



Persistent Personal Data Vaults Empowering a Secure and Privacy
Preserving Data Storage, Analysis, Sharing and Monetisation Platform

D1.2

The DataVaults Core Semantic/Data Model

Editor(s)	Simon Dutkowski
Lead Beneficiary	FRAUNHOFER
Status	Final
Version	1.1
Due Date	31/07/2020
Delivery Date	25/08/2020
Dissemination Level	PU

This deliverable has been submitted to the EC and is pending approval



DataVaults is a project co-funded by the European Commission under the Horizon 2020 Programme (H2020-ICT-2019-2) under Grant Agreement No. 871755 and is contributing to the BDV-PPP of the European Commission.

Project	DataVaults – 871755
Work Package	WP1 - DataVaults Data Value Chain Elaboration and Concept Fusion
Deliverable	D1.2 – The DataVaults Core Semantic/Data Model
Editor(s)	FRAUNHOFER – Simon Dutkowski
Contributor(s)	FRAUNHOFER – Kyriakos Stefanidis FRAUNHOFER – Yury Glikman TECNALIA – María Jose Lopez Osa TECNALIA – Valentin Sanchez UBITECH – Giannis Ledakis ANDAMIAN7 – Vincent Keunen MIWENERGIA – Pablo Barrachina MIWENERGIA – Ramon Ruiz OLYMPIAKOS – Giorgos Totomis PIRAEUS – Michalis Bourbos PRATO – Elena Palmisano PRATO – Paolo Boscolo
Reviewer(s)	ATOS –Javier Villazán SUITE5 - Sotiris Koussouris

Abstract	<p>This document is a deliverable of WP1 and defines the data model and the lifecycle of the data within DataVaults. Based on semantic web technology the data model is build out of the demonstrator input, representing the current state of the DataVaults MVP. The document defines the data model using the resource description framework and identifies related ontologies, concepts and vocabularies. The core model is specified as profile of the general data catalogue vocabulary (DCAT). This document is a living document and will be improved during the first half of the project to support the development of DataVaults.</p>
Disclaimer	<p>The information and views set out in this publication are those of the author(s) and do not necessarily reflect the official opinion of the European Communities. Neither the European Union institutions and bodies nor any person acting on their behalf may be held responsible for the use which may be made of the information contained therein.</p> <p>© Copyright in this document remains vested with the DataVaults Partners</p>

Executive Summary

This document defines the data model and the lifecycle of the data within DataVaults. The data model is based on semantic web technology, is defined using the resource description framework (RDF) [1] and identifies related ontologies, concepts and vocabularies. The core DataVaults model is specified as profile of the general data catalogue vocabulary (DCAT) [2] which is an RDF vocabulary designed to facilitate interoperability between data catalogues published on the Web. Due to the open nature of RDF, the data model can be extended without breaking the system or APIs under development.

The DataVaults DCAP profile reuses the main DCAT classes (Catalog, Dataset, DataService, Distribution and CatalogRecord) and extends them in order to facilitate the storage of personal data. Whenever necessary, existing ontologies are adopted that describe domain specific properties.

The data model contains six main sections. The first two are the basic profile data and location data. Then, there is a section that is used for access control followed by three domain specific sections, namely HealthCare data, Social and Activity data, Smarthome and Energy data. The domains have been chosen based on the input of the project's demonstrators and the chosen properties follow the demonstrator's requirements while also adding more general properties that are needed for all scenarios.

Apart from the data model, this deliverable describes the data lifecycle as part of DataVaults. The lifecycle refers to both the management of the personal data as well as the workflows for data analytics and compensation of data assets. The main input for the data lifecycle is the high-level usage scenarios that have been defined in T1.5.

This document is a living document and will be improved during the first half of the project to support the development of DataVaults. In M18, the result will be transferred in deliverable D1.4 containing the final data model.

Table of Contents

1	Introduction	8
1.1	Document structure	8
2	DataVaults Main (Meta)Data Model.....	9
2.1	Profiling DCAT	9
2.1.1	Catalogue.....	12
2.1.2	Datasets.....	12
2.1.3	Data Services	13
2.1.4	Catalogue Records.....	14
2.2	Profile Data	14
2.2.1	Person.....	15
2.2.2	Contact Point.....	15
2.2.3	Additional Information.....	15
2.3	Access Control	16
2.3.1	Open Digital Rights Language.....	17
2.3.2	Data Privacy Vocabulary.....	17
2.4	Location Data.....	17
2.4.1	Downloadable File(s).....	18
2.4.2	Service or API.....	19
2.4.3	Geolocation for Individual.....	20
2.5	Healthcare Data	20
2.5.1	Structured User Data for the Health Domain	21
2.5.2	Time Series	22
2.5.3	Electronic Health Records	23
2.6	Social Data / Activity Data	24
2.6.1	Membership to online communities (social media)	24
2.6.2	Collections of Posts	24
2.7	Smarthome / Energy Data	25
3	Lifecycles	27
4	Conclusions and Future Work.....	31
5	References	31

List of Figures

Figure 1: DCAT Class Summary.....	10
Figure 2: DataVaults DCAT Profile Overview	11
Figure 3 Datasets and Distributions	12
Figure 4: Dataset and Data Service	14
Figure 5: ODRL Information Model	16
Figure 6: Lifecycle for Data Management in DataVaults	27
Figure 7: Lifecycle steps regarding searching and data analytics in DataVaults.....	28
Figure 8: Lifecycle steps regarding compensation in DataVaults.....	29

List of Tables

Table 1: Common namespaces used in DataVaults	11
Table 2: Properties of the Catalogue class	12
Table 3: Properties of the Dataset class.....	13
Table 4: Properties of the Distribution class	13
Table 5: Properties of the Data Service class	14
Table 6: Properties of the Catalogue Record class.....	14
Table 7: Properties of the Person class	15
Table 8: Properties of the Contact Point class	15
Table 9: Properties of the Address class	15
Table 10: Additional Properties for an Individual Profile.....	16
Table 11: Properties of the Data Privacy Vocabulary	17
Table 12: Properties of datasets containing location data	19
Table 13: Properties of distributions containing location Data	19
Table 14: Data Service for Location Data	19
Table 15: Spatial Property in Catalog and Dataset.....	20
Table 16: Properties of the Location class	20
Table 17: Health related Properties of foaf:Person	21
Table 18: Properties of dvdm:Pulse class.....	21
Table 19: Dataset properties for time series health data	22
Table 20: Distribution properties for time series health data	22
Table 21: DataService properties for time series health data	22
Table 22: Dataset properties for health data.....	23
Table 23: Distribution properties for health data	23
Table 24: Online community participation	24
Table 25: Dataset properties for social media posts.....	25
Table 26: Distribution properties for social media posts.....	25
Table 27: Social media post structure	25
Table 28: Properties of Smart Home Energy Dataset	26
Table 29: Properties of Smart Home Energy Distribution.....	26
Table 30: Example of a Smart Home Energy Dataset.....	26

Terms and Abbreviations

AMI	Atomic Medical item
API	Application Programming Interface
BDVA	Big Data Value Association
DCAT	Data Catalog Vocabulary
DCAT-AP	Data Catalog Vocabulary Application Profile
ESCO	European Skills, Competences, qualifications and Occupations
FOAF	Friend of a Friend
GPS	Global Positioning System
HVAC	Ventilation and Air Conditioning
MVP	Minimum Valuable Platform
ODRL	Open Digital Rights Language
OEMA	Ontology for Energy Management Applications
OGC	Open Geospatial Consortium
PDV	Private Data Vocabulary
RDF	Resource Description Framework
SPARQL	SPARQL Protocol and RDF Query Language
URI	Uniform Resource Identifier
URL	Unified Resource Locator
W3C	World Wide Web Consortium

1 INTRODUCTION

DataVaults aims to deliver a solution for collecting, storing, managing, sharing, analysing and monetizing personal data and its derivatives. Furthermore, data queries, analysis and experimentation in DataVaults will allow the linking and merging of data from various sources and the combination of those with personal data, based on the DataVaults core data model. These activities, which rely on the semantic annotation of data and the curation of those to make them linkable, will allow to raise the economic value of both personal and of other kind of data, as more detailed and interesting insights will be generated.

