The International Data Spaces Information Model – An Ontology for Sovereign Exchange of Digital Content

Sebastian Bader[1,7][0000−0003−1328−704X], Jaroslav Pullmann[2][0000−0003−3360−0822], Christian Mader[3], Sebastian Tramp[4][0000−0003−4707−2864], Christoph Quix[5,9][0000−0002−1698−4345], Andreas W. Müller[6], Haydar Akyürek[5][0000−0001−6874−1671], Matthias Böckmann[1][0000−0002−1753−0791], Benedikt T. Imbusch[7], Johannes Lipp[5,8][0000−0002−2639−1949], Sandra Geisler[5][0000−0002−8970−6282], and Christoph Lange[5,8][0000−0001−9879−3827]

[1] Fraunhofer IAIS, Schloss Birlinghoven, 53757 Sankt Augustin , Germany, {sebastian.bader,matthias.boeckmann}@iais.fraunhofer.de
[2] Stardog Union, 2101 Wilson Blvd #800, Arlington, VA 22201 , United States, jaro.pullmann@stardog.com
[3] MANZ Solutions GmbH, Johannesgasse 23, 1010 Vienna , Austria, christian.mader@manz.at
[4] eccenca GmbH, Hainstr. 8, 04109 Leipzig , Germany, sebastian.tramp@eccenca.com
[5] Fraunhofer FIT, Schloss Birlinghoven, 53757 Sankt Augustin , Germany, {haydar.akyuerek, christoph.quix, johannes.lipp, sandra.geisler, christoph.lange-bever}@fit.fraunhofer.de
[6] Schaeffler Technologies, Herzogenaurach , Germany, andreas_w.mueller@schaeffler.com
[7] University of Bonn, Endenicher Allee 19a, 53115 Bonn , Germany, benedikt.imbusch@uni-bonn.de
[8] RWTH Aachen University, Ahornstraße 55, 52074 Aachen , Germany
[9] Hochschule Niederrhein, Germany

Abstract. The International Data Spaces initiative (IDS) is building an ecosystem to facilitate data exchange in a secure, trusted, and semantically interoperable way. It aims at providing a basis for smart services and cross-company business processes, while at the same time guaranteeing data owners' sovereignty over their content. The IDS Information Model is an RDFS/OWL ontology defining the fundamental concepts for describing actors in a data space, their interactions, the resources exchanged by them, and data usage restrictions. After introducing the conceptual model and design of the ontology, we explain its implementation on top of standard ontologies as well as the process for its continuous evolution and quality assurance involving a community driven by industry and research organisations. We demonstrate tools that support generation, validation, and usage of instances of the ontology with the focus on data control and protection in a federated ecosystem.

Keywords: data model · digital ecosystems · data sovereignty · federated architecture · ontology

1 Introduction: IDS Key Principles

Seamless collaboration and information exchange are the foundations of digital business models. Huge internet-based platforms have emerged, connecting people around the world and exchanging information in unprecedented speed. While end-users got used to such convenient communication and data exchange in their private interactions, they expect similar characteristics in their professional environment. However, data exchange in business-to-business relations faces a significant amount of still unresolved challenges. One example is the typical dilemma of digital strategies – sharing valuable data involves the risk of losing the company's competitive advantage, whereas not participating prevents innovative business models and undermines upcoming revenue opportunities.

There is currently no standardised, widely accepted means for a trustful exchange of business data that ensures traceability, data owner's privacy and sovereignty. Privacy concerns and protection of proprietary information are critical factors of future data infrastructures [7]. Such an infrastructure is a key prerequisite for a secure, standardised and fine-grained sharing of sensitive business data, unlocking the potential for novel value creation chains and the inception of intermediation platforms [9].

The International Data Spaces initiative[10] (IDS; formerly "Industrial Data Space") targets the requirements mentioned above by promoting a standard for virtual data spaces for reliable data exchange among business partners. To achieve the goal of sovereign data exchange, aspects of data management, semantic data integration, and security have to be addressed. The IDS proposes a message-based approach to bridge syntactic differences. Still, a successful exchange of data objects requires sufficient understanding of its content and meaning. A shared information model is therefore needed. The *IDS Information Model* (IDS IM) is an RDFS/OWL ontology, which defines the general concepts depicted in Fig. 1 along with roles required to describe actors, components, roles and interactions in a data space. This ontology serves two purposes, (1) as a catalogue of machine-readable terms and data schema for IDS components and (2) as a shared language for all stakeholders. Each involved player needs to understand

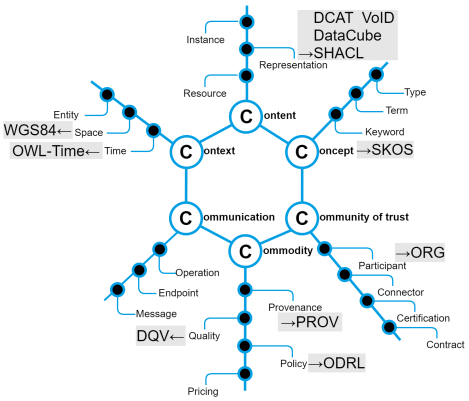

Fig. 1. Partitions of the ontology by concern (pointing to standards reused).

