PROJECT DELIVERABLE REPORT

Grant Agreement Number: 101058732

Joint Industrial Data Exchange Platform

Type: R

D3.3 Report on Developed Search Services

Issuing partner	UNITN
Participating partners	UCAM, TVS
Document name and revision	D3.3 Report on developed search services
Author(s)	Simone Bocca (UNITN) Rasel Ahmed (TVS)
Deliverable due date	2023-December-01
Actual submission date	

Project Coordinator	Vorarlberg University of Applied Sciences
Tel	+43 (0) 5572 792 7128
E-mail	florian.maurer@fhv.at
Project website address	www.jidep.eu

Dissemination Level		
PU	Public	✓
PP	Restricted to other programme participants (including the Commission services)	
CO	Confidential, only for members of the consortium (including the Commission services)	
SEN	Sensitive, limited under the conditions of the Grant Agreement	

Content

1. Executive Summary

This is the first version of the deliverable describing the data sharing and data search approach adopted by the JIDEP project. More in detail, such an approach is defined by and is part of, the iTelos methodology (see D3.1, final version). The deliverable describes which are the resources produced by the JIDEP project that can be distributed for future reuse, and exploited for new purposes. Following the above-mentioned data-sharing approach, such resources are described by a detailed set of metadata, defined over a specific metadata schema designed for the JIDEP project. The metadata produced are then published online by exploiting a dedicated data catalogue, developed for the project, and based on a lightweight data catalogue framework called JKAN. The deliverable reports the description of the JIDEP catalogue application, by detailing how the application is structured, and the functionalities that it is able to offer. The JIDEP catalogue described in this deliverable, together with the iTelos data-sharing approach, defines the strategy, adopted by the JIDEP project, for the development of data sharing and data search services.

The current section of the deliverable reports the executive summary of the document together with the description of the intended audience. Section 2 describes the iTelos data sharing approach, by highlighting which are the requirements to be satisfied to distribute quality data. Section 3 describes which are the resources that the JIDEP project aims to share, and the metadata defined to describe such resources. Then, Section 4 defines the JIDEP catalogue application, by describing the application components and providing the instructions to be followed to set up and update the catalogue and its content. Sections 5 and 6 conclude the deliverable with the conclusions and a summary of the details that have been updated from the previous version of the deliverable. At the end of the deliverable, two appendices report more details about the management of the catalogue application.

1.1 Intended Audience

The intended audience for this report is composed of the following subjects:

- JIDEP project technical roles: subjects involved in the JIDEP project, having technical competencies, and mainly working over the implantation of the DLT platform.
- Data scientists: technical subjects involved in the iTelos methodology process, by acting over the data management activities (data collection and data sharing/distribution).
- Knowledge engineers: technical subjects involved in the iTelos methodology process, by acting over the knowledge management activities (metadata and metadata schema definition).

2. iTelos Data Sharing

In this section, it is reported the data-sharing approach as part of the iTelos methodology described in D3.1. The distribution of data is fundamental for the reuse of the information. By distributing the data, it is possible to reduce the effort to be paid in developing new services and applications, through the reuse of the data shared. The approach, proposed by iTelos, aims at producing data with a high level of quality and interoperability. During the methodology, external standardised data, and knowledge (schema and ontologies), are collected and reused to build purpose-specific Knowledge Graphs (KGs). Being built by reusing reference standard resources, the KGs produced are in turn compatible within the domains where the reused standard resources are more exploited. Nevertheless, building reusable KGs is not enough to reduce the effort in building new data to support new services and applications. One more step is required regarding the share and distribution of such resources, in a way that they can be found, searched and clearly interpreted to understand if, and which, resources can be suitable for further projects. The "*reuse and share*" approach defined by the iTelos methodology has the objective of reducing the effort in building KGs to support applications and services. It is important to notice that, following this approach, there is no reuse without sharing. The key idea under the iTelos methodology is to bootstrap a circular data economy, where the high-quality data produced by a project has a twofold benefit, (i) to satisfy the relative project purpose, and (ii) to reduce the effort in building further projects, by reusing the data distributed.

Nevertheless, it is not always the case that the data distributed can be exploited for future reuse. Many times, it happens that the data doesn't respect the quality and interoperability criteria required to concretely reuse that, thus increasing the effort to be paid to find, collect and exploit such data for other projects. This is not the case with iTelos where the requirements for data reuse and sharing are respected. Regarding the reuse requirements, the reader can refer to the deliverables D3.1 and D3.2, where the construction of KG by using the iTleos, is described from the methodological point of view, as well as describing the concrete process implementation. Regarding the data sharing or data distribution requirements, there are some criteria to be followed, which are reported below. Currently, one of the most important references for interoperability and reusability is the FAIR principles [1]. Such a set of principles defines the requirements that the data has to satisfy in order to be Findable, Accessible, Interoperable and Reusable. Below is a brief description of such requirements:

- **Findability**: the capacity of data to be found is based on the definition of identifiers and metadata. It is crucial to identify the information carried by a data, by associating it with unique identifiers. The identifiers, as well as other important information, can be defined and metadata associated with the data to be shared and distributed (see section 4.2 for more detail about the metadata adopted in the JIDEP project). The quality of the metadata set for a dataset (or any kind of information to be distributed) defines how much the data is findable

by search services, as well as by the human interpretation of the metadata itself.