The DataVaults data model needs to describe a holistic Personal Data Value Chain addressing all the aspects of personal data management. This includes data protection, security, GDPR compliance, IPR management (compensation schemes etc.) and representation of the main value flows in data marketplaces. It should be based on existing standards.

This document presents the first iteration of this model allowing description of DataVaults datasets to make them discoverable and include the domain specific information from the DataVaults demonstrators. The data model is defined using the resource description framework (RDF) [1] as a profile of the Data Catalog Vocabulary (DCAT) [2]. It is an RDF vocabulary designed to facilitate interoperability between data catalogues published on the Web. Defining a DataVaults specific profile of DCAT allows extension with elements required for DataVaults at a later point in time.

1.1 DOCUMENT STRUCTURE

Section 2 starts with a description of the actual data model based on DCAT. All used ontologies, concepts and vocabularies are listed and briefly explained. In subsections each domain from the demonstrators are covered in more depth. In the following Section 3 a detailed view on lifecycles of the data and metadata is outlined. also based on the high-level usage scenarios from task T1.5.

Section 4 concludes the document and gives an outlook for future work. Additionally, it contains a short explanation what extensibility in the RDF domain means.

2 DATAVAULTS MAIN (META)DATA MODEL

The Big Data Value Association (BDVA) reference model [3] differentiates between the following six types of data:

- Structured data
- Time series data
- Geospatial data
- Media, Image, Video and Audio data
- Text data, including Natural Language Processing data and Genomics representations
- Graph data, Network/Web data and Metadata

The main task of the DataVaults data model is provide a clear distinction between data and metadata. The data model will not specify how data has to be deposited, since this would restrict options for domain specific analysis or searching operations. For example, full-text search requires comprehensive indexing, so a dedicated search engine will be the database of choice. In contrast, geo information may require geometric calculations for two-dimensional search, which is facilitated by a different kind of storage engine. On the other hand, a lot of information (such as name, address, nationality) is related to the individual and is described as part of the structured metadata.

The Data Model for DataVaults described in this document is based on the specification of the Data Catalog Vocabulary (DCAT) [2], developed under the responsibility of the Government Linked Data Working Group at W3C. DCAT is an RDF vocabulary designed to facilitate interoperability between data catalogues published on the Web. Additional classes and properties from other well-known vocabularies are re-used whenever possible.

2.1 PROFILING DCAT

*“A **DCAT profile** is a specification for a data catalog that adds additional constraints to DCAT”* [2]. A profile can be seen as a named set of constraints based on DCAT and specific to the related domain. E.g. DCAT-AP is a profile for data portals in Europe. Additional constraints mean [2]:

- Cardinality constraints, including a minimum set of required metadata fields
- Sub-classes and sub-properties of the standard DCAT classes and properties
- Classes and properties for additional metadata fields not covered in DCAT vocabulary specification
- Controlled vocabularies or URI sets as acceptable values for properties
- Requirements for specific access mechanisms (RDF syntaxes, protocols) to the catalogue’s RDF description

The DataVaults DCAT Profile will try to honour these rules, so in the end any DataVaults Catalogue will also be a DCAT compliant catalog. A summary of the DCAT classes is shown in Figure 1.

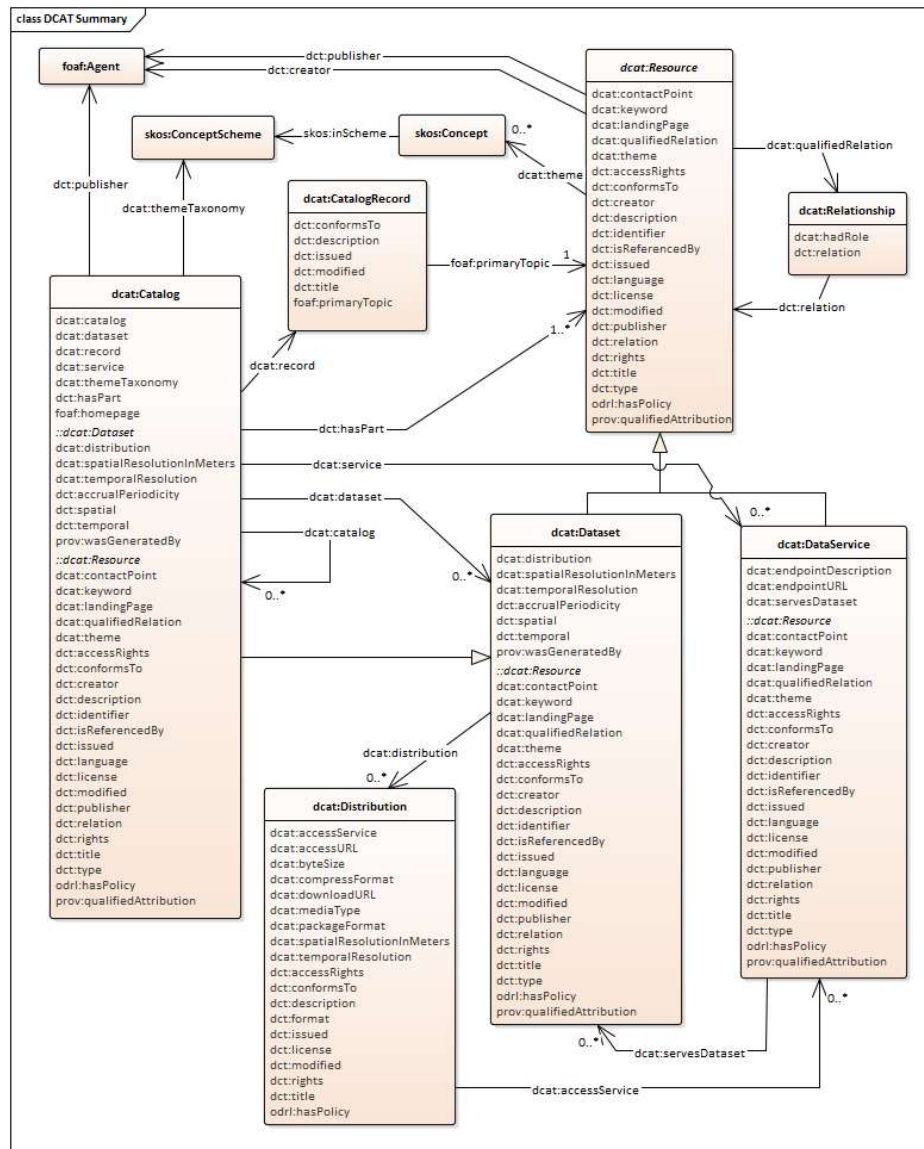


Figure 1: DCAT Class Summary

DataVaults will reuse the Catalog, Dataset, DataService, Distribution and CatalogRecord classes. Figure 2 explains the relationship between them. The basic idea for using DCAT is that in the context of DataVaults an individual can be seen as a collection of datasets, i.e. a catalogue. The metadata for the individual is attached to the catalogue via the “*publisher*” property, information and contact information via “*contactPoint*”.