[10] `https://internationaldataspaces.org`

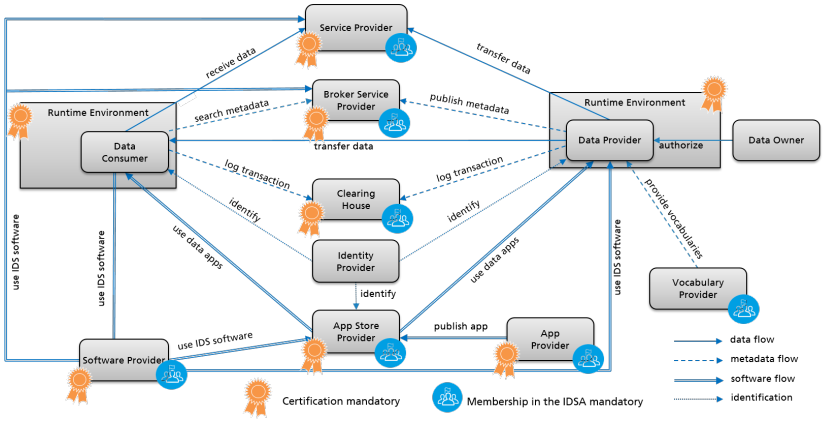

Fig. 2. IDS Reference Architecture with its main roles and interactions.

and be able to interpret this set of terms, thus enabling semantic interoperability in federated environments. The IDS IM therefore presents the backbone and common denominator for the data-sovereign ecosystem as envisioned by the IDS.

This paper presents version 4.0.0 of the IDS IM. Recent advances over earlier publications ([22,18]) especially include the elaborated coverage of enforceable permissions and restrictions as the foundation of data usage policies, a significantly enhanced supply of interaction messages, as well as improved tool support for instance creation and validation.

Section 2 outlines the IDS environment and explains the fundamental concepts. Sections 3 and 4 explain the implementation using standard ontologies as well as the continuous evolution and quality assurance methods, followed by a presentation of tools for generation, validation, and usage of instances of the ontology in Section 5. Section 6 reviews current adoption and Section 7 reviews related work and similar approaches. Section 8 concludes the paper and outlines next steps.

2 Governance and Context of the IDS Information Model

The IDS has been designed in a systematic process with broad involvement of industrial stakeholders [17]. Its specification and reference implementations are maintained and supported by the International Data Spaces Association (IDSA), a non-profit organisation to disseminate and evolve the IDS views and principles. The IDSA, with more than 100 member organisations meanwhile, serves as the institutional body for promoting the IDS in research projects and industrial applications. In particular, via its sub-working group (SWG) 4 “Information Model”, the IDSA ensures the sustainability of the ontology and provides the resources for future extensions (cf. Section 3.2 for details).

The IDS Reference Architecture Model (RAM) defines the roles assumed and the responsibilities of organisations interacting in a data space [18]. Fig. 2 shows,

Table 1. Key facts about the IDS Information Model and related resources.

General	**Licence**	Apache License 2.0
	Size	278 classes, 149 object properties, 115 data properties, 684 individuals
	Total size	3912 triples
Reuse	**Reused ontologies**	CC, DCAT, DCMI Terms, FOAF, ODRL, OWL-Time, VoID, etc.
Documentation	**Ontology documentation**	`https://w3id.org/idsa/core/`
	Element description	Using `rdfs:label`, `rdfs:comment`
Availability	**Namespace**	ids: `https://w3id.org/idsa/core/` idsc: `https://w3id.org/idsa/code/`
	Serialisations	Turtle, RDF/XML, JSON-LD, N-Triples
	GitHub	`https://github.com/International-Data-Spaces-Association/InformationModel/`
	VoCol Instance	`http://vocol.iais.fraunhofer.de/ids/`

for a broad initial overview, the core *interactions* and *roles* in the IDS.Data Providers exchange messages with Data Consumers via standardised software interfaces, and use multiple services to support this. They can, for example, publish metadata about resources to a directory ("broker") and thus allow others to find these. At the heart of every IDS interaction is the adherence to the usage rules – accomplished by the connection of machine-readable usage policies with each interaction and the application of certified, trustworthy execution environments. The so-called IDS Connectors interpret and enforce the applied policies, thus creating a federated network for a trustworthy data exchange.

The IDS IM specifies the domain-agnostic common language of the IDS. The IM is the essential agreement shared by the participants and components of the IDS, facilitating compatibility and interoperability. It serves the stakeholders' requirement "that metadata should not be limited to syntactical information about data, but also include data ownership information, general usage conditions, prices for data use, and information about where and how the data can be accessed" [17] by supporting the description, publication and identification of (digital) resources. It is, like other elementary IDS software components, available as open source to foster adoption (cf. Tab. 1). The ontology, the normative implementation of the declarative UML representation in the IDS RAM, was originally created in 2017 and first released in 2018.

