EJP RD

European Joint Programme on Rare Diseases

H2020-SC1-2018-Single-Stage-RTD
SC1-BHC-04-2018
Rare Disease European Joint Programme Cofund

Grant agreement number 825575

Del 11.6

Virtual platform of RD resources annotated with EJP ontological model

Organisation name of lead beneficiary for this deliverable:

Partner 1 – INSERM-Orphanet

Contributors: AMC, BBMRI-ERIC, ELIXIR/EMBL-EBI, GUF, LUMC, UMCG, UPM

Due date of deliverable: month 12

Dissemination level: Public

Table of contents

List of figures

1. Introduction

Across the multitude of RD resources available, there is a significant need for a standardized representation of data to map different concepts in order to support development of computational tools that will enable robust data analysis and integration[1]. Semantic interoperability among terminologies, data elements, and information models is fundamental and critical for sharing information[2]. Key problems and questions in RD clinical practice and research require intelligent data-mining, and integrative analysis of data from multiple sources to be addressed efficiently in terms of time and personnel.

This deliverable describes the first implementation of the EJP RD ontological model and metadata model (described in deliverable D11.1 "First Ontological model of resources metadata"). The ontological model and the metadata model provide standard vocabulary terms that are used by catalog providers to describe and expose their metadata elements. They were made operational within a "Linked Data Platform" (LDP). This LDP, in combination with the backend triple store (a semantic database), provides a unified rich query environment over which simple user-oriented applications can be built, based on the semantic model defined by the deliverable D11.1. At a first stage, catalog providers (RD-connect, BBMRI, JRC ERDRI, Orphanet…) populate this LDP (central endpoint). A federated endpoint will be setup in the future using the same approach.

The LDP constitutes the first version of a central component of the EJP RD Virtual platform: it allows users to find registries and biobanks represented as a standard model by one or several of their metadata elements through a query endpoint (http://ejprd.fair-dtls.surf-hosted.nl:8890/sparql/). This first version has been set up as an evolutionary model that will be refined and populated further during the project therefore allowing for new query possibilities, according to end-users' needs.

Through the use of the EJP RD ontological model and metadata model, the definition of collection, storage, annotation, and communication standards will ensure that the data exposed in various catalogs can be found, queried and analyzed without risking, for example, miscount errors that could be introduced due to data content overlaps or missed links between datasets. This harmonization is an important step to facilitate research and to improve patient care by accelerating data retrieval, analysis, sharing,

1 Kibbe, W. A., Arze, C., Felix, V., Mitraka, E., Bolton, E., Fu, G., et al. (2014). Disease Ontology 2015 update: an expanded and updated database of human diseases for linking biomedical knowledge through disease data. Nucleic Acids Research, 43(D1), D1071–D1078.

2 Jiang, G., Solbrig, H. R., Iberson-Hurst, D., Kush, R. D., & Chute, C. G. (2010). A collaborative framework for representation and harmonization of clinical study data elements using semantic MediaWiki. Summit on Translational Bioinformatics, 2010, 11.

and exploitation across different sources, such as patient records, databases, registries, and biobanks[3,4,5].

2. Approach

Based on the model described in D11.1, a first "Linked Data Platform" (LDP) was set up that allows semantic queries according to this model.

A LDP is a structured data specification defining a set of integration patterns for building RESTful HTTP services that are capable of reading and writing RDF (Resource Description Framework) data, i.e. a specification for 'speaking' in terms of interoperable, machine interpretable data. The LDP allows use of RESTful HTTP API to consume, create, update and delete RDF resources.

Each catalog had to export its metadata, aligned with the model. Validation of the alignment to the model was done by using a "validator" based on SHEX[6]/SHACL[7].

Shape Expression (SHEX) and Shapes Constraint Language (SHACL) are designed to validate RDF graphs. A SHACL validation engine takes a data graph and a graph containing shapes declarations as input and produces a validation report that can be consumed by tools. All these graphs can be represented in any Resource Description Framework (RDF) serialization format including JSON-LD (JSON Linked Data) or Turtle (Terse RDF Triple Language, a syntax and file format for expressing data in the Resource Description Framework (RDF) data model).

The selected catalogs for this task were the Orphanet catalog of registries & biobanks[8], RD-connect Biobank and registry finder[9], the RD-connect Sample catalogue[10] and the ERDRI directory of registries[11]. As the RD-Connect biobank and registry finder are being integrated in BBMRI-ERIC catalog, working out the former was considered a necessary milestone to describe the latter. The first integration cycle was achieved during a

3 Maiella, S., Olry, A., Hanauer, M., Lanneau, V., Lourghi, H., Donadille, B., et al. (2018). Harmonising phenomics information for a better interoperability in the rare disease field. European Journal of Medical Genetics, 61(11), 706–714.

4 Kodra, Y., Weinbach, J., Posada de la Paz, M., Coi, A., Lemonnier, S. L., van Enckevort, D., et al. (2018). Recommendations for Improving the Quality of Rare Disease Registries. (Vol. 15, p. 1644). Presented at the International journal of environmental research and public health.

5 Gainotti, S., Torreri, P., Wang, C. M., Reihs, R., Mueller, H., Heslop, E., et al. (2018). The RD-Connect Registry & Biobank Finder: a tool for sharing aggregated data and metadata among rare disease researchers. European Journal of Human Genetics: EJHG, 26(5), 631–643.

6 https://github.com/shexSpec/shex/wiki/ShEx

7 https://www.w3.org/TR/shacl/

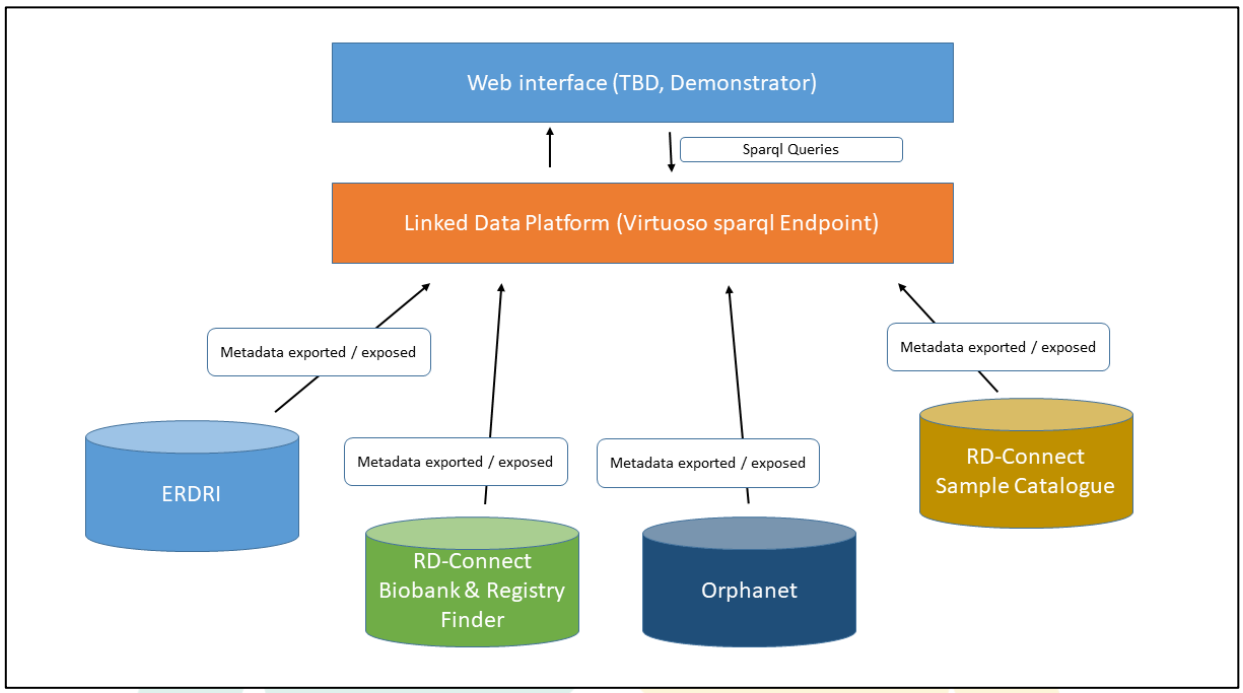
8 https://www.orpha.net/consor/cgi-bin/index.php

9 https://rd-connect.eu/what-we-do/phenotypic-data/rb-finder-for-registries/

10 https://samples.rd-connect.eu

11 https://eu-rd-platform.jrc.ec.europa.eu/erdridor/

"Hackathon" in Paris (July 15-16, 2019) to set up the integration process (Figs. 1 and 2).