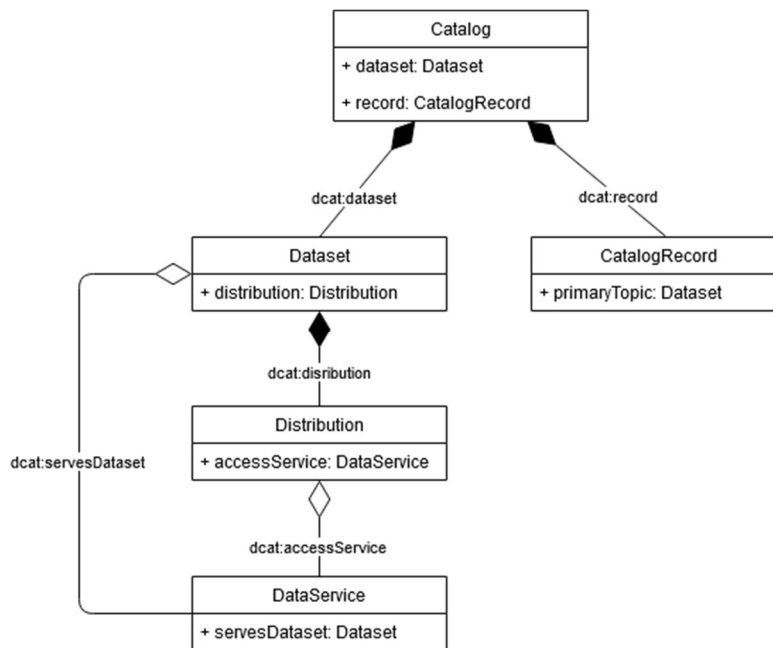


Figure 2: DataVaults DCAT Profile Overview

The DCAT Profile for DataVaults reuses terms from various existing specifications. Classes and properties specified in the next sections have been taken from the following namespaces:

Prefix	URI
adms	http://www.w3.org/ns/adms#
dcat	http://www.w3.org/ns/dcat#
dct	http://purl.org/dc/terms/
dpv	http://www.w3.org/ns/dpv#
dvdm	http://datavaults.eu/ns/dm#
esco	http://data.europa.eu/esco/model#
foaf	http://xmlns.com/foaf/0.1/
locn	http://www.w3.org/ns/locn#
odrl	http://www.w3.org/ns/odrl/2/
rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#
rdfs	http://www.w3.org/2000/01/rdf-schema#
skos	http://www.w3.org/2004/02/skos/core#
time	http://www.w3.org/2006/time#
vcard	http://www.w3.org/2006/vcard/ns#
xsd	http://www.w3.org/2001/XMLSchema#
infrastructure	http://www.purl.org/oema/infrastructure/
enaeg	http://www.purl.org/oema/eaeg/

Table 1: Common namespaces used in DataVaults

In this document the main classes of the DataVaults DCAT Profile will be used without their namespace prefixes, e.g. DataService is equivalent to dcat:DataService. The following sections contain tables specifying the allowed fields for each of the aforementioned classes. The URI thereby describes the property, i.e. the predicate in RDF terms. The permitted values, along with their formatting if applicable, are listed in the Range column. Finally, the cardinality

indicates how often a property may be used. For example, there can only be one publisher of a catalogue, but a catalogue can have multiple datasets.

2.1.1 Catalogue

A catalogue contains a list of datasets alongside catalogue records which store meta-metadata for each dataset. Also, contact information about the publishing individual of a catalogue is attached (publisher). This included addresses (contactPoint), e.g. one for home and one for work. Optionally, control access permissions and rules on a person's global level can be attached as an ODRL [4] description (hasPolicy). Alternatively, access policies can also be attached to individual datasets, which is covered in the next section.

Property	URI	Range	Card.
Publisher	dct:publisher	foaf:Person	1..1
Contact Point	dcat:contactPoint	vcard:Individual	1..n
Dataset	dcat:dataset	dcat:Dataset	0..n
Record	dcat:record	dcat:CatalogReord	0..n
Access Policy	odrl:hasPolicy	odrl:Policy	0..1
Location	dcat:spatial	dct:Location	0..1

Table 2: Properties of the Catalogue class

2.1.2 Datasets

A dataset is used to describe data that is provided as one or more downloadable files. The data itself is stored separately. It is insignificant whether the files are precomputed or generated on demand. Each dataset has one or more distributions. A distribution describes a concrete representation of the data. This relationship is illustrated in Figure 3.

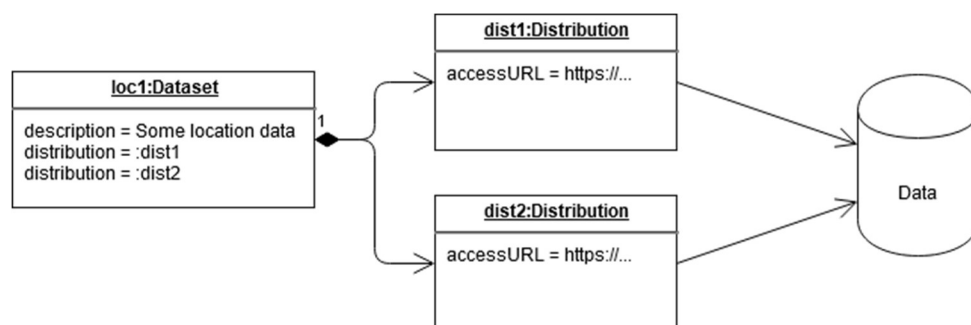


Figure 3 Datasets and Distributions

The permitted properties of both datasets and distributions are listed in Table 3 and Table 4 respectively. Note that multiple titles and descriptions may only be assigned for different languages. Distributions should have either an access URL or service, or both.

Property	URI	Range	Card.
Title	dct:title	rdfs:Literal	1..n
Description	dct:description	rdfs:Literal	1..n
Type	dct:type	dvd:DatasetType	0..1
Distribution	dcat:distribution	dcat:Dataset	0..n
Conforms to	dct:conformsTo	dct:Standard	0..n

Issued	dct:issued	rdfs:Literal typed as xsd:date or xsd:dateTime	0..1
Modified	dct:modified	rdfs:Literal typed as xsd:date or xsd:dateTime	0..1
Policy	odrl:hasPolicy	odrl:Policy	0..1
Keyword	dcat:keyword	rdfs:Literal	0..n

Table 3: Properties of the Dataset class

Property	URI	Range	Card.
Title	dct:title	rdfs:Literal	1..n
Description	dct:description	rdfs:Literal	0..n
Format	dct:format	rdfs:Resource	0..1
Media Type	dcat:mediaType	rdfs:Resource	0..1
Access URL	dcat:accessURL	rdfs:Resource	0..1
Package Format	dcat:packageFormat	dct:MediaType	0..1
Compress Format	dcat:compressFormat	dct:MediaType	0..1
Conforms to	dct:conformsTo	dct:Standard	0..1
Issued	dct:issued	rdfs:Literal typed as xsd:date or xsd:dateTime	0..1
Modified	dct:modified	rdfs:Literal typed as xsd:date or xsd:dateTime	0..1
Access Service	dcat:accessService	dcat:DataService	0..1

Table 4: Properties of the Distribution class

2.1.3 Data Services

According to DCAT, a Data Service may be understood as a “[...] collection of operations that provides access to one or more datasets or data processing functions.”¹ The relationship of data services to datasets and distributions in the DataVault context are shown in Figure 4.

¹ https://www.w3.org/TR/vocab-dcat-2/#Class:Data_Service

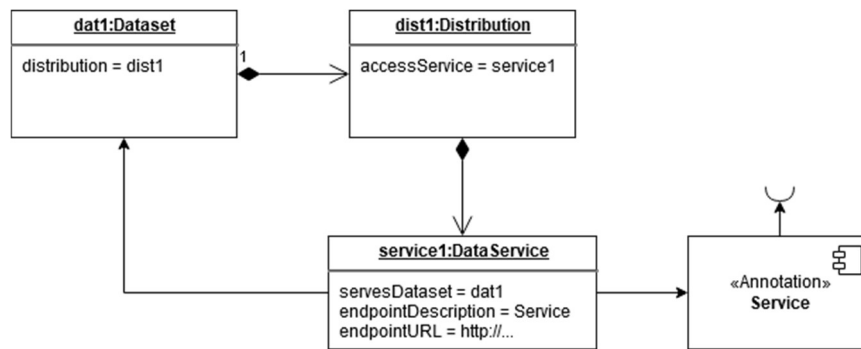


Figure 4: Dataset and Data Service

The permitted properties for defining a data service are listed in Table 5.

