4

i3-MARKET Semantic Models

Taking into consideration the nature of the project, we worked on the following:

A. The definition, creation, and collection of semantic data models that allow to share a common description of the data assets (as per the case of shared data offering description model), operations, services, data details, credentials, contracts, pricing, and actors.

B. Development and implementation of semantic engine system and storage for the management of such information, creation of data offering description, management of controlled registries, mapping of information, interfaces among components, links of data and actors, discovery and retrieval of necessary information, compiling of smart contracts, and other operations.

C. Share the semantic models with the community to make use of them and work with people to improve and maintain the models for the present and future.

The use of standardized semantic metadata models and interaction patterns is important to enabling interoperability between nodes, user-friendly services, exchangeability of data assets, representation of actors (marketplaces, providers, consumers, and owners), and data exchange between different instances in the infrastructure ecosystem. A variety of standards already exist for sub-specific topics and domains; the most suitable ones to set up a common information model are selected and integrated into a high-level collection of vocabularies and ontologies. On top of these models, we created i3-MARKET core models to define the missing parts and for the main operational interactions and links among entities. Within i3-MARKET Backplane, the information ecosystem and the infrastructure ecosystem have to be combined to enable a seamless exchange of information and operations in a federated distributed architecture.

From a meta-modelling perspective, the i3-MARKET has raised certain requirements that go beyond the simple main description of datasets, adding information models to define other entities, operations actors, sharing agreements, and data details. While the existing semantic models cover only partially the requirements for the Backplane scopes, we imported, linked, and just in case extended common vocabularies and created the i3-MARKET semantic core model, pricing model, and contractual model for data sharing agreements and service agreements for contracts to compile a collection of semantic information models in O-CASUS models to cover the needs.

i3-MARKET Semantic Models

This section introduces the implementation of i3-MARKET semantic models that comprise the definition, creation, and collection of data models that allow to share a common description of the data assets (as per the case of dynamic data offering descriptions, operations, services, data details, credentials, contracts, pricing, actors, etc.). This section also deals with the definition and implementation of (meta)data management systems and registries catalogues to manage information and meta data descriptions. The main objective of the i3-MARKET Semantic Models is to share the data models with the community to make use of them and work with people on improving and maintaining the models for the present and future.

The i3-MARKET semantic models allow and facilitate the creation of data offering descriptions, management of controlled registries, mapping of information, and distribution of info and details among other components. The models and descriptions provide the links of data assets, metadata, and actors for discovering and retrieving of necessary information, compiling of smart contracts, and other operations.

4.1 i3-MARKET Model Specifications

Specifications for i3-MARKET semantic model solutions comprise the following:

- The definition, creation, and collection of data models that allow to share a common description of the data assets (as per the case of dynamic data offering descriptions, operations, services, data details, credentials, contracts, pricing, actors, etc.).

- The development and implementation of (meta)data management systems and registry catalogues to manage information and metadata descriptions.
- Share the data models with the community to make use of them and work with people on improving and maintaining the models for the present and future.
- The creation of data offering descriptions, management of controlled registries, mapping of information, and interfaces among other components.
- Provide the links of data and actors for discovering and retrieving of necessary information, compiling of smart contracts, and other operations.

We propose the i3-MARKET semantic core model and the semantic models imported and extended that create the collection of O-CASUS models based on the terminologies, definitions, and vocabularies needed to represent the i3-MARKET domain entities and operations. These concepts and their relationships are explained in more detail, including additional sub-concepts.

The O-CASUS semantic models comprise a collection of ontologies and vocabularies to cover the concepts used in the Backplane to define the following:

- i3-MARKET semantic core model
- W3c Data Catalog Vocabulary (DCAT and DCAT-AP)
- W3c Vocabulary of Interlinked Datasets (VoID)
- W3c Verifiable Credentials and DID
- SKOS Simple Knowledge Organization System
- IT Service Ontology
- EU Vocabularies Frequency Named Authority List
- EU Vocabularies File Type Named Authority List
- EU Vocabularies Languages Named Authority List
- EU Vocabularies Continents Named Authority List
- ADMS licence type vocabulary
- Distribution availability vocabulary
- Domain annotations

One of the key aspects when designing a semantic model is the reuse of knowledge. Once a semantic model is created for a domain, it should be (at least to some degree) reusable for other applications in the same domain. To simplify both semantic model development and reuse, a modular design is beneficial. Based on the project specification and the domain environment, the semantic models can be modularized according to their scope, as follows:

- Organization module
- Market module
- Provider module
- Consumer module
- Owner model
- Query module
- Data offering module
- Contractual parameters module
- Data exchange module
- Dataset information module
- Links to pricing module and the other vocabularies and ontologies to cover the various parts of the i3-MARKET O-CASUS sematic information models

A data marketplace is an online transactional location or store that facilitates the buying and selling of data. As many businesses seek to augment or enrich internal datasets with external data, cloud-based data marketplaces are appearing at a growing rate to match data consumers with the right data sellers.

Typical data types for sale in a data marketplace can range from business intelligence and research, demographic, firmographic, and market data to business intelligence and public data. A data marketplace is a more public (and sometimes commercial or monetized) form of data sharing. Data sharing has a long history in academic, research, and public policy circles but in recent years has made enormous inroads into private enterprises, from big business to analyst, consulting, and market intelligence firms. Data consumers include government, analyst, big business, and market intelligence firms. As data volumes continue to explode and machine learning and AI become more important in decision-making, data marketplaces are helping organizations reduce the effort and cost involved in locating required datasets and helping data providers extend their market reach.

However, big data is supported by continuous heterogeneity of underlying data sources (e.g., in IoT spaces), devices and communication technologies, and interoperability in different layers, from communication and seamless integration of platforms to interoperability of data to a global scale.

In a white paper on interoperability [67], it is discussed that many layers of interoperability exist:

- Technical interoperability
- Syntactical interoperability
- Semantic interoperability
- Organizational interoperability
- Dynamic interoperability
- Static interoperability

Discovery, understanding, and collaboration at this level require more than just an ability to interface and to exchange data. Interoperability is “the ability of two or more systems or components to exchange data and use Information” [68], whereas semantic interoperability “means enabling different agents, services, and applications to exchange information, data and knowledge in a meaningful way, on and off the Web” [67][68].

Semantic interoperability is achieved when interacting systems attribute the same meaning to an exchanged piece of data, ensuring consistency of the data across systems regardless of individual data information. This consistency of meaning can be derived from pre-existing standards or agreements on the description and meaning of data or it can be derived in a dynamic way using shared vocabularies either in a schema form or in an ontology-driven approach.

In i3-MARKET, we are aiming at an innovative approach for semantic data, metadata, and modelling activities as represented in Figure 4.1.

To lead the concept of O-CASUS, which is an idea based on the data lifecycle process, we:

- compile vocabularies and taxonomies in relation to marketplaces metadata, operation, and management;
- formalize the state of current marketplaces by using best practices and standards;

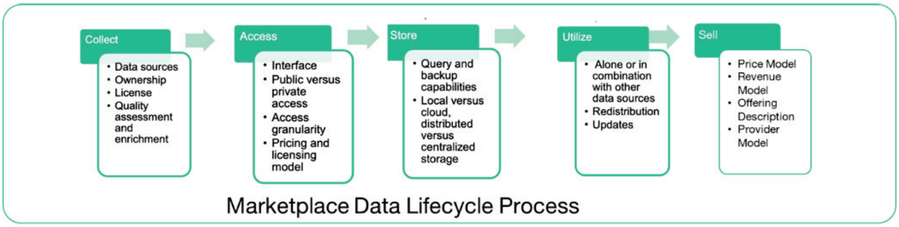

Collect
• Data sources
• Ownership
• License
• Quality assessment and enrichment
Access
• Interface
• Public versus private access
• Access granularity
• Pricing and licensing model
Store
• Query and backup capabilities
• Local versus cloud, distributed versus centralized storage
Utilize
• Alone or in combination with other data sources
• Redistribution
• Updates
Sell
• Price Model
• Revenue Model
• Offering Description
• Provider Model
Marketplace Data Lifecycle Process

Figure 4.1 i3-MARKET data model and the data lifecycle process.

- compile an ontology for collecting, accessing, storing, utilizing, and selling data.

4.2 i3-MARKET Semantic Core Models

The Figure 4.2 illustrates the high level of i3-MARKET semantic models that include all the basic conceptual entities and their relationship to all modules.

Details of each module are presented in the following subsections as shown in Figure 4.3. The i3-MARKET Semantic Core Models provide an overview of the i3-MARKET classes of resources that can be members of data offerings and the relationships between them. Except where it does not provide cardinality constraints as they are shown in the Figure 4.4 respectively.

One of the main contributions of the semantic models (vocabularies/ontologies) is the consolidation of the i3-MARKET models and the integration and extensions of other common sematic models to enable the

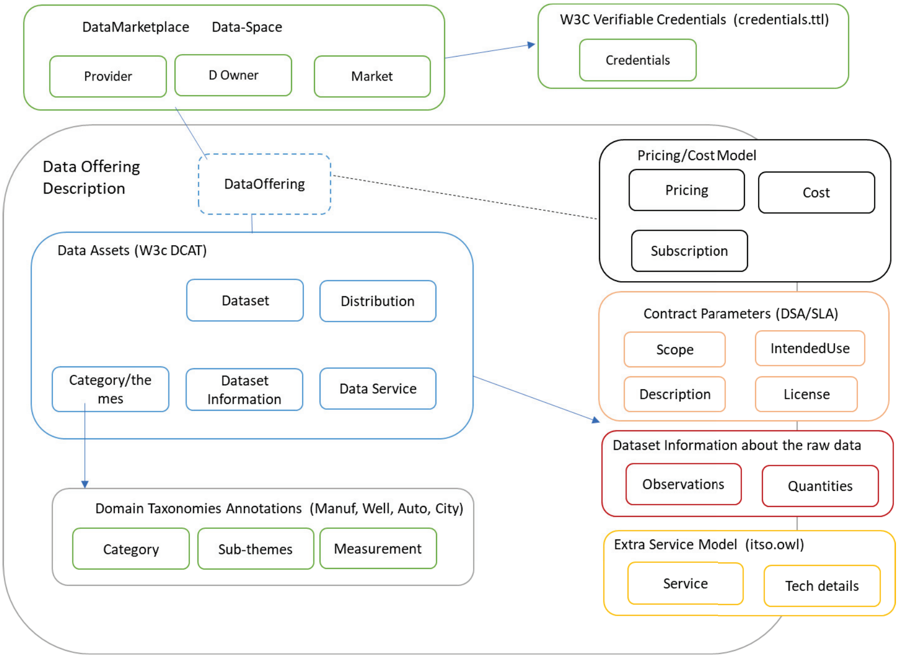


Figure 4.2 High-level semantic model structure of main modules.

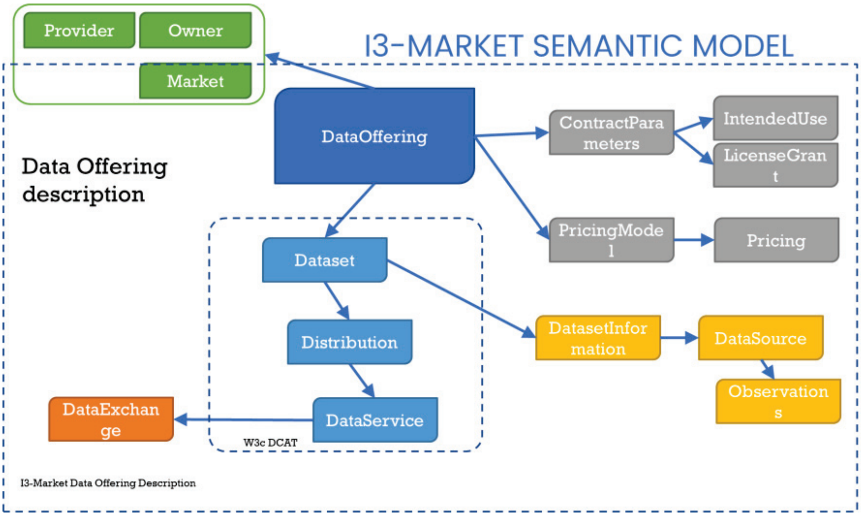


Figure 4.3 Main classes' block diagram of the i3-MARKET semantic model.

mapping of the metadata describing the data assets, contracts, and operations, provided from i3-MARKET stakeholders, to the model/ontology concepts to capture the structural and semantic characteristic of the metadata in relation to the various entities that corresponds to the different data assets and data offerings respectively.