TBD: To Be Defined

Figure 1. Scheme of the LDP loading process

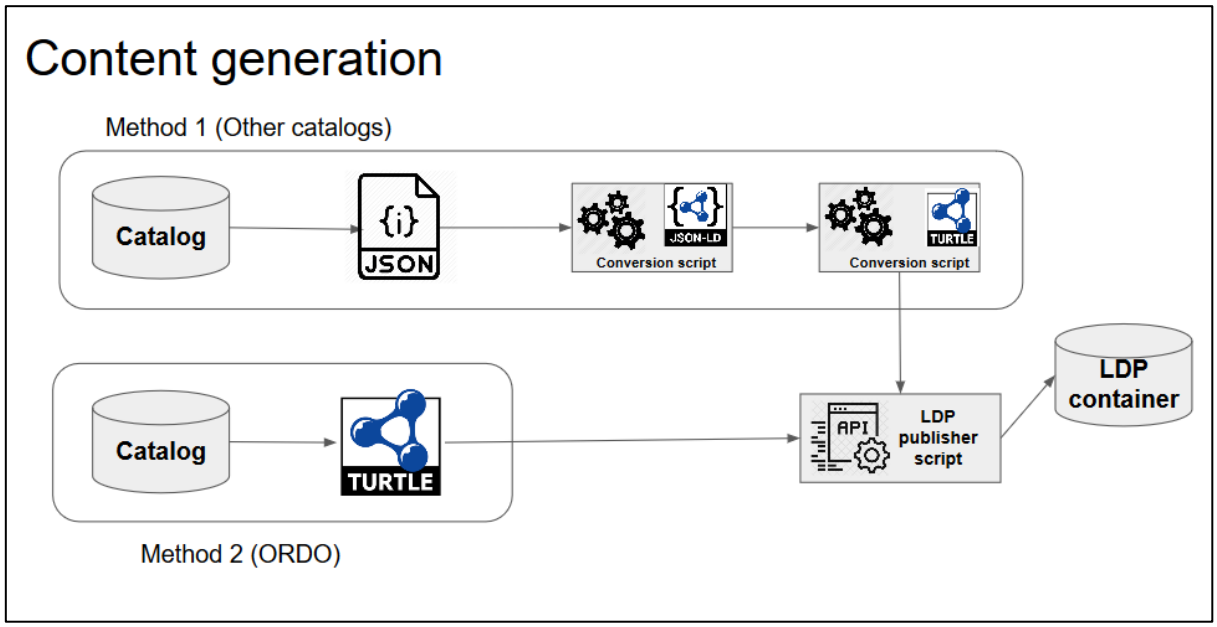


Figure 2. LDP content generation

3. Linked Data Platform

http://ejprd.fair-dtls.surf-hosted.nl:8890/DAV/home/LDP/

The Linked Data Platform (LDP) was designed to facilitate the emergence of a uniformly-behaving read/write Web.

Historically, the Web was primarily readable, with documents being passed via the HTTP protocol. With "Web 2.0", the world saw the emergence of increasingly user-created content (e.g. social media platforms like Facebook, Twitter, Linked-In, etc.). Each platform, however, had its own API that allowed the user-content to be added. With the advent of "Web 3.0" - the Semantic/Linked-Data Web - where machines, rather than people, will be creating much of the Web's content, there was an increasing need to harmonize the way content is added/manipulated. In 2015, The World Wide Web Consortium published its recommendation for LDP - a lightweight, unifying interface, that could be used as the common basis for all read/write Web resources. The LDP is grounded in the core standards of REST, and Resource Description Framework (RDF), and defines an extremely lightweight set of behaviors that a server should exhibit - for example, that there are "containers", and that these "containers" may contain either additional "containers", or may contain RDF-based or non-RDF-based files (e.g. images). The path from a container to its content is explicitly defined, and can be mapped onto whatever equivalent data-structures exist in a pre-existing repository. In this way, exploration of the data can be fully automated.

The simplicity, and standards-compliance, of LDP made it a clear choice for the EJP RD project. One limitation of LDP, however, was that it does not provide support for "search" operations - it facilitates automated exploration, but it cannot be queried. OpenLink Software recently added support for LDP into its widely used Virtuoso database software. In the Virtuoso implementation, a searchable database acts as the back-end storage for the Linked Data Platform server, thus providing for the searchability requirement of EJP RD. Finally, with respect to FAIR-compliance, we note that the first published exemplar [https://peerj.com/articles/cs-110] of high-quality FAIR data publishing used LDP as its basis, thus this technology choice further ensures that EJP RD will be deeply FAIR.

To facilitate the uptake and exploration of LDP within (and beyond) the EJP RD platform, we have created a pre-configured Virtuoso LDP server, and published it as a Docker image (https://hub.docker.com/r/markw/ldp_server). Several instances of this server are now running, including the primary server for EJP RD Pillar 2 research and development (http://ejprd.fair-dtls.surf-hosted.nl:8890/DAV/home/LDP/) as well as an additional public server that can be used for training purposes (http://w3id.org/FAIR_Training_LDP), which is provided via the UPM partner, and their collaborating company, FAIR Data Systems S.L. (Spain), in-kind.

4. Catalogs processing for machine-reading

4.1. Orphanet catalog processing

Orphanet and the Orphanet Rare Diseases Ontology (ORDO) provide knowledge information about rare diseases. ORDO integrates a nosology (classification of rare diseases), relationships (gene-disease relations, epidemiological data) and connections with other terminologies (MeSH, UMLS, MedDRA), databases (OMIM, UniProtKB, HGNC, ensembl, Reactome, IUPHAR, Genatlas) or classifications (ICD-10). Orphanet provides also several directories of relevant resources for rare diseases including information about registries and biobanks. Those directories of resources are available and displayed on the Orphanet website (https://www.orpha.net).

In the context of the EJP RD, and in order to be aligned with others catalogs, Orphanet has developed a process to extract the metadata about registries and biobanks and web services to expose this extraction accordingly to the metadata model described in D11.1 (Figure 3).

