467–1474). ACM. Retrieved from https://doi.org/10.1145/3167132.3167288 doi: 10.1145/3167132.3167288

Eclipse. (2024). *Papyrus.* Retrieved from https://www.eclipse.org/papyrus/ (Accessed: 2023-05-11)

European Commission, E. (2022). Proposal for a regulation of the european parliament and of the council on the european health data space. *COM (2022) 197 Final*, *140*, 1–122.

Hoffmann, M. R., Janiczak, B., & Mandrikov, E. (2024). *Eclemma - java code coverage for eclipse.* Retrieved from https://www.eclemma.org/ (Accessed: 2024-06-31)

IDSA, I. D. S. A. (2024a). *Dataspace protocol.* Retrieved from https://docs.internationaldataspaces.org/ids-knowledgebase/v/dataspace-protocol (Accessed: 2024-01-20)

[4] https://github.com/International-Data-Spaces-Association/IDS-testbed
[5] https://projects.eclipse.org/projects/technology.dataspacetck
[6] https://docs.internationaldataspaces.org/ids-ram-5-working-draft

IDSA, I. D. S. A. (2024b). *Ids reference architecture model 4.0.* Retrieved from https://docs.internationaldataspaces.org/ids-knowledgebase/v/ids-ram-4 (Accessed: 2024-09-26)
INNOPAY, & Sitra. (2020). *Data sovereignty and soft infrastructures: key enablers for the next phase of the european data economy* (White Paper). INNOPAY and Finnish Innovation Fund Sitra. Retrieved from https://www.innopay.com/en/publications/white-paper-data-sovereignty-and-soft-infrastructures-are-key-enablers-next-phase (Accessed: 2024-01-02)
Jürjens, J. (2001). Modelling audit security for smart-card payment schemes with UMLsec. In *Ifip international information security conference* (pp. 93–107).
Jürjens, J. (2002). Umlsec: Extending uml for secure systems development. In J.-M. Jézéquel, H. Hussmann, & S. Cook (Eds.), *Uml 2002 — the unified modeling language* (pp. 412–425). Berlin, Heidelberg: Springer Berlin Heidelberg.
Jürjens, J. (2005). *Secure systems development with UML.* Springer. Retrieved from https://doi.org/10.1007/b137706 doi: 10.1007/b137706
Jürjens, J., & Wimmel, G. (2001). Formally testing fail-safety of electronic purse protocols. In *Proceedings 16th annual international conference on automated software engineering (ase 2001)* (pp. 408–411).
Lauf, F., Scheider, S., Meister, S., Radic, M., Herrmann, P., Schulze, M., … et al. (2021). *Data sovereignty and data economy - two repulsive forces?. position paper.* Retrieved from https://publica.fraunhofer.de/handle/publica/301000 doi: 10.24406/isst-n-634865
Mildebrath, H. (2020). *The cjeu judgement in the schrems ii case.* Retrieved from https://www.europarl.europa.eu/RegData/etudes/ATAG/2020/652073/EPRS_ATA(2020)652073_EN.pdf (Accessed: 2024-02-13)
Object Management Group, OMG. (2017, December). *OMG® Unified Modeling Language® (OMG UML®) Version 2.5.1.* Retrieved from https://www.omg.org/spec/UML/2.5.1/PDF (Accessed: 2024-09-23)
Otto, B., Steinbuss, S., Teuscher, A., & Lohmann, S. (2019, April). *Ids reference architecture model, version 3.0.* Zenodo. Retrieved from https://doi.org/10.5281/zenodo.5105529 doi: 10.5281/zenodo.5105529
Peikert, A. (2023, October). *Entwicklung eines umlsec-profils zur darstellung von sicherheitsmaßnahmen in industriellen datenökosystemen* (Bachelor's Thesis). Koblenz.
Schneider, K., Knauss, E., Houmb, S., Islam, S., & Jürjens, J. (2012). Enhancing security requirements engineering by organizational learning. *Requirements Engineering*, *17*(1), 35–56.
SOPHIST. (2016a). *Master: Schablonen für alle fälle.* Retrieved from https://www.sophist.de/fileadmin/user_upload/Bilder_zu_Seiten/Publikationen/Wissen_for_free/RE-Broschuere_Englisch_-_Online.pdf (Accessed: 2025-01-29)
SOPHIST. (2016b). *The sophists: A short re primer.* Retrieved from https://www.sophist.de/fileadmin/user_upload/Bilder_zu_Seiten/Publikationen/Wissen_for_free/RE-Broschuere_Englisch_-_Online.pdf (Accessed: 2025-01-29)
Travizano, M., Sarraute, C., Ajzenman, G., & Minnoni, M. (2018). *Wibson: A decentralized data marketplace.* Retrieved from https://arxiv.org/abs/1812.09966
Turkmayali, A., & Gras, N. (2024, July). *Making the Dataspace Protocol an international standard.* Zenodo. Retrieved from https://doi.org/10.5281/zenodo.12663036 doi: 10.5281/zenodo.12663036

About the authors

Sanjeev Sun Shakya is a Software Engineer at Zenjob SE, Berlin. He earned his Master's degree in Web and Data Science from the University of Koblenz in 2024, with his thesis focusing on model-based sovereignty analysis of data spaces. You can reach him at sanjeev.shakya@zenjob.com.

Qusai Ramadan is an Associate Professor at the Centre for Industrial Software (CIS) at University of Southern Denmark (SDU). His research focuses on software engineering, with a particular emphasis on engineering trustworthy software and model-based development. From 2020 to 2024, he worked as a postdoctoral researcher in the Software Engineering Research Group at University of Koblenz in Germany. You can contact the author at qura@mmmi.sdu.dk.

Alexander Peikert is currently pursuing a Master's degree at the University of Vienna, Austria. Since 2023, he is working as a research assistant in the Software Engineering Research Group at the University of Koblenz, Germany, where he finished his Bachelor studies in model-based security engineering. You can contact the author at alexpeikert@uni-koblenz.de.

Julian Flake is a researcher at the Institute for Software Technology at University of Koblenz, Germany. He started his position in Koblenz, after he earned his Diploma in Computer Science from the Technical University Dortmund, Germany, and had positions at the chair for Software Engineering at TU Dortmund and at the Fraunhofer Institute for Software and System Engineering (ISST), Dortmund. His research interests cover data protection compliance, business process management and model based engineering in general. You can contact Julian Flake via flake@uni-koblenz.de.