European Commission

Directorate - General for Informatics
Directorate B – Digital Services
DIGIT B2 - Interoperability

Digital Europe Programme

A07.01 – Development of a SEMIC registry of data models

Action 'Promoting semantic interoperability amongst the European Union Member States (SEMIC)'

Specific Contract 098 under Framework Contract BEACON

Date:	26/04/2024
Doc. Version:	v0.1

DOCUMENT CONTROL INFORMATION

Settings	Value
Document Title:	**Development of a SEMIC registry of data models**
Project Title:	Action 'Promoting semantic interoperability amongst the European Union Member States (SEMIC)'
Document Author(s):	Nathan Ghesquière Emidio Stani Joren Verleyen
Project Owner (European Commission):	Alexandra Balahur
Project Manager (European Commission):	Claudio Baldassarre
Contractor's Project Manager (CPM):	Florian Barthélemy
Doc. Version:	0.1
Sensitivity:	Limited
Date:	26/04/2024

REVISION HISTORY

Date	Version	Description	Author(s)
26/04/2024	0.1	First version	Nathan Ghesquière Emidio Stani Joren Verleyen
XX/02/2024	0.1	Review and approval	Claudio Baldassarre Alexandra Balahur

DOCUMENT LOCATION

The latest version of this controlled document is stored in https://citnet.tech.ec.europa.eu/CITnet/confluence/display/SEMIC/A07.01+Development+of+a+SEMIC+Registry+of+Data+Models

Table of Contents

1 Introduction

1.1 Context and scope of this document

In the rapidly evolving landscape of digital governance and interoperability initiatives within the European Union (EU), there arises an imperative need for harmonizing semantic models across Member States. Currently, the fragmentation of standardization efforts and limited collaboration between Member States poses significant challenges to efficient data exchange and interoperability between various public administrations and stakeholders. Recognizing this challenge, SEMIC endeavours to address the need for a centralized and standardized repository of semantic models at the EU level.

1.2 Technical requirements workshop

In the context of laying out foundations for a future registry pilot for Q3 and Q4 of 2024, we have looked at adoption requirements in the first workshop, which was held in March 2024. An overview of the first workshop, including the established requirements, can be found below. Afterwards, this document will dive deeper into the technical requirements, which will be presented during the second workshop in May. These technical requirements consist of two parts:

- Metadata: In this section, six different repositories were investigated to make a comparison of the technical components each repository possesses and the features they have that could potentially be valuable for the registry.
- Architecture: The architectural components encompassing the registry give a clear summary of the connection the registry could have to national repositories, how storage of the models could be organised and potential tools that can be leveraged.

2 Summary of the first workshop

During the first workshop, the adoption requirements of the EU-wide registry, which aims to promote semantic interoperability among the European Union Member States, was discussed. During this workshop, the potential users and roles in the registry were also highlighted. Use cases were provided to illustrate how Member States can make models available on the registry, customize models, access and download models, and co-create models. Emphasis was put on the difference between a registry and a repository, with the registry serving as a centralized directory of metadata and the repository as a storage facility for actual data artifacts. Lastly, it is mentioned that national repositories can be connected to the EU-wide registry to facilitate the storage and access of semantic models.

2.1 Captured requirements

This section summarizes the most important requirements brought up by the working group. These identified requirements are crucial for ensuring the effectiveness and usability of the EU-wide registry and taking in the feedback given by the Member States. The following requirements were determined:

- The registry pilot should enable domain specific filtering of models when looking for a semantic model.
- The semantic modelling done on the European Commission level should be findable on the registry.
- The ability to compare a model with its previous versions and thus having an overview of the changes implemented with each iteration of the model.
- Publishing a model to the registry should be possible through two methods:

    - Using an API to expose information from national repositories to the registry
    - Metadata harvesting from national repositories
- Within the features the registry could offer, the most important ones that should be prioritized are searching and filtering.
- The contact details of the creator(s) of a certain model should be retrievable from that models' metadata, which gives Member States the chance to contact the creator(s) directly.
- Not every model would be accessible to anyone, so access rights is considered important. Open access is preferred but restricting access should be explored.

3 Metadata

The main functionality of the registry is to be able to search for existing data models in the different repositories. To do so, the registry must store metadata about the models and the respective repositories.

To reach the objectives of the registry, this metadata should be foreseen by Member States and transmitted to the European Registry. To explore the metadata further, below are some functional aspects needed for the registry explained by analysing existing repositories from a set of Member States or other third parties. The repositories analysed are LOV, TNO (Netherlands), SGIT (Italy), DVT (Finland), SGF (France) and DNN (Norway).