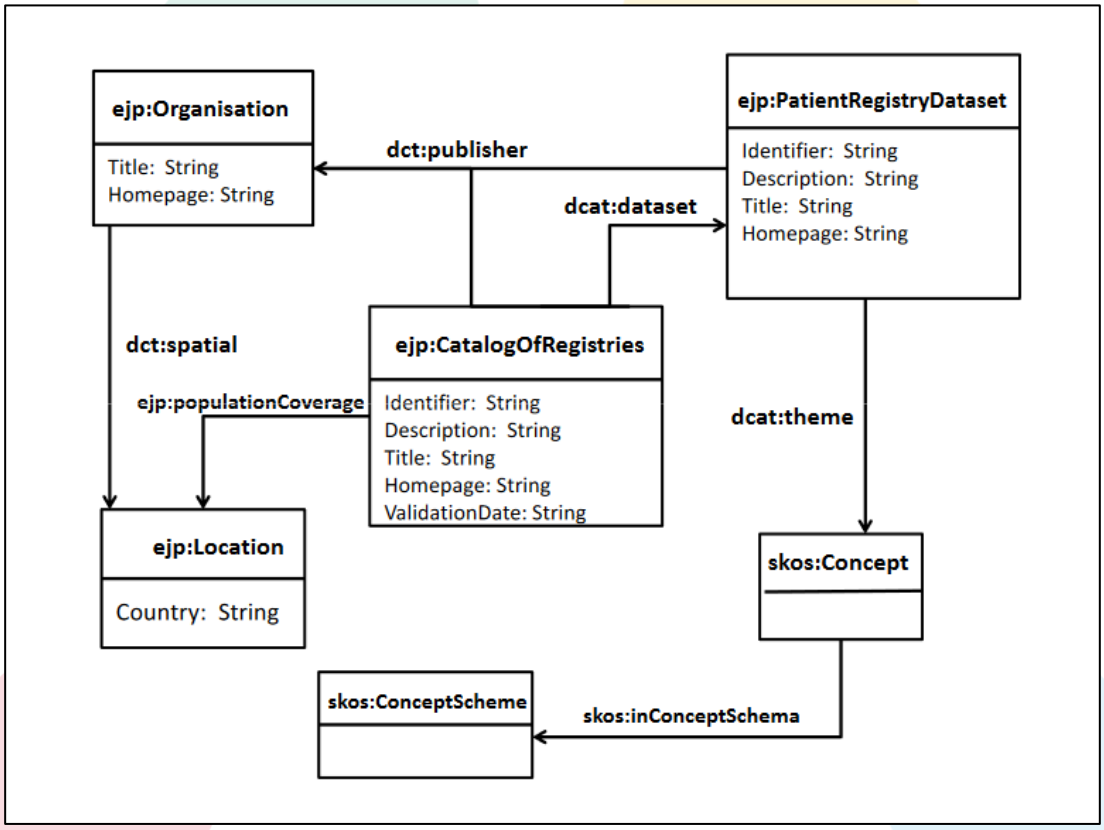


Figure 3. Metadata model used to annotate Orphanet's registries datasets

The following figure (Figure 4) shows the high-level architecture of Orphanet prototype. Two main steps have oriented the software development process: the conversion of the Orphanet resources into RDF triples format and the development of tools to visualize and export the results. The results are implemented as a knowledge base (a blazegraph RDF dataset) which can be queried using SPARQL queries.

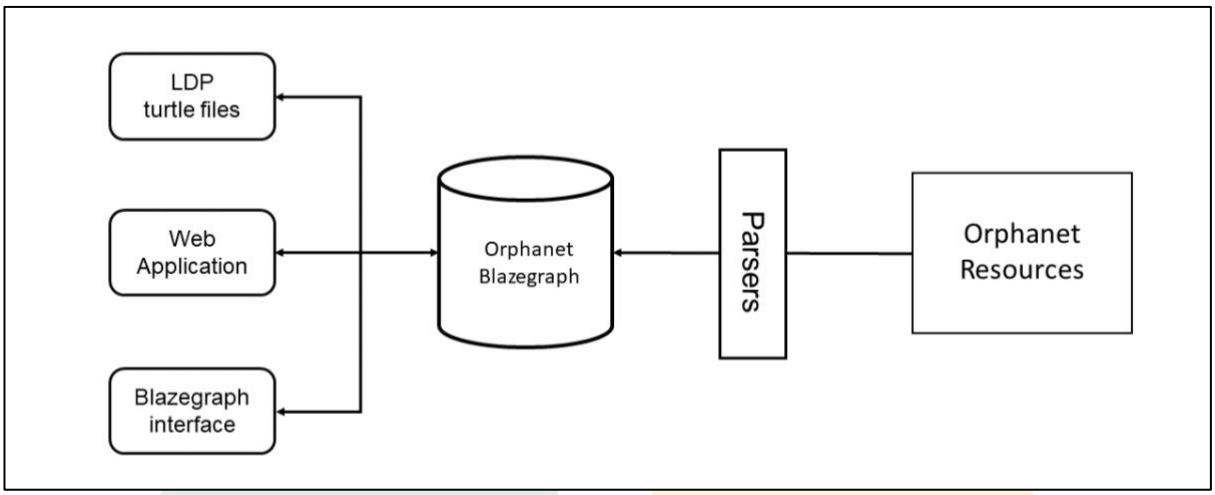


Figure 4. Orphanet's data exposition process Architecture

Basically, Orphanet resources are implemented in xml format and represent an export of the actual information available on the internal Orphanet database. First, parsers were developed to populate the knowledge base by creating individuals (instances) in each class as an ontology, adding data properties between the instances and their literal values, as well as establishing object properties between instances in different classes. The objective of this step is to convert the entire content of the Orphanet resources in order to produce RDF triples format.

Once the blazegraph is populated automatically, RESTful-Web services were developed to expose extractions as Turtle files accordingly to the metadata model described in D11.1. Simple Java Web-based application (a set of Servlets and JSP pages) was also developed and deployed on a Tomcat 9 application server that provides a user-friendly interface allowing to view all the content of the blazegraph. Furthermore, the blazegraph web user interface allows specifying and executing SPARQL queries and to view or download sets of results in several formats (XML, CSV, etc.).

Please note that all Orphanet tools related to EJP RD are available at the EJP RD Github repository https://github.com/ejp-rd-vp/orphanet_prototype

To populate the Linked Data Platform, we have first exported the definition of the Orphanet's catalog:

```
@prefix :  <http://purl.org/ejp-rd/vocabulary/> .
@prefix dct:   <http://purl.org/dc/terms/> .
@prefix rdf:   <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix owl:   <http://www.w3.org/2002/07/owl#> .
@prefix xsd:   <http://www.w3.org/2001/XMLSchema#> .
@prefix ejp:   <http://purl.org/ejp-rd/vocabulary/> .
@prefix ordo:  <http://www.orpha.net/ORDO/> .
@prefix rdfs:  <http://www.w3.org/2000/01/rdf-schema#> .
@prefix dcat:  <http://www.w3.org/ns/dcat#> .
@prefix foaf:  <http://xmlns.com/foaf/0.1/> .
@prefix dc:          <http://purl.org/dc/elements/1.1/> .

ejp:Orphanet_Location
            a           ejp:Location ;
            ejp:country  "FRANCE" .

ejp:Orphanet  a    ejp:Organisation ;
            dct:spatial  ejp:Orphanet_Location ;
            foaf:page           "https://www.orpha.net" .

ejp:CatalogOfRegistries_Orphanet
            a                   ejp:CatalogOfRegistries ;
            dct:publisher       ejp:Orphanet ;

ejp:CatalogOfRegistries_Orphanet
            a                   ejp:CatalogOfRegistries ;
            dct:publisher       ejp:Orphanet ;
            dct:title           "Orphanet" , "The portal for rare diseases and orphan drugs" ;

dcat:dataset       ejp:PatientRegistryDataset_Orphanet_67122 , ejp:BiobankDataset_Orphanet_104017  , //
dcat:themeTaxonomy  <http://www.orphadata.org/data/ORDO/ordo_orphanet.owl> .
```

The "dcat:dataset" contains all references to the Orphanet's catalogs.

In the process and webservices development, Orphanet has directly exported its content using TTL (Terse RDF Triple Language, aka Turtle) as shown in Figure 2 "LDP content generation".

Therefore, each reference has been exported accordingly to the EJP RD's model.

```
@prefix :          <http://purl.org/ejp-rd/vocabulary/> .
@prefix dct:   <http://purl.org/dc/terms/> .
@prefix rdf:   <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix owl:   <http://www.w3.org/2002/07/owl#> .
@prefix xsd:   <http://www.w3.org/2001/XMLSchema#> .
@prefix ejp:   <http://purl.org/ejp-rd/vocabulary/> .
@prefix ordo:  <http://www.orpha.net/ORDO/> .
@prefix rdfs:  <http://www.w3.org/2000/01/rdf-schema#> .
@prefix dcat:  <http://www.w3.org/ns/dcat#> .
@prefix foaf:  <http://xmlns.com/foaf/0.1/> .
@prefix dc:          <http://purl.org/dc/elements/1.1/> .

ejp:PatientRegistryDataset_Orphanet_4084_Organisation_Location
            a           ejp:Location ;
            ejp:country  "FRANCE" .

ejp:PatientRegistryDataset_Orphanet_4084_Organisation
            a           ejp:Organisation ;
            dct:spatial  ejp:PatientRegistryDataset_Orphanet_4084_Organisation_Location .

ejp:PatientRegistryDataset_Orphanet_4084
            a                   ejp:PatientRegistryDataset ;
            dc:creator          "Orphanet" ;
```

```
        dc:identifier          "4084" ;
        dct:publisher          ejp:PatientRegistryDataset_Orphanet_4084_Organisation ;
        dct:title              "EURECHINOREG: European registry of alveolar echinococcosis" ;
        ejp:populationCoverage  ejp:PatientRegistryDataset_Orphanet_4084_PopulationCoverage ;
        ejp:validationDate             "2004-11-29 00:00:00.0" ;
        dcat:theme             ordo:Orphanet_284 ;
        foaf:page              "http://www.orpha.net/consor/cgi-bin/OC_Exp.php?lng=en&Expert=10188" .

ejp:PatientRegistryDataset_Orphanet_4084_PopulationCoverage
        a           ejp:Location ;
        ejp:country  "European" .
```