More specifically, the core uses of these models are as follows:

1) Data registration of metadata descriptions, which corresponds to the data harmonization process. In this way, each provided data asset is registered in our registry with concepts from the i3-MARKET data offering model in a semi-automatic way.
2) Metadata linking where any provided data asset metadata will be linked with other relevant sources (or data assets) that exist in the Backplane.
3) Data discovery (for local or federated registries) that involves the development of algorithms and software for supporting the selection of the most appropriate metadata that best match user preferences.
4) Management of information related to smart contract, data access and transfer, pricing models, identity and credential identifications, and notifications.

The i3-MARKET models are used for capturing the structural and semantic metadata characteristics of the various entities involved in the i3-MARKET Backplane domain, whereas the underlying conceptual models

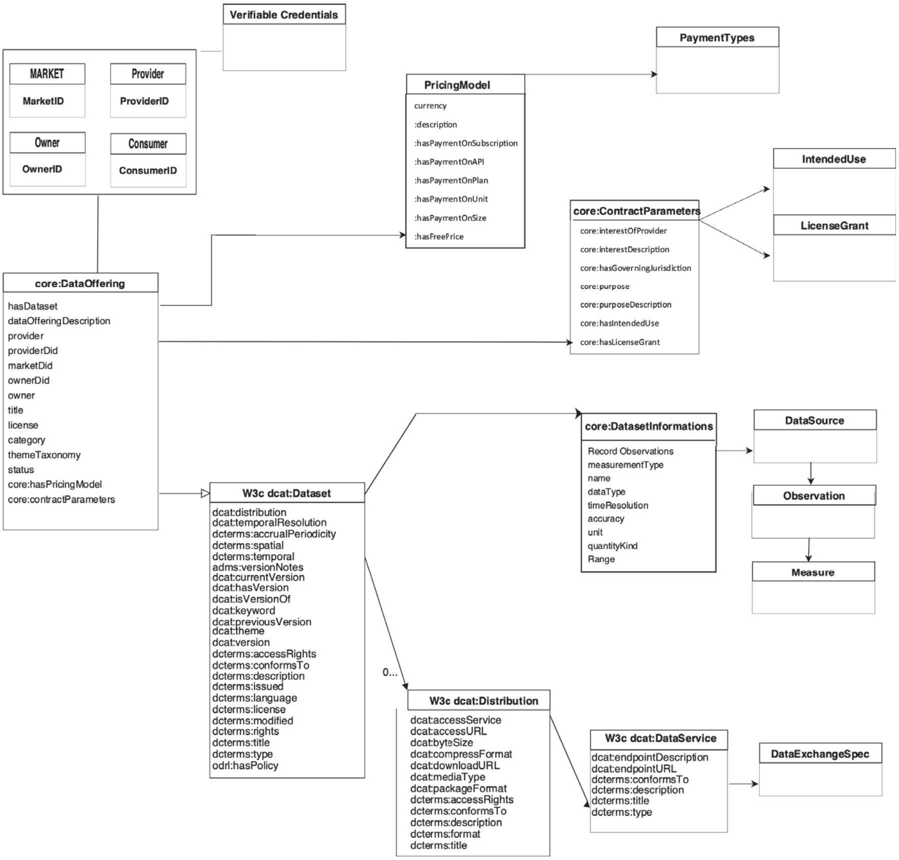


Figure 4.4 Overview of the i3-MARKET semantic model.

facilitate the use of lightweight reasoning during the discovery and operational process, e.g., for contracts and service/agreements, data access/transfer operations, etc.

4.3 Data Marketplace and Data Space Actors

- **Provider module:**

A provider can be a marketplace, data space, or service instance that offers available DataOfferings. A provider is described through the *core:Provider* class. At this stage, each provider has a name and ID (core:providerId) and its organization as shown in Table 4.1. More information about the provider can be added in the future.

Table 4.1 Provider properties.

Property name	Data types	Description
core:providerId	String	Provider ID
providerDescription	String	A description of the provider
providerName	String	Name of the provider
:sourceOrganization	core:Organization	The provider's organization

- **Organization module:**

A provider may also describe its organization. The provider's organization has been an instance of the *schema.org model (Organization Ontology class particularly)*. The connection between the provider and the organization is the sourceOrganization property. Table 4.2 presents some basic properties of the organization class, e.g., at the moment example taken from the *schema.org model and particularly Organization Class.*

Table 4.2 Organization properties.

Property name	Data types	Description
core:organizationId	String	Organization ID
organizationName	String	Name of the organization
:address	String	Physical address of the organization
contactPoint	schema:ContactPoint	A contact point for the organization
organizationDescription	String	A description of the organization

- **Consumer module:**

A consumer can be an entity, application, or service instance that requires access to data resources in order to implement an intended service or function. In the consumer model, we create the *core:Consumer* class that represents the i3-MARKET consumers. Same as the provider, the consumer is also linked to the organization. Table 4.3 presents some basic properties of the core:Consumer.

- **Owner module:**

The actual owner of the data sources provided by marketplace, data space, or service instance that offers available DataOfferings. An owner is

Table 4.3 Consumer properties.

Property name	Data types	Description
core:consumerId	String	Consumer ID
core:dataOfferingQuery	core:DataOfferingQuery	Query to i3-MARKET of consumer
consumerDescription	String	A description of the consumer
consumerName	String	Name of the consumer
:sourceOrganization	schema:Organization	The consumer's organization
core:subscribedTo	core:DataOffering	Data offering IDs the consumer subscribes to

described through the *core:Owner* class. At this stage, each owner has a name (schema:name) and ID (core:ownerID) as shown in Table 4.4. More information about the provider can be added in the future.

Table 4.4 Owner properties

Property name	Data types	Description
core:ownerId	String	Owner ID
ownerDescription	String	A description of the provider
ownerName	String	Name of the provider
:sourceOrganization	schema:Organization	The provider's organization

- **Data market module:**

  Information on the connected data marketplace is given in Table 4.5.

Table 4.5 Market module.

Property name	Data types	Description
core:dataMarketId	String	Data Market ID
dataMarketDescription	String	A description of the data marketplace
dataMarketName	String	Name of the data marketplace
dataMarketNode	String	Info of the data market node

4.4 Data Offering

Data offering description:

The i3-MARKET enables providers to offer or trade access to datasets via the Backplane. A data offering is defined by a "data offering description", which describes via metadata a set of resources offered via the i3-MARKET Backplane. It typically encompasses a set of related information. A data offering description provides a semantic description of the datasets provided to a consumer once the data offering is registered. The description also entails context and meta information about the distribution, including information about the pricing for accessing the resource(s), the license of the information provided, contractual parameters, and service description as URL for data access.

As illustrated in Figure 3.12, the data offering module represents the initial conceptualization, which is built around the DataOffering Class and its metadata. All the core concepts of this module are defined as follows.

A provider registers its offerings on the marketplace by providing an offering description. An offering description is an instance of the data offering class (which can be mapped to the common subclass of schema:Offer). It contains the information about the data assets, data service, categories of data assets, subclass components of catalogues and resources, data services, and categories of the offering (:category). All relevant communication metadata are provided on how the offering can be accessed through the data service and service extension descriptions.

Details of all classes and their properties in the offering module are presented in the following sections.

- **Data Offering description:**

To describe the data assets, contractual parameters, rights, licenses, pricing models, data service, endpoints, format of data, domain annotations, related actors, and other information that describe the datasets, we defined shared "data offering descriptions".

We use W3c Data Catalog Vocabulary (DCAT) – Version 3 vocabulary related to parts such as: dataset, distribution, and DataService used in data offering description (https://w3c.github.io/dxwg/dcat/).

It is recommended to use the description and specifications of DCAT for all the information related to dataset, distribution, and DataService used in data offering description (https://www.w3.org/TR/vocab-dcat-3/).

DCAT enables a publisher to describe datasets and data services in a catalogue using a standard model and vocabulary that facilitates the consumption and aggregation of metadata from multiple catalogues. This can increase the discoverability of datasets and data services. It also makes it possible to have a decentralized approach to publishing data catalogues and makes possible federated search for datasets across catalogues in multiple sites using the same query mechanism and structure.

- **Data Offering class:**

Definition: High-level class in the i3-MARKET core model that introduces the data offering description of dataset resources in Table 4.6.

Table 4.6 Data Offering properties.

Property	Data types	Description
hasDataset	Dataset	Links the data offering in core to a DCAT (-AP) dataset
dataOfferingDescription	String	Contains a free-text account of the *DataOffering*
Provider	Provider	Refers to an entity (organization) responsible for making the data offering available
providerDiD	DID	This is the provider DID, registered in VC and i3-MARKET, which is uniquely identified as a provider in IDM and Wallet
Owner	Owner	Refers to an entity that have source ownership of the data
ownerDiD	DID	This is the owner DID, registered in VC and i3-MARKET, which is uniquely identified as an owner in IDM and Wallet
Marketid	Market	This is the market name ID, which is uniquely identified as a marketplace
marketDiD	DID	This is the market DID, registered in VC and i3-MARKET, which is uniquely identified as a marketplace in IDM and Wallet
dataOfferingTitle	String	Contains a name given to the catalogue
License	LicenseDocument	This property refers to the license under which the catalogue can be used or reused
Category	ConceptScheme	Refers to a knowledge organization system used to classify the data offering categories for datasets The high-level category terms and the URI used are defined in the scheme file DataOfferingCategory.ttl
themeTaxonomy	skos:ConceptScheme	This property refers to a knowledge organization system used to classify the DataOffering's datasets

Table 4.6 *Continued.*

Property	Data types	Description
Active	String	Flag to set if the DataOffering is activated/available by the provider to be checked/searched by, e.g., the consumer
core:ownerConsentForm		Hashtag string to report the information about the explicit user consent form documentations
core:inSharedNetwork		Boolean to define if the DataOffering is shared by the marketplace to be visible and consumable by all actors in the i3-MARKET network
core:status		To define the data offering status
core:dataOffering ExpirationTime		Expiration time of DataOffering in case
core:lastModified		Most recent date on which the data offering was changed, updated, or modified
dcat:previousVersion		The previous version of a resource in a lineage (PAV)
Version		To define the “version” of the registered data offering
Core:datasetInformation		For the module that describes the information related to the details of the raw data (with info on origin of data, measurements, data types, devices, units, etc.)
core:hasPricingModel	pricingmodel: PricingModel	The pricing model for the data offering
core:contractParameters	core: ContractParameters	Some specific contract parameters related to data offering

- **(DCAT) Dataset class:**

Definition: A collection of data published or curated by a single agent and available for access or download in one or more representations, as shown in Table 4.7.

- **DatasetInformation class:**

Definition: Extended specific annotations to add extra information related to a dataset. This information is used to give providers the possibility to describe with more granularity the source and types of data in datasets

Table 4.7 DCAT dataset main properties.

Property	Data types	Description
Description	String	Contains a free-text account of the dataset
Title	String	Contains a name given to the dataset
Keyword	String	Contains a keyword or tag describing the dataset
core:datasetInformation	core:DatasetInformation	Some specific information annotations of dataset metadata information types, which represent attributes of observations, measurements, fields, etc. in the dataset
core:datasetRecord	core:DatasetRecord	In case data records types that represent attributes of fields, in the dataset
datasetDistribution	Distribution	Links the dataset to an available distribution
geographicalCoverage	Location	Refers to a geographical area covered by the dataset
temporalCoverage	PeriodOfTime	Refers to a temporal period that the dataset covers
Category	Concept	Refers to a category of the dataset. A dataset may be associated with multiple categories
accessRights	RightsStatement	Refers to information that indicates whether the dataset is open data, has access restrictions, or is not public
Frequency	Frequency	Refers to the frequency at which the dataset is updated
Documentation	Documentation	Refers to a page or document about this dataset
hasVersion	Dataset	Refers to a related dataset that is a version, edition, or adaptation of the described dataset
Creator	Agent	Refers to the entity primarily responsible for producing the dataset
dcat:theme	skos:Concept	This property refers to a category of the dataset. A dataset may be associated with multiple themes

and annotations related to specific domains (see Table 4.8). Also consult Appendix B for an extended version of information details to be used to describe the raw original data for consumers' understanding).

Table 4.8 Main properties of the DatasetInformation class.

Property	Data types	Description
core:measurementType	String	The data types that represent attributes of observations, measurements in the dataset
core:measurementChannelType	String	The data measurement channel types in the dataset
core:sensorID	String	Sensor ID
core:deviceID	String	Device ID
core:cppType	String	Cyber−physical systems cpp type
core:sensorType	String	Sensor type

- **(DCAT) Distribution class:**

  Definition: A specific representation of a dataset in Table 4.9.

Table 4.9 DCAT distribution main properties.