The following topics about metadata will be discussed:

- The metadata models that were adopted by these repositories
- Common metadata found between each repository

3.1 Adopted metadata models

The following table presents the analysis of the file format in which the metadata is represented, the reused metadata models and supported file formats across researched repositories. Additional details regarding this analysis are provided in the table below:

Repositories	Metadata file format	Metadata model reuses	File format
SGIT (Italy)	RDF	ADMS-AP_IT (aligned to dcat:Dataset/Distribution, reuses dcat:accessURL, dcat:downloadURL) DCAT (reuses dcat:contactPoint, dcat:keyword, dcat:theme)	• RDF
DNN (Norway)	RDF	ModellDCAT-AP-NO (reuses dcat:Catalogue and dcat:Resource)	• RDF
DVT (Finland)	RDF	IOW DCAM DCAP	• RDF • XSD • JSON schema • OpenAPI • JSON-LD context
LOV	RDF	VOAF DCAT for distributions (versions over time)	• RDF

		Up to the model owner	
SGF (France)	YAML (+ minimum metadata per file format)	Validation of minimum metadata per file format Up to the model owner	• XSD • JSON schema • Table schema • Other (RDF)
TNO (Netherlands)	RDF	DCAT-AP (reuses dcat:Catalogue, dcat:Dataset, dcat:Distribution) Up to the model owner (minimum metadata, aligned to DCAT-AP via library)	• RDF • XSD • JSON schema • OpenAPI • Excel

From the above table it is possible to note that:

- Repositories use mostly RDF to describe metadata
- DCAT/DCAT-AP are the most common vocabularies, in some cases LOV, SGF and TNO leave to the model owners to add other metadata
- Certain repositories do not contain exclusively semantic models, like SGF or TNO while others, DVT, export to other types of formats to be used in other context like APIs

3.2 Minimum common metadata

Next, this set of repositories are compared to identify a minimum set of features thought to be useful to be implemented within the EU-wide registry. These features and whether they are present in each repository researched are outlined in the table below.

Metadata	SGIT	DNN	DVT	LOV	SGF	TNO
Name	X (M)	X (M)	X (M)	X	X	X
Description	X (M)	X (M)	X (M)	X	X	X
Version_info	X (M)	X		(X)	X	X
Creation date	X (DT)		X (DT)		X (D)	
Publication data		X (D)		X (D)		
Modified date	X (DT)	(X) (D)	X (DT)	(X) (D)	(X) (D)	
languages	X (OP)	X (OP)	X	X		
URI identifier	X	X	X	(X)		
contributor	X		X	(X)	X	
publisher	X	X		(X)		
theme	X (OP)	X (URI)	X (URI)	X		
Main concepts	X (URI)	X (URI)	X (URI)			
status	X	X (URI)	X			
Used by	X (project)			(X) (dataset / vocabulary		(X) (project)

Contact point	X	X			X	

(M) = multilingual (X) = optional (OP) = using OP classification (D) = Date (DT) = Datetime

The different colours within the previous table represent various levels of importance and frequency within the investigated repositories:

Mandatory fields

In the table above, the mandatory fields, Name and Description, are shown with a light-blue colour. They are multilingual in the Italian, Norwegian and Finnish repositories with English as a secondary language. Data models described in multiple languages have more chance to be reused. For example, the data.europa.eu portal supports language translation: even if the search keyword used is typed in English; it can find the related dataset given in another language, which is because the metadata is translated. The registry could just rely on the multilingual name and descriptions for the time being, and propose a translation service later.

Uncommon fields

The remainder of the colours in the table above are uncommon fields, not found in every repository. They are, in order of frequency and importance, categorized per colour. Some features, such as creation, publication or modification dates for example, are not used by every repository and thus create a first notable difference between these repositories. These features would be difficult to summarize for the EU-wide registry, as each repository uses different formats. For example, dates are expressed in “datetime” or “date” format. This could be resolved when homogenization is put in place.

Another interesting feature is the language in which a data model is described. Some repositories use code lists adopted by the Publications Office, while others do not. Therefore, homogenization is necessary. The same is needed to describe the “theme” or the “status” of a model. On top of that, in the Norwegian repository, contributor is part of the metadata and publisher is part of the Finnish repository. Ideally, the registry would have both in the metadata.