The resultant RDF for each dataset complies to the defined model as example below:

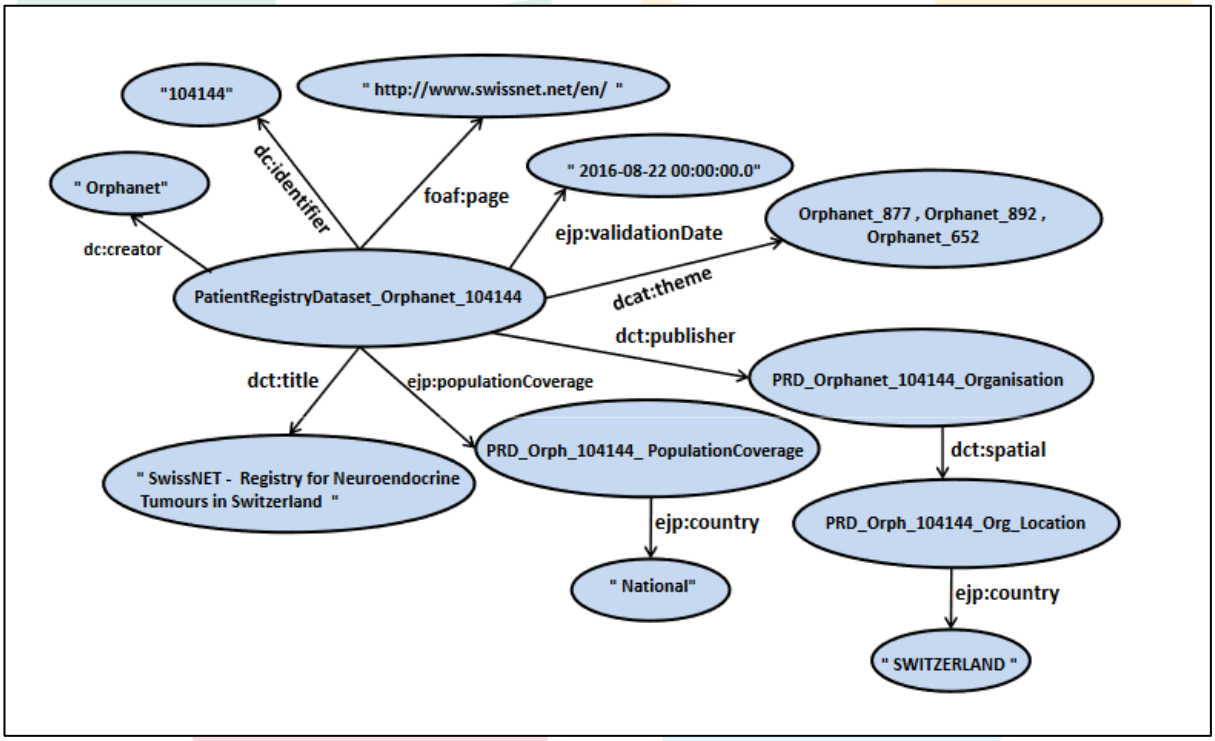


Figure 5. Schema representing a specific RDF description on a particular registry in the Orphanet's catalog, accordingly to the model described in deliverable 11.1

The Linked Data platform was populated with **1024 registries and biobanks metadata**, each linked to one or many rare diseases using "dcat:theme (ordo)" allowing queries in SPARQL. This is used as part of the "Demonstrator" (Figure 6).

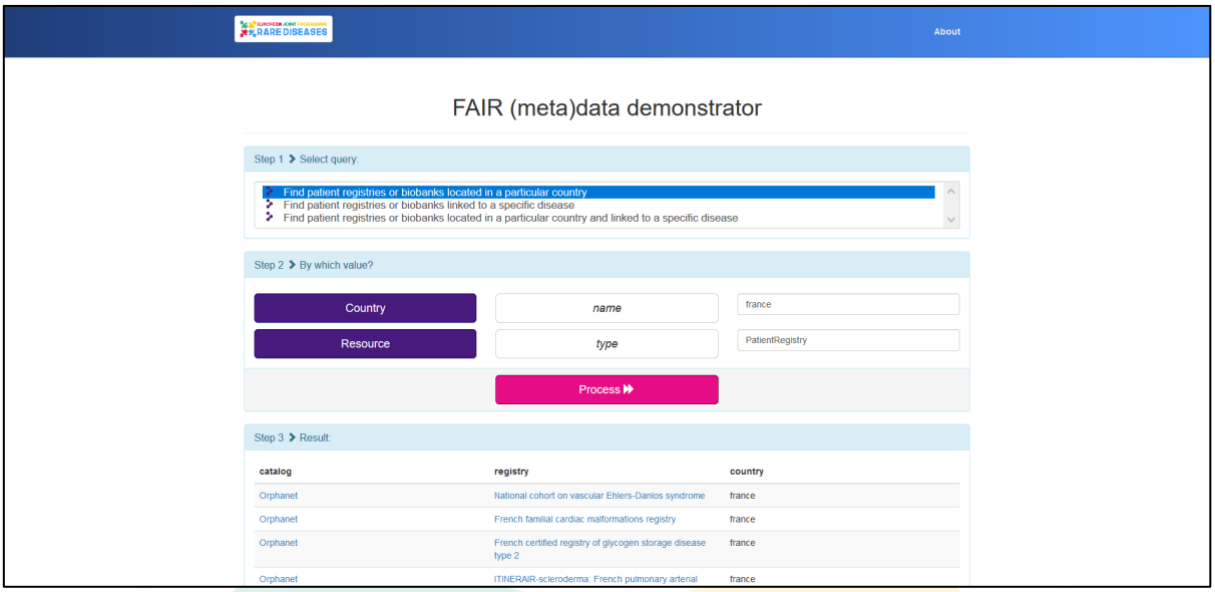


Figure 6. Demonstrator with list of Patient Registries from Orphanet Catalog for "France"

4.2. RD-Connect Registries and Biobanks finder processing

The RD-Connect Registries and Biobank finder contains content data and aggregated data from Biobanks and patient Registries. Each biobank and patients' registry is represented by an ID-Card that contains information about the organisation, contact details and aggregated data of samples/donors found in the Biobank or patients' Registry.

An API and export functionality was implemented as a prototype for feeding the data into the conversion scripts. All aggregated data for registries and biobanks where exported into a common JSON-LD format.


```
{
  "theme" : [
    {
      "@id" : "http:\/\/www.orpha.net\/ORDO\/Orphanet_2020"
    }
  ],
  "publisher" : {
    "name" : "MRC Centre for Neuromuscular Diseases BioBank London\nDubowitz",
    "location" : {
      "country" : "",
      "city" : "-"
    }
  },
  "numberOfPatients" : 1,
  "name" : "MRC Centre for Neuromuscular Diseases BioBank London",
  "@type" : "BiobankDataset",
  "@id" : "http:\/\/localhost:8080\/dataset\/?disease=urn:miriam:orphanet:2020&biobank=168144"
}
```


Also, the Scaleus [https://link.springer.com/article/10.1007/s10916-017-0705-8], a semantic web migration tool, was reactivated and updated as a SPARQL endpoint.

Furthermore, the existing API for connection to other directories and services was extended to fit the data required for the hackathon.

4.3. RD-Connect Sample Catalogue processing

The RD-Connect sample catalogue contains metadata for about sixty-seven thousand biosamples from patients with a rare disease from fourteen different Rare Disease Biobanks in Europe. For each sample metadata including minimal clinical information stored to help researchers find the relevant samples based on disease, sample type or other parameters and to enable the researcher to request access to the samples from the biobank.

A prototype for an adaptor was developed that functions as a converter between the internal MOLGENIS JSON representation of the data and a JSON-LD representation that can be ingested by the conversion scripts mentioned above to incorporate the sample data into the EJP RD Virtual Platform (Figure 7).