Property	Data types	Description
Description	String	Contains a free-text account of the distribution
accessURL	Resource	Contains a URL that gives access to a distribution of the dataset
Availability	Concept	Indicates how long it is planned to keep the distribution of the dataset available
Format	MediaTypeOrExtent	Refers to the file format of the distribution
downloadType	String	Download type (it means frequency as "Stream" or "Bulk" dataset can be downloaded)
License	LicenseDocument	Refers to the licence under which the distribution is made available
accessService	DataService	Refers to a data service that gives access to the distribution of the dataset
byteSize	Double	Size of a distribution in bytes
Documentation	Documentation	Refers to a page or document about this distribution
downloadURL	Resource	URL that is a direct link to a downloadable file in a given format
releaseDate	DateTime	Contains the date of formal issuance (e.g., publication) of the distribution

- **(DCAT) DataService class:**

Definition: A collection of operations that provides access to one or more datasets or data processing functions is shown in Table 4.10.

Table 4.10 DataService properties.

Property	Data types	Description
Description	String	Contains a free-text account of the data service
endpointURL	Resource	The root location or primary endpoint of the service (an IRI)
Title	String	Contains a name given to the data service
servesDataset	Dataset	Refers to a collection of data that this data service can distribute
License	LicenseDocument	Refers to the licence under which the data service is made available
accessRights	RightsStatement	Includes information regarding access or restrictions based on privacy, security, or other policies
serviceID	String	Service ID
serviceSpecs	ServiceSpecs	Service specification reference to ITSO extra service model specifications

- **ContractParameters class:**

Definition: A collection of parameters that provides information about the use and scope of the DataOffering/dataset in Table 4.11.

Table 4.11 DataService properties.

Property name	Data types	Description
core:interestOfProvider	Literal	This property is used to identify the interest of the data owner. The following possibilities exist: Free sharing quotation; selling of data (e.g., just earning money by selling the data, no specific feedback on these data by a data consumer expected)
core:interestDescription	Literal	Data provider can specify which sort of quotation he wants exactly, e.g., quotation for maintenance service or quotation for optimization of production
core:hasGoverningJurisdiction	Literal	Jurisdiction

Table 4.11 *Continued.*

Property name	Data types	Description
core:purpose	Literal	Purpose for the use of the dataset
core:purposeDescription	Literal	Description of the purpose for the use of the dataset
core:hasIntendedUse	core:IntendedUse	To intended use class/properties
core:hasLicenseGrant	core:LicenseGrant	To license grant class/properties

- **LicenseGrant class:**

Definition: Definition of the type of license is associated with the data asset in Table 4.12.

Table 4.12 LicenseGrant properties.

Property name	Data/object types	Description
core:paidUp	Boolean	If licence grant to paidUp
core:transferable	Boolean	Transferable (true or false)
core:exclusiveness	Boolean	License of exclusiveness (true or false)
core:revocable	Boolean	License revocable (true or false)
core:processing	Boolean	If licence grant data to be processed
core:modifying	Boolean	If licence grant data to be modified
core:analyzing	Boolean	If licence grant data to be analysed
core:storingData	Boolean	If licence grant to store data
core:storingCopy	Boolean	If licence grant to store copy of data
core:reproducing	Boolean	If licence grant to reproduce data
core:distributing	Boolean	If licence grant to distribute data
core:loaning	Boolean	If licence grant to loan data
core:selling	Boolean	If licence grant to sell data
core:renting	Boolean	If licence grant to rent data
core:furtherLicensing	Boolean	If licence grant for further licensing
core:leasing	Boolean	If licence grant to lease data

- **IntendedUse class:**

Definition: What the data provider allows the consumer to be the intended use of the data assets in Table 4.13.

Table 4.13 IntededUse properties.

Property name	Data/object types	Description
core:processData	Boolean	Process data (true or false)
core:shareDataWithThirdParty	Boolean	Share data with third party (true or false)
core:editData	Boolean	Edit data (true or false)

- **DataExchangeSpec class:**

Definition: Information inside the accessService block for data exchange specifications that serve the distributions of the datasets, used also by the data access and transfer system in Table 4.14.

Table 4.14 DataExchange properties.

Property name	Data/object types	Description
core:encAlg	Boolean	Encryption algorithm used to encrypt blocks. Either AES-128-GCM (’A128GCM’) or AES-256-GCM (’A256GCM)
core:signingAlg	Boolean	Signing algorithm used to sign the proofs. Like ECDSA secp256r1 with key lengths: either “ES256”, “ES384”, or “ES512”
core:hashAlg	Boolean	Hash algorithm used to compute digest/commitments. It is SHA2 with different output lengths: either “SHA-256”, “SHA-384”, or “SHA-512”
core:ledgerContractAddress		The ledger smart contract address (hexadecimal) on the DLT
core:ledgerSignerAddress		The orig (data provider) address in the DLT (hexadecimal)
core:pooToPorDelay		Maximum acceptable delay between the issuance of the proof of origin (PoO) by the orig and the reception of the proof of reception (PoR) by the orig
core:pooToPopDelay		Maximum acceptable delay between the issuance of the proof of origin (PoO) by the orig and the reception of the proof of publication (PoP) by the dest
core:pooToSecretDelay		If the dest (data consumer) does not receive the PoP, it could still get the decryption secret from the DLT. This defines the maximum acceptable delay between the issuance of the proof of origin (PoO) by the orig and the publication (block time) of the secret on the blockchain

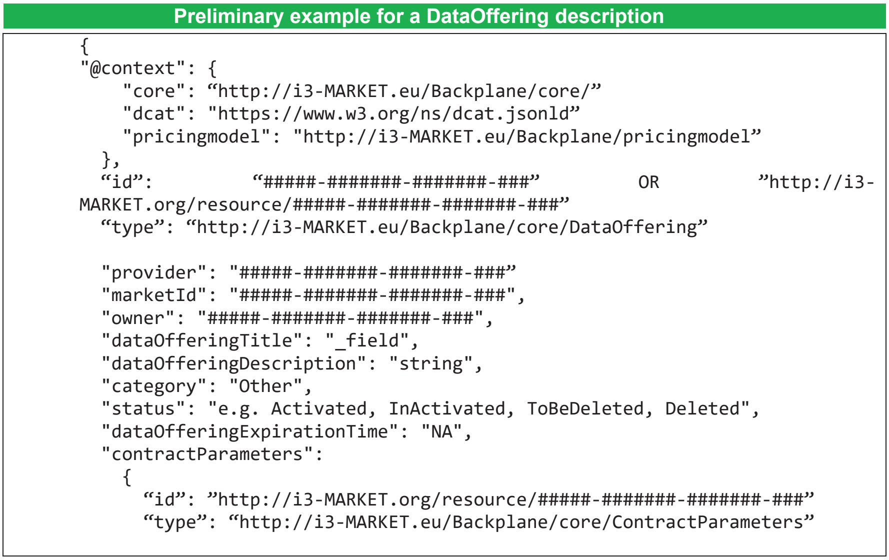
For a more complete list of classes and attributes that are used for the data offering description and details on their definitions, please see Tables 4.15 and 4.16.

Table 4.15 Preliminary example for a metadata description.

```
##
PREFIX rdfs: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdf: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd:  <http://www.w3.org/2001/XMLSchema#>
PREFIX dct:  <http://purl.org/dc/terms/>
PREFIX dcat: <http://www.w3.org/ns/dcat#>
PREFIX pricingmodel: <http://i3-MARKET.eu/Backplane/pricingmode/>
PREFIX core: <http://i3-MARKET.eu/Backplane/core/>
PREFIX : <http://i3-MARKET.org/resource/>

:Mindsphere
  a core:Provider ;
  rdfs:label "Mindsphere"@en ;
  core:dataOffering ex:DataOffering-1  .
```

Table 4.16 Preliminary example for a DataOffering description.



Preliminary example for a DataOffering description

```
{
"@context": {
    "core": “http://i3-MARKET.eu/Backplane/core/”
    "dcat": "https://www.w3.org/ns/dcat.jsonld”
    "pricingmodel": "http://i3-MARKET.eu/Backplane/pricingmodel”
  },
  “id”:         “#####-#######-#######-###”         OR         ”http://i3-
MARKET.org/resource/#####-#######-#######-###”
  “type”: “http://i3-MARKET.eu/Backplane/core/DataOffering”

  "provider": "#####-#######-#######-###”
  "marketId": "#####-#######-#######-###",
  "owner": "#####-#######-#######-###",
  "dataOfferingTitle": "_field",
  "dataOfferingDescription": "string",
  "category": "Other",
  "status": "e.g. Activated, InActivated, ToBeDeleted, Deleted",
  "dataOfferingExpirationTime": "NA",
  "contractParameters":
    {
      “id”: ”http://i3-MARKET.org/resource/#####-#######-#######-###”
      “type”: “http://i3-MARKET.eu/Backplane/core/ContractParameters”
```

Table 4.16 *Continued.*

```
      "contractParametersId": "string",
      "interestOfProvider": "NA",
      "interestDescription": "NA",
      "hasGoverningJurisdiction": "NA",
      "purpose": "NA",
      "purposeDescription": "NA",
      "hasIntendedUse":
       {
        “id”: ”http://i3-MARKET.org/resource/#####-#######-#######-###”
        “type”: “http://i3-MARKET.eu/Backplane/core/IntendedUse”

        "intendedUseId": "string",
        "processData": "true OR false",
        "shareDataWithThirdParty": "true OR false",
        "editData": "true OR false"
       }      ,
      "hasLicenseGrant":
       {
        “id”: ”http://i3-MARKET.org/resource/#####-#######-#######-###”
        “type”: “http://i3-MARKET.eu/Backplane/core/LicenseGrant”

        "licenseGrantId": "string",
        "copyData": "true OR false",
        "transferable": "true OR false",
        "exclusiveness": "true OR false",
        "revocable": "true OR false"
       }
     }  ,

    "hasDataset":
     {
      “id”: ”http://i3-MARKET.org/resource/#####-#######-#######-###”
      “type”: “http://www.w3.org/ns/dcat#Dataset”

      "datasetId": "string",
      "title": "_field",
      "keyword": "_field",
      "dataset": "_field",
      "description": "_field",
      "issued": "date-time",
      "modified": "date-time",
      "temporal": "_field",
      "language": "_field",
      "spatial": "_field",
      "accrualPeriodicity": "_field",
      "temporalResolution": "_field",
      "distribution": [
       {
        “id”: ”http://i3-MARKET.org/resource/#####-#######-#######-###”
        “type”: “http://www.w3.org/ns/dcat#Distribution”

        "distributionId": "string",
        "title": "_field",
        "description": "_field",
        "license": "_field",
        "accessRights": "_field",
```

Table 4.16 *Continued.*

```
          "downloadType": "_field",
          "conformsTo": "_field",
          "mediaType": "_field",
          "packageFormat": "_field",
          "accessService":
            {
              “id”:   ”http://i3-MARKET.org/resource/#####-#######-#######-
###”
              “type”: “http://www.w3.org/ns/dcat#DataService”

              "dataserviceId": "string",
              "conformsTo": "_field",
              "endpointDescription": "_field",
              "endpointURL": "_field",
              "servesDataset": "_field",
              "serviceSpecs": "_field"
            }
        }
      ],
      "datasetInformation": [
        {
          “id”: ”http://i3-MARKET.org/resource/#####-#######-#######-###”
          “type”: “http://i3-MARKET.eu/Backplane/core/DatasetInformation”

          "datasetInformationId": "string",
          "measurementType": "_field",
          "measurementChannelType": "_field",
          "sensorId": "_field",
          "deviceId": "_field",
          "cppType": "_field",
          "sensorType": "_field"
        }
      ],
      "theme": [
        "_field"
        "_field"
        "_field"
      ]
    }
}
```

4.4.1 Controlled vocabularies suggested to be used for particular annotations

In Table 4.17, a number of properties are listed with controlled vocabularies that should be used for the listed properties. The declaration of the following controlled vocabularies as high recommendation (in DCAT_AP specifications are listed as mandatory) ensures a minimum level of interoperability.

Table 4.17 Preliminary example for annotations.