2.1 Motivating Example

We use the example of the provider of financial intelligence data, the 'Business Intel Inc.', which collects, verifies, and processes stock market data for investment companies. One of their top seller is a cleared dataset of all Wall Street rates, which high frequency traders use to train their AI models. In order to further

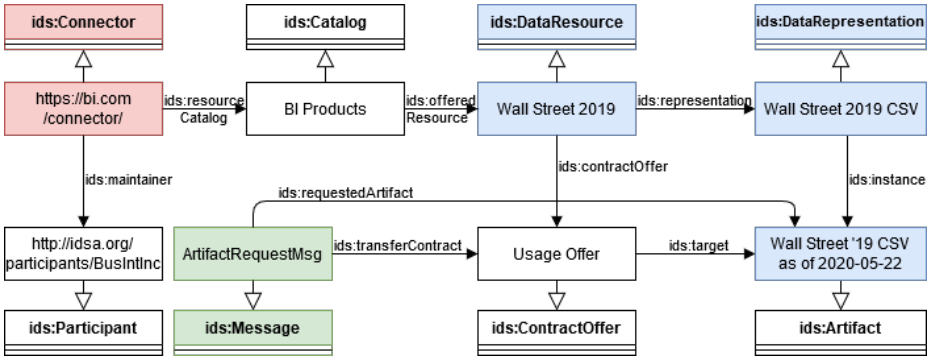

Fig. 3. IDS core classes and their instances in the running example.

automate their selling process, 'Business Intel Inc.' provides their dataset in an IDS ecosystem, through their IDS Connector at `https://bi.com/connector/` to ensure to (1) transforming existing data into economic value, while (2) restricting access and subsequent usage, and thus (3) ensuring the sovereignty over their data. These translate to four requirements:

(R1) Describe the data resource to make it discoverable (to potential, still unknown costumers)
(R2) Create business value through data exchange
(R3) Describe intended and prevent unintended usages
(R4) Control the usage over the complete digital life cycle

The description and announcement of the data resource (R1) is shown in Listing 1.[11] The unambiguous metadata is understood by every other participant in the IDS. In addition, as Attard et al. explain, added value from digital data can only be created through value co-creation [2]. Therefore, data resources must be made available at the right time to the right consumer. The requirement (R2) is fulfilled by the IDS infrastructure and the involved components, which are able to interpret a data resource based on its self-description and understand the described relations (cf. Fig. 3).

Listing 1. Stock market data modelled as an IDS DataResource.

```
_:StockData a ids:DataResource ;
  ids:title "Wall Street Stock Prices 2019"@en ;
  ids:description "This dataset contains the complete stock market prices
    of all 2019 Wall Street listed companies by milliseconds."@en ;
  ids:keyword "stock price", "Wall Street", "2019" ;
  ids:publisher <http://idsa.org/participants/BusIntInc>;
  ids:temporalCoverage [ a ids:Interval ;
    ids:begin [ a ids:Instant ;
```

[11] We abbreviate URIs following `http://prefix.cc/`.

```
    ids:dateTime "2019-01-01T00:00:00.000-04:00"^^xsd:dateTimeStamp ];
  ids:end [ a ids:Instant ;
    ids:dateTime "2019-12-31T23:59:59.999-04:00"^^xsd:dateTimeStamp]];
ids:language idsc:EN ;
ids:representation [ ids:instance _:StockDataCSV ; ids:mediaType
  <https://www.iana.org/assignments/media-types/text/csv>];
ids:resourceEndpoint [ a ids:ConnectorEndpoint ; ids:accessURL
  "https://bi.com/connector/reports/2019_wall_street.csv"^^xsd:anyURI];
ids:contractOffer _:StockDataOffer .
```

3 Methodology

3.1 Design Principles

The IDS overall has been designed as an alliance-driven multi-sided platform [17]. The basic process is aligned with the *eXtreme Design* method [21] with a strong focus on agile and collaborative workflows. The role of a customer is filled through a dedicated *ontology owner*, an experienced ontology expert who acts as the link to the developer community. In addition, the IDS IM is driven by the initial requirements originally collected and described in the RAM [18], and later on represented through publicly accessible *issues*. Furthermore, the *eXtreme Design* proposal to use separate ontology modules has let to the partitions shown in Fig. 1. As demanded by [10], the ontology development process needs to be test-driven, which is implemented by an automated syntax validation together with a semi-automated code generation pipeline (cf. Sec. 5.1). This code is integrated into several runtime components, in particular IDS Connectors, serving as test environment for each and every update.

Deep integration with state of the art software development platforms (Git, continuous integration, build agents, sprint-based development) enables an agile, iterative release management. Combining these characteristics with Semantic Web best practices led to the core design principles of the IDS IM:

Reuse: The body of existing work is evaluated and reused by refining terms of standard vocabularies, many of them being W3C Recommendations.

Linked Data: The IM is published under a stable namespace, in common RDF serialisations together with a human-readable documentation and interlinked with external resources.

FAIR: The ontology as a whole follows the FAIR principles (findable, accessible, interoperable, reusable [23]).

Separation of concerns: Each module of the ontology addresses a dedicated concern that applies to a digital resource (cf. Fig. 1).

3.2 Maintenance and Update Process

As stated, the IM's development within the IDSA SWG4 follows an agile methodology involving different stakeholder groups. Interested IDSA members support

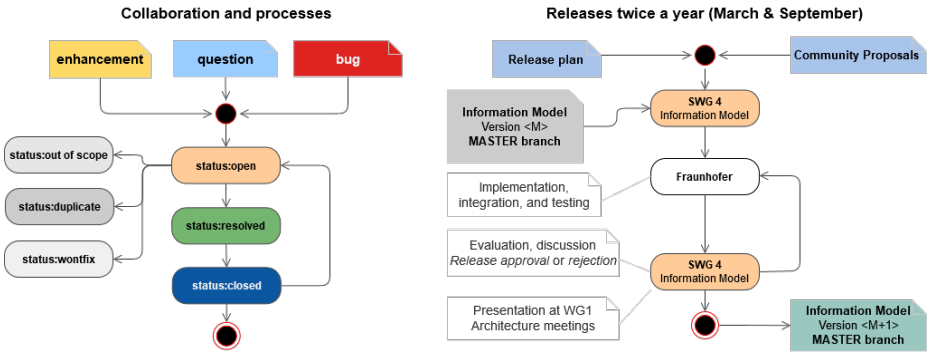

Fig. 4. The IDS IM update and release process.

the core modelling team by supplying domain knowledge, providing use cases and validating the model against their requirements. It has followers and contributors from 13 different IDSA member organisations and represents one of the most active IDSA communities.