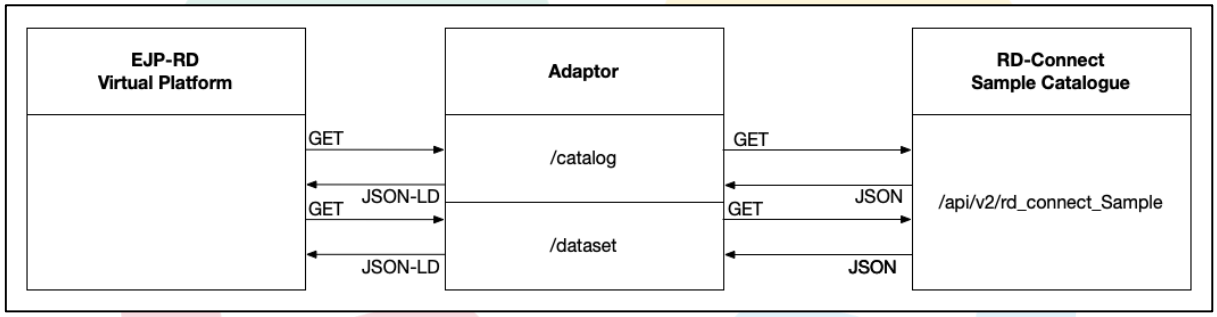


Figure 7. Processing of RD-Connect Sample Catalog

Data in the RD-Connect Sample Catalogue was aggregated per biobank and disease using the MOLGENIS REST API:

- https://samples.rd-connect.eu/api/v2/rd_connect_Sample?aggs=x==BiobankID;y==Disease

The resulting JSON document contains a two-dimensional array, providing the numbers of samples per combination of a disease and biobank. The empty cases were filtered to produce a list of datasets per combination of biobank and disease, which was published as JSON-LD in the /catalog endpoint of the adaptor.


```
{
        "@id": "https://samples.rd-connect.eu/",
        "type": "CatalogOfRegistries",
        "datasets": [
                "https://rd-connect.vapor.cloud/dataset/?disease=ORPHA:119&biobank=168144",
                "https://rd-connect.vapor.cloud/dataset/?disease=ORPHA:2020&biobank=168144",
                "https://rd-connect.vapor.cloud/dataset/?disease=ORPHA:206966&biobank=168144",
                "https://rd-connect.vapor.cloud/dataset/?disease=ORPHA:206973&biobank=168144",
                "https://rd-connect.vapor.cloud/dataset/?disease=ORPHA:262&biobank=168144",
                "https://rd-connect.vapor.cloud/dataset/?disease=ORPHA:265&biobank=168144",
                "https://rd-connect.vapor.cloud/dataset/?disease=ORPHA:267&biobank=168144",
                "https://rd-connect.vapor.cloud/dataset/?disease=ORPHA:280671&biobank=168144",
                "https://rd-connect.vapor.cloud/dataset/?disease=ORPHA:34515&biobank=168144",
```

```
                "https://rd-connect.vapor.cloud/dataset/?disease=ORPHA:457074&biobank=168144",
                "https://rd-connect.vapor.cloud/dataset/?disease=ORPHA:52428&biobank=168144",
                ...
                "https://rd-connect.vapor.cloud/dataset/?disease=ORPHA:99938&biobank=87919"
        ],
        "publisher": {
                "name": "University Medical Center Groningen",
                "location": {
                        "city": "Groningen",
                        "country": "The Netherlands"
                }
        }
}
```


Each of the datasets points to the /dataset endpoint of the adaptor with the disease and biobank ID's as the parameters. Based on the given disease and biobank ID the adaptor queries the RD-Connect Sample Catalogue using the MOLGENIS REST-API to retrieve the number of patients for the given disease in the biobank, e.g.: https://samples.rd-connect.eu/api/v2/rd_connect_Sample?aggs=x==BiobankID;y==Disease;distinct==ParticipantID&q=Disease==urn:miriam:orphanet:119;BiobankID==87919

The resulting JSON document contains a two-dimensional array with the number of patients for the disease in the biobank, as well as the metadata related to the biobank to provide the dataset record. Based on this information the JSON-LD dataset record is published by the adaptor.


```
{
        "@type": "BiobankDataset",
        "@id": "https://rd-connect.vapor.cloud/dataset/?disease=ORPHA:119&biobank=87919",
        "numberOfPatients": 1,
        "theme": [
                {
                        "@id": "http://www.orpha.net/ORDO/Orphanet_119"
                }
        ],
        "name": "Newcastle MRC Biobank for Rare and Neuromuscular Diseases",
        "publisher": {
                "name": "The John Walton Muscular Dystrophy Research Centre",
                "location": {
                        "country": "United Kingdom",
                        "city": "Newcastle upon Tyne"
                }
        }
}
```


A choice was made for the development of an adaptor as a separate service next to the RD-Connect Sample Catalogue to solve several issues. Above all it allowed to rapidly prototype the functionality without affecting the production deployment of the RD-Connect Sample Catalogue. Besides that, the adaptor enabled us to do internal data and API transformations to adapt internal representation of the RD-Connect Sample Catalogue to the models and APIs used by the EJP RD Virtual Platform. Specifically there was a need to transform the notation of the diseases from a short code to the corresponding resolvable Internationalized Resource Identifiers (IRIs), e.g. 119 to http://www.orpha.net/ORDO/Orphanet_119. The code of the adaptor is available on the EJP RD Github repository https://github.com/ejp-rd-vp/rd-

connect-sample-catalogue-adaptor and the service has been deployed on the Vapor cloud at https://rd-connect.vapor.cloud/catalog. In the future the adaptor will be integrated with the RD-Connect Sample Catalogue itself, once the APIs and data structures have sufficiently matured.

4.4. JRC-ERDRI processing

The European Rare Diseases Registry Infrastructure contains metadata about existing European rare disease patient registries. At the current stage ERDRI provides two different sources of information:

- the Directory of Registries (ERDRI.DoR) containing information about a registry itself;
- the Metadata Repository (ERDRI.MDR) holding data element specifications describing every item collected in the respective registry.

In the context of D11.6 data stored in the DoR is appropriate to be integrated into the described Linked Data Platform as information on catalogue level.

Currently ERDRI.dor does not have a RESTful interface, hence there is no possibility to retrieve catalogue data as well as integrating ERDRI.dor into the Linked Data Platform. However, the necessary extension of ERDRI.dor by implementing a RESTful interface is already worked on together with the JRC. In the case of the ERDRI.mdr this has already been done for the demonstration instance and demonstration metadata on data element level can be retrieved via https://eu-rd-platform.jrc.ec.europa.eu/mdr-training/rest/api/mdr/

Currently, all the registry specific information can be accessed, and after logging in, via the graphical user interface of ERDRI.dor, e.g.:

- https://eu-rd-platform.jrc.ec.europa.eu/erdridor/register/2851 or,
- https://eu-rd-platform.jrc.ec.europa.eu/erdridor/register/2478

As there is currently no interface for the retrieval of a structured representation, below is a tabular representation as an example:

	PCD Registry
General Information	
Acronym	PCD Registry
Medical area	Pulmonology
Type of Registry	Epidemiology, Clinical Registry, Basic Research, Patient Registry, Healthcare planning
Other type	
Description	Primary Ciliary Dyskinesia (PCD) is a rare disorder of mucociliary clearance caused by defective hair like organelles (cilia). The purpose of the PCD Registry is to measure, survey and compare different aspects of PCD manifestation, course and treatment, to provide data for epidemiological research and to identify special patient groups suitable for multi-centre trials.
Registry is member of Eurocat	No
Website	https://www.pcdregistry.eu/
Sponsors	
Rare diseases	

Name of the disease (multiple)	[137628] Cardiac anomalies-heterotaxy syndrome
	[363250] Ciliopathy
	[275742] Genetic infertility
	[244] Primary ciliary dyskinesia
	[98861]Primary ciliary dyskinesia, Kartagener type
Structure	
Inclusion and exclusion criteria	
Recruitment area	International
Name of the recruitment area	Global
Recruitment	As of 01/05/2013
Current number of cases	761 at 01/03/2019
Data source	University hospital, Non university hospital, Physician, Research Institution
Other data source	
Data management	central
Link to the privacy policy	
Ethical review committee	
Availability for future collaborations/studies	yes
Registry information	
Institution	University Hospital Muenster of the Westfalian University of Muenster (WWU)
Facility	
Department	Department of General Pediatrics
Street & number	Albert-Schweitzer-Campus 1
Postcode City	D-48149 Muenster
Country	Germany
Responsible for the registry	
Contact person	
Position	
E-Mail address	
Phone number	