Property URI	Used for class	Vocabulary name	Vocabulary URI	Usage note
dcat:mediaType	Distribution	IANA Media Types [5]	http://www.iana.org/assignments/media-types/media-types.xhtml	
dcat:theme	Dataset	Dataset Theme Vocabulary	http://publications.europa.eu/resource/authority/data-theme	The values to be used for this property are the URIs of the concepts in the vocabulary
dcat:themeTaxonomy	Catalogue	Dataset Theme Vocabulary	http://publications.europa.eu/resource/dataset/data-theme	The value to be used for this property is the URI of the vocabulary itself, i.e., the concept scheme, not the URIs of the concepts in the vocabulary
dct:accrualPeriodicity	Dataset	EU Vocabularies Frequency Named Authority List [70]	http://publications.europa.eu/resource/authority/frequency	
dct:format	Distribution	EU Vocabularies File Type Named Authority List [71]	http://publications.europa.eu/resource/authority/file-type	
dct:language	Catalogue, dataset, catalogue record, distribution	EU Vocabularies Languages Named Authority List [72]	http://publications.europa.eu/resource/authority/language	
dct:publisher	Catalogue, dataset	EU Vocabularies Corporate Bodies Named Authority List [73]	http://publications.europa.eu/resource/authority/corporate-body	The corporate bodies NAL must be used for European institutions and a small set of international organizations. In case of other types of organizations, national, regional, or local vocabularies should be used
dct:spatial	Catalogue, dataset	EU Vocabularies Continents Named Authority List [74], EU Vocabularies Countries Named Authority List [75], EU Vocabularies Places Named Authority List [76], Geonames	http://publications.europa.eu/resource/authority/continent/,~http://publications.europa.eu/resource/authority/country,~http://publications.europa.eu/resource/authority/place/, http://sws.geonames.org/	The EU Vocabularies Name Authority Lists must be used for continents, countries, and places that are in those lists; if a particular location is not in one of the mentioned Named Authority Lists, Geonames, URIs must be used
adms:status	Distribution	ADMS status vocabulary	http://purl.org/adms/status/	The list of terms in the ADMS status vocabulary is included in the ADMS specification [77]
dct:type	Agent	ADMS publisher type vocabulary	http://purl.org/adms/publishertype/	The list of terms in the ADMS publisher type vocabulary is included in the ADMS specification

Table 4.17 *Continued.*

Property URI	Used for class	Vocabulary name	Vocabulary URI	Usage note
dct:type	Licence document	ADMS licence type vocabulary	http://purl.org/adms/licencetype/	The list of terms in the ADMS licence type vocabulary is included in the ADMS specification
dcatap:availability	Distribution	Distribution availability vocabulary	http://data.europa.eu/r5r/availability/	The list of terms for the availability levels of a dataset distribution is included in the DCAT-AP specification

4.4.2 Pricing model

Here we present the general representation of a pricing model to describe the pricing information attached to the data assets related to legacy information of pricing specification in the marketplaces.

Pricing models associated with the DataOffering class is shown in Table 4.18.

- **Base class pricingmodel:PricingModel:**

Table 4.18 PricingModel basic properties.

Property name	Data types	Description
currency	xyz	Currency type
:description	String	A description
:name	String	Name
:hasPaymentOnSubscription	pricingmodel: PaymentOnSubscription	PaymentOnSubscription
:hasPaymentOnAPI	pricingmodel:PaymentOnAPI	PaymentOnAPI
:hasPaymentOnPlan	pricingmodel:PaymentOnPlan	PaymentOnPlan
:hasPaymentOnUnit	pricingmodel:PaymentOnUnit	PaymentOnUnit
:hasPaymentOnSize	pricingmodel:PaymentOnSize	PaymentOnSize
:hasFreePrice	pricingmodel:FreePrice	FreePrice

For payment categories from marketplace terms, we can have like:
pricingmodel:PaymentOnPlan, pricingmodel:PaymentOnAPI,
pricingmodel:PaymentOnUnit,
pricingmodel:PaymentOnSize,
pricingmodel:PaymentOnSubscriptiOn, and pricingmodel:FreePrice.

- **PaymentOnPlan class:**

The payment type class pricingmodel:PaymentOnPlan is shown in Table 4.19.

Table 4.19 PaymentOnPlan basic properties.

Property name	Data types	Description
:hasPlanPrice	String	Plan price
:description	String	A description
:name	String	Name
: :planDuration	String	Plan duration

- **PaymentOnAPI class:**

The payment type class pricingmodel:PaymentOnAPI is shown in Table 4.20.

Table 4.20 PaymentOnAPI basic properties.

Property name	Data types	Description
:hasAPIPrice	String	Basic price
:description	String	A description
:name	String	Name
pricingmodel:numberObject		Number of objects moved via API

- **PaymentOnUnit class:**

The payment type class pricingmodel:PaymentOnUnit is shown in Table 4.21.

Table 4.21 PaymentOnUnit basic properties.

Property name	Data types	Description
:hasUnitPrice	String	Basic price
:description	String	A description
:name	String	Name
pricingmodel:dataUnit		Data unit type
:unitID	String	:unit ID

- **PaymentOnSize class:**

The payment type class pricingmodel:PaymentOnSize is shown in Table 4.22.

Table 4.22 PaymentOnSize basic properties.

Property name	Data types	Description
:hasSizePrice	String	Basic price
:description	String	A description
:name	String	Name
:dataSize		Size of data

Table 4.23 PaymentOnSubscription basic properties.

Property name	Data types	Description
:hasSubscriptionPrice	String	Subscription price
:description	String	A description
:name	String	Name
:timeDuration	Time	Subscription d duration
:fromValue	Date time	Subscription validity starting point
:toValue	Date time	Subscription validity ending point
:repeat	pricingmodel:RepeatBy	In case the subscription is repeatable

- **PaymentOnSubscriptiOn class:**

The payment type class pricingmodel:PaymentOnSubscriptiOn is shown in Table 4.23.

- **FreePrice class:**

The payment type class pricingmodel:FreePrice is shown in Table 4.24.

Table 4.24 FreePrice basic properties.

Property name	Data types	Description
:hasFreePrice	String	Free option

Suggested data pricing-value model:

The pricing information is useful to compile the necessary details for smart contracts and other auditable information (Figure 4.5), plus we use parameters for helping users evaluate the possible best suggested prices for their assets that have to be traded/shared (Figure 4.6).

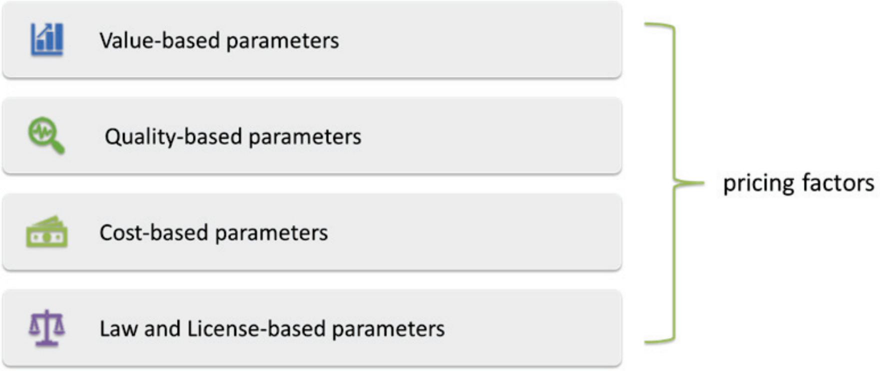
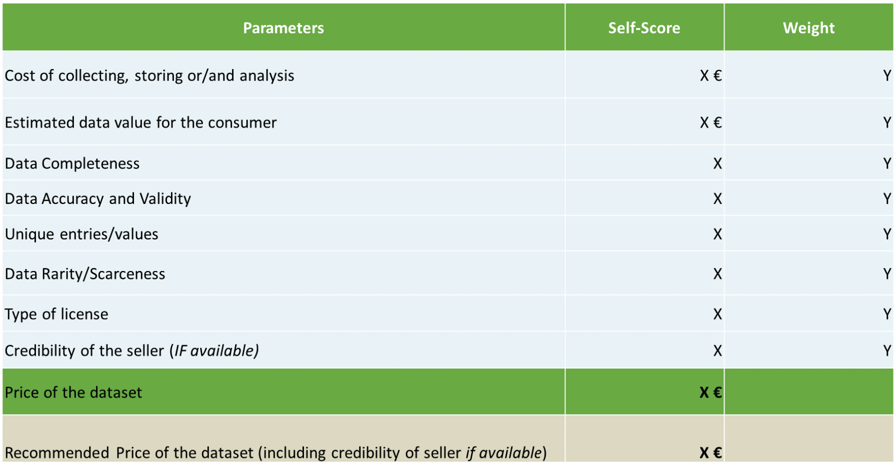


Figure 4.5 Data pricing-value model.

Parameters used in our price recommendation calculator tool are shown in Figure 4.6.



Parameters	Self-Score	Weight
Cost of collecting, storing or/and analysis	X €	Y
Estimated data value for the consumer	X €	Y
Data Completeness	X	Y
Data Accuracy and Validity	X	Y
Unique entries/values	X	Y
Data Rarity/Scarceness	X	Y
Type of license	X	Y
Credibility of the seller (*IF available)*	X	Y
Price of the dataset	**X €**	
Recommended Price of the dataset (including credibility of seller *if available*)	**X €**	

Figure 4.6 Data pricing-value model parameters.

4.4.3 Domain categorization/taxonomies for domain-specific annotations of datasets

Property: core:category and dcat:theme:

The dcat:theme is used to give annotation and information about the domain categorization of the datasets. In i3-MARKET, we use the themes as sub-categories to give more granularity in defining the domain annotations. In DCAT 1, the domain of *dcat:theme* was *dcat:Dataset*, which limited the use of this property in other contexts. The domain has been relaxed in later revisions.

We also added a upper level property for a data offering to annotate directly the high-level type of category the data offering belongs to as core:category.

Semantic attribute:	**dcat:theme**
Definition:	A category of the resource. A resource can have multiple themes
Sub-property of:	dct:subject
Range:	skos:Concept
Usage note:	The set of skos:Concepts used to categorize the resources are organized in a skos:ConceptScheme describing all the categories and their relations in the catalogue

Class: ConceptScheme:

Semantic class:	**skos:ConceptScheme**
Definition:	A knowledge organization system (KOS) used to represent themes/categories of datasets in the catalogue

Class: Concept:

Semantic Class:	textbfskos:Concept
Definition:	A category or a theme used to describe datasets in the catalogue
Usage note:	It is recommended to use either skos:inScheme or skos:topConceptOf on every skos:Concept used to classify datasets to link it to the concept scheme it belongs to. This concept scheme is typically associated with the catalogue using dcat:themeTaxonomy

We are using skos:ConceptScheme via skos:Concept to create taxonomies to annotate high-level types of annotations for domain themes/categories classifications.

Example of category terms as in i3-MARKET DataCategory.ttl schema.

```
Data Categories

<http://i3.market.eu/auth/dataCatagory/Manufacturing>
                skos:prefLabel "Manufacturing"@en.
<http://i3.market.eu/auth/dataCatagory/Automotive>
                skos:prefLabel "Automotive"@en.
<http://i3.market.eu/auth/dataCatagory/Wellbeing>
                skos:prefLabel "Wellbeing"@en.
<http://i3.market.eu/auth/dataCatagory/Agriculture>
                skos:prefLabel "Agriculture,   fisheries,   forestry   and
food"@en.
<http://i3.market.eu/auth/dataCatagory/Culture>
                skos:prefLabel "Culture and sport"@en.
<http://i3.market.eu/auth/dataCatagory/Economy>
                skos:prefLabel "Economy and finance"@en.
<http://i3.market.eu/auth/dataCatagory/Education>
                skos:prefLabel "Education"@en.
<http://i3.market.eu/auth/dataCatagory/Energy>
                skos:prefLabel "Energy"@en.
<http://i3.market.eu/auth/dataCatagory/Environment>
                skos:prefLabel "Environment"@en.
<http://i3.market.eu/auth/dataCatagory/Government>
                skos:prefLabel "Government and public sector"@en.
<http://i3.market.eu/auth/dataCatagory/Health>
                skos:prefLabel "Health"@en.
<http://i3.market.eu/auth/dataCatagory/International>
                skos:prefLabel "International issues"@en.
<http://i3.market.eu/auth/dataCatagory/Justice>
                skos:prefLabel "Justice, legal system and public safety"@en.
<http://i3.market.eu/auth/dataCatagory/Regions>
                skos:prefLabel "Regions and cities"@en.
<http://i3.market.eu/auth/dataCatagory/society>
                skos:prefLabel  "Population and society"@en.
<http://i3.market.eu/auth/dataCatagory/Science>
                skos:prefLabel "Science and technology"@en.
<http://i3.market.eu/auth/dataCatagory/Transport>
                skos:prefLabel "Transport"@en.
```

4.4.4 W3C Verifiable Credentials data model

For representing the Verifiable Credentials, the Backplane follows the W3c Verifiable Credentials Data Model 1.0.

Credentials are a part of our daily lives; driver's licenses are used to assert that we are capable of operating a motor vehicle, university degrees can be used to assert our level of education, and government-issued passports enable us to travel between countries. These credentials provide benefits to us when used in the physical world, but their use on the Web continues to be elusive.