Property	URI	Range	Card.
Dataset	dcats:servesDataset	dcats:Dataset	0..n
Endpoint Description	dcats:endpointDescription	rdfs:Resource	0..1
Endpoint URL	dcats:endpointURL	rdfs:Resource	1..n
Title	dct:title	rdfs:Literal	1..n
Conforms to	dct:conformsTo	dct:Standard	0..n
Serves Dataset	dcats:servesDataset	dcats:Dataset	0..n

Table 5: Properties of the Data Service class

2.1.4 Catalogue Records

The record element contains meta-metadata to track the lifecycle of each dataset or data service resource entry. For now, only a minimum set of properties are defined, allowing the tracing of the actual lifetime of a dataset within the catalogue of the individual. These properties are shown in Table 6.

Property	URI	Range	Card.
Issued	dct:issued	rdfs:Literal typed as xsd:dateTime	0..1
Modified	dct:modified	rdfs:Literal typed as xsd:dateTime	1..1
Primary Topic	foaf:primaryTopic	dcats:Dataset or dcats:DataService	1..1

Table 6: Properties of the Catalogue Record class

2.2 PROFILE DATA

A catalogue represents the top-level class assigned to an individual. The metadata of a catalogue as defined in the core DCAT vocabulary contains properties that allows description of profiles of individuals.

There are two ontologies commonly used for describing an individual's master file data. They have many overlapping properties. In DataVaults the focus has been laid on the FOAF ontology, which is complemented by the vCard [5] ontology. All FOAF [6] related fields are added via the dct:publisher property, encoded as a foaf:Person element. All vCard related values can be found as a vcard:Individual attached via the dcats:contactPoint property.

Information that currently cannot be assigned to one of the two elements will be specified in a separate table and structured in further steps.

2.2.1 Person

In the DataVaults context a person is defined by the properties listed in Table 7.

Property	URI	Range	Card.
First Name	foaf:firstName	rdfs:Literal	1..1
Last Name	foaf:lastName	rdfs:Literal	1..1
Birthday	foaf:birthday	rdfs:Literal typed as xsd:date	0..1
Place of birth	dvdn:placeOfBirth	skos:Concept	0..1
Gender	foaf:gender	skos:Concept	0..1
Nationality	dvdn:nationality	skos:Concept	0..1

Table 7: Properties of the Person class

2.2.2 Contact Point

A contact point contains all personal information (email addresses, phone numbers and social media accounts). The relevant properties of both contact points and addresses are shown in Table 8 and Table 9 respectively.

Property	URI	Range	Card.
Address	vcard:hasAddress	vcard:Address	0..n
Email	vcard:hasEmail	Email URI	0..n
Phone Number	vcard:hasTelephone	Tel URI	0..n

Table 8: Properties of the Contact Point class

Property	URI	Range	Card.
Type	rdf:type	vcard:Home or vcard:Work	1..1
Street	vcard:street-address	rdfs:Literal	0..1
Postal Code	vcard:postal-code	rdfs:Literal	0..1
Country	vcard:country-name	rdfs:Literal	0..1

Table 9: Properties of the Address class

2.2.3 Additional Information

Several identifiers, memberships, and other information that is requested or provided from the demonstrators is listed here as part of the catalogue. This assignment is temporary and will be redefined in the future. In addition, it needs to be examined if ontologies or vocabularies with semantically more appropriate definitions exist.

Property	URI	Range	Card.
----------	-----	-------	-------

National Insurance Number	dvdm:nationalInsuranceNr	rdfs:Literal	0..1
Member Number	dvdm:memberNumber	rdfs:Literal	0..1
Social Security Number	dvdm:socialSecurityNumber	rdfs:Literal	0..1
Occupation	dvdm:occupation	esco:Occupation	0..1
Qualification	dvdm:qualification	esco:Qualification	0..1
Certificate	dvdm:certificate	foaf:Document	0..n
Transportation Means	dvdm:transportation	skos:Concept	
Cultural Interest	dvdm:cutluralInterest	rdfs:Literal	0..n
Disabilities	dvdm:disabilities	rdfs:Literal	0..n
Civil Status	dvdm:civilStatus	skos:Concept	0..1

Table 10: Additional Properties for an Individual Profile

2.3 ACCESS CONTROL

The scheme for representing access control policies in DataVaults is based on the hasPolicy property, which links the Resource class to the ODRL Information Model [4]. The general model is shown in Figure 5.

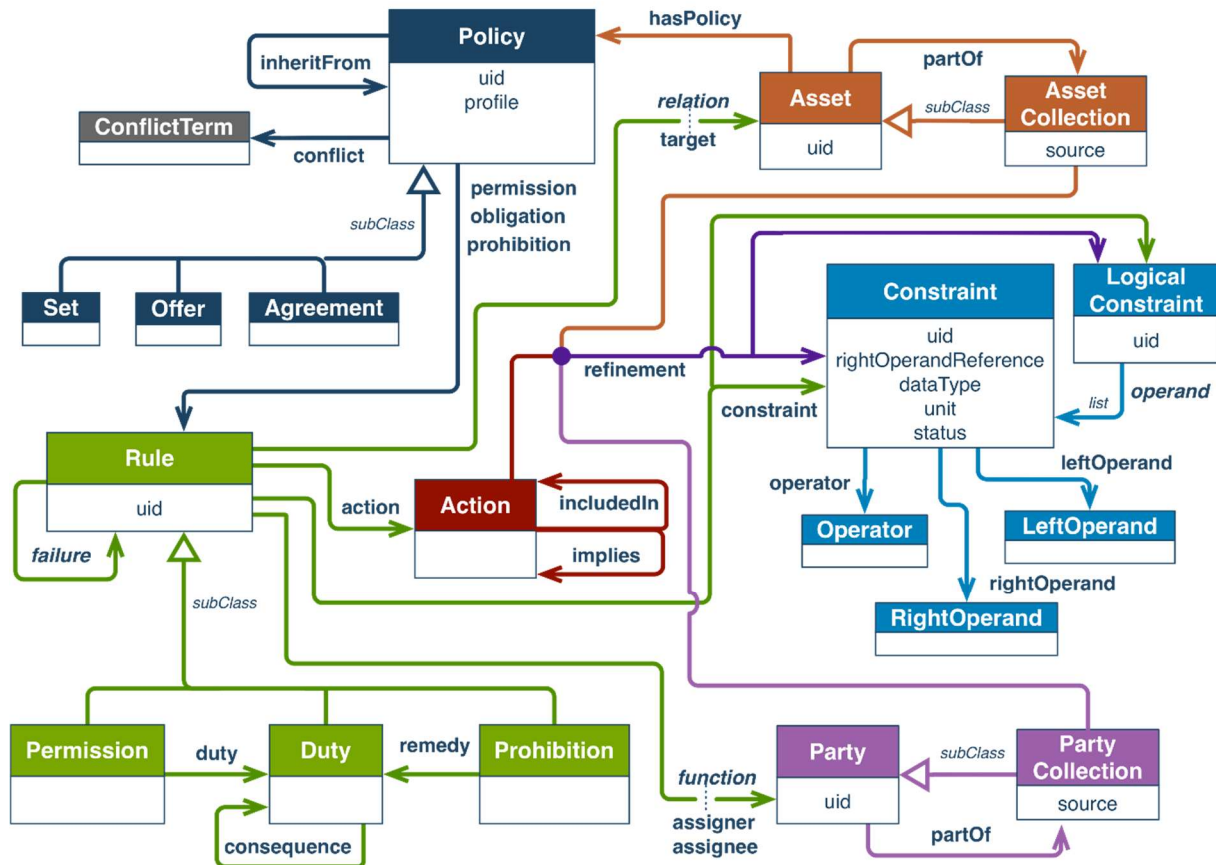


Figure 5: ODRL Information Model

The policy will have some rules in order to set permissions, prohibitions, and obligations associated to the access to a specific resource.

As the Resource class in the diagram is the subject of the Rule class, it can be considered as a DataVaults DCAT resource. This is the data gathered about an individual by the DataVaults ecosystem.

The rules associated to a Policy will be derived by the individuals and created through the policy access editor. This allows introduction of conditions to be fulfilled by seekers for accessing the data. The information will be provided by those seekers that are interested in the data through the DataVaults platform.