An example of how a catalogue metadata set could look like following the metadata model developed in the deliverable 11.1 is shown below:


```
{
 "@id": "https://www.pcdregistry.eu/",
 "@type": "PatientRegistryDataset",
 "name": "PCD Registry",
 "disease_cases": [
            {
            "disease_code": [
            {"@id": "http://www.orpha.net/ORDO/Orphanet_137628"},
            {"@id": "http://www.orpha.net/ORDO/Orphanet_363250"},
            {"@id": "http://www.orpha.net/ORDO/Orphanet_275742"},
            {"@id": "http://www.orpha.net/ORDO/Orphanet_244"},
            {"@id": "http://www.orpha.net/ORDO/Orphanet_98861"}
            ]
            }
 ],
 "publisher": [
            {
            "name": "University Hospital Muenster of the Westfalian University of Muenster (WWU)",
            "location": {
            "city": "D-48149 Muenster",
```

```
            "country": "Germany"
            }
            }
  ]
}
```

5. SHEX/SHACL validation

In order to ensure quality of data and conformity of datasets accordingly to the EJP RD metadata model, it is necessary to automatize and control entries feeding the VP endpoint.

To do so, specific tools which will check datasets are provided.

This validation tool is based on Shape Expression (ShEX) and Shapes Constraint Language (SHACL).

ShEx is a language for validating and describing RDF. ShEx expressions can be used both to describe RDF and to automatically check the conformance of RDF data. The syntax of ShEx is similar to Turtle and SPARQL while the semantics is inspired by regular expression languages.

An example of a ShEx-representation to constrain the properties that can or must be specified for a registry:

```
PREFIX : <https://www.ejprarediseases.org/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

Start = @<RegistryShape>

<RegistryShape> { # A Registry has:
  :id xsd:string+ ;     # one or more IDs
  :name  xsd:string ;             # a name
  :acronym xsd:string? ;# an optional acronym
  :type ["Epidemiology" "Clinical Registry" "Basic Research" "Patient Registry" "Healthcare Planning"]+; #   one or
more types, restricted to those mentioned.
  :othertype xsd:string*         #  any number of other types.
}
```

An example of a description that validates against this shape:

```
PREFIX : <https://www.ejprarediseases.org/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

<Registry>
:id "myRegistry" ;
:name "PCD Registry" ;
:type "Clinical Registry" ;
```

SHACL is a World Wide Web Consortium (W3C) specification for validating graph-based data against a set of conditions. Among others, SHACL includes features to express conditions that constrain the number of values that a property may have, the type of such values, numeric ranges, string matching patterns, and logical

combinations of such constraints. SHACL also includes an extension mechanism to express more complex conditions in languages such as SPARQL.

6. SPARQL Queries

[1]Linked Data Platform (LDP): http://ejprd.fair-dtls.surf-hosted.nl:8890/sparql

SPARQL (a recursive acronym for SPARQL Protocol and RDF Query Language) is an RDF query language to retrieve and manipulate data stored in Resource Description Framework (RDF) format. SPARQL allows for a query to consist of triple patterns, conjunctions, disjunctions, and optional patterns. As a query language, SPARQL is accurate for computer analysts and data stewards and can be used for exploring queries in conjunction with stakeholders. It's not intended as a user interface for every type of stakeholder neither end-users.

This section provides examples of SPARQL queries interrogating the information currently present in the Linked Data Platform (LDP)[1]. Each example URL ('http://…') below can be clicked or copied to a browser to run, under the condition that the 'sandbox' LDP server is active[12]. Please note that the development of additional queries will be possible following the planned extension of the Linked Data Platform's dataset (e.g. number of patients in a registry diagnosed with a particular disease) to be integrated in a future update of the LDP.

6.1. SPARQL queries examples

The following SPARQL query will return a list of all patient registries or biobanks present in the LDP, from all catalogs, displaying the "Titles":

```
prefix ejp:<http://purl.org/ejp-rd/vocabulary/>
prefix dct:<http://purl.org/dc/terms/>
select ?ressources ?titles
where {
{?ressources a ejp:PatientRegistryDataset}
UNION
{?ressources a ejp:BiobankDataset} ?ressources dct:title ?titles}
```

http://ejprd.fair-dtls.surf-hosted.nl:8890/sparql?default-graph-uri=&query=prefix+ejp%3A%3Chttp%3A%2F%2Fpurl.org%2Fejp-rd%2Fvocabulary%2F%3E%0D%0Aprefix+dct%3A%3Chttp%3A%2F%2Fpurl.org%2Fdc%2Fterms%2F%3E%0D%0Aselect+%3Fressources+%3Ftitles++%0D%0Awhere+%7B%0D%0A%7B%3Fressources+a+ejp%3APatientRegistryDataset%7D%0D%0AUNION%0D%0A%7B%3Fressources+a+ejp%3ABiobankDataset%7D+%3Fressources+dct%3Atitle+%3Ftitles%0D%0A%7D%0D%0A%0D%0A&should-sponge=&format=text%2Fhtml&timeout=0&debug=on&run=+Run+Query+

The following SPARQL query will return a list of all the Orphanet catalog's datasets present in the LDP:

```
prefix dcat:  <http://www.w3.org/ns/dcat#>
prefix dct:   <http://purl.org/dc/terms/>
prefix ejp:<http://purl.org/ejp-rd/vocabulary/>
select ?ressources ?titles ?urls  where
 { ?catalogue dcat:dataset ?ressources. ?catalogue  dct:publisher ejp:Orphanet. ?ressources dct:title ?titles.
```

[12]Considering the experimental status of current services, this is not guaranteed.

```
?ressources foaf:page ?urls}
```

http://ejprd.fair-dtls.surf-hosted.nl:8890/sparql?default-graph-uri=&query=prefix+dcat%3A++%3Chttp%3A%2F%2Fwww.w3.org%2Fns%2Fdcat%23%3E%0D%0Aprefix+dct%3A+++%3Chttp%3A%2F%2Fpurl.org%2Fdc%2Fterms%2F%3E%0D%0Aprefix+ejp%3A%3Chttp%3A%2F%2Fpurl.org%2Fejp-rd%2Fvocabulary%2F%3F%0D%0Aselect+%3Fressources+%3Ftitles+%3Furls++where%0D%0A+%7B+%3Fcatalogue+dcat%3Adataset+%3Fressources.+%3Fcatalogue++dct%3Apublisher+ejp%3AOrphanet.+%3Fressources+dct%3Atitle+%3Ftitles.+%3Fressources+foaf%3Apage+%3Furls%7D%0D%0A%0D%0A%0D%0A&should-sponge=&format=text%2Fhtml&timeout=0&debug=on&run=+Run+Query+

The following SPARQL query asks for patient registries or biobanks linked to a specific disease (for instance the Usher syndrome) from the different catalogs:

```
prefix ejp:<http://purl.org/ejp-rd/vocabulary/>
prefix ordo:<http://www.orpha.net/ORDO/>
prefix dcat:<http://www.w3.org/ns/dcat#>
prefix dct:   <http://purl.org/dc/terms/>
select ?ressources  ?titles where
{?ressources dcat:theme ordo:Orphanet_886 . ?ressources dct:title ?titles}
```

http://ejprd.fair-dtls.surf-hosted.nl:8890/sparql?default-graph-uri=&query=prefix+ejp%3A%3Chttp%3A%2F%2Fpurl.org%2Fejp-rd%2Fvocabulary%2F%3E%0D%0Aprefix+ordo%3A%3Chttp%3A%2F%2Fwww.orpha.net%2FORDO%2F%3E%0D%0Aprefix+dcat%3A%3Chttp%3A%2F%2Fwww.w3.org%2Fns%2Fdcat%23%3F%0D%0Aprefix+dct%3A+++%3Chttp%3A%2F%2Fpurl.org%2Fdc%2Fterms%2F%3F%0D%0Aselect+%3Fressources++%3Ftitles+where+%0D%0A%7B%3Fressources+dcat%3Atheme+ordo%3Aorphanet_886+.+%3Fressources+dct%3Atitle+%3Ftitles+%7D%0D%0A&should-sponge=&format=text%2Fhtml&timeout=0&debug=on&run=+Run+Query+