Currently, it is difficult to express education qualifications, healthcare data, financial account details, and other sorts of third-party verified machine-readable personal information on the Web. The difficulty of expressing digital credentials on the Web makes it challenging to receive the same benefits through the Web that physical credentials provide us in the physical world.

This specification provides a standard way to express credentials on the Web in a way that is cryptographically secure, privacy respecting, and machine verifiable.

Also, in i3-MARKET, the SSI& IAM subsystems use DIDs that follow the W3c decentralized identifiers (DIDs) v1.0 specifications.

Decentralized identifiers (DIDs) are a new type of identifier that enables verifiable, decentralized digital identity. A DID refers to any subject (e.g., a person, organization, thing, data model, abstract entity, etc.) as determined by the controller of the DID. In contrast to typical, federated identifiers, DIDs have been designed so that they may be decoupled from centralized registries, identity providers, and certificate authorities. Specifically, while other parties might be used to help enable the discovery of information related to a DID, the design enables the controller of a DID to prove control over it without requiring permission from any other party. DIDs are URIs that associate a DID subject with a DID document allowing trustable interactions associated with that subject.

4.4.5 Smart contracts for data sharing agreements

How to create a fair and trusted ecosystem around the sharing/trading of data?

- ✓ Contracts set the basis of the data/sharing/trading.
- ✓ Contractual agreements are critical to ensure the fair and trustworthy sharing/trading of data.
- ✓ Smart contracts offer new affordances and opportunities to enhance trust in trading and sharing of data.

Most common clauses for DSAs:

- ○ General legal provisions:
    - ■ Parties and roles.
    - ■ Preamble, scope of the agreement, and definitions.

    - ■ Description/specification of the subject matter > description/scoping of the data and data trading and the agreement.
    - ■ Duties/obligations and rights > terms & conditions for data trading (room for flexibility).
    - ■ Intended use.
    - ■ Warranties > room for flexibility.
    - ■ Liability and dispute resolution mechanisms > room for flexibility.
    - ■ Duration and termination > performance, breach, modification, conditions for premature termination, and consequences.
  - ○ Specific license types:
    - ■ License grants.
    - ■ Intended use.

4.5 Online i3-MARKET Semantic Model Repository and Community Management

The results are shared not only with project partners but also with stakeholders and community in open-source repositories. As part of open-source assets, the data models, documentations, and files used in the i3-MARKET project are made available, such as the following.

- The i3-MARKET data pack is the set of files, schemas, and metadata model diagrams that represent the way the i3-MARKET semantics are organized and structured; it also contains the metadata in two different formats, e.g., ttl and Jason-ld. owl.
- The i3-MARKET semantic model info is the documentation that describes in detail all the taxonomies and vocabularies from needed domains used in i3-MARKET and that describes and represents all the relationships between them to build the graph representation of the i3-MARKET semantic model.
- The support repo is the mechanism for how the data model is maintained following the interoperability requirements in i3-MARKET. If you want to contribute or have any suggestion for improving the semantic models, visit the open-source repositories and contact the authors and members.
- The model files are shared in i3-MARKET GitHub/Gitlab repositories with release versions where each section contains the online machine-readable files in OWL and other format for online accessibility. The files

are maintained and updated regularly to keep the latest version of the models files up to date.

The code as well the models and vocabularies are available open-source via the establishment of the i3-MARKET spaces on Gitlab available at: https://gitlab.com/i3-MARKET-V3-public-repository/, and GitHub available at: https://github.com/i3-MARKET-V3-public-repository/.

The i3-MARKET semantic models governance process, which is defined as the support and evaluation process to include semantic improvements, is as follows:

- **Request for changes or updates:** Identify any changes previous to a major release, which should be considered private and usually is on testing and pre-consensus/staging.
- **The evaluation of any type of update request:** A review from editors and community approves participation and updates. In particular terms, vocabularies, ontologies or initiate a model in i3-MARKET OSS.
- **The communication of the results from technical experts:** A tagging version using alpha, beta, and gamma versions and then tagged as major is used here.
- **Evaluation of contributions for new commits:** Technical experts, PM, TM, TPMs, WPLs, and TaskLs, Assess and Evaluate the Contribution Includes documentation at the initiated project in i3-MARKET OSS.
- **Reports and changes report:** The technical board issues a short report, explaining the rational on the rejection in exceptional cases. This step can include rejection/cancellation of project participation.

It is possible to find a more complete definition of the attributes used in the data offerings description schema template as used in the Semantic Engine API in Appendix A.

4.6 Data Offerings Description–Schema Definitions in the API Template

When creating resources as per the data offering description, you fill the attribute fields to describe the traded/shared assets and datasets in the templates that are registered in the registry catalogue and allow the collection of information that are used by the engine and other components to retrieve details for search and retrieval of data for information systems and operations.

You can find the main semantic data model files for i3-MARKET in open-source GitHub and Gitlab repository project, e.g., at https://github.com/i3-MARKET-V3-Public-Repository/SemanticsDataModels.

Definitions for semantic description of data offerings in relation to the API template:

DataOffering:
{
"marketId":

Semantic attribute:	core:marketId
Definition:	This is the market name ID, which is uniquely identified a marketplace
Range:	Marketplace identifier: xsd:string
Usage note:	n/a
See also:	n/a

"provider":

Semantic attribute:	core:provider
Definition:	Provider of the DataOffering
Range:	Provider identifier: xsd:string
Usage note:	Should be the identifier of the provider in the i3-MARKET system verification that should be done with registered providers. All other providers shall be rejected.Return an error message in case an unregistered provider is specified.
See also:	Maybe connected with the IDs in identity manager. As the actual registration is by the Marketplaces/DataSpaces, they have the knowledge and responsibility to have the name/identity of the providers (that have knowledge of the owners) whom they would know are the providers

"owner":

Semantic attribute:	core:owner
Definition:	Owner of the DataOffering
Range:	Owner identifier: xsd:string
Usage note:	Should be the identifier of the owner in the i3-MARKET system. Owners are not registered in i3-MARKET. Optional parameter. Not to be verified.
See also:	n/a

"marketDid": (could be automatically filled by, e.g., WEB-RI in the creation moment of the data offering)

Semantic attribute:	core:marketDid
Definition:	This is the market DID, registered in VC and i3-MARKET, which is uniquely identified a marketplace
Range:	Marketplace identifier: DID
Usage note:	This ID is generated at the marketplace level, and inserting into an offering automatically by the marketplace itself rather than by a user.
See also:	

"providerDid": (could be automatically filled by, e.g., WEB-RI in the creation moment of the data offering)

Semantic attribute:	core:providerDid
Definition:	Provider of the DataOffering DID, registered in VC and i3-MARKET, which is uniquely identified
Range:	Provider identifier: DID
Usage note:	Should be the identifier of the provider in the i3-MARKET system. Verification should be done with registered providers. All other providers shall be rejected.Return an error message in case an unregistered provider is specified.
See also:	linked to VC

"ownerDid": (could be automatically filled by, e.g., WEB-RI in the creation moment of the data offering)

Semantic attribute:	core:ownerDid
Definition:	Owner of the DataOffering DID, registered in VC and i3-MARKET, which is uniquely identified
Range:	Owner identifier: DID
Usage note:	Should be the identifier of the owner in the i3-MARKET system. Owners are not registered in i3-MARKET. Optional parameter. Not to be verified.
See also:	Maybe connected with the IDs in identity manager

"ownerConsentForm": (should be implemented allowing the indication for user consent form hash details)

Semantic attribute:	core:ownerConsentForm
Definition:	Hashtag string to report the information about the explicit user consent form documentations
Range:	
Usage note:	Should be the Hashtag string to report the information about the explicit user consent form documentations.
See also:	n/a

"active":

Semantic attribute:	core:active
Definition:	Boolean to define if the DataOffering is ready to be visible
Range:	
Usage note:	Should be the Boolean to define if the DataOffering is ready to be visible. True or false.
See also:	n/a

"inSharedNetwork":

Semantic attribute:	core:inSharedNetwork
Definition:	Boolean to define if the DataOffering is shared by the marketplace to be visible and consumable by all actors in the i3-MARKET network
Range:	n/a
Usage note:	Should be the Boolean to define if the DataOffering is shared by the marketplace to be visible and consumable by all actors in the i3-MARKET network. True or false.
See also:	n/a

“personalData”:

Semantic attribute:	core:personalData
Definition:	Boolean: To define if the data offering offers a dataset that contains personal data
Range:	n/a
Usage note:	Should be the Boolean to define if the data offering offers a dataset that contains personal data.
See also:	n/a

“dataOfferingTitle”:

Semantic attribute:	core:dataOfferingTitle
Definition:	The title of the DataOffering
Range:	xsd:string
Usage note:	A name to identify the DataOffering. A few words only that summarize the offering.
See also:	n/a

“dataOfferingDescription”:

Semantic attribute:	core:dataOfferingDescription
Definition:	A description of the DataOffering
Range:	xsd:string
Usage note:	Used to have description text to describe what the data offering is about.This can be a long block of text. At least 1000 characters shall be reserved for this.
See also:	n/a

“category”:

Semantic attribute:	core:category
Definition:	A category to have a high-level classification of domain for the DataOffering
Range:	xsd:anyURI
Usage note:	Use the categories naming schema defined for high-level categories as URIs:Categories should only be added by extending the categories list. This is done by the community.The parameter should be checked against this list. If it does not match, return an error.

See also:	Categories in table below prefix: dataCatagory <http://i3.market.eu/auth/dataCatagory> dataCatagory:Automotive Data categories (as per definitions in Gitlab file:https://gitlab.com/i3-MARKET/code/data-models/-/blob/master/Version-1/DataOfferingCategory.ttl) <http://i3.market.eu/auth/dataCatagory/Manufacturing> <http://i3.market.eu/auth/dataCatagory/Automotive> <http://i3.market.eu/auth/dataCatagory/Wellbeing> <http://i3.market.eu/auth/dataCatagory/Agriculture> <http://i3.market.eu/auth/dataCatagory/Culture> <http://i3.market.eu/auth/dataCatagory/Economy> <http://i3.market.eu/auth/dataCatagory/Education> <http://i3.market.eu/auth/dataCatagory/Energy> <http://i3.market.eu/auth/dataCatagory/Environment> <http://i3.market.eu/auth/dataCatagory/Government> <http://i3.market.eu/auth/dataCatagory/Health> <http://i3.market.eu/auth/dataCatagory/International> <http://i3.market.eu/auth/dataCatagory/Justice> <http://i3.market.eu/auth/dataCatagory/Regions> <http://i3.market.eu/auth/dataCatagory/Society> <http://i3.market.eu/auth/dataCatagory/Science> <http://i3.market.eu/auth/dataCatagory/Transport> See also file DataOfferingCategory.ttl

“status”:

Semantic attribute:	core:status
Definition:	To define the DataOffering status
Range:	xsd:string
Usage note:	Possible values:“Inactive”: The offer is not visible but still exists and can be activated again.“ToBeDeleted”: Data offer is still available and visible but will be deleted once the last contract on this offer expired. No new purchases are allowed on it.“Deleted”: The offer is not visible and cannot be activated again. It is no longer available for consumers or providers.
Note:	Rename this field to “*Status*”. Possible values:“Inactive”: The offer is not visible but still exists and can be activated again.“ToBeDeleted”: Data offer is still available and visible but will be deleted once the last contract on this offer expired. No new purchases are allowed on it.“Deleted”: The offer is not visible and cannot be activated again. It is no longer available for consumers or providers.

"dataOfferingExpirationTime":

Semantic attribute:	core:dataOfferingExpirationTime
Definition:	Expiration time of DataOffering in case
Range:	Can be: xsd:dateTime
Usage note:	The dateTime data type is used to specify a date and a time. The dateTime is specified in the following form "YYYY-MM-DDThh:mm:ss" where: • YYYY indicates the year • MM indicates the month • DD indicates the day • T indicates the start of the required time section • hh indicates the hour • mm indicates the minute • ss indicates the second **Note:** All components are required! The following is an example of a dateTime declaration in a schema: "2002-05-30T09:00:00".
See also:	n/a

"dataOfferingCreated": (this can be created automatically by the system at registration time, by engine timestamp, instead of manually by market...)

RDF property	core:dataOfferingCreated
Definition:	Date of formal issuance (e.g., publication) of the data offering
Range:	Encoded using the relevant ISO 8601 date and time compliant string (DATETIME) and typed using the appropriate XML schema datatype (XMLSCHEMA11-2) (xsd:dateTime)
Usage note:	This property *should* be set using the first known date of issuance. The date of the initial publication of this data offering in i3-MARKET.
See also:	Property: release date

"lastModified": (this can be created automatically by the system at registration time, by engine timestamp, instead of manually by market...)