The IM is provisioned in two parallel ways. The stable releases, reflecting major model updates, are provided once or twice a year (see also the *eXtreme Design's* integration loop). In the meantime, new features and bug fixes, which have been filed in the GitHub issue tracker, are addressed in monthly sprints as part of the module development loop. Those yield, besides incremental versions, nightly builds and snapshots. The overall process is depicted in Fig. 4. The community as a whole initiates change requests by creating tickets and proposing updates. The IDSA SWG4 then acts as the final authority, reviewing and merging the proposals.

3.3 Ontology Build Process and Quality Control

The continuous evolution of the IDS IM is supported by Continuous Integration and Continuous Deployment (CI/CD) mechanisms with automated quality assessment. As the IM is developed in its Turtle source code representation in a Git repository, CI/CD is realised similar to software projects. We currently run three test stages:

- A syntax check validates the syntactical correctness of all component files (currently 220 Turtle files).
- A reasoner[12] checks for logical inconsistencies, such as disjoint classes that subsume each other, or cycles in the class hierarchy.
- A set of RDFUnit[13] [13] test cases is used to find code smells and report common errors.

[12] currently Pellet: `https://github.com/stardog-union/pellet`
[13] `https://github.com/AKSW/RDFUnit`

For new versions and releases, human-readable documentation is generated by semi-automatic invocations of Widoco, as explained below in Section 4. The Widoco process includes further quality checks provided by the OOPS! ontology pitfall scanner web service[14] [20], such as identifying broken links to external vocabularies ("namespace hijacking"). The quality is furthermore ensured by a code review process, where each change request must be evaluated and approved by at least one reviewer not involved in the creation of the change request. Major updates are additionally discussed in the IDSA SWG4 and require unanimous consent.

Instances of the IM can also be validated against its schema by using SHACL shapes. Every class has its corresponding shape, stating the required properties, their cardinality requirements and value types[15]. The shapes are used to (1) validate incoming data objects but also (2) to describe the restrictions on class attributes. Thereby, the SHACL representations are used both as the enabler for instance validation and as a further extension to the schema description, for instance for cardinality restrictions.

4 Implementation and Reuse of Standards

The declarative representation of the IDS IM is provisioned as 5-Star Linked Data and conforms to the FAIR principles. It is accessible under an open license, in a stable namespace and maintained in a public GitHub repository (cf. Tab. 1). Dereferencing the namespace URLs redirects the client to either a human-readable website supplying the ontology documentation page, generated in W3C style by Widoco[16] [8], or directly to a serialisation in one of the supported formats (RDF/XML, Turtle, JSON-LD, N-Triples). Further documentation is available at a public, read-only instance of the VoCol vocabulary collaboration environment. This instance includes views of the evolution of the ontology and a public SPARQL endpoint[17]. All classes, properties and instances are enhanced with descriptions and, wherever appropriate, links to further information sources.

4.1 Relations to External Ontologies

Terms from external ontologies are individually imported by extending the respective class (using *rdfs:subClassOf*) or property (*rdfs:subPropertyOf*), in order to adapt its axiomatisation (e.g., *rdfs:range*) or insufficient specification from the IDS context (cf. Tab. 1 and Fig. 1). The adoption of external concepts into the IDS namespace is necessary, as those concepts are further refined according to the IDS characteristics and facets. For instance, the *DigitalContent* class

[14] `http://oops.linkeddata.es`
[15] `https://github.com/International-Data-Spaces-Association/InformationModel/tree/master/testing`
[16] `https://zenodo.org/record/3519946`
[17] `https://vocol.iais.fraunhofer.de/ids/`

Table 2. Examples for Information about Data Structure and Semantics.

Information	Standard	User Story
Use of vocabularies	VoID (*void:vocabulary*, *void:classPartition*, *void:propertyPartition*, etc.)	"This resource mainly contains information about average and minimum temperatures," or "This resource mainly contains instances of the W3C SOSA/SSN sensor data ontology."
Data structure	Data Cube (*qb:structure*, *qb:component*, *qb:dimension*, *qb:attribute*, etc.)	"This resource consists of a three-dimensional matrix with temperature measurements in degrees centigrade in the dimensions 1. time, 2. geo-coordinates, and 3. sensor used."
Graph structure	SHACL (*sh:shapesGraph*)	"This resource contains measurements of average and minimum temperature in a specific place at a specific time, measured by sensor X"

captures the type and semantics of a binary content in an abstract, format-independent way, extending *dcat:Dataset*. Among others, it records the context in terms of spatial, temporal and real-world entity coverage, the (SKOS) concepts related to the content (theme), and the provenance of the content by leveraging the PROV-O[18] vocabulary. IDS components, however, require specific attributes and relations, which are not stated – and not intended to be – in the original vocabularies.