The previous SPARQL query can be expanded to display other diseases concept linked to the same patients' registries/biobanks:

```
prefix ejp:<http://purl.org/ejp-rd/vocabulary/>
prefix ordo:<http://www.orpha.net/ORDO/>
prefix dcat:<http://www.w3.org/ns/dcat#>
prefix dct:   <http://purl.org/dc/terms/>
select ?ressources  ?titles ?disease where
{?ressources dcat:theme ordo:Orphanet_886 . ?ressources dct:title ?titles . ?ressources dcat:theme ?disease}
```

The following SPARQL query will return from the different catalogs a list of patient registries or biobanks located in a particular country and linked to a specific disease (e.g. Italian registries and biobanks linked to Cystic fibrosis):

```
prefix ejp:<http://purl.org/ejp-rd/vocabulary/>
prefix dcat:<http://www.w3.org/ns/dcat#>
prefix ordo:<http://www.orpha.net/ORDO/>
prefix dct:<http://purl.org/dc/terms/>
select  ?ressources ?titles  where {
?ressources dcat:theme ordo:Orphanet_586.
{?ressources dct:publisher ?organization. ?organization dct:spatial ?location. ?location dcat:country-name "Italy"}
UNION
{?ressources dct:publisher ?organization. ?organization dct:spatial ?location. ?location ejp:country "ITALY"}.
 ?ressources dct:title ?titles}
```

http://ejprd.fair-dtls.surf-hosted.nl:8890/sparql?default-graph-uri=&query=prefix+ejp%3A%3Chttp%3A%2F%2Fpurl.org%2Fejp-rd%2Fvocabulary%2F%3E%0D%0Aprefix+dcat%3A%3Chttp%3A%2F%2Fwww.w3.org%2Fns%2Fdcat%23%3E%0D%0Aprefix+ordo%3A%3Chttp%3A%2F%2Fwww.orpha.net%2FORDO%2F%3E%0D%0Aprefix+dct%3A%3Chttp%3A%2F%2Fpurl.org%2Fdc%2Fterms%2F%3E%0D%0Aselect++%3Fressources+%3Ftitles+where+%7B%0D%0A%3Fressources+dcat%3Atheme+ordo%3AOrphanet_586.%0D%0A%7B%3Fressources+dct%3Apublisher+%3Forganization.+%3Forganization+dct%3Aspatial+%3Flocation.+%3Flocation+dcat%3Acountry-name+%22Italy%22%7D%0D%0AUNION%0D%0A%7B%3Fressources+dct%3Apublisher+%3Forganization.+%3Forganization+dct%3Aspatial+%3Flocation.+%3Flocation+ejp%3Acountry+%22ITALY%22%7D.%0D%0A+%3Fressources+dct%3Atitle+%3Ftitles+%0D%0A%7D%0D%0A&should-sponge=&format=text%2Fhtml&timeout=0&debug=on&run=+Run+Query+

The SPARQL query can be used as "back-end" query language. The Linked Data Platform can be used as an API (Application Program Interface). In the previous examples we choose to display the result into HTML arrays for human readability. As the semantic technologies are used to setup machine readable systems, results can be displayed in many other formats, including JSON, RDF/XML or TTL.

7. Demonstrator human-readable interface

To demonstrate the usefulness of the above SPARQL queries and enable end users to provide feedback, FAIR Data Demonstrator tools developed in the RD-Connect project were reused. The FAIR Data Demonstrator is a simple web application that contains predefined template of SPARQL queries described by human readable text (See figure 8). When a user selects one of these queries the tool creates a webform. The webform contains one or more dropdown menus. The number of dropdown menus in a webform varies based on the complexity of the queries (See figure 9). The user can make use of the autocomplete option to select the values of the dropdown menus. This step prepares a SPARQL query that can be executed on the LDP. After this step the prepared SPARQL query can be executed on the LDP by clicking on the process button. The results of the queries will be displayed as a table (See figure 10). In the case of a no result query, a warning message is displayed to the user. The FAIR Data Demonstrator deployment specific to this deliverable can be accessed via this link.