Semantic attribute:	core:lastModified
Definition:	Most recent date on which the data offering was changed, updated, or modified

Range:	Encoded using the relevant ISO 8601 date and time compliant string (DATETIME) and typed using the appropriate XML schema datatype (XMLSCHEMA11-2) (xsd:dateTime)
Usage note:	The value of this property indicates a change to the data offering record. An absent value *may* indicate that the item has never changed after itsinitial publication, that the date of last modification is not known, or that the item is continuously updated.
See also:	Property: frequency, Property: update/modification date, and Property: update/modification date in DCAT 3 webpage

"versionNotes":

Semantic attribute:	adms:versionNotes
Definition:	A description of changes between this version and the previous version of the resource (VOCAB-ADMS)
Range:	xsd:string
Usage note:	In case of backward compatibility issues with the previous version of the resource, a textual description of them *should* be specified by using this property.
See also:	Property: current version, Property: has version, Property: is replaced by, Property: is version of, Property: previous version, Property: release date, Property: replaces, Property: status, and Property: version notes

"previousVersion":

Semantic attribute:	dcat:previousVersion
Definition:	The previous version of a resource in a lineage (PAV)
Range:	xsd:anyURI
Usage note:	This property is meant to be used to specify a version chain, consisting of snapshots of a resource. The notion of version used by this property is limited to versions resulting from revisions occurring to a resource as part of its lifecycle. One of the typical cases here is representing the history of the versions of a dataset that have been released over time.
See also:	Property: current version, Property: has version, Property: is replaced by, Property: is version of, Property: previous version, Property: release date, Property: replaces, Property: status, and Property: version notes

“replaces”:

Semantic attribute:	dcterms:replaces
Definition:	A related resource that is supplanted, displaced, or superseded by the described resource (DCTERMS)
Range:	xsd:anyURI
Usage note:	Resource replaced.
See also:	Property: current version, Property: has version, Property: is replaced by, Property: is version of, Property: previous version, Property: release date, Property: replaces, Property: status, and Property: version notes

“previousVersion”:

Semantic attribute:	dcat:previousVersion
Definition:	The previous version of a resource in a lineage (PAV)
Range:	xsd:anyURI
Usage note:	This property is meant to be used to specify a version chain, consisting of snapshots of a resource. The notion of version used by this property is limited to versions resulting from revisions occurring to a resource as part of its lifecycle. One of the typical cases here is representing the history of the versions of a dataset that have been released over time.
See also:	Property: current version, Property: has version, Property: is replaced by, Property: is version of, Property: previous version, Property: release date, Property: replaces, Property: status, and Property: version notes

“contractParameters”:
{
“interestOfProvider”:

Semantic attribute:	core:interestOfProvider
Definition:	This property is used to identify the interest of the data owner/provider related to the trading/sharing of their data assets. The following possibilities exist: • Free sharing • Quotation • Selling of data (e.g., just earning money by selling the data, no specific feedback on these data by a data consumer expected)
Range:	xsd:string
Usage note:	It could be simple notations like: Free Sharing – Quotation − Selling of data; or we can decide to have specific definitions for our system.
See also:	n/a

“interestDescription”:

Semantic attribute:	core:interestDescription
Definition:	Data provider can specify which sort of quotation he wants exactly, e.g., quotation for maintenance service or quotation for optimization of production
Range:	xsd:string
Usage note:	More text description of the interest of the data owner/provider related to the trading/sharing of their data assets.Example: “This data is shared only for the purpose of creating a quotation for maintenance for the production machines described in the dataset. Any other use of this data is not permitted”.
Note:	n/a

“hasGoverningJurisdiction”:

Semantic attribute:	core:hasGoverningJurisdiction
Definition:	The file format of the distribution
Range:	xsd:string (or xsd:anyURI)
Usage note:	Can be string naming like: GLOBAL US-JURISDICTION EU-JURISDICTION (or we use URIs to define the specific terms for jurisdictions) To be extended to define a list of jurisdictions that are valid here.
See also:	n/a

“purpose”:

Semantic attribute:	core:purpose
Definition:	Purpose of the agreement
Range:	xsd:string
Usage note:	Short label for the purpose. In case we could have specific terminology for define list of @purpose@ terms.
Note:	This parameter is part of the contractual parameters. Ask contract partners, what this is for (Susanne).

“purposeDescription”:

Semantic attribute:	core:purposeDescription
Definition:	In case full text description of describing the reasons behind the creation of the agreement
Range:	xsd:string
Usage note:	Text description.
Note:	This parameter is part of the contractual parameters. Ask contract partners, what this is for (Susanne).

“hasIntendedUse”:

{

“processData”: “true OR false”

Semantic attribute:	core:processData
Definition:	If consumer allowed to process data
Range:	xsd:boolean
Usage note:	The value space of xsd:boolean is true and false. Its lexical space accepts true, false, “TRUE”, or “FALSE”.
Note:	Part of contractual parameters. Ask contract partners, what this is for.Make this parameter to type Boolean.

“shareDataWithThirdParty”: “true OR false”

Semantic attribute:	core:shareDataWithThirdParty
Definition:	If consumer allowed to share data with third parties
Range:	xsd:boolean
Usage note:	The value space of xsd:boolean is true and false. Its lexical space accepts true, false, “TRUE”, or “FALSE”.
Note:	Part of contractual parameters. Ask contract partners, what this is for.Make this parameter to type Boolean.

“editData”: “true OR false”

Semantic attribute:	core:editData
Definition:	If consumer is allowed to edit the data
Range:	xsd:boolean
Usage note:	The value space of xsd:boolean is true and false. Its lexical space accepts true, false, “TRUE”, or “FALSE”.
Note:	Part of contractual parameters. Ask contract partners, what this is for.Make this parameter to type Boolean.

},

"hasLicenseGrant":

{

"paidUp": "true OR false"

Semantic attribute:	core:paidUp
Definition:	If licence grant to paidUp
Range:	xsd:boolean
Usage note:	The value space of xsd:boolean is true and false. Its lexical space accepts true, false, "TRUE", or "FALSE".
Note:	Part of contractual parameters. Ask contract partners, what this is for.

"transferable": "true OR false"

Semantic attribute:	core:transferable
Definition:	If licence is transferable
Range:	xsd:boolean
Usage note:	The value space of xsd:boolean is true and false. Its lexical space accepts true, false, "TRUE", or "FALSE".
See also:	n/a

"exclusiveness": "true OR false"

Semantic attribute:	core:exclusiveness
Definition:	If licence grant exclusiveness
Range:	xsd:boolean
Usage note:	The value space of xsd:boolean is true and false. Its lexical space accepts true, false, "TRUE", or "FALSE".
See also:	n/a

"revocable": "true OR false"

Semantic attribute:	core:revocable
Definition:	If licence is revocable
Range:	xsd:boolean
Usage note:	The value space of xsd:boolean is true and false. Its lexical space accepts true, false, "TRUE", or "FALSE".
See also:	n/a

"processing": "true OR false"

Semantic attribute:	core:processing
Definition:	If licence grant data to be processed
Range:	xsd:boolean
Usage note:	The value space of xsd:boolean is true and false. Its lexical space accepts true, false, "TRUE", or "FALSE".
Note:	Part of contractual parameters. Ask contract partners, what this is for.

"modifying": "true OR false"

Semantic attribute:	core:modifying
Definition:	If licence grant data to be modified
Range:	xsd:boolean
Usage note:	The value space of xsd:boolean is true and false. Its lexical space accepts true, false, "TRUE", or "FALSE".
Note:	Part of contractual parameters. Ask contract partners, what this is for.

"analyzing": "true OR false"

Semantic attribute:	core:analyzing
Definition:	If licence grant data to be analysed
Range:	xsd:boolean
Usage note:	The value space of xsd:boolean is true and false. Its lexical space accepts true, false, "TRUE", or "FALSE".
Note:	Part of contractual parameters. Ask contract partners, what this is for.

"storingData": "true OR false"

Semantic attribute:	core:storingData
Definition:	If licence grant to store data
Range:	xsd:boolean
Usage note:	The value space of xsd:boolean is true and false. Its lexical space accepts true, false, "TRUE", or "FALSE".
Note:	Part of contractual parameters. Ask contract partners, what this is for.

"storingCopy": "true OR false"

Semantic attribute:	core:storingCopy
Definition:	If licence grant to store a copy data
Range:	xsd:boolean
Usage note:	The value space of xsd:boolean is true and false. Its lexical space accepts true, false, "TRUE", or "FALSE".
Note:	Part of contractual parameters. Ask contract partners, what this is for.

"reproducing": "true OR false"

Semantic attribute:	core:reproducing
Definition:	If licence grant to reproduce data
Range:	xsd:boolean
Usage note:	The value space of xsd:boolean is true and false. Its lexical space accepts true, false, "TRUE", or "FALSE".
Note:	Part of contractual parameters. Ask contract partners, what this is for.

"distributing": "true OR false"

Semantic attribute:	core:distributing
Definition:	If licence grant to distribute data
Range:	xsd:boolean
Usage note:	The value space of xsd:boolean is true and false. Its lexical space accepts true, false, "TRUE", or "FALSE".
Note:	Part of contractual parameters. Ask contract partners, what this is for.

"loaning": "true OR false"

Semantic attribute:	core:loaning
Definition:	If licence grant to loan data
Range:	xsd:boolean
Usage note:	The value space of xsd:boolean is true and false. Its lexical space accepts true, false, "TRUE", or "FALSE".
Note:	Part of contractual parameters. Ask contract partners, what this is for.

“selling”: “true OR false”

Semantic attribute:	core:selling
Definition:	If licence grant to sell data
Range:	xsd:boolean
Usage note:	The value space of xsd:boolean is true and false. Its lexical space accepts true, false, “TRUE”, or “FALSE”.
Note:	Part of contractual parameters. Ask contract partners, what this is for.

“renting”: “true OR false”

Semantic attribute:	core:renting
Definition:	If licence grant to rent data
Range:	xsd:boolean
Usage note:	The value space of xsd:boolean is true and false. Its lexical space accepts true, false, “TRUE”, or “FALSE”.
Note:	Part of contractual parameters. Ask contract partners, what this is for.

“furtherLicensing”: “true OR false”

Semantic attribute:	core:furtherLicensing
Definition:	If licence grant for further licensing
Range:	xsd:boolean
Usage note:	The value space of xsd:boolean is true and false. Its lexical space accepts true, false, “TRUE”, or “FALSE”.
Note:	Part of contractual parameters. Ask contract partners, what this is for.

“leasing”: “true OR false”

Semantic attribute:	core:leasing
Definition:	If licence grant to lease data
Range:	xsd:boolean
Usage note:	The value space of xsd:boolean is true and false. Its lexical space accepts true, false, “TRUE”, or “FALSE”.
Note:	Part of contractual parameters. Ask contract partners, what this is for.

```
    }       },
```

“hasPricingModel”:

{

“pricingModelName”:

Semantic attribute:	pricingmodel:pricingModelName
Definition:	The name to define the legacy, by marketplace, pricing model related to the data offering
Range:	xsd:string
Usage note:	Pricing models are individually defined by marketplaces. No common pricing model will be defined for i3-MARKET. Maybe try to generalize existing pricing models.
See also:	

“basicPrice”:

Semantic attribute:	pricingmodel:basicPrice
Definition:	The generic basic price for the traded data for basic cost of trade
Range:	xsd:double
Usage note:	Number related to price.
See also:	

“currency”:

Semantic attribute:	pricingmodel:currency
Definition:	The file format of the distribution
Range:	xsd:string
Usage note:	Using ISO 4215 currency terminology.
See also:	lis-ISO-4217-Currencyt_one.xml See XML file for three-letter abbreviations. lis-ISO-4217-Currencyt_one.xml

“hasPaymentOnSubscription”:

{

“timeDuration”:

Semantic attribute:	pricingmodel:timeDuration
Definition:	Time duration of subscription
Range:	xsd:anyURI

Usage note:	Or generic xsd:string text with labels for duration vocabulary or URIs with vocabulary like: "http://reference.data.gov.uk/def/intervals/Day" "http://reference.data.gov.uk/def/intervals/Hour" "http://reference.data.gov.uk/def/intervals/Minute" "http://reference.data.gov.uk/def/intervals/Month" "http://reference.data.gov.uk/def/intervals/Quarter" "http://reference.data.gov.uk/def/intervals/Second" Price is per timeDuration. For example, if parameter is "Second" here, then the specified price is per second (€/ s).
See also:	Terms in intervals.rdf

"description":

Semantic attribute:	dcterms:description
Definition:	The description of payment on subscription
Range:	xsd:string
Usage note:	Text description.
See also:	n/a

"repeat":

Semantic attribute:	pricingmodel:repeat
Definition:	If subscription can be repeated define the frequency, e.g., daily, monthly, etc.
Range:	xsd:anyURI
Usage note:	We can use specific vocabulary For example, in freq.ttl definitions like: http://purl.org/cld/freq/daily freq:monthly freq:weekly …….
See also:	See also freq.ttl or frequency.ttl.txt

"hasSubscriptionPrice":

Semantic attribute:	pricingmodel:hasSubscriptionPrice
Definition:	Price allocated to subscription payment type
Range:	xsd:double
Usage note:	Price.
See also:	n/a

} ,

"hasPaymentOnPlan":

{

There may be things like basic plan, premium plans, etc. that gives access to certain types of data. Which are difficult to implement in i3-MARKET.