4.2 Expressing Data Structure and Domain-specific Semantics

The IDS IM is independent of concrete application domains and thus does not provide terminology for the *content* of data resources. However, as the IDS encourages interoperability and extensible ecosystems, it encourages the use of RDF and domain ontologies for Representations (cf. [22] for a sample scenario using a taxonomy of steel grades). In this context, it is desirable to include information about the domain-specific semantics and, similarly, the structure of content into the metadata of a Resource or some of its Representations – for example, to be able to retrieve more relevant data resources. To this end, the IM reuses VoID, the Data Cube Vocabulary, and SHACL[19], as explained in Table 2 and detailed by examples in the GitHub repository.

5 Tool Support

While the IDS IM serves as the shared language throughout a data space, its adoption is usually challenging for component developers not familiar with the Semantic Web. A set of tools therefore supports the implementers and aims at preventing pitfalls as much as possible.

5.1 Java API to Generate Instances

[18] `https://www.w3.org/TR/prov-o/`
[19] `https://www.w3.org/TR/{void,vocab-data-cube,shacl}/`

```
DataResource metadata = new DataResourceBuilder()
  ._title_(Util.asList(new TypedLiteral("Wall Street ... 2019","en")))
  ._description_(Util.asList(new TypedLiteral("This dataset...", "en")))
  ._keyword_(Util.asList(new PlainLiteral("stock price"), [...]))
  ._publisher_(URI.create("http://idsa.org/participants/BusIntInc"))
  ._temporalCoverage_(Util.asList(new IntervalBuilder().[...].build()))
  ._language_(Util.asList(Language.EN))
  ._representation_(Util.asList(new RepresentationBuilder()[...])))
  ._resourceEndpoint_(Util.asList(new ResourceEndpointBuilder()[...]))
  ._contractOffer_(<data_restrictions>).build();
```

Listing 2. Java representation of the running example.

Instantiating the IDS IM concepts is crucial when running IDS components in practice, e.g., when sending messages, creating metadata descriptions or specifying usage restrictions (cf. Listing 2). Developers of Connectors within the early IDS projects found it inconvenient and error-prone to create these instances directly on the level of RDF data structures. Therefore, a software stack has been developed to transform the declarative representation of the IDS IM into a Java class library. The code generation process takes the ontology's Turtle source files as input and automatically validates, compiles and pushes the Java library files. To the best of our knowledge, the IDS IM is the only ontology with such a representation directly in executable code.

The Java API is publicly deployed via two channels: it is pushed to a Maven repository[20], and a nightly release is made available as a ZIP file on GitHub[21] as a part of the CI/CD pipeline (cf. Section 3.3). Besides the Java API, this ZIP file contains the Turtle sources, an UML-like visualisation of the ontology as well as a parser and serializer for IM instances[22]. This package helps to onboard developers faster and to give them as much support as possible. As the adoption in the developer community is crucial for the success of the IDS in general and the IDS IM in particular, we have also created a thin web application which guides the user through the modelling process.

5.2 GUI for Instance Management

This so-called IDS Semantic Instance Manager (cf. Fig. 5) supports non-RDF expert developers and system architects in expressing their required entities by a template-based GUI driven by the Java API introduced in Section 5.1. Instances can be exported to the common RDF serialisations. In the case of

[20] `https://maven.iais.fraunhofer.de/artifactory/eis-ids-public/de/fraunhofer/iais/eis/ids/infomodel/java/`

[21] `https://github.com/International-Data-Spaces-Association/InformationModel/releases`

[22] Demo project available at `https://jira.iais.fraunhofer.de/stash/projects/ICTSL/repos/ids-infomodel-demo/browse`

Messages, they can be sent directly to the target, thus turning the Semantic Instance Manager into a GUI for interactive control of a data space. Most IDS concepts have required properties, for example, a timestamp when the message was issued or the URI of the issuer. The GUI supports and guides the users to formulate valid IM instances, thus drastically lowering the entry barrier for constructing, e.g., messages, component and data descriptions or usage policies. Changes in the evolving IM, such as the introduction of new message types, do not require adapting the GUI, as it is dynamically built from the Java library using reflection.

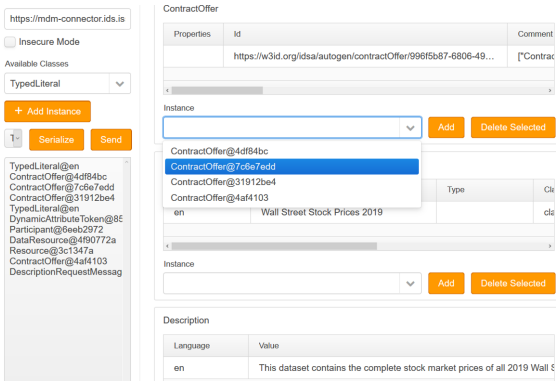

Fig. 5. IDS Semantic Instance Manager GUI.