The Constraint class will include the leftOperand and rightOperand properties, among others. These types are taken from the Data Privacy Vocabulary (DPV) [7]. The values will be validated against the information provided by the companies when they contact DataVaults.

2.3.1 Open Digital Rights Language

The ODRL allows specifying policies which can then be anchored by the odrl:hasPolicy property in either a dcat:Catalog, dcat:Dataset, or dcat:DataService. Basically, policies are made up of various constraints. These constraints can be defined using operands, similar to mathematical equations. Each constraint is made up of at least a right-hand and a left-hand operand.

The Policies to be implemented as part as the DataVaults ecosystem will follow the general ODRL information model and it will be the basic model used for adapting it to the DataVaults requirements. The further tasks related to the access control will work on the definition of the conditions that will take part of those policies. As a preliminary analysis, the constraint class should define the elements to be checked for allowing access to the data sharing.

2.3.2 Data Privacy Vocabulary

DataVaults includes the Data Privacy Vocabulary as an extension of the constraint related properties in the policy model. These properties are related to the values the seekers should provide to the DataVaults platform when requesting data. The hasProcessing property connects an instance of processed data with proof of consent. Likewise, the hasPurpose property connects a reason for data handling with proof of consent. These properties must be part of the policies to be implemented. As this is a very preliminary analysis of the model, the number of properties can vary. The properties are shown in Table 11.

Property	URI	Range	Cardinality
Has Processing	dvp:hasProcessing	dvp:Processing	1..n
Has Purpose	dvp:hasPurpose	dvp:Purpose	1..n

Table 11: Properties of the Data Privacy Vocabulary

2.4 LOCATION DATA

Location data can be seen as one of the major data types. It is not surprising that three of the five DataVaults demonstrators have interest in evaluating location information. The demonstrators have requirements on providing or using location information in their scenarios as defined in D1.1 [8]:

One demonstrator provides

- Live location of municipal buses (Prato)

Demonstrators need

- User position (live)

Location data is usually of type Geospatial and temporal (BDVA) [3]. Whereas the actual data should be stored separately in a way that allows easy and better analytics as part of the value chain, only the metadata describing the data is specified as part of the DataVaults data model.

There are two main use cases how data is requested from a potential data seeker, and how the data can be provided by the DataVaults platform. One is via a “download link” where the data is provided as a file containing the requested location information. The second way involves exposing an API or service emitting the real time data dynamically. There, consumers can access or collect the location information continuously.

We can distinguish between these two use cases where the data is either available as:

- Simple downloadable file(s)
- Live or real time data via an API or services

DataVaults will primarily focus on the first use case, but the DataVaults data model will also allow the description of datasets that can only be accessed through an API service (second use case). The current DataVaults demonstrators are primarily interested in location data (e.g. GPS tracking), which is expected as real-time data. However, the DataVaults MVP will also focus on sharing data as snapshots, even if the data is collected and stored in real time.

Besides the usual properties to describe general datasets available as file download, some of the properties associated with the `dc:Dataset` class aid in describing geo location datasets. Therefore, real time data accessible through a service is described by the `dc:Dataset` in the DataVaults data model.

2.4.1 Downloadable File(s)

In case data can be provided as a downloadable files the dataset can be described as `dc:Dataset` with corresponding distributions. Assuming the DataVaults platform will store location data as MVP, these datasets containing location data will typically have only one distribution, which describes the original data format and media type. However, this does not mean that the data can also be accessed and provided in different representation or package formats. If there are requirements to restrict the provisioning to a limited set of file formats the dataset can also contain several distributions, each describing one format. Sometimes it could also be useful to split the data in pieces e.g. for different time spans. In these cases, a dataset contains several distributions, each pointing to one slice of the data. The metadata properties shown in Table 12 are relevant for datasets containing location data. Table 13 lists the properties of the corresponding distribution.

Property	URI	Range	Card.
Spatial resolution in meters	<code>dc:spatialResolutionInMeters</code>	<code>rdfs:Literal</code> typed as <code>xsd:decimal</code>	0..1

Temporal resolution	dcats:temporalResolution	rdfs:Literal typed as xsd:duration	0..1
----------------------------	--------------------------	------------------------------------	------

Table 12: Properties of datasets containing location data

Property	URI	Range	Card.
Spatial resolution in meters	dcats:spatialResolutionInMeters	rdfs:Literal as xsd:decimal	0..1
Temporal resolution	dcats:temporalResolution	rdfs:Literal as xsd:duration	0..1
Frequency	dct:accrualPeriodicity	dct:Frequency	0..1

Table 13: Properties of distributions containing location Data

Example:

```
@prefix dcat: <http://www.w3.org/ns/dcat#> .
@prefix dct: <http://purl.org/dc/terms/> .
@prefix dvdm: <http://datavaults.eu/ns/dm#> .
@prefix dv: <http://datavaults.eu/> .

dv:individual1
  a dcat:Catalog ;
  dct:title "Individual 1 Catalogue of Datasets"@en
  dcat:dataset dv:individual1-location-dataset .

dv:individual1-location-dataset
  a dcat:Dataset
  dcat:spatialResolutionInMeters "10"^^xsd:decimal ;
  dcat:temporal
  dcat:distribution dv:dist1 .

dv:dist1
  a dcat:Distribution ;
  dcat:temporalResolution "PT1H"^^xsd:duration ;
  dct:accrualPeriodicity "<>" ;
  dcat:accessURL "<>"
```

2.4.2 Service or API

When the location data is provided via a service or an API the distribution should point to a `dcat:DataService` object containing more information about how the service or API has to be used. Note that a data service may also be shared between several datasets. In this case the `DataService` should clearly indicate which datasets it serves through the `dcat:servesDataset` property. Table 14 lists the properties describing a `dcat:DataService`.

Property	URI	Range	Cardinality
Endpoint description	dcats:endpointDescription	rdfs:Resource	1..1
endpointURL	dcats:endpointURL	rdfs:Resource	1..1

Table 14: Data Service for Location Data

The endpoint description may be expressed in several ways, for example as an OpenAPI (Swagger) description [OpenAPI], an OGC [9] GetCapabilities response, or a SPARQL [10] Service Description. If a formal representation is not possible depending of the service or API provided a text-based description may also be chosen.

2.4.3 Geolocation for Individual

In case a single static location needs to be stored for an individual or a dataset of an individual, both a `dcatalog:Catalog` and a `dcatalog:Dataset` have the property `dcatalog:spatial` which allows the association of a location. The property is of range `dct:Location`, as shown in Table 15.

Property	URI	Range	Card.
Spatial	<code>dcatalog:spatial</code>	<code>dct:Location</code>	0..1

Table 15: Spatial Property in Catalog and Dataset

A location must contain at one of the following three, mutually exclusive properties: `dcatalog:bbox`, `dcatalog:centroid` or `locn:geometry`. Geometry can be used in cases where the location needs to be described as one or more complex shapes. The property `dct:temporal` is optional. The relevant properties are listed in Table 16.