Example for other usage: Deliver data only once a month or once every *x* period.Optional parameter does not have to be used.

"description":

Semantic attribute:	pricingmodel:planDescription
Definition:	The text description of plan
Range:	Xsd:string
Usage note:	Description text.
See also:	n/a

"planDuration":

Semantic attribute:	pricingmodel:planDuration
Definition:	The duration of the plan
Range:	xsd:anyURI
Usage note:	Or generic xsd:string text with labels for duration vocabulary or URIs with vocabulary like: "http://reference.data.gov.uk/def/intervals/Day" "http://reference.data.gov.uk/def/intervals/Hour" "http://reference.data.gov.uk/def/intervals/Minute" "http://reference.data.gov.uk/def/intervals/Month" "http://reference.data.gov.uk/def/intervals/Quarter" "http://reference.data.gov.uk/def/intervals/Second"
See also:	Terms in intervals.rdf

"hasPlanPrice": "string"

Semantic attribute:	pricingmodel:hasPlanPrice
Definition:	The price of the plan
Range:	xsd:double
Usage note:	Price.
See also:	n/a

} ,

“hasPaymentOnApi”:

{

“description”:

Semantic attribute:	Dcterms:description
Definition:	The text description of payment type
Range:	Xsd:string
Usage note:	Description text.
Note:	Optional. Useful for Agora.

“numberOfObject”:

Semantic attribute:	pricingmodel:numberObject
Definition:	Number of objects for API handle payments
Range:	Xsd:double
Usage note:	
Note:	Optional. Useful for Agora.

“hasAPIPrice”: “string”

Semantic attribute:	pricingmodel:hasAPIPrice
Definition:	The price of the API payment type
Range:	xsd:double
Usage note:	Price.
Note:	Optional. Useful for Agora.

}   ,

“hasPaymentOnUnit”:

{

"description":

Semantic attribute:	Dcterms:description
Definition:	The text description of payment type
Range:	Xsd:string
Usage note:	Description text. Purchase a cluster of data. Sets of data. One cluster is a group of datasets.
See also:	n/a

“dataUnit”:

Semantic attribute:	pricingmodel:dataUnit
Definition:	Data unit type handle by service
Range:	Xsd:string
Usage note:	Define what the unit resembles. Example: A predefined dataset. A “Unit” of transaction as indicated in specification of the service method of exchange.
See also:	Data unit type – In telecommunications, a *protocol data unit (PDU)* is a single unit of information transmitted among peer entities of a computer network. For example, the data unit in which data are packeted when transmitted in streams. Also, e.g., a data unit that contains one or many stream data objects.

“hasUnitPrice”: “string”

Semantic attribute:	pricingmodel:hasUnitPrice
Definition:	The price of the unit by payment type
Range:	xsd:double
Usage note:	Price per data unit.
See also:	n/a

} ,

“hasPaymentOnSize”:

{

“description”:

Semantic attribute:	Dcterms:description
Definition:	The text description of payment type
Range:	Xsd:string
Usage note:	Description text.
See also:	n/a

“dataSize”:

Semantic attribute:	pricingmodel:dataSize
Definition:	The size of data exchanged for payment
Range:	Typically typed as xsd:nonNegativeInteger
Usage note:	The size in bytes can be approximated (as a non-negative integer) when the precise size is not known. While it is recommended that the size be given as an integer, alternative literals such as “1.5 MB” are sometimes used.
See also:	We can decide to use a specific vocabulary

"hasSizePrice": "string"

Semantic attribute:	pricingmodel:hasSizetPrice
Definition:	The price of the unit by payment type
Range:	xsd:double
Usage note:	Price, e.g., pay per megabyte of data.
See also:	n/a

} ,

"hasFreePrice":

{

"hasPriceFree": "FREE"

Semantic attribute:	pricingmodel:hasPriceFree
Definition:	The data is shared for free
Range:	Xsd:string
Usage note:	"FREE". Data is for free, no payment needed.
See also:	We might use an URI as Pricingmodel:Free as unique term

} } ,

"hasDataset":

{ (Dataset description)

Description of the datasets contained. Note: This is not a description of the individual data items but an overview.

"title":

Semantic attribute:	dcterms:title
Definition:	A name given to the dataset
Range:	Xsd:string [rdfs:Literal]
Usage note:	Title.
See also:	n/a

"keyword":

Semantic attribute:	dcat:keyword
Definition:	A keyword or tag describing the resource
Range:	Xsd:string [rdfs:Literal]
Usage note:	Text keywords (in case we can decide to have a selection of terminologies to set as keywords). One or more keywords describing the data.
See also:	To have multiple keywords, you can have multiple instances of the property "keyword"

"description":

Semantic attribute:	dcterms:description
Definition:	A free-text account of the dataset
Range:	Xsd:string [rdfs:Literal]
Usage note:	Description text of dataset.
See also:	n/a

"issued":

RDF property	dcterms:issued
Definition:	Date of formal issuance (e.g., publication) of the distribution
Range:	Encoded using the relevant ISO 8601 date and time compliant string (DATETIME) and typed using the appropriate XML schema datatype (XMLSCHEMA11-2) (xsd:dateTime)
Usage note:	This property *should* be set using the first known date of issuance. The date of the initial publication of this dataset in i3-MARKET.
See also:	§ 6.4.7 Property: release date

"modified":

Semantic attribute:	dcterms:modified
Definition:	Most recent date on which the item was changed, updated, or modified
Range:	Encoded using the relevant ISO 8601 date and time compliant string (DATETIME) and typed using the appropriate XML schema datatype (XMLSCHEMA11-2) (xsd:dateTime)
Usage note:	The value of this property indicates a change to the actual item, not a change to the catalogue record. An absent value *may* indicate that the item has never changed after itsinitial publication, that the date of last modification is not known, or that the item is continuously updated.
See also:	§ 6.6.2 Property: frequency, § 6.5.4 Property: update/modification date, and § 6.8.4 Property: update/modification date in DCAT 3 webpage

"temporal":

Semantic attribute:	dcterms:temporal
Definition:	The temporal period that the dataset covers
Range:	In general, used singularly can be used URIs as in intervals vocab OR dcterms:PeriodOfTime (an interval of time that is named or defined by its start and end dates)

Usage note:	In case we extend the model to serve the temporal coverage of a dataset may be encoded as an instance of dcterms:PeriodOfTime, or may be indicated using an IRI reference (link) to a resource describing a time period or interval.
	For example, as [a dcterms:PeriodOfTime]
	dcat:startDate “2016-03-04”∧∧xsd:dateTime;
	dcat:endDate “2018-08-05”∧∧xsd:dateTime;

See also:	Intervals.rdf

“language”:

Semantic attribute:	dcterms:language
Definition:	A language of the item. This refers to the natural language used for textual metadata (i.e., titles, descriptions, etc.) of a catalogued resource (i.e., dataset or service) or the textual values of a dataset distribution
Range:	Resources defined by the Library of Congress (ISO 639-1, ISO 639-2) *should* be used If an ISO 639-1 (two-letter) code is defined for language, then its corresponding IRI *should* be used; if no ISO 639-1 code is defined, then IRI corresponding to the ISO 639-2 (three-letter) code *should* be used
Usage note:	Repeat this property if the resource is available in multiple languages.
See also:	Also if representations of a dataset are available for each language separately, define an instance of dcat:Distribution for each language and describe the specific language of each distribution using dcterms:language (i.e., the dataset will have multiple dcterms:language values and each distribution will have just one as the value of its dcterms:language property).

“spatial”:

Semantic attribute:	dcterms:spatial
Definition:	The geographical area covered by the dataset
Range:	Xsd:anyURI to use in case using a IRI reference (link) to a resource describing a location. It is recommended that links are to entries in a well-maintained gazetteer such as Geonames Or a dcterms:Location (a spatial region or named place)
Usage note:	The spatial coverage of a dataset may be encoded as an instance of dcterms:Location. Or may be indicated using an IRI reference (link) to a resource describing a location. It is recommended that links are to entries in a well-maintained gazetteer such as Geonames.

See also:	For example, for bbox dcterms:spatial [[a dcterms:Location] dcat:bbox """POLYGON[[3.053 47.975 , 7.24 47.975 , 7.24 53.504 , 3.053 53.504 , 3.053 47.975]]""" ;]

"accrualPeriodicity":

Semantic attribute:	dcterms:accrualPeriodicity
Definition:	The frequency at which a dataset is published
Range:	xsd:anyURI
Usage note:	We can use specific vocabulary For example, in freq.ttl definitions like: http://purl.org/cld/freq/daily freq:monthly freq:weekly.
See also:	See also freq.ttl or at frequency.ttl.txt

"temporalResolution":

Semantic attribute:	dcat:temporalResolution
Definition:	Minimum time period resolvable in the dataset
Range:	xsd:duration
Usage note:	If the dataset is a time-series, this should correspond to the spacing of items in the series. For other kinds of dataset, this property will usually indicate the smallest time difference between items in the dataset.
See also:	n/a

"theme": [

Semantic attribute:	dcat:theme
Definition:	A (sub-)category of the resource. A resource can have multiple themes
Range:	It would be better to have xsd:anyURI with URIs that represent the various terms in a vocabulary (to be defined with pilot partners for terms related to domains)
Usage note:	Use this for domain-specific categories. For example, subcategories like production machines, assembly lines, etc.To be defined by each application domain.Theme can be used multiple times to provide multiple subcategories. The set of themes used to categorize the resources are organized in a skos:ConceptScheme, skos:Collection, owl:Ontology, or similar, describing all the categories and their relations in the catalogue.
See also:	

],

“distribution”: (Distribution: A specific representation of a dataset. A dataset might be available in multiple serializations that may differ in various ways, including natural language, media-type or format, schematic organization, temporal and spatial resolution, and level of detail or profiles [which might specify any or all of the above]).

{

“title”:

Semantic attribute:	dcterms:title
Definition:	A name given to the distribution
Range:	Xsd:string [rdfs:Literal]
Usage note:	Title.
See also:	n/a

“description”:

Semantic attribute:	dcterms:description
Definition:	A free-text account of the distribution
Range:	Xsd:string [rdfs:Literal]
Usage note:	Description text of dataset.
See also:	n/a

“license”:

Semantic attribute:	dcterms:license
Definition:	A legal document under which the distribution is made available
Range:	dcterms:LicenseDocument
Usage note:	For interoperability, it is recommended to use canonical IRIs of well-known licenses such as those defined by Creative Commons. Information about licenses and rights *should* be provided on the level of distribution. Information about licenses and rights *may* be provided for a dataset in addition to but not instead of the information provided for the distributions of that dataset. Providing license or rights information for a dataset that is different from information provided for a distribution of that dataset *should* be avoided as this can create legal conflicts. See also guidance at §9. License and rights statements.
See also:	Property: rights Property: license **ToDo: Describe a list of possible licenses here.**

"accessRights":

Semantic attribute:	dcterms:accessRights
Definition:	Information about who can access the resource or an indication of its security status
Range:	dcterms:LicenseDocument
Usage note:	Information about licenses and rights *may* be provided for the resource. To express statements concerning only access rights (e.g., whether data can be accessed by anyone or just by authorized parties). Access rights can also be expressed as code lists/taxonomies. Examples include the access rights code list (EUV-AR) used in (DCAT-AP) and the Eprints Access Rights Vocabulary Encoding Scheme.
See also:	Property: rights dcterms:accessRights <http://publications.europa.eu/resource/authority/access-right/PUBLIC> ; dcterms:conformsTo <http://www.opengis.net/def/serviceType/ogc/csw> ;

"downloadType":

Semantic attribute:	core:downloadType
Definition:	Information about download type (it means "Stream" or "Bulk" dataset download)
Range:	xsd:string
Usage note:	To use a set of words like "Stream" and "Bulk".
See also:	n/a