6 Adoption

This section gives a brief overview of common use cases in which the IDS IM enables semantic interoperability in data spaces. The adoption processes in general are organised in five main verticalisation initiatives, which map the generic IDS specifications with the domain-specific requirements. These initiatives involve industrial manufacturing community, which is strongly related with the Plattform Industrie 4.0 and the Industrial Internet Consortium, the medical, energy, and material data space, as well as IDS in Smart Cities. In addition, at least seven commercially driven implementation processes are known to the authors. For instance, the public tender on re-implementing the German national mobility data platform explicitly enforces the IDS specifications[23] to ensure a self-sovereign landscape of equally empowered participants.

We regard the IDS IM as a reference ontology for trustworthy, data-driven architectures. It is a cornerstone of any IDS-related implementation and thus used in all related publicly funded projects, and impacts several industry platforms. The IDSA highlights 14 real-world use cases, the majority of them being realised with an investment from companies and contributing to their business success; furthermore, 10 EU research projects alone involve the IDSA (plus several of its members).[24] On a more technical level, at least 11 different Connector implementations with explicit support for a defined IDS IM version are currently known to the authors. Further adoption of the IDS IM among component developers is fostered at the quarterly IDSA Plugfest[25].

[23] https://www.evergabe-online.de/tenderdetails.html?0&id=322425
[24] https://www.internationaldataspaces.org/success-stories/
[25] https://www.internationaldataspaces.org/get-involved/#plugfest

Listing 3. Exemplary policy (*ids:ContractOffer* class; cf. Fig. 3) that grants read access to members of a certain organisation.

```
_:StockDataOffer a ids:ContractOffer ;
    ids:permission [ ids:target _:StockData ;
        ids:action idsc:READ ;
        ids:constraint [ ids:leftOperand idsc:USER;
             ids:operator idsc:MEMBER_OF;
             ids:rightOperandReference <http://whiterock-invest.com/> ];
        ids:postDuty [
            ids:action [ ids:includedIn idsc:COMPENSATE ;
                ids:actionRefinement [ ids:leftOperand idsc:PAY_AMOUNT ;
                    ids:operator idsc:EQ ;
                    ids:rightOperand "5000000"^^xsd:double ] ] ] [...] ] .
```

6.1 Community of Trust and Usage Control

Technically implemented trust and data sovereignty are at the heart of the IDS. Unambiguous description of usage restrictions and definition of the required attributes are therefore one of the most important use cases of the IDS IM. As the vocabulary presents the shared understanding of all involved parties – combining the different domains to one consistent ecosystem – it connects their security, certification, governance and interoperability models with each other.

Key challenges in the context of data usage control are the formal description of permissions and obligations. In our example, the Business Intel Inc. is able to present its intended restrictions in terms of a machine-readable policy (see R3). The Open Digital Rights Language (ODRL [11]) provides the terms and concepts for these statements. The IDS IM further details these constructs and defines their implications, focusing on their publication, negotiation, acknowledgement and enforcement. These additional steps enhance the solely *descriptive* ODRL vocabulary to legally *binding and enforceable* statements. Thus, the IDS IM not only allows to state permissions, e.g., that a data asset can be read (*ids:Permission* to *idsc:READ*) by certain users, but it can also express them in decidable terms for usage control engines such as MyData[26]. Such tools independently evaluate the agreed usage policies and, for instance, grant or deny access to individual resources. Modelling usage policies, contracts and the mappings between declarative and technically enforceable policies is a crucial prerequisite for the implementation of the IDS value proposition, to maintain the complete *sovereignty* of data owners with regard to their content.

Listing 3 shows a policy for the example data resource. The IDS IM is – to the best of our knowledge – the only vocabulary to cover the actual enforcement of usage restrictions. Established standard languages, for instance XACML, only focus on access control or, as for instance ODRL, only allow the description and

[26] `https://www.mydata-control.de/`

exchange of policies. The IDS IM closes this gap with detailed instructions on how to interpret each attribute, how to resolve statements and how to relate given policies to a system environment [3]. This is one aspect of solving R4. Listing 3 further shows how R2 (Business Value) is expressed. The *postDuty* clause describes and enforces a compensation for using the dataset, thereby combining business and data security statements in one representation.

Furthermore, the clear semantic of the allowed Action (READ) tells every interested buyer that the usage in its own IT landscape is covered (R3.1: describe intended usages), while any further distribution or reselling will and must be prohibited by the Usage Control Framework (R3.2: prevent unintended usage). The contract gives the Business Intel Inc. the tool to enforce its business model in the technical landscape of the customer (R4: control over the complete life cycle), which of course must also be supported by the execution environment of the customer. This however is ensured by the signed certification claims and can be checked on the fly.

6.2 Trust through Certified Attribute Declarations

In order to evaluate these claims, participants and components are subject to a *certification process* – an additional means to establish trust within and across data spaces. Organisational structures, methodologies, and standards underlying that process are detailed in [12]. A normative, tamper-proof reference of certification, security and identity attributes is maintained by IDS infrastructure components, operated as part of an Identity Provider (cf. Fig. 2).

Being part of the Public Key Infrastructure (PKI), this component augments the core attributes of an identity proof by a set of dynamic IDS-specific attributes. These attributes are defined in the IM for purposes of a single-truth maintenance, bridging the gap between the certification process during the *design time* of components, the security onboarding at *deployment time*, and the automated validations during interactions at *runtime*, not only at a data provider but at any intermediary IDS system throughout the complete data life cycle (R4). As Listing 4 shows, the basic JSON Web Token structure from RFC 7519 has been extended with additional attributes for usage control systems. Most relevant is here the *securityProfile* property, which contains crucial information on the trustworthiness of the target system. To the best of our knowledge, no other data model supports such a holistic approach and combines the various requirements.