http://purl.org/ejp-rd/fair-metadata-demo

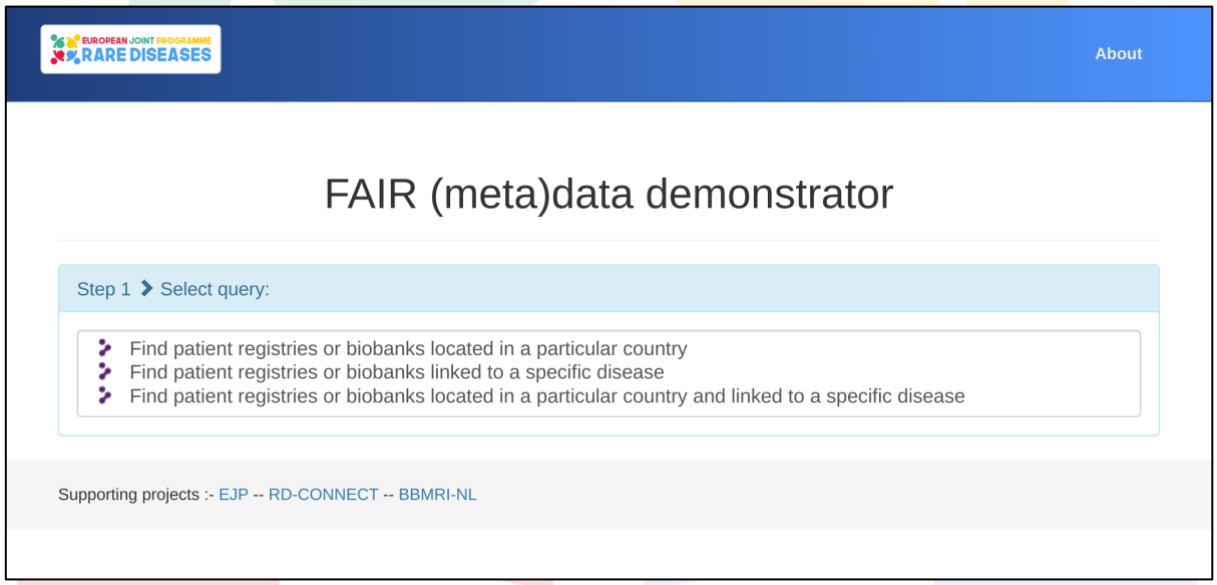


Figure 8. List of predefined queries

Figure 9. Webform based on the complexity of the SPARQL query

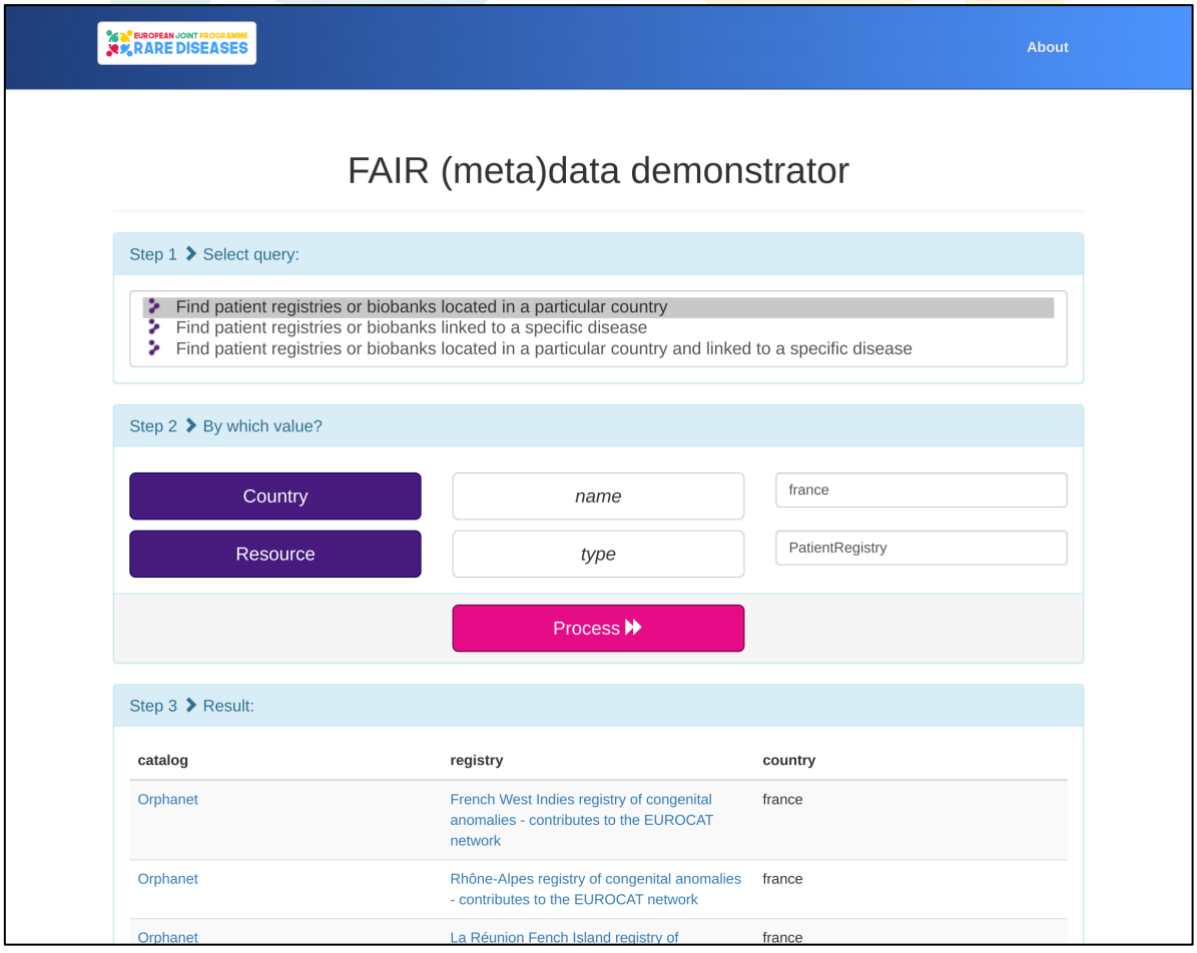


Figure 10. Result table for a query

8. Next steps and discussion

This deliverable represents one of the first steps needed to setup a complete Virtual Platform.

The metadata model, described in deliverable D11.1 and used to provide the Linked Data Platform with catalogs of registries and biobanks metadata, is a "framework model" that can be expanded to other types of resources and different layers of "granularity". Nevertheless, several aspects still need to be addressed:

8.1. Curation process

This model and its implementation are an important first step in setting up the Virtual Platform by providing a common language to multiple resources. A clinician looking for a particular registry obtains access to all LDP-exposed catalog datasets that contain registries and/or biobanks. However, this element of the budding VP is still limited in querying power. Indeed, a single registry entered as individual entry in multiple catalogs can still be labelled differently in each of these catalogs despite a common metadata model (registries labels, despite being constrained by the metadata model in terms of structure—string of characters—can still differ in the wording). If that is the case, these multiple entries will be considered by a query as separate entities and counted twice despite being the same registry. An alignment of concepts is therefore necessary to obtain a fully operational Virtual Platform. This alignment can be automated to a certain extent, but a manual curation is inevitable to ensure that entries which unicity identification relies mostly on metadata that are not easily transformed and standardized automatically, such as labels, are properly mapped across all resources. This mapping task is already ongoing and should improve future VP versions and their querying capabilities.

Furthermore, this will improve as well the Quality of Data (QoD) of each participating repository, which is already one of the positive consequences of the process building this first version of the VP.

8.2. Use of the Orphanet classification of rare disorders, and other hierarchically organized vocabularies

Another challenge is the current level of information that can be inferred from the data. In the current model, only information directly linked to a diagnosis (i.e. disorder, group of disorders or subtypes of disorders) can be retrieved and exposed to the user. One additional feature that will be added to the ulterior versions of the VP to significantly improve data exploitation would be the possibility to use the Orphanet classification of rare disorders. It will allow the retrieval of relevant information based on the diseases hierarchy. Queries exploiting the classification will be able to return for example not only registries linked to a particular disease, but also all registries linked to the subtypes of this disease; aggregated results using groups of diseases will also be allowed.

Improving the query tool such way will allow to reproduce this behaviour exploiting other hierarchical/ontological resources such as the Human Phenotype Ontology (HPO) when the model will be extended to the registries themselves.

8.3. VP architecture / Use cases reciprocal illumination

The correct development of the VP is dependent on continuous interactions between the end users (clinicians, researchers, etc.) defining use-cases and the VP developers. The metadata model chosen and its implementation in this first version of the VP was based on two basic use-cases: (1) how to retrieve registries/biobanks linked to a particular disease, and (2) how to retrieve registries/biobanks of a particular country, and the combination of both (registries/biobanks of particular disease in a particular country). Both are "catalog-level" use-cases, i.e. their focus is on the catalog itself—there is no need to access data from within the registries/biobanks. The intra-registry/biobank data is considered to be a "record-level" data and concerns the patients directly (e.g. number of patients diagnosed, outcomes of patients, etc.).

The next step in the development of the VP is to incorporate metadata description of record-level data into the EJP RD model to integrate into the VP. This will allow the resolution of more complex use-cases (often at the patient level) such as counting the number of pediatric patients with a specific disease, which requires a federated counting query to 'visit' standardized patient information (age and diagnosis in this example).

The collection of relevant use-cases is a task undertaken by the work focus "use-cases", which is gradually creating a use-cases database from the inputs of all stakeholders (researchers, patient advocates, clinicians, policy makers, industry representatives, etc.) to be used for future developments in the VP. These use-cases will be prioritized based on a logical course of actions so that the VP will be able to provide answers to progressively complex multi-resource and multi-level questions. The VP needs to be "field-tested" by various stakeholders to further develop the query tool and the VP interface. The continuous cycle of interaction between 'users in the loop' and VP developers will ensure the production of a VP that meets the expectations and answers the needs of all stakeholders.

8.4. Go from a central to a federated approach

While the demonstrator currently provides a central "warehouse" for metadata representing all participating registries and biobanks, this is clearly not a scalable, sustainable, nor desirable solution. As such, in the next phase of the EJP RD project, we will move from a "warehouse" model to a "distributed" model, where metadata about a registry/biobank is published at-source by each of the participants. Our selection of RDF+SPARQL as the representation language and publishing format for this metadata, provides a means to unify metadata from all of these resources at query-time, since SPARQL is designed to support cross-resource queries by design (known as "federated queries"), spanning multiple independent Web metadata resources. Thus, in the next phase, from Year 2, we will work towards a scalable and sustainable solution to the problem of harmonized metadata exploration and query.

8.5. Update process

This first implementation of several metadata from different catalogs was based on a centralised approach. As mentioned above, the "distributed" model is the main aim. In the distributed model each resource needs to expose its data at the source in a FAIR manner. That will ensure the availability of the most up-to-date version of datasets across the Virtual Platform. Meanwhile, an update process of the Linked Data Platform needs to be performed on a regular basis. At their level, catalogs should provide facilities to do so (API, webservices…), and an automated process, which is compliant with SHEX/SHACL validation step, needs to be developed.

8.6. Link with FAIRification process

Making resources interoperable and machine readable through annotation with metadata models is an important step in the FAIRification process. This step of how to apply the metadata models that are developed (or endorsed) by the metadata work focus (within WP12) will be investigated in a 'learning-by-doing' approach as part of the FAIRification work focus. First, local data stewards will be guided in achieving optimal annotation of their resources by organising collaboration with EJP RD interoperability stewards. Secondly, an inventory of tools is being built that can support data stewards in the resource annotation step. This includes evaluation of existing tools, such as registry software, for their capability to incorporate the annotation step as a less invasive means to achieve high quality annotation. Also, the alignment between annotation of centralised resources and that of federated resources will be investigated to ensure efficient communication between the centralised and federated resources that comprise the EJP RD infrastructure.