Property	URI	Range	Card.
Bounding box	<code>dcatalog:bbox</code>	<code>rdfs:Literal</code>	0..1
Centroid	<code>dcatalog:centroid</code>	<code>rdfs:Literal</code>	0..1
Geometry	<code>locn:geometry</code>	<code>rdfs:Literal</code>	0..1
Time	<code>dct:temporal</code>	<code>dct:PeriodOfTime</code>	0..1

Table 16: Properties of the Location class

Example:

```
@prefix dcat: <http://www.w3.org/ns/dcat#> .
@prefix dct: <http://purl.org/dc/terms/> .
@prefix dv: <http://datavaults.eu/> .

<dv:individual1>
  a dcat:Catalog ;
  dct:spatial [ a dct:Location ;
    dcat:bbox "POLYGON((-180 90,180 90,180 -90,-180 -90,-180
90))"^^gsp:wktLiteral ;
    dcat:temporal [ a dct:PeriodOfTime ;
      dcat:startDate "2020-06-29T12:00:00"^^xsd:dateTime ;
      dcat:endDate "2020-07-05T16:00:00"^^xsd:dateTime ] ;
    dcat:spatialResolutionInMeters "10"^^xsd:decimal ;
```

2.5 HEALTHCARE DATA

The DataVaults data model for health information is based on the Andaman7 demonstrator, and therefore tries to follow the Andaman7 A7 protocol as closely as possible. *“A7 defines both an exchange protocol and a very simple, basic, data structure. It is model dependent (medical domain), but also as generic as possible.”* [11]

Following this approach, the concepts of A7 and the DataVaults DCAT Profile need to be mapped. The basic concept of the A7 is an AMI (Atomic Medical Item), which is so fine grained that information like first name of a person or age can be stored. Basically, an AMI is a key-value pair which is annotated with a source and a timestamp. Additionally, the AMI has qualifiers, which themselves are AMIs. Many AMIs can be mapped directly to DataVaults core profile information (section 2.2). In these cases, annotations are either dropped or implicitly define the subject of the corresponding RDF triple. Related qualifiers are mapped to the corresponding object type of the RDF or dropped if no mapping can be found. If the relevance

of source, timestamp and qualifiers is high and must be kept, then they need to be mirrored to a corresponding dataset. A full list of AMIs can be found in the AMI dictionary².

Following the demo scenario of Andaman7 and with regards to the BDVA reference model, the DataVaults data model must be able to handle the following types of data:

- **Structured Data**
Structured data is for User Profile information and can be part of DataVaults data model.
- **Time series**
Usually from wearable devices, e.g. measuring the heartbeat. For this, the data model defines the required metadata as part of the data model.
- **Structured and Media Data**
This is related to EHRs and doctor consultation reports. They are not stored in the data model. However, the DataVaults data model must be able to include related metadata.

2.5.1 Structured User Data for the Health Domain

Like any other data in A7, structured user data is described as AMIs, usually tagged with “*tag.administrative*”. Most of these AMIs can be mapped to already defined properties of the User Profile described in section 2.2. Table 17 defines all health-related properties as they are defined in the AMI dictionary. Table 18 lists the properties of the `dvdm:pulse` class. These properties are located inside the `foaf:Person` class.

Property	URI	Range	Card.
Blood Group	<code>dvdm:bloodGroup</code>	<code>skos:Concept</code>	0..1
Birth date	<code>foaf:birthday</code>	<code>rdfs:Literal</code> typed as <code>xsd:date</code>	0..1
Birth place	<code>dvdm:placeOfBirth</code>	<code>skos:Concept</code>	0..1
Age of Kid	<code>dvdm:ageOfKid</code>	<code>rdfs:Literal</code> typed as <code>xsd:decimal</code>	0..n
Civil Status	<code>dvdm:civilStatus</code>	<code>skos:Concept</code>	0..1
Organ Donor	<code>dvdm:organDonor</code>	<code>rdfs:Literal</code> typed as <code>xsd:Boolean</code>	0..1
Pulse	<code>dvdm:pulse</code>	<code>dvdm:Pulse</code>	0..n
Gender	<code>foaf:gender</code>	<code>skos:Concept</code>	0..1

Table 17: Health related Properties of `foaf:Person`

Property	URI	Range	Card.
Timestamp	<code>dvdm:bloodGroup</code>	<code>rdfs:Literal</code> typed as <code>xsd:dateTime</code>	0..1
Pulse	<code>rdf:value</code>	<code>rdfs:Literal</code> typed as <code>xsd:decimal</code>	0..1

Table 18: Properties of `dvdm:Pulse` class

² <http://developers.andaman7.com/guide/medical-data/types.html#amis>

2.5.2 Time Series

A time series dataset contains data that is frequently or periodically collected over time. The data model of DataVaults does not prescribe how or where this data is actually stored. Instead the data should be stored in a place where it can efficiently be provided, analysed or otherwise processed. DataVaults data model only describes the metadata of this type of dataset. Time series are potentially also real time data. Therefore, these datasets could also be accessed via a service or an API. The DataVaults MVP will not contain such a functionality and so the Dataset and Distribution classes will be sufficient. Nevertheless, in case the data should also be provided continuously in real time, the access to a corresponding service can be described with the DataService class.

Besides the usual and common properties, there are a few properties that are specifically useful for time series related data. These properties are shown in Table 19 (datasets), Table 20 (distributions), and Table 21 (service descriptions).

Property	URI	Range	Card.
Temporal	dc:temporal	dct:PeriodOfTime	0..1
Accrual Periodicity	dc:accrualPeriodicity	dct:Frequency	0..1
Temporal Resolution	dc:temporalResolution	xsd:duration	0..n

Table 19: Dataset properties for time series health data

Property	URI	Range	Card.
Temporal Resolution	dc:temporalResolution	xsd:duration	0..n

Table 20: Distribution properties for time series health data

Property	URI	Range	Card.
Endpoint description	dc:endpointDescription	rdfs:Resource	1..1
endpointURL	dc:endpointURL	rdfs:Resource	1..1

Table 21: DataService properties for time series health data

The endpoint description may be expressed as an OpenAPI (Swagger) description [OpenAPI], or informally if a formal representation is not possible depending of the service or API provided.

Example:

```
@prefix dcat: <http://www.w3.org/ns/dcat#> .
@prefix dct: <http://purl.org/dc/terms/> .
@prefix dv: <http://datavaults.eu/> .
@prefix dvdata: <http://datavaults.eu/set/data/> .

dv:dataset-joes-wearable1
  a dcat:Dataset ;
  dct:type "heartbeat" ;
  dct:title "Joe`s Heartbeats"@en ;
  dct:description "The heartbeats measured from Joe`s smartwatch"@en ;
  dcat:distribution dv:dist-joes-hb-1, dv:dist-joes-hb-2 ;

dv:dist-joes-hb-1
```

```

a          dcat:Distribution ;
dct:title  "Joe's downloadable heartbeat data"@en
dcat:accessURL <https://datavaults.eu/data/11a/dist1.csv>
dct:format <http://publications.europa.eu/resource/authority/file-type/CVS> ;

dv:dist-joes-hb-2
a          dcat:Distribution ;
dct:title  "Joe's heartbeat real time as real time"@en
dct:accessService dv:service-joes-hb ;

dv:service-joes-hb
a          dcat:DataService ;
dct:title  "Joe's heartbeat real time service"@en
dcat:endpointURL <https:// datavaults.eu/api >

```

2.5.3 Electronic Health Records

EHRs and Consultations in the context of A7 are collections or sets of AMIs. These sets can be stored in DataVaults in different ways. The A7 protocol allows the retrieval of these datasets in different standardized formats and is out of scope for the DataVaults data model. However, the corresponding metadata of these datasets are described with Dataset and Distribution classes. Beyond the common properties for all datasets, like title or format, Table 22 and Table 23 contain the dataset and distribution properties relevant for health-related data respectively.

Property	URI	Range	Card.
Release date	dct:issued	rdfs:Literal typed as xsd:date or xsd:dateTime	1..1
Modification date	dct:modified	rdfs:Literal typed as xsd:date or xsd:dateTime	0..1
Provenance	dct:provenance	dct:ProvenanceState ment	0..n

Table 22: Dataset properties for health data

Property	URI	Range	Card.
Release date	dct:issued	rdfs:Literal typed as xsd:date or xsd:dateTime	1..1
Modification date	dct:modified	rdfs:Literal typed as xsd:date or xsd:dateTime	0..1
Policy	odrl:hasPolicy	odrl:Policy	0..1

Table 23: Distribution properties for health data

Example

```

@prefix dcat: <http://www.w3.org/ns/dcat#> .
@prefix dct:  <http://purl.org/dc/terms/> .
@prefix dv:   <http://datavaults.eu/> .
@prefix dvdata: <http://datavaults.eu/set/data/> .

<dv:dataset-joes-ehr>
a          dcat:Dataset ;
dct:type   "ehr" ;
dct:title  "Joe's EHR"@en
dct:description "The Electronic Health Record of Joe"@en
dcat:distribution <dv:dist-joes-ehr-1>

```

```
<dv:dist-joes-ehr-1>
  a          dcat:Distribution ;
  dct:title   "Joe's EHR in RDF format"@en
  dcat:accessURL <dvdata:joes-ehr>
  dct:format  <http://publications.europa.eu/resource/authority/file-type/RDF> ;
```

2.6 SOCIAL DATA / ACTIVITY DATA

Social and activity data usually contains activities from social platforms like Facebook, Twitter or Instagram. This type of personal data is the most requested data regardless of use case. Almost all DataVaults demonstrators request access to social media presence of an individual.