- **Accessibility**: even if the data is well described by a quality set of metadata, such metadata has to be first accessed by those who are looking for data resources to be reused. The capacity of data to be accessed depends on the services and protocols adopted to distribute the data and the relative metadata. The data distribution service has to be open, thus accessible and exploitable by any kind of user. The protocols adopted to provide information (metadata) over the resources distributed, have to be fully explanatory regarding the accessible data, but at the same time, they have to protect the sensible data from any kind of malicious use. The services and the protocols have to maintain the metadata for specific resources even if the resources are no longer available.

- **Interoperability**: data interoperability depends on how the data have been built and shaped. To be Interoperable the data, as well as the metadata used to describe such data, have to be represented by using reference standards. Reference standards can be applied both at the data (and metadata) and dataset level. At the data, and metadata, level such standards define the formats and the syntax of the information values within the datasets to be shared. For example, the metadata indicating the date of creation of certain datasets has to use the ISODate format to express the date value. At the datasets level the standards defined the shape of the datasets containing the information to be distributed. For example, a dataset defined in CSV (open format) is more interoperable than other datasets defined using proprietary formats (like Microsoft Excel). It is important to notice how the interoperability touches both the data (and datasets) and the relative metadata.

- **Reusability**: the reusability of a dataset depends on the quality of the information within that, as well as on the quality of the metadata defined for such a dataset. While the quality of a dataset regards the process used to build the dataset itself, the quality of metadata can be improved also after the dataset creation. To increase the reusability of data, it is important to describe it with a rich set of metadata including mandatory, recommended and optional metadata. Some examples of mandatory metadata are the size of the datasets, the date of creation and publication, and the author. A very important (mandatory) metadata is the provenance, specifying the origin of the dataset and, if it is the case, how it has been already handled (eventually modified) until the current moment. Recommended metadata can be for example the number of rows in a tabular dataset, while optional metadata can be the number of users who already visualised the dataset. Nevertheless, the metadata can be mandatory, recommended and optional depending on the domain of the information, as well as the purpose for which the data has been created. Moreover, reusability is strongly dependent on the licence assigned to the data. There are licences that allow the exploitability of the data, and others that limit

their reuse. It is important to choose the right licence considering which information (sensible or not) should be distributed.

Commented [FB(1]: Maybe I understood this wrong, is this meant to be 'sensitive'?

In the JIDEP EU project context, the iTelos methodology is adopted to produce quality data, shaped as KGs, about industrial products and recyclable materials. The project's industrial partners provide data, shaped in their internal formats, that are then transformed into reusable KGs, thanks to the tools described in D3.2, integrated into the JIDEP DLT platform (see D3.4 for more details about the DLT platform). The data produced can be adopted for different new purposes (such as research or industrial projects). To be exploited such data has to be compliant with the above FAIR principles, and it has to be distributed by dedicated services, allowing the users to search, find and access the information they need.

3. JIDEP Shareable Resources

Describing the FAIR principles, in the previous section, shows how the metadata plays a crucial role in the data distribution. A quality set of metadata defined for data to be distributed improves its ability to be found, accessed and reused. In this section, the deliverable describes which types of resources the JIDEP project aims to share, and how such resources are described by a dedicated set of metadata, defined over a specific metadata schema.

3.1 Type of Resources

One of the objectives of the JIDEP project is to produce reusable quality data to be shared. To this end, the project exploits the iTelos methodology that produces reusable purpose-specific KGs. Nevertheless, the approach adopted by iTelos involves the generation of different types of resources, to better represent and shape the information to be produced and shared. Below is a list of all the types of resources considered by the methodology, and thus handled within the JIDEP project.

- **Input Datasets:** these datasets are provided by the industrial partners. These datasets have a high level of heterogeneity to be handled. They can be provided in multiple formats, depending on the format adopted by the data owners. They could be represented through non-standard data schemas.

- **Ontologies**: The iTelos methodology follows a specific knowledge modelling approach, reported in D2.3 and D3.1. According to this approach, purpose-specific ontologies are built, to represent the information finally included in the KGs produced by the JIDEP project. Such ontologies are concrete (knowledge) resources produced by the project. Moreover, to build such purpose-specific ontologies, the iTelos knowledge modelling approach considers the reuse of reference standard ontologies, largely adopted in the domain of interest, thus enhancing the reuse of the purpose-specific ontologies built by the methodology. The distribution of the ontologies produced in the project is crucial to support the iTelos reuse and share approach described in section 2 of the current deliverable.