"dataStream":

Semantic attribute:	core:dataStream
Definition:	Boolean attribute to check if the dataset is offered as a stream or not
Range:	
Usage note:	Should be the Boolean attribute to check if the dataset is offered as stream or not in the "Distribution" class block.
See also:	n/a

“conformsTo”:

Semantic attribute:	dcterms:conformsTo
Definition:	An established standard to which the distribution conforms (very optional)
Range:	dcterms:Standard (A basis for comparison; a reference point against which other things can be evaluated.)
Usage note:	This property *should* be used to indicate the model, schema, ontology, view, or profile that this representation of a dataset conforms to. This is (generally) a complementary concern to the media-type or format. This is a link to a specific file that describes the data in a domain specific format. It can also be a text in a freely definable format.
See also:	Property: format, Property: media type Also check file-type.ttl.txt

“mediaType”:

Semantic attribute:	dcat:mediaType
Definition:	The media-type of the distribution as defined by IANA (IANA-MEDIA-TYPES)
Range:	Xsd:anyURI [dcterms:MediaType]
Usage note:	dcat:mediaType *should* be used if the type of the distribution is defined by IANA (IANA-MEDIA-TYPES). https://www.iana.org/assignments/media-types/ For example, mediaType <http://www.iana.org/assignments/media-types/application/ld+json> For example, a link to a XML, csv, or JSON file, to describe the data format.
See also:	Property: media type, Property: conforms to Check also file-type.ttl.txt

“packageFormat”:

Semantic attribute:	dcat:packageFormat
Definition:	The package format of the distribution in which one or more data files are grouped together, e.g., to enable a set of related files to be downloaded together
Range:	Xsd:anyURI [dcterms:MediaType]
Usage note:	In case it is compressed, this could be .zip, .rar, etc. This property to be used when the files in the distribution are packaged, e.g., in a TAR file, a Frictionless Data Package, or a Bagit file. Theformat *should* be expressed using a media-type as defined by IANA (IANA-MEDIA-TYPES), if available.
See also:	Property: compression format.

"accessService": (info inside distribution for service that serves the distributions of the datasets)

{

"conformsTo":

Semantic attribute:	dcterms:conformsTo
Definition:	An established standard to which the distribution conforms
Range:	dcterms:Standard (A basis for comparison; a reference point against which other things can be evaluated.)
Usage note:	This property *should* be used to indicate the model, schema, ontology, view, or profile that this representation of a dataset conforms to. This is (generally) a complementary concern to the media-type or format.
See also:	Property: conforms to

"endpointDescription":

Semantic attribute:	dcat:endpointDescription
Definition:	A description of the services available via the endpoints, including their operations, parameters, etc.
Range:	xsd:string
Usage note:	The endpoint description gives specific details of the actual endpoint instances, while dcterms:conformsTo is used to indicate the general standard or specification that the endpoints implement. An endpoint description may be expressed in a machine-readable form, such as an OpenAPI (Swagger) description (OpenAPI), an OGC GetCapabilities response (WFS), (ISO-19142), (WMS), (ISO-19128), a SPARQL service description (SPARQL11-SERVICE-DESCRIPTION), an (OpenSearch) or (WSDL20) document, a Hydra API description (HYDRA), and else in text or some other informal modes if a formal representation is not possible.
See also:	n/a

"endpointURL":

Semantic attribute:	dcat:endpointURL
Definition:	The root location or primary endpoint of the service (a Web-resolvable IRI)
Range:	xsd:anyURI
Usage note:	The URL address of the resource via service.
See also:	n/a

“servesDataset”:

Semantic attribute:	dcat:servesDataset
Definition:	A collection of data that this data service can distribute. The dataset ID or name and files
Range:	xsd:string
Usage note:	To point to the datasets that are served via the data service.
See also:	n/a

“serviceSpecs”: “string”

Semantic attribute:	core:serviceSpecs
Definition:	Description of service specification for more details on the data service implementations
Range:	
Usage note:	To extend in case the description of data service to add more detailed descriptions on the system.To describe more details about the service, e.g. QoS, etc.
See also:	n/a

“dataExchangeSpec”: (info inside accessService block for data exchange specifications that serve the distributions of the datasets)

{

“encAlg”: “string”

Semantic attribute:	core:encAlg
Definition:	Encryption algorithm used to encrypt blocks. Either AES-128-GCM (’A128GCM’) or AES-256-GCM (’A256GCM)
Range:	
Usage note:	Encryption algorithm used to encrypt blocks. Either AES-128-GCM (’A128GCM’) or AES-256-GCM (’A256GCM).
See also:	n/a

“signingAlg”: “string”

Semantic attribute:	core:signingAlg
Definition:	Signing algorithm used to sign the proofs. Like ECDSA secp256r1 with key lengths: either “ES256”, “ES384”, or “ES512”
Range:	n/a
Usage note:	Signing algorithm used to sign the proofs. It is ECDSA secp256r1 with key lengths: either “ES256”, “ES384”, or “ES512”.
See also:	n/a

"hashAlg": "string"

Semantic attribute:	core:hashAlg
Definition:	Hash algorithm used to compute digest/commitments. It is SHA2 with different output lengths: either "SHA-256", "SHA-384", or "SHA-512"
Range:	
Usage note:	Hash algorithm used to compute digest/commitments. It is SHA2 with different output lengths: either "SHA-256", "SHA-384", or "SHA-512".
See also:	n/a

"ledgerContractAddress": "string"

Semantic attribute:	core:ledgerContractAddress
Definition:	The ledger smart contract address (hexadecimal) on the DLT
Range:	n/a
Usage note:	The ledger smart contract address (hexadecimal) on the DLT.
See also:	n/a

"ledgerSignerAddress": "string"

Semantic attribute:	core:ledgerSignerAddress
Definition:	The orig (data provider) address in the DLT (hexadecimal)
Range:	n/a
Usage note:	The orig (data provider) address in the DLT (hexadecimal).
See also:	n/a

"pooToPorDelay": "number"

Semantic attribute:	core:pooToPorDelay
Definition:	Maximum acceptable delay between the issuance of the proof of origin (PoO) by the orig and the reception of the proof of reception (PoR) by the orig
Range:	n/a
Usage note:	Maximum acceptable delay between the issuance of the proof of origin (PoO) by the orig and the reception of the proof of reception (PoR) by the orig.
See also:	n/a

“pooToPopDelay”: “number”

Semantic attribute:	core:pooToPopDelay
Definition:	Maximum acceptable delay between the issuance of the proof of origin (PoP) by the orig and the reception of the proof of publication (PoR) by the dest
Range:	
Usage note:	Maximum acceptable delay between the issuance of the proof of origin (PoP) by the orig and the reception of the proof of publication (PoR) by the dest.
See also:	n/a

"pooToSecretDelay": "number"

Semantic attribute:	core:pooToSecretDelay
Definition:	If the dest (data consumer) does not receive the PoP, it could still get the decryption secret from the DLT. This defines the maximum acceptable delay between the issuance of the proof of origin (PoP) by the orig and the publication (block time) of the secret on the blockchain
Range:	n/a
Usage note:	If the dest (data consumer) does not receive the PoP, it could still get the decryption secret from the DLT. This defines the maximum acceptable delay between the issuance of the proof of origin (PoO) by the orig and the publication (block time) of the secret on the blockchain.
See also:	n/a

}   }

}   ],

“datasetInformation”: (a description of types that represent attributes of observations, measurements, fields, etc., in the dataset to describe the information and structure of the raw real data in the datasets)

{

"measurementType":

Semantic attribute:	core:measurementType
Definition:	The data types that represent attributes of observations and measurements in the dataset
Range:	xsd:anyURI
Usage note:	Simple text strings or the use of specific vocabularies collected to support domains For example, like the vocabulary created for wellbeing. For example, <http://www.i3-MARKET.eu/wellbeing_annotations/Sleep_count_micro_awakenings>. Specific types of measurements for a certain domain. Parameter can be put multiple times in the API call.
See also:	See also example for Wellbeing in DataRecords_Annotations_for_Wellbeing_datasets_measurements_02.ttl attached to this page but also in gitlab https://gitlab.com/i3-MARKET/code/data-models/-/blob/master/Version-1/DataRecords_Annotations_for_Wellbeing_datasets_measurements_02.ttl

"measurementChannelType":

Semantic attribute:	core:measurementChannelType
Definition:	The data measurement channel types in the dataset
Range:	xsd>string or xsd>anyURI
Usage note:	Simple text strings or the use of specific vocabularies collected to support domains.
See also:	n/a

"sensorId":

Semantic attribute:	core>sensorID
Definition:	Sensor ID
Range:	xsd>string
Usage note:	ID used to identify the sensors in original datasets source.
See also:	n/a

"deviceId":

Semantic attribute:	core>deviceID
Definition:	Device ID
Range:	xsd>string
Usage note:	ID used to identify the devices in original datasets source.
See also:	n/a

“cppType”:

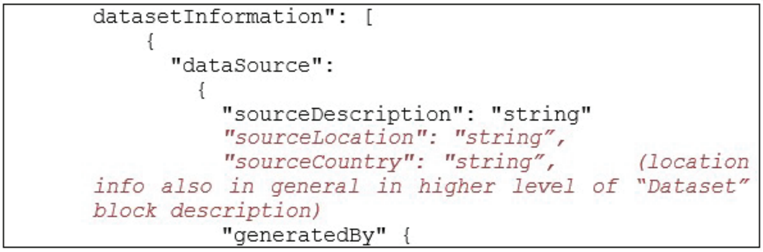
Semantic attribute:	core:cppType
Definition:	The cpp types in the dataset. Derived from AGORA requirements
Range:	xsd>string or xsd>anyURI
Usage note:	Simple text strings or the use of specific vocabularies collected to support domains.
See also:	n/a

“sensorType”: “string”

Semantic attribute:	core:sensorType
Definition:	The cpp types in the dataset. Derived from wellbeing and AGORA requirements
Range:	xsd>string or xsd>anyURI
Usage note:	Simple text strings or the use of specific vocabularies collected to support domains.
See also:	n/a

4.7 Extended Version of Structure for DatasetInformation

The DatasetInformation module may be extended to add more specific description and structure information related to the raw original data that is contained in the assets/datasets delivered by the providers. This way, users can check and have a better understanding of the underlying data model and associated metadata that describe the data that are transferred. The proposed data model description of the source of the data, each observation and data item in the data with their details like quantity type, data type, unit, resolution, range, etc.



```
datasetInformation": [
     {
       "dataSource":
         {
           "sourceDescription": "string"
           "sourceLocation": "string”,
           "sourceCountry": "string”,       (location
info also in general in higher level of “Dataset”
block description)
           "generatedBy" {
            "platform": "string”,
            "device": "string”,
            "sensor": "string”,
            "sensorResolution": "string”,
            "procedure": "string”
          }
        }
      "document": {
             "name":     "string",
             "documentDescription":     "string",
             "comment": "string”,
             "timeResolution":     "string"
      }
      "observation" (dataItem): [
        {
          "observationNname":    "string",
          "observationDdescription":    "string",
          "observedProperty":    "string",
          "dataType":    "string",
          "accuracy":     "string",
          "timeResolution":     "string",
          "measure": {
                        “quantityKind”:
"string",
                        “unit”:       "string",
                        “value”:     "string"
            }
          "range":
            {
              "rangeType": "string",
              "min": "string",
              "max": "string"
            }
           "subObservation" (subDataItem) [
           {
                  "observationNname":    "string",
                  "observationDdescription":
"string",
                  "observedProperty":    "string",
                  "dataType":    "string",
                  "accuracy":     "string",
                  "timeResolution":     "string",
                  "measure": {
                        “quantityKind”:
"string",
                        “unit”:       "string",
                        “value”:     "string",
            }
                  "range":
            {
              "rangeType": "string",
              "min": "string",
              "max": "string",
            }    }
            ]
 } ]
 } ]
```

For the definition and semantic annotations related to quantities and units, we can refer to:

- The Ontology of Units of Measure (OM) 2.0 models concepts and relations are focus on units, quantities, measurements, and dimensions. http://www.ontology-of-units-of-measure.org/page/om-2
- The Quantities, Units, Dimensions and Data Types Ontologies http://www.qudt.org/2.1/catalog/qudt-catalog.html
    - https://qudt.org/
    - https://www.qudt.org/2.1/catalog/qudt-catalog.html
    - https://www.qudt.org/doc/DOC_SCHEMA-DATATYPES.html
    - https://www.qudt.org/doc/DOC_VOCAB-UNITS.html
    - https://www.qudt.org/doc/DOC_VOCAB-QUANTITY-KINDS.html