The Italian, Norwegian and Finnish repositories express the main concepts from each model, that would help a better find of the models. However, the concepts are provided by their URI, while their label would be better, therefore an extraction/transformation mechanism should be in place. Additionally, the “used by” relation could be helpful to find out how the models are used in practice. Finally, a “contact point” could be used to reach out to the owner of the model to eventually contribute to the model.

Looking at the table, the Italian repository (SGIT) metadata covers most of the needs that would be relevant and valuable for the EU-wide registry. Other repositories, such as LOV, can be considered as the minimum needed for the registry to work properly.

The registry could implement such metadata by making the most used features mandatory and the other ones could be made optional.

Example

With the following example, taken from the SGIT repository, the different metadata are explained, and an overview of each metadata is given about the format they are stored in.

Metadata

- *Name*: INAIL Ontology about Injuries and Occupational Diseases”
- *Description*: “This is the ontology of the INAIL Organization about…”
- *Version_info / string*: “Version 2.1 - Modified on 01 December 2023”

- *Creation date*: 2022-09-06T00:00:00+00:00
- *Publication date*: https://w3id.org/italia/work-accident/onto/adm_serv
- *Modified date*: 2023-12-01T00:00:00+00:00
- *Languages*: <http://publications.europa.eu/resource/authority/language/ENG>
- *URI identifier*: “https://w3id.org/italia/work-accident/onto/adm_serv”
- *Contributor*: <https://w3id.org/italia/work-accident/data/organization/td >
- *Publisher*: <https://w3id.org/italia/work-accident/data/organization/inail>
- *Theme*: <http://publications.europa.eu/resource/authority/data-theme/SOCI>
- *Main concepts*: INAILService:EventoLesivoINAIL, INAILService:GestioneAssicurativaINAIL, ...
- *Status*: "catalogued", "published"
- *Used by*: <https://w3id.org/italia/data/project/NDC> (Object with label)
- *Contact point*: <https://w3id.org/italia/work-accident/data/organization/ufficioV> (Object including email)

4 Registry architecture

The architectural components needed for the EU-wide registry for semantic models encompass a range of elements, including the architectural components, connection between the registry and national repositories, storage and existing tools to leverage.

This section will delve into the significance of these architectural building blocks and shed light on why they are relevant for the EU-wide registry for semantic models. By understanding the purpose and functionality of these components, stakeholders can appreciate the value they bring to the registry and the benefits they offer in terms of data exchange and collaboration across the European Union.

Some context however is needed to further explain what these architectural components mean in regard to the EU-wide registry. The context given in the following section will also explain why it is relevant to think about these technical components.

4.1 Architectural components

Firstly, the architectural components of these repositories are analysed. Each repository has a different approach on how they connect to their respective database and national registry. These approaches are explained in the following table:

Repositories	Internal Database	SPARQL endpoint	API	Validation service
SGIT	MySQL (configuration only/optional)	X (public, based on Virtuoso)	X (aggregated + REST based on OpenAPI)	
DNN		X (public, based on Fuseki)		
DVT	PostgreSQL	X (private, based on Fuseki, behind API)	X (REST based on OpenAPI)	
LOV	MongoDB *	X (public, based on Fuseki *)	X (REST)	
SGF				X (for the 3 supported schemas)
TNO	MySQL		X (REST for validation)	X (for XML schema)

* According to the original paper

Most of the repositories have an internal database, a SPARQL endpoint and an API. Italy, Norway and LOV have a SPARQL endpoint that is public, while Finland has one that is private and behind a Rest API.

More details on Rest API's and SPARQL endpoints will be discussed in 4.1, where the different connection options will be explained further.

Some, like SGF and TNO, provide a validation service to validate the metadata before uploading.

4.2 Connection between registry and repositories

Repositories and registry can exchange data in an automated way via API. Depending on the upstream or downstream approach taken, the connection could change.

In the upstream case, Member States could upload the metadata about their models to the registry that would have an API. In such case, there can be at least three approaches:

- **Rest API**

Within the API landscape, the most adopted is the OpenAPI specification that is based on REST principles a common architectural approach based on the HTTP protocol. This approach is used by data.europa.eu to upload metadata on the website, for example by uploading dataset's metadata on a catalogue. The downside of OpenAPI is that the data exchanged is fixed to a data model, thus there is data overfetching, e.g. the end user might need to know just the name of a model and not all the information related (publisher, creation date, etc.).