- **Output Knowledge Graphs**: produced by exploiting the iTelos tool (see D3.2), the KGs resources are the output of the methodology, and so are the main output data of the JIDEP project. These resources are produced by the methodology, by merging the purpose-specific ontologies and the cleaned and formalised input datasets. Due to their high adaptability over different kinds of services and applications, as well as thanks to their high scalability and reduced cost of maintenance, the KGs are quality resources to be distributed. They are the key resources to support further industrial and research projects, as well as to leverage the circular data economy supported by the iTelos methodology.

3.2 JIDEP Metadata

The above types of resources are described by a specific set of metadata. The metadata sets need to be defined for each kind of resource to describe the characteristics of input datasets, ontologies and KGs. Below the metadata for each type of resource is described.

- **Input Datasets Metadata**: the metadata for the resources collected (both data and knowledge) by the iTelos methodology, is already defined by, and depends on, the data source. Different data sources adopt different metadata schemas for their datasets. This is another reason why iTelos defines its own metadata schema for the (knowledge and data) resources produced, to reduce the heterogeneity in the description of the resources distributed.

- **Ontologies Metadata**: iTelos has a specific approach for the definition of the ontologies to structure the final KG (see D2.3 and D3.1 for more details). For this reason, the ontologies produced have a specific metadata schema, highlighting their own characteristics. Here below the metadata attributes of such a schema.
    - **Licence**: this attribute encodes the licence of the resource (e.g., CC-BY-SA-4.0).
    - **URL**: this attribute encodes the dereferenceable URL of the resource.
    - **Keyword**: this attribute encodes the keywords which can quickly convey the topic of the resource.
    - **Publisher**: this attribute encodes the publisher of the resource.
    - **Creator**: this attribute encodes the creator of the resource.
    - **Owner**: this attribute encodes the owner of the resource.
    - **Language**: this attribute encodes the natural language(s) in which the resource's information is represented.
    - **Size**: this attribute encodes the byte size of the resource.
    - **Name**: this attribute encodes the name of the resource in a natural language.
    - **PublicationTimestamp**: this attribute encodes the timestamp of the publication of the resource in the respective catalogue.
    - **Description**: this attribute encodes the description of the resource in a natural language.
    - **Version**: this attribute encodes the version of the resource.

- **Domain**: this attribute encodes the domain to which the resource belongs (e.g., society, territory, automotive).
- **FileFormat**: this attribute encodes the file format of the resource (e.g., OWL-RDF, XML).
- **KnowledgeType**: type of the Knowledge resource, according to the iTelos knowledge modelling approach (e.g., lightweight ontology, teleontology, teleology).

- **Output Knowledge Graphs metadata**: as the final output of the methodology, the KGs have their own specific metadata schema, having the objective of highlighting the characteristics of the data produced by the JIDEP project. Here below the metadata attributes of such a schema.
  - **Name**: this metadata property encodes the name of the resource in a natural language.
  - **Trade Name**: this metadata property encodes the name that indicates the business or company name of the component.
  - **Brand Name**: this metadata property encodes the name that indicates the brand (or the name, if no brand is specified) of the product/component.
  - **URL**: this metadata property encodes the locatable unique identifier of the resource on the web.
  - **Manufacturer Name**: this metadata property encodes the name of the company that produces the product/component.
  - **Manufacturer Registration Number**: this metadata property encodes the registration number of the company that produces the product/component.
  - **Manufacturer Registration Country**: this metadata property encodes the registration country of the company that produces the product/component.
  - **Image**: this metadata property encodes the URL of the image of a product or component (if it exists).
  - **Domain**: this metadata property encodes the domain to which the resource belongs (e.g., society, territory, automotive).
  - **Type**: this metadata property encodes in, natural language, the type of product/component, in the above-specified Domain.
  - **Creation Date**: this metadata property encodes the date (in ISODate format) of the creation for the specific product/component.
  - **Updation Date**: this metadata property encodes the date (in ISODate format) of the last update for the specific product/component.
  - **Owner**: this metadata property encodes, in natural language, the name of the owner of the described resource (KG-dataset).
  - **Status**: this metadata property encodes, in natural language, the status of the described resource (e.g., final, draft, to-be-updated).
  - **Maintainer**: this metadata property encodes in natural language the name of the maintainer of the described resource (KG-dataset).

Commented [FB(2]: Update?

- **Maintainer Email**: this metadata property encodes in natural language the email of the maintainer of the described resource (KG-dataset).
- **Reference Ontology**: this metadata property encodes the URL of the knowledge resource (ontology) used to represent the information in the relative KG, according to the iTelos methodology.
- **Reference Domain Language**: this metadata property encodes the URL of the language resource (reference standard domain vocabulary) used to represent the information in the relative KG, according to the iTelos methodology.

The above described metadata are defined within a dedicated metadata ontology describing all the metadata information that can be used to distribute and search the resources that the JIDEP project aims to share.

4. JIDEP Catalogue

This section reports how the iTelos data-sharing approach, defined above, is implemented through a dedicated data catalogue. The objective of such an implementation is described below, together with the structure of the catalogue and its functionalities.