Listing 4. Serialised Dynamic Attribute Token (DAT) in JSON-LD.

```
{ "@context" : "https://w3id.org/idsa/contexts/context.jsonld",
  "@id" : "http://w3id.org/idsa/DatPayload/A51317560",
  "@type" : "ids:DatPayload",
  "referringConnector" : { "@id": "http://bi.com/connector" },
  "iss": "65:43:5D:E8...:keyid:CB:8C:...AE:2F:31:46",
  "sub": "65:43:5D:E8...:keyid:CB:8C:...DD:CB:FD:0B",
```

```
"iat": 1589982066, "nbf": 1590154866, "exp": 1590759666,
"aud": { "@id": "idsc:IDS_CONNECTOR_ATTRIBUTES_ALL" },
"scope": "ids_connector_attributes",
"securityProfile": { "@id": "idsc:BASE_SECURITY_PROFILE" }}
```


7 Related Work

Several consortia have been formed to standardise (industrial) data exchange. The most prominent ones so far include the German Plattform Industrie 4.0 (PI4.0) and the US-American Industrial Internet Consortium (IIC). The PI4.0 focuses on physical assets and provides an extensive data model, called the Asset Administration Shell [4]. Nevertheless, this model does not sufficiently reflect the requirements of sovereign data interactions. The IIC focuses on the aspects of interoperable systems and architectures but also specifies a brief vocabulary [5], intended to enable discussions between experts but not to serve as a formal information model for a machine to machine interactions.

A huge amount of semantic description languages for interfaces and federated systems has been proposed. The SOAP technology stack and its service description language WSDL has been extended with the WSMO and WSMO-Light ontologies [6]. OWL-S is a similar OWL-based ontology for semantic descriptions of services. Furthermore, description languages for REST APIs have recently gained popularity, most prominently OpenAPI[27]. The IDS IM's definition of an *Interface*, e.g., of a Resource, is technology-agnostic, comparable to Web Service Interfaces in WSDL 2.0[28] and the concept of Service Profiles in OWL-S ontology[29]. Still, the focus is on the functionality of the endpoints itself, disregarding the challenges proposed through data protection and trust requirements.

The Data Catalog Vocabulary (DCAT) [14] is a related W3C Recommendation making use of well-established vocabularies to describe the distribution of (static) data sets. The limited expressivity of DCAT 1 was a major motivation for the IDS IM to extend it by versioning or temporal and spatial context. DCAT also neither includes relations to originating organisations nor allows for the description of data-related service APIs. *dcat:DataService* is just one example of how many of these limitations have recently been addressed with DCAT 2[30].

In addition to plain description languages, several ecosystems have been designed to seamlessly exchange data. bIoTope[31] aims at enabling interoperability between *vertical IoT silos* via a standardised open API. Data integration is supposed to be based on vocabularies to describe the different data sources. FIWARE[32] provides data through a RESTful API with RDF semantics called

[27] https://swagger.io/docs/specification/about/
[28] https://www.w3.org/TR/wsdl20/#Interface
[29] https://www.w3.org/Submission/OWL-S/#4
[30] https://www.w3.org/TR/vocab-dcat-2/
[31] http://www.biotope-project.eu
[32] https://www.fiware.org

NGSI-LD[33]. Besides the claim to reduce JSON payload costs and a full REST adoption, it offers a more powerful query language, especially for geospatial queries. FIWARE has been used to implement the IDS architecture [1]. Being RESTful, NGSI-LD serves a different purpose than the message-based IDS IM; however, they have in common the "Context" concern of data, e.g., in a spatio-temporal sense. Nevertheless, these ecosystems do not sufficiently express the conditions and restrictions imposed through digital information exchange.

The terminology of authorisations, obligations, and conditions introduced by the influential UCON_{ABC} [19] usage control model has been adopted by many later models. Together with RFC 2904 and the introduction of the different *policy points*, these two works form the theoretical foundation of usage control. However, neither proposes a vocabulary to specify distinct permissions or prohibitions. This task is, to some, degree covered by XACML [15,16]. Still, XACML only focuses on *access* control, not on the more holistic usage control.

The Data Privacy Vocabulary (DPV[34]) provides terms to annotate and categorise instances of legally compliant personal data handling according to the GDPR, including the notions of data categories, data controllers, purposes of processing data, etc. We are considering it as a candidate for extending the IDS IM by terminology for describing privacy aspects of data or software resources.

8 Conclusion and Future Work

We introduced an Information Model for data space ecosystems with a focus on supporting data sovereignty. We described how to support model development, documentation, and usage by different representations for various groups of stakeholders. We further demonstrated the usage of design principles that helped us to advance state-of-the-art models underlying our work.

The IDS IM is available openly on GitHub and comprises the patterns and features necessary to describe and implement digital sovereignty in a federated ecosystem. It shows how semantic technologies can be enhanced with security and trust to pave the way for enforceable, self-determined, i.e., sovereign data management across organisations. The comprehensive view on the challenge of addressing data owners' legitimate concerns while enabling productive data usage by other parties is a requirement for upcoming data-driven business cases.