In addition, there are different tools built upon such specification, acting as gateways, that allow to monitor API usage, providing caching and add security measures (authentication, authorization, etc.) [2] to access such WSO2 API Manager, Gravitee.io, Kong, etc. Italy is going to adopt an API gateway in front of SPARQL endpoint for example to monitor for example the usage of code list.

- **SPARQL protocol**

Another way is the use of SPARQL protocol, based as well on the HTTP protocol, and integrated in many RDF graph databases. SPARQL could be good for advanced users that by knowing the data model behind, can perform custom queries. However, the direct access to a graph database should be controlled. This is not just for the resources (memory, cpu, etc.) used, but also for the data access (monitoring and security). Certain graph databases have security mechanism such as GraphDB, Virtuoso but the security model can vary.

Indeed data.europa.eu has a SPARQL endpoint (based on Virtuoso), that allows to search for datasets, but no security measure is in place.

- **Combination of these 2**

A third approach, could be combine the 2 above approaches, taking advantages of both, Rest API for common usage and SPARQL endpoint for the advanced users by putting the SPARQL endpoint behind the Rest API in turn behind an API Gateway to have more control. This approach is used for example by DVT (Finland) or by SANTE (France) and in the future by SGIT.

In the case of a downstream connection, it would be up to the Registry to connect to the different member States' endpoints, for example:

- LOV provides Rest API to just get the data in JSON format and a SPARQL endpoint that provides metadata about vocabularies according to the VOAF vocabulary.

- SGIT provides only a SPARQL endpoint (based on Virtuoso) that could be queried according to the ADMS-AP_IT and DCAT data models.
- DVT provides Rest API (following OpenAPI documentation) to access the data provided in multiple formats (RDF, XML, JSON), the RDF is based on DCAM, DCAP and its IOW data models. It is also possible to query the SPARQL endpoint behind via the RestAPI but access is restricted.

In addition:

- SGF provides a website of models that can be in three different formats (Table Schema, JSON schema or XML schema) that need to pass a validation service (minimum metadata) and need to be stored in a GitHub repository.
- DNN is a portal collecting different type of data (API concepts, information models, public services). In particular, the portal describes information models (that can be conceptual, logical and physical) in the form of linked data, partially based on DCAT-AP and ADMS, so their metadata can be queried with a SPARQL endpoint (based on Fuseki).

As it is possible to see there are different data models, therefore a potential harvester should be:

- Adapted for each Member States' endpoint in terms of data exchange;
- Modified whenever a data model is updated;
- Executed at a frequency adapted to each Member States' availability service.

Therefore, this operation will need more effort than simply creating an API for upstream. However, changes in the API for upstream can have an impact on the Member States.

Considering that Member States in the upstream will need to provide metadata about their data models it will probably be into RDF but using their model, thus a transformation process will be needed to homogenise to the one of the registry.

4.3 Storage

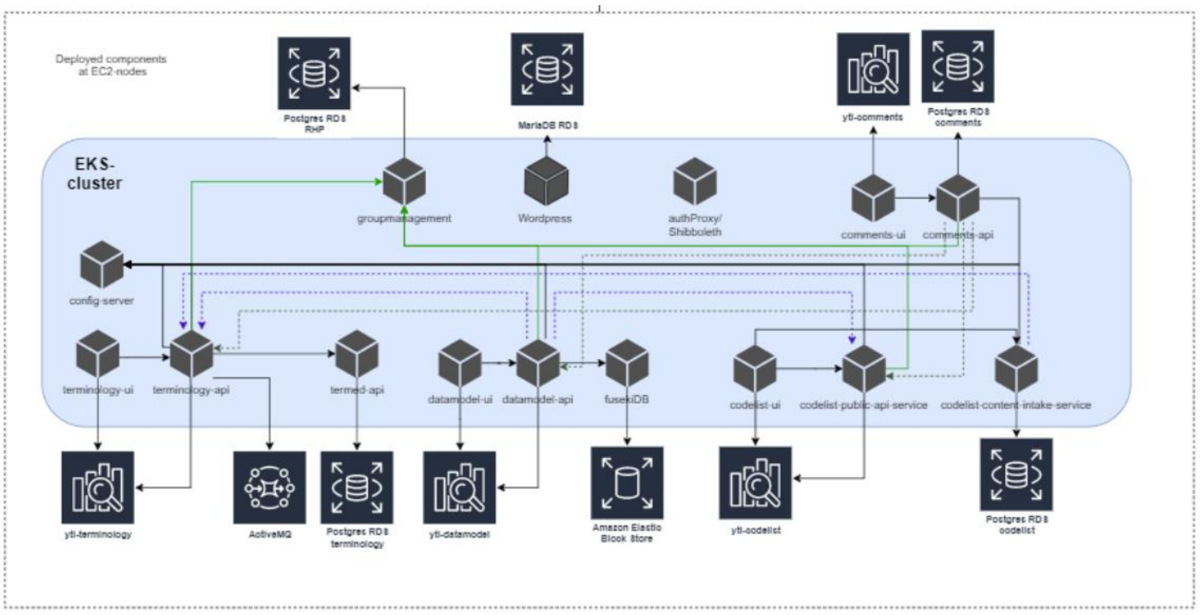
The storage of a registry partially depends on how a data model will be used. From the above repositories, it is possible to see that a SPARQL endpoint is used to query models that it means the related model is described as RDF. The data could be stored in the form of RDF in a SPARQL endpoint to allow Member States in performing custom queries.