More specifically, social and activity data can be split into two main categories. First, the membership in online communities, meaning what social media communities an individual is involved in as well as their account and profile data. This also includes friend networks. The second category contains contributions, i.e. posts, by the individuals in each of their online communities.

The above data is expressed using the Semantically Interlinked Online Communities (SIOC) Core Ontology [12], which describes the information that communities have about their structure and contents. This helps find related information and new connections between content items and other community objects. Additional information about individuals and content creators is described using the FOAF Vocabulary [6].

2.6.1 Membership to online communities (social media)

The memberships in online communities is expressed via the `sioc:Community` class. The basic attributes that are stored for each online community participation is the name of the community (Linked.in, Facebook, etc.), account name, profile data and list of friends. The description of the properties can be found in Table 24:.

Property	URI	Range	Card.
Site	<code>dct:hasPart</code>	<code>sioc:Community</code>	0..n
Account Name	<code>foaf:accountName</code>	<code>sioc:User</code>	0..1
Personal Profile	<code>foaf:page</code>	<code>foaf:PersonalProfile</code> <code>Document</code>	0..1
Knows	<code>foaf:knows</code>	<code>foaf:Person</code>	0..n

Table 24: Online community participation

2.6.2 Collections of Posts

The collections of posts from a specific social media community for a specific time frame are expressed as datasets. Each dataset is described by the online community that the posts belong to, the timeframe of the posts and (optionally) a single topic. The structure of the dataset metadata can be found in Table 25:.

Each dataset contains distributions for specific time frames. The structure of the distribution metadata can be found in Table 26:.. Each distribution contains the list of posts. The structure of the data that is recorded for each post can be found in Table 27:.

Property	URI	Range	Card.
----------	-----	-------	-------

Temporal Coverage	dct:temporal	rdfs:Literal typed as xsd:duration	0..1
Site	dct:hasPart	sioc:Site	1..1
Account Name	foaf:accountName	sioc:User	1..1

Table 25: Dataset properties for social media posts

Property	URI	Range	Card.
Temporal Coverage	dct:temporal	rdfs:Literal typed as xsd:duration	0..1
Topic	sioc:topic	skos:Concept	0..n
Site	dct:hasPart	sioc:Site	1..1
Account Name	foaf:accountName	sioc:User	1..1

Table 26: Distribution properties for social media posts

Property	URI	Range	Card.
Title	dc:title	rdfs:Literal	1..1
Created	dcterms:created	rdfs:Literal typed as xsd:date or xsd:dateTime	1..1
Topic	sioc:topic	skos:Concept	0..n
Location	dcat:spatial	dct:Location	
Content	sioc:content	sioc:Post	1..1
Number of Replies	sioc:num_replies	xsd:Integer	1..1
Number of Likes	dvdm:num_likes	xsd:Integer	1..1
Related to	sioc:related_to	sioc:Post	0..n
Has Reply	sioc:has_reply	sioc:Item	0..n

Table 27: Social media post structure

2.7 SMARTHOME / ENERGY DATA

One data type relevant to the DataVaults pilots is that of smart home energy data. Datasets regarding smart home energy data contain measurements from the various apartment devices.

A dataset contains sets of measurements for each device and for different timeframes. Each distribution contained in a dataset is characterized both by the device that it represents and the timeframe that the included measurements took place. Metadata that describe the apartments and devices are also relevant on both dataset and distribution level.

The two main ontologies that will be used to describe the required entities are:

- The OEMA Energy and Equipment ontology that represents energy equipment such as building automation system resources (sensors, actuators/controllers and HVAC systems), industrial equipment (construction and manufacturing equipment), energy generators (EVs, Home Power Plants, etc.), loads (white and brown goods), power storage/energy carriers, among others. The ontology also represents energy equipment features such as devices' power curves and profile or device state.
- The OEMA Infrastructure ontology represents information about infrastructures and buildings. This includes, among others, infrastructure and building types (household,

microgrid, etc.), technical data (material, surface, etc.), spaces data (floors, rooms, etc.), and geometrical data (floor area).

The relevant metadata properties of a smart home energy dataset can be found in the Table 28. The property `dvdm:infrastructure` points to one or more Infrastructure elements from the OEMA infrastructure ontology which is related to the dataset.

Property	URI	Range	Card.
Temporal Coverage	<code>dct:temporal</code>	<code>rdfs:Literal</code> typed as <code>xsd:duration</code>	0..1
Infrastructure	<code>dvdm:infrastructure</code>	<code>infrastructure:Infrastructure</code>	0..n
Consumption Profile	<code>dvdm:consumptionProfile</code>	<code>rdfs:Literal</code>	0..1

Table 28: Properties of Smart Home Energy Dataset

The relevant metadata properties of a distribution contained in the aforementioned dataset can be found in Table 29.

Property	URI	Range	Card.
Temporal Coverage	<code>dct:temporal</code>	<code>rdfs:Literal</code> as <code>xsd:duration</code>	0..1
Frequency	<code>dct:accrualPeriodicity</code>	<code>dct:Frequency</code>	0..1
Device Code	<code>enaq:deviceCode</code>	<code>rdfs:Literal</code>	0..1
Gather Method	<code>dvdm:gatherMethod</code>	<code>rdfs:Literal</code> typed as <code>xsd:string</code>	0..1
Billing Code	<code>dvdm:billingCode</code>	<code>rdfs:Literal</code> typed as <code>xsd:integer</code>	0..1
Vendor Code	<code>enaq:vendorCode</code>	<code>rdfs:Literal</code>	0..1
Winter Power Time Frame	<code>enaq:winterPowerTimeFrame</code>	<code>rdfs:Literal</code> typed as <code>xsd:integer</code>	0..1

Table 29: Properties of Smart Home Energy Distribution

While the metadata describe the dataset/distribution adequately, the structure of the data in each dataset contains only the actual measurements and the timestamp for each measurement.

Property	Data Type	Description
ID	Integer	Autoincrement
Timestamp	DateTime	Timestamp of current measurement
InputActiveMagnitude	Double	Measurement
OutputActiveMagnitude	Double	Measurement
Quadrant1ReactiveMagnitude	Double	Measurement
Quadrant2ReactiveMagnitude	Double	Measurement
Quadrant3ReactiveMagnitude	Double	Measurement
Quadrant4ReactiveMagnitude	Double	Measurement

Table 30: Example of a Smart Home Energy Dataset

3 LIFECYCLES

In addition to the data model described in previous sections, a high-level description of the data lifecycle as part of DataVaults is presented hereafter. This lifecycle is related to the management of the personal data, but also related to workflows for data analytics and compensation of data assets, and will be used to set the main lines of the DataVaults methodology, that will be delivered as part of deliverable D1.3 [13] of the project.

The first part of DataVaults lifecycle is for the data management. This includes the data sharing but also data deletion and revocation. The suggested lifecycle is depicted in Figure 6 below.