4.1 Catalogue Objective

One of the objectives of the JIDEP project is to distribute data about products, components and materials, to enhance the future reuse (recycle) of the objects described by such data. By adopting the iTelos methodology not only the product, components and material can be reused, but also the data (the information that such data carries) are distributed in order to be reused for further projects (like data analysis, statistics, research, data integration, and other purposes), thus enabling circular data reuse. As already anticipated in the above sections the data distribution plays a crucial role in this circular approach. The JIDEP project aims to support such an approach by providing a dedicated data catalogue (described in detail in the following section), where the resources produced by the iTelos methodology (KGs) are published. By adopting iTelos, the KGs produced are composed of three kinds of resources described below.

- The datasets transformed into KGs including the data values carrying the information.
- The unique ontology used to represent the information included in the above datasets (produced as a composition of existing standard ontologies, to increase the interoperability of the final KG).
- The reference domain vocabulary used to represent the information in the final KG (produced as a composition of existing standard vocabularies, to increase the interoperability of the final KG).

The JIDEP data catalogue aims at distributing all the above three kinds of resources. More in detail, by accessing the main JIDEP catalogue it is possible to navigate and search the KGs resources available, by querying the metadata used to describe the

KGs published. Then, by accessing the single resources, it is possible to find, and subsequently explore, the ontology and the vocabulary used to represent the information in the relative KG. This is possible thanks to the KGs metadata published in the JIDEP catalogue called "*Reference Ontology*" and "*Reference Domain Language*", respectively (see the previous section for more details about the metadata). In other words, the JIDEP catalogue is an entry point of a stratified data distribution environment which allows the distribution of quality resources at different levels (KGs, ontologies and vocabularies).

It is important to highlight an important feature of the data catalogue adopted to implement the iTelos data distribution. The catalogue does not contain directly the resources to be distributed but only the metadata describing them. This means that by navigating the catalogue the users can search and get information about the resources, through their metadata. Once identified the interested resource, the users will access the actual resource file by downloading it directly, or by making an explicit request to the data owner/maintainer. This approach keeps the data separated from its metadata, thus protecting the real resources from any kind of threats. Notice how a precise and detailed definition of metadata for the resources to be distributed, is crucial to allow the users to find the interested resources.

4.2 JIDEP Catalogue Application

The JIDEP catalogue's objective, described in the previous section, is achieved by the implementation of a data catalogue application based on the JKAN framework [2]. The JKAN framework is a quick and smart solution that allows a user (with basic technical skills) to set up and put in production a data catalogue in a short time (less than 2 hours). The framework base technology is GitHub, in fact, each catalogue developed by using JKAN, is based on its own GitHub repository. Such a repository is then turned into the catalogue GUI, by exploiting the GitHub pages functionality, which produces a static webpage starting from the content of a (github) repository. For this reason, the content of the catalogue repository is divided into two categories of files described in the following subsections. Figure 1 depicts the JIDEP catalogue GUI that is accessible by this link, while the JIDEP catalogue repository can be visualised through this link.

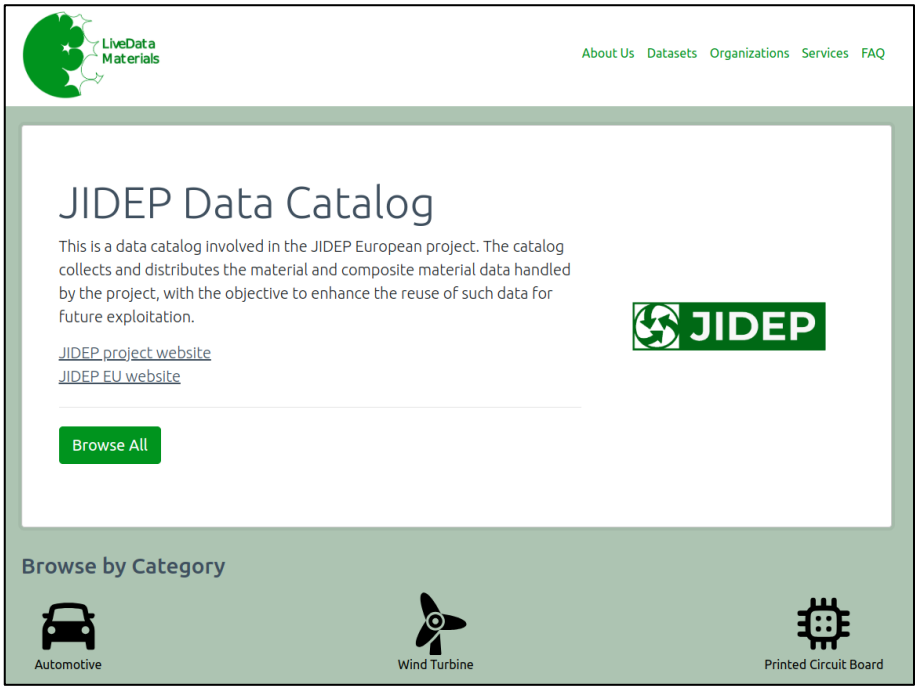

Figure 1 - JIDEP Catalogue GUI

4.2.1 Application Resources

These resources are directories and files defining the structure of the catalogue. More in detail those files that define the graphic appearance of the catalogue, as well as the business logic for its functionalities, like the search of datasets and the filters defined over the metadata (see section 4.3 for more details about the catalogue functionalities). Below is a description of the most important application resources that can be found in the repository.