Content Management Systems (CMS), based on an RDF model, are scarce, one of the few examples is Linked Data Hub.

However, a SPARQL endpoint might not be the only choice, a relational database could be sufficient for the management of the registry. In this sense, a generic CMS such as Drupal or WordPress, having an integrated authentication mechanism and based on a relational database, could be used.

A relational database could be used:

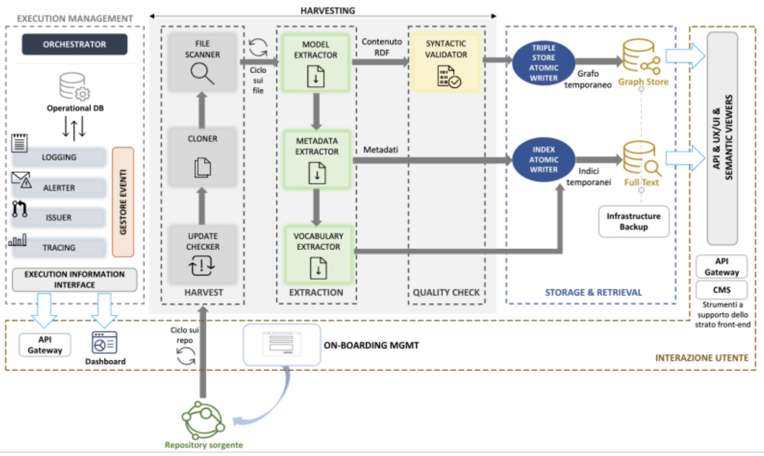
- To store the data and directly used by a CMS, in such case one must take in account, that the metadata about the model could change thus the interface of the Content Management System too.
- To store just configuration for CMS needed to create a portal. In such case the portal could rely on an indexing engine (Apache Solr, OpenSearch, Elastic Search) where data is stored to be effectively searched.



The architecture of DVT, shown as example, there is indeed a mix of relational and graph database:

- The relational (PostgreSQL) used for group management (access), CMS (Wordpress) associated with a MariaDB database
- The graph database (FusekiDB) used for the datamodel website (DVT)

As such the SPARQL endpoint could be behind an API gateway to monitor and secure its usage as it will be used in SGIT. From the image below we can also note that metadata is indexed in a search engine for full text searches.



4.4 Existing tools to leverage

In the below table it is possible to see the tool identified in the previous sections:

Functionality	Tools
SPARQL Endpoint	Virtuoso, Fuseki DB
Relational database	PostgreSQL, MySQL/MariaDB, MongoDB
CMS	Wordpress
API Documentation	Swagger

The applications developed by the repositories are in general custom made. There might be a need for API Gateway and indexing engines:

Functionality	Possible tools
API Gateway	WSO2 API Manager, Gravitee.io, Kong, etc.
Indexing Engines	Apache Solr, OpenSearch, Elastic Search, etc.

In addition, for the registry, there should be a transformation process to homogenize the metadata coming from different repositories, that would be needed to be custom made. However, it could rely on existing transformations engines like Linked Pipes ETL, Sparql-Generate, Sparql Anything, OpenRefine, etc.

5 Next steps

This document will be verified with the active working group in May 2024. After the verification of this document, the working group will come together in June for a closing meeting to have an overall summary of the workshops held and to give closing remarks. This final feedback would then be brought along for a future pilot project later this year, for which we would like to invite you to participate. Any feedback regarding the content of this document is welcomed in the ‘issues’ tab of this GitHub page.