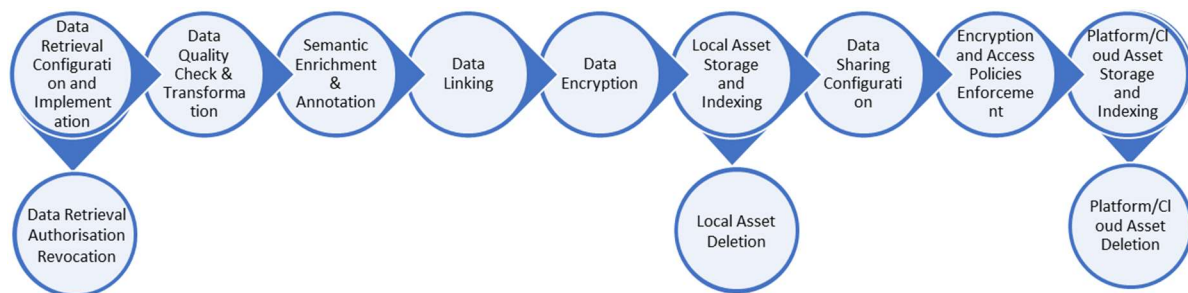


Figure 6: Lifecycle for Data Management in DataVaults

The steps of the lifecycle are the following:

I. Data Retrieval Configuration and Implementation

This operation is responsible for configuring the access to the personal data of the Individual and the actual implementation of the connection of the Personal DataVaults App with the data source.

II. Data Quality Check & Transformation

Data Quality Check process is responsible to discover inconsistencies and other anomalies in the data through validation rules and various statistical checks, while Data Transformation aims to ensure compliance with the available DataVaults schemas through data filtering and data cleansing techniques.

III. Semantic Enrichment & Annotation

Semantic Enrichment and Annotation is responsible for the attachment of additional, machine-readable information to the retrieved data.

Data Linking

In addition to the semantic enrichment, data linking is also used to upgrade the dataset discoverability.

IV. Data Encryption

Encryption of Data prior to share to the DataVaults Cloud Platform

V. Local Asset Storage and Indexing

Indexing and persisting the encrypted personal data in the form of data assets at the side of the Individual.

VI. Data Sharing Configuration

When data are to be shared through the platform, details about the data sharing should be provided.

VII. Encryption and Access Policies Enforcement

Make data available for searchable encryption while respecting the data sharing configuration.

VIII. Platform/Cloud Asset Storage and Indexing

Indexing and persisting the encrypted personal data in the form of data assets at the side of the DataVaults Platform.

IX. Data Asset Retrieval Authorisation Revocation

Revocation of authorization previously provided to DataVaults for collecting personal data from data sources. Such personal data are assumed to have been collected by DataVaults and stored at the user's side but not yet configured for sharing, shared and uploaded to DataVaults Cloud.

X. Local Asset Deletion

Local Asset Deletion concerns the process of the complete removal of a data asset that has been collected by DataVaults.

XI. Platform/Cloud Asset Deletion

Asset Deletion step is related to the complete removal of a data asset that has been uploaded to the DataVaults platform, if the data asset is not part of any active contract.

In addition to these core steps, further workflows for the analysis of the data and for the payment of data have been considered.

Regarding the analysis of data, lifecycle covers the required processing of the data when it has been already uploaded in the DataVaults platform, as depicted in Figure 7.



Figure 7: Lifecycle steps regarding searching and data analytics in DataVaults

The corresponding steps are:

- Asset Searching

We consider this first step for the processing required for searching over the encrypted DataVaults assets shared by Individuals, and retrieve any information that is eligible for sharing.

- Asset Export

Similarly, to searching, the usage of data and the accompanying policies is required, and in addition the usage of the data of contracts for allowing the export of datasets.

- Cloud Based Data Analysis

Data Analysis step reflects the application of data analytics algorithms on selected data assets.

- Data Visualization

Data Visualization is not affecting the data; however, it allows the graphical representation of the data.

A part of the lifecycle that is not directly related to the data, but it is related to the access of users in data is the creation of smart contracts and the compensation management.



Figure 8: Lifecycle steps regarding compensation in DataVaults

The steps of this workflow are the following:

- Private Contract Creation

The creation of a contract that encompass the sharing configurations made by the Individual regarding the various sharing aspects.

- Public Seeker – Platform Contract Creation

When a user decides to acquire a data asset that is available through the DataVaults cloud, a contract is issued between DataVaults and the user.

- Compensation Management

The subsequent compensation of the user providing the data with cryptocurrency, as defined by the pricing and sharing terms, in order to register the financial transaction.

- Contract Issuing

A contract between the user providing the data and the DataVaults Cloud Platform shall be created upon successful completion of any data asset transaction of data from the cloud to a user seeking data. This contract entails the terms defined by the data asset sharing configuration and is validated after the data asset has been transferred to the Data Seeker and the related cryptocurrency transaction has been completed successfully

More details regarding how these steps are envisioned as part of DataVaults and the use cases is provided in deliverable D1.3 [13], where the phases and operations included in these phases are described.

4 CONCLUSIONS AND FUTURE WORK

The current state of the DataVaults data model is a first draft and needs to be fine-tuned as planned, during the course of WP1 and the advancement of the MVP and the overall definition of the project's architecture. Furthermore, it is expected that during the project development, requirements related to the data model will change, new requirements will be revealed or even some will be removed, and therefore, the document at hand is considered as the first version of a "living" document that will be used to guide the initial technical design and development phases of the project, and will accommodate the changes that are propagated back from these processes.

With this in mind, the DataVaults data model in its current version has been articulated in such a manner so to be easily extended without breaking the system or APIs under development due to the open nature of RDF. RDF, which defines data as a graph with a few primitive construction elements, subject, predicate and object, is predestined for future extensions. Adding just a new property will not require any adjustments on existing and mature code.

In future iterations the data model and the underlying DataVaults DCAT Profile will be sharpened by:

- Adjusting cardinalities
- Using additional vocabularies
- Using domain specific ontologies
- Redefining and finetuning the DCAT DataVaults Profile.
- Adapting existing Ontologies
- Considering GeoDCAT-AP for location-based datasets

Besides the continuous improvement of the data model, one open issue that needs to be addressed and is currently not covered is the "Preferences and Recommendations". If there is no ontology currently available, a new one needs to be designed. Alternatively, this information can be externalised as dataset instead being part of the user profile.

Finally, it is noted that, as planned in the DataVaults DoA, the revised and final version of the data model will be provided as part of D1.4, in M18 as part of the updated DataVaults concept (accompanied by the final MVP version), incorporating all the changes that will be imposed by the progress of the other WPs.

5 REFERENCES

- [1] W3C, "RDF 1.1 Concepts and Abstract Syntax," [Online]. Available: <https://www.w3.org/TR/rdf11-concepts/>.

- [2] W3C, “Data Catalog Vocabulary (DCAT) - Version 2,” [Online]. Available: <https://www.w3.org/TR/vocab-dcat-2/>. [Accessed 07 2020].
- [3] Big Data Value Association, “Big Data Value Strategic Research and Innovation Agenda (SRIA),” 2017.
- [4] W3C, “ODRL Information Model 2.2,” 2018. [Online]. Available: <https://www.w3.org/TR/2018/REC-odrl-model-20180215/>.
- [5] W3C, “vCard Ontology - for describing People and Organizations,” 2014. [Online]. Available: <http://www.w3.org/TR/vcard-rdf/>.
- [6] L. M. Dan Brickley, “FOAF Vocabulary Specification 0.99,” 2014. [Online]. Available: <http://xmlns.com/foaf/spec/>.
- [7] W3C, “Data Privacy Vocabulary v0.1,” 2019. [Online]. Available: <https://w3.org/ns/dpv>.
- [8] DataVaults, “D1.1 DataVaults Data Value Chain Definition,” 2020.
- [9] OGC, “The Home of Location Technology Innovation and Collaboration,” [Online]. Available: <https://www.ogc.org/>. [Accessed 2020].
- [10] W3C, “SPARQL 1.1 Query Language,” 2013. [Online]. Available: <http://www.w3.org/TR/sparql11-query/>.
- [11] Andaman7, “A7 Protocol,” [Online]. Available: <http://developers.andaman7.com/a7-protocol.html>. [Accessed August 2020].
- [12] SIOC Project, “SIOC Core Ontology Specification,” 2018. [Online]. Available: <http://rdfs.org/sioc/spec/>.
- [13] DataVaults, “D1.3 DataVaults MVP and Usage Scenarios,” 2020.