- **_data**: This directory contains the yml configuration files used to define the metadata schema of the resources handled by the catalogue (within the “schemas” directory), as well as the categories that can be used to categorise the resources in the catalogues, and the custom services which can be accessed to the catalogue.
- **_includes**: This directory contains the website pages (html files) that are shown in the catalogue web page.
- **_organisations**: This directory contains the markdown files defining the organisation that uploads datasets in the catalogue (the data owners).
- **css**: This directory contains the catalogue main.css style file.

- **img**: This directory contains the images used in the catalogue website.
- **scripts**: This directory contains the business logic of the JKAN catalogue. More in detail, in the “src” subdirectory we can find the javascript code divided into the different files, containing different specific functions. While in the “dist” subdirectory we can find the bundle.js. Such a file is created after building the code in the src subdirectory, by using the Jekyll static website generator [3]. The bundle.js file is the only file that is actually executed in order to apply the JKAN business logic to the catalogue. This means that all the modification performed in the source code (src subdirectory) needs to be reported in the bundle.js file too, this happens automatically when running the catalogue build process (see section 4.3 for the details about how to make a new build for catalogue updates).
- **_config.yml**: This is the main config file of the JAKN catalogue. In this file, the base configuration is specified, like the title and URL of the catalogues, as well as the logo and the main navigation bar. Moreover, important configurations are defined in this file, such as the name of the yml file to be used as a reference for the metadata schema of each resource in the catalogue.
- **datasets.json**: This file defines the structure of each metadata set (dataset of metadata) that can be searched within the catalogue. This file is used for the search functionality mostly, this means that, in order to search a metadata set in the catalogue, it needs to respect the field structure in this file, which can be updated in order to add new search criteria.
- **Index.html**: this is the main HTML file of the catalogue webpage. The visualisation of the catalogue website, as well as its navigation, starts from this file.

The above structure of the repository, plus the metadata resources described in the next subsection, is provided through a dedicated template that can be accessed at this link.

4.2.2 Metadata Resources

These resources are the metadata sets (datasets of metadata), namely files storing the metadata for each resource to be distributed by the catalogue. The metadata-sets are maintained into specific markdown (.md) files, and defined through a specific schema (see section 3.2). For each resource to be distributed in the catalogue a relative metadata-set file exists. All the metadata-sets files are stored in a dedicated folder in the catalogue repository.

- **_datasets**: this directory stores the markdown file describing (following a precise metadata schema) each dataset uploaded in the catalogue.

The metadata-set files are read by the catalogue business logic (Jekyll) and displayed in the catalogue GUI, both when they are listed together (see Figure 2), and when the

single metadata set is visualised through a dedicated webpage, as depicted in Figure 3.

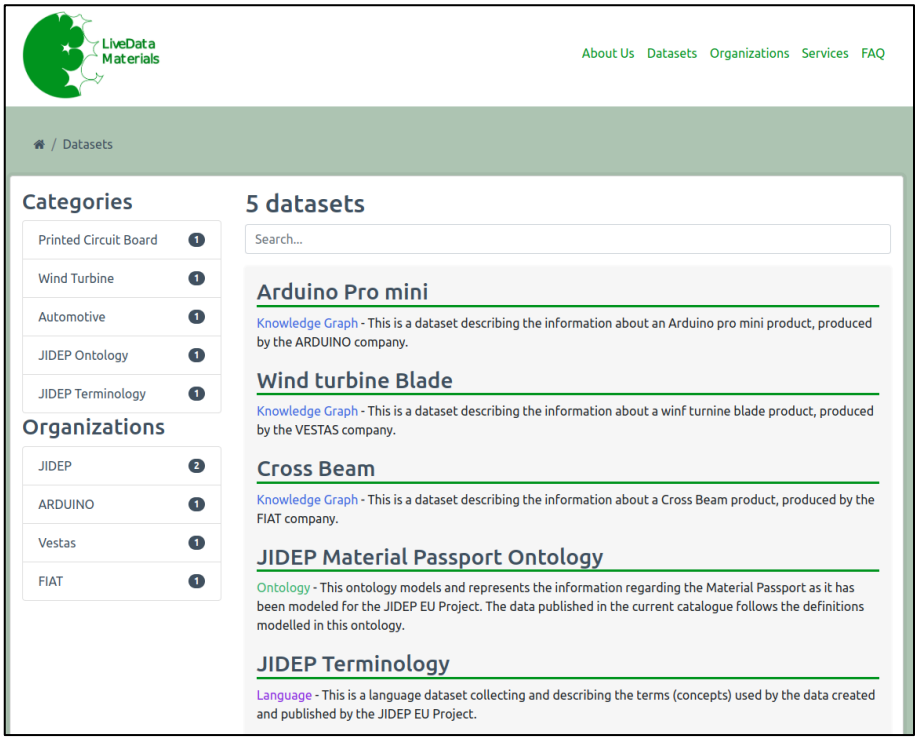

Figure 2 - JIDEP Catalogue list of metadata set published

Figure 3 also highlights two particular types of metadata, already mentioned and partially described in section 3.2, which are crucial for the data-sharing approach defined by the iTelos methodology. The instantiation of such metadata is described below.

- *Reference Ontology*: the value of this metadata is a link to the JIDEP catalogue web page that describes the metadata set of the JIDEP ontology that is used to model all the data (KGs) published. Thanks to this metadata, the user who is navigating the JIDEP catalogue can directly understand the base (ontological) structure of the data that are provided as KGs. The direct link between data and ontology, allows for a semantic disambiguation (disambiguation based on the data schema) of the data to be reused, making it more interoperable.
- *Reference Domain Language*: the value of this metadata is a link to the JIDEP catalogue web page that describes the metadata set of the JIDEP terminology

file, which includes the terms (concepts) used to represent all the data (KGs) published. Thanks to this metadata, the user navigating in the JIDEP catalogue can directly understand the terminology (vocabulary) adopted by the data published in the JIDEP catalogue, thus being able to understand the exact meaning of each term used in the data itself. The direct link between data and its terminology, allows for a linguistic disambiguation (disambiguation based on the data language) of the data to be reused, making it more interoperable.

Figure 3 - JIDEP Catalogue single metadata set webpage

4.3 Catalogue Functionalities

In this section, the deliverable describes the main functionalities offered by the JIDEP data catalogue. Nevertheless, before the description of such functionalities, it is necessary to describe how the JIDEP catalogue has been set up and deployed.

The first step for the creation of the catalogue has been the cloning of the JKAN template repository, already mentioned above. Once the new repository has been

created, the administrator of the catalogue has to do some preliminary modifications in the _config.yml file available in the repository created. More in detail, the following fields have to be modified in the _config.yml file:

- **title**: the title of the catalogue webpage.
- **greeting**: the name of the catalogue, that will appear on the catalogue landing page.
- **description**: the description of the catalogue, that will appear in the catalogue landing page.
- **baseurl**: the base catalogue URL, which must be composed as follows "/<name_of_the_repository_created>".

After these modifications, the catalogue is ready to be used. New metadata sets can be uploaded in the proper directory and automatically displayed as resources available to be distributed. It is possible to make updates over the structure of the catalogue, as well as over the metadata schema used to define each metadata set. For example, a new metadata field can be added to such a schema to allow the catalogue users to search resources based on that new metadata information. Appendices A and B, at the end of this deliverable, provide instructions about how to perform such kinds of updates.

4.3.1 Publish Resources

The first main functionality of the JIDEP catalogue is the publication of new resources to be distributed. Each new resource has to be described by its own metadata-set (markdown file), defined following the specific schema (described in the previous sections) and stored together with the other metadata-sets files in the *_datasets* repository directory. An example of metadata set markdown file content is shown below:

```
---
schema: default
title: Wind Turbine Blade Model B45
domain: wind_turbine
category:
    - wind_turbine
organisation: Siemens Gamesa Renewable Energy Ltd
brand_name: Siemens Gamesa Renewable Energy
trade_name: Siemens Gamesa Renewable Energy
manufacturer_name: Siemens Gamesa Renewable Energy Ltd
manufacturer_registration_number: 76334232
manufacturer_registration_country: Germany
type: product
status: active
owner: CRF
—
```

Once the markdown file is stored in the proper directory, and eventually pushed into the remote GitHub catalogue repository, if the catalogue update is performed locally,

the new resource will be automatically available in the catalogue GUI, described by its metadata.

4.3.2 Search Functionalities

The second main functionality of the JIDEP catalogue is the ability to search through/for published resources. More in detail, the search functionality acts over the metadata sets describing each resource. A search bar is the most basic search functionality offered by the JIDEP catalogue. The users can type, in the search bar (see Figure 4), some keywords related to the resources they are looking for. The strings typed in the search bar are then matched with the metadata values of each metadata set in the catalogue. The search result is composed of the list of resources, represented by their metadata sets, for which the above match was positive.

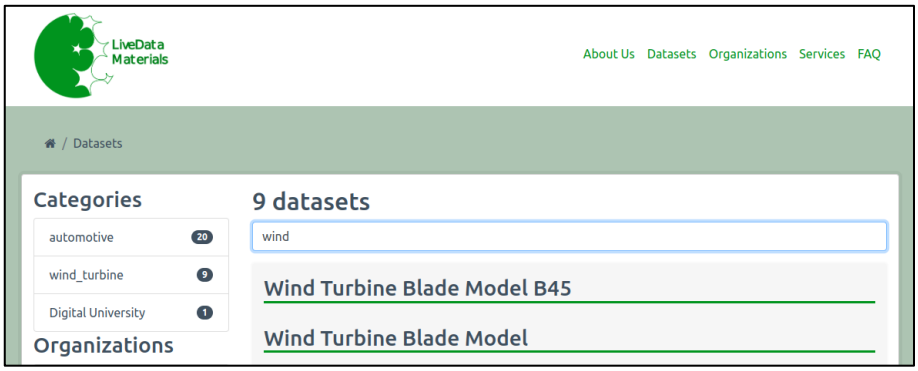

Figure 4 - JIDEP Catalogue search functionality

Moreover, the JIDEP catalogue offers two more advanced search functionalities defined as filters that can be applied to the research of resources. The filters are described as follows.

- *Organisation filter*: each metadata record published in the JIDEP catalogue contains a metadata value that is used to identify the organisation that provided the published information. The value of such metadata must be selected from the list of organisations registered in the catalogue. A portion of such organisations list is depicted in Figure 5. The organisation filter can be used from the catalogue web page where all the resources are listed together. On the left-hand side of such a web page, there is a list of organisation names (see Figure 6). By clicking on one of the items in the list, the filter is applied by selecting (and displaying) all (and only) the metadata sets describing the information provided by the selected organisation.

- *Category filter*: the JIDEP metadata catalogue defines three categories of resources that can be published, based on the domain associated with such resources. The data domains considered by the catalogue, and thus the resource categories, are actually defined by the three JIDEP use cases, namely

the automotive sector, the wind turbine sector and the printed circuit board (PCB) sector. The three catalogue categories are summarised in the catalogue landing page (see Figure 1) and are used as a filter to group the resources in relation to a specific data domain during the user's data search actions. As with the previous filter, the three categories are also provided in the catalogue web page, where all resources are listed together, with the category name list of selectable items located on the left side of the web page layout (see Figure 7). Clicking on one of the items in the list applies the filter by selecting (and displaying) all (and only) metadata resources for the selected data domain (JIDEP use case).

In addition, the JIDEP catalogue allows the combination of the organisation filter and the category filter, so that both can be used for more advanced searches to retrieve the resources provided by a specific organisation in a specific data domain.

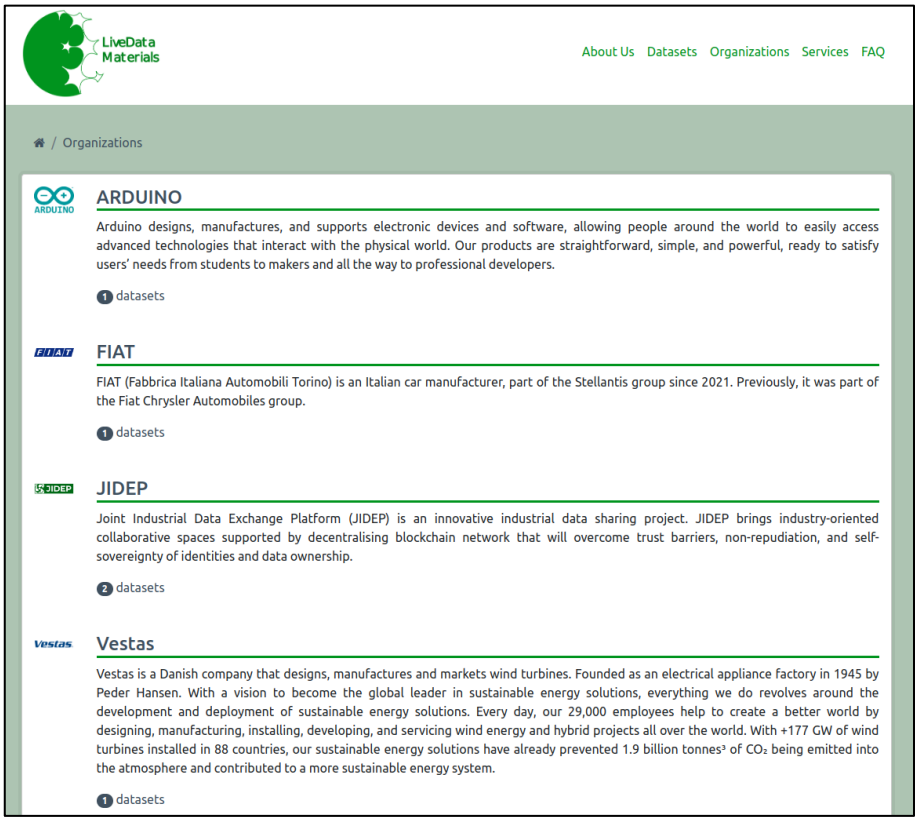

Figure 5 - JIDEP Catalogue organisation list

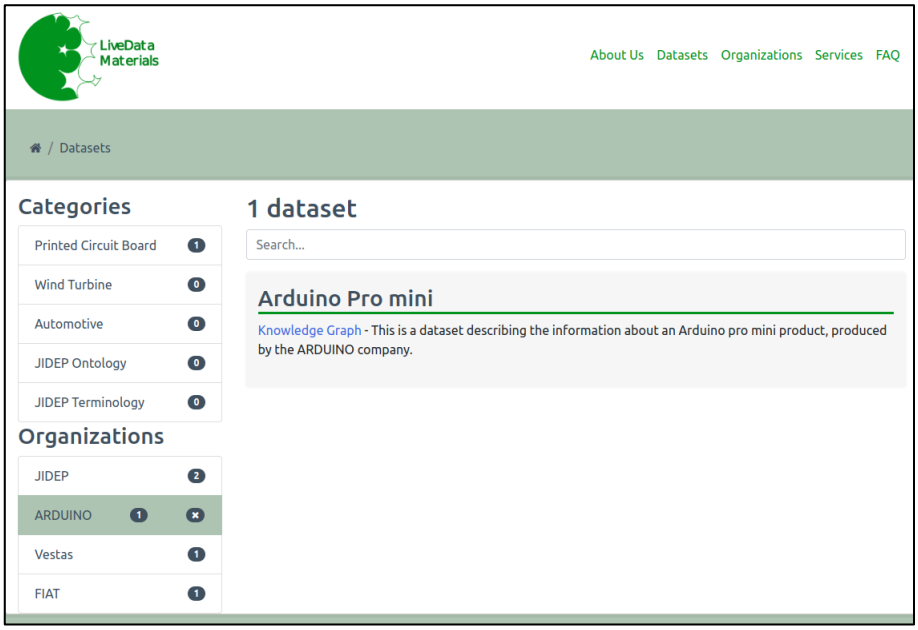

Figure 6 - JIDEP Catalogue organisation filter

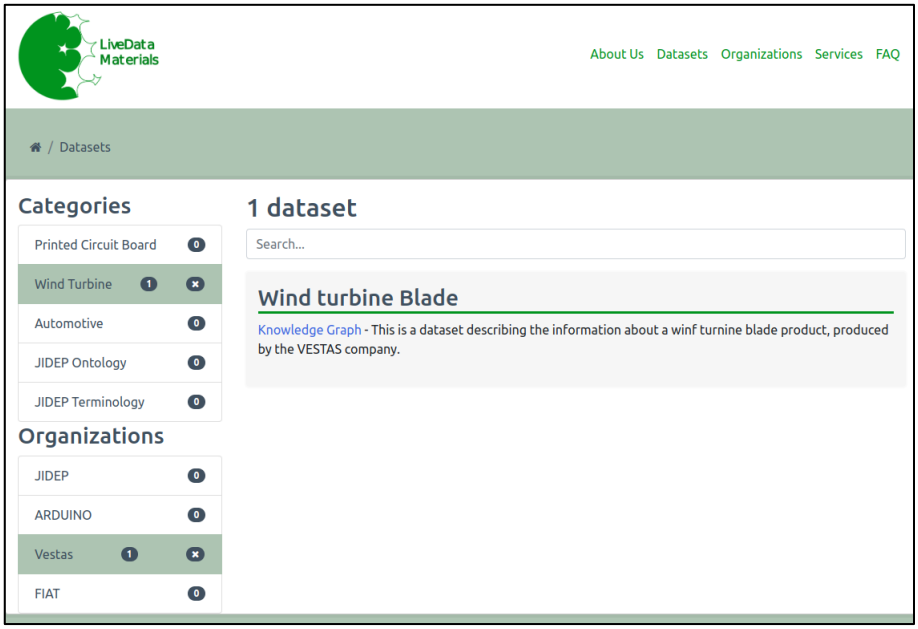

Figure 7 - JIDEP Catalogue category filter

4.3.3 Download Resources

The resources distributed by the JIDEP catalogue are described in the catalogue GUI by their metadata set. According to the iTelos data-sharing approach described in Section 4.1, the files that actually contain the interoperable data, are not stored in the same repository as the catalogue metadata resources. The data source is a different repository that can be defined using a different technology from the one used by the JIDEP catalogue repository (GitHub). For this reason, the download of resources from the JIDEP catalogue is defined by linking the data source repository and the catalogue application itself. The data download links are provided on each specific resource web page (see the link in the red rectangle below in Figure 8), together with the rest of the metadata set defined for the relative resource. Next to the link, for each downloadable data, the format of the file being downloaded is indicated (i.e. in Figure 8, RDF-TTL is the format of the KG data being downloaded). The current implementation of the JIDEP catalogue, for sharing the interoperable material passport data produced by the project, maintains the data to be downloaded in a separate repository. Such a repository is the DLT data storage (see deliverable D3.4), a centralised system where all KG-related data is securely stored, managed, and accessed. It ensures data integrity, supports efficient data retrieval, and facilitates collaboration by providing a single source of truth for all stakeholders.