Following the described contribution methodology, the IM is continuously evolved with industry stakeholders via the IDSA. We thus ensure that it is in line with emerging requirements of data ecosystems concerned with maintaining data sovereignty down to implementation specifications. Thus, the IM also promotes Semantic Web standards in disciplines where there is little awareness so far.

Next steps include developing tools for automated extraction of IDS IM metadata from the content of data resources, and to fully support retrieving data resources with a defined structure or domain-specific semantics.

[33] `https://fiware-datamodels.readthedocs.io/en/latest/ngsi-ld_howto/`
[34] `https://www.w3.org/ns/dpv`

Acknowledgements

This research was funded by the German Federal Ministries of Education and Research (grant number 01IS17031) and Transport and Digital Infrastructure (19F2083A), the EU H2020 projects BOOST4.0 (780732), DEMETER (857202) and TRUSTS (871481) and the Fraunhofer Cluster of Excellence "Cognitive Internet Technologies". We thank Eva Corsi, Anna Kasprzik, Jörg Langkau and Michael Theß, who have also contributed to the IDS IM via the IDSA SWG4.

References

1. Alonso, Á., Pozo, A., Cantera, J., de la Vega, F., Hierro, J.: Industrial data space architecture implementation using FIWARE. Sensors **18**(7), 2226 (2018)
2. Attard, J., Orlandi, F., Auer, S.: Data value networks: Enabling a new data ecosystem. In: International Conference on Web Intelligence (WI). IEEE (2016)
3. Bader, S.R., Maleshkova, M.: Towards Enforceable Usage Policies for Industry 4.0. LASCAR Workshop at ESWC (2019), http://ceur-ws.org/Vol-2489/
4. Boss, B., Hoffmeister, M., Deppe, T., Pethig, F., Bader, S., Barnstedt, E., et al.: Details of the Asset Administration Shell Part 1. Tech. rep., ZVEI (2019)
5. Bournival, E., Simmon, E., Buchheit, M., Baudoin, C., Hirsch, F., Boss, B., et al.: The Industrial Internet of Things Vocabulary. IIC (2019), https://hub.iiconsortium.org/vocabulary (accessed on 28.11.2019)
6. Domingue, J., Roman, D., Stollberg, M.: Web Service Modeling Ontology (WSMO)-An Ontology for Semantic Web Services (2005)
7. Finn, R., Wadhwa, K., Grumbach, S., Fensel, A.: Byte Final Report and Guidelines. Tech. Rep. D7.3, BYTE Project (2017), http://new.byte-project.eu/wp-content/uploads/2014/02/D7.3-Final-report-FINAL.pdf
8. Garijo, D.: WIDOCO: a wizard for documenting ontologies. In: International Semantic Web Conference. pp. 94–102. Springer (2017)
9. Grumbach, S.: Intermediation Platforms, an Economic Revolution. ERCIM News **2014**(99) (2014)
10. Hitzler, P., Gangemi, A., Janowicz, K.: Ontology engineering with ontology design patterns: foundations and applications, vol. 25. IOS Press (2016)
11. Ianella, R., Villata, S.: ODRL Information Model 2.2. Tech. rep., W3C ODRL Community Group (2018), https://www.w3.org/TR/odrl-model/
12. IDSA: Whitepaper certification. Technical report, IDSA (2018), https://www.internationaldataspaces.org/publications/whitepaper-certification/
13. Kontokostas, D., Westphal, P., Auer, S., Hellmann, S., Lehmann, J., Cornelissen, R., Zaveri, A.: Test-driven evaluation of linked data quality. In: WWW (2014)
14. Maali, F., Erickson, J., Archer, P.: Data Catalog Vocabulary (DCAT). Tech. rep., W3C (2014), https://www.w3.org/TR/vocab-dcat/
15. Mazzoleni, P., Crispo, B., Sivasubramanian, S., Bertino, E.: XACML Policy Integration Algorithms. Transactions on Information and System Security **11**(1) (2008)
16. Moses, T., et al.: eXtensible Access Control Markup Language (XACML). OASIS Standard (February 2005)
17. Otto, B., Jarke, M.: Designing a multi-sided data platform: findings from the international data spaces case. In: Electronic Markets. vol. 29. Springer (2019)
18. Otto, B., et al.: Reference Architecture Model. IDSA (2019), https://www.internationaldataspaces.org/ressource-hub/publications-ids/, version 3.0

19. Park, J., Sandhu, R.: The UCON ABC usage control model. ACM Transactions on Information and System Security (TISSEC) **7**(1), 128–174 (2004)
20. Poveda-Villalón, M., Gómez-Pérez, A., Suárez-Figueroa, M.C.: Oops!(ontology pitfall scanner!): An on-line Tool for Ontology Evaluation. IJSWIS **10**(2) (2014)
21. Presutti, V., Daga, E., Gangemi, A., Blomqvist, E.: eXtreme Design with Content Ontology Design Patterns. In: Proc. Workshop on Ontology Patterns (2009)
22. Pullmann, J., Petersen, N., Mader, C., Lohmann, S., Kemeny, Z.: Ontology-based Information Modelling in the Industrial Data Space. In: ETFA. IEEE (2017)
23. Wilkinson, M.D., Dumontier, M., Aalbersberg, I.J., Appleton, G., Axton, M., Baak, A., Blomberg, N., et al.: The FAIR Guiding Principles for Scientific Data Management and Stewardship. Scientific Data **3** (Mar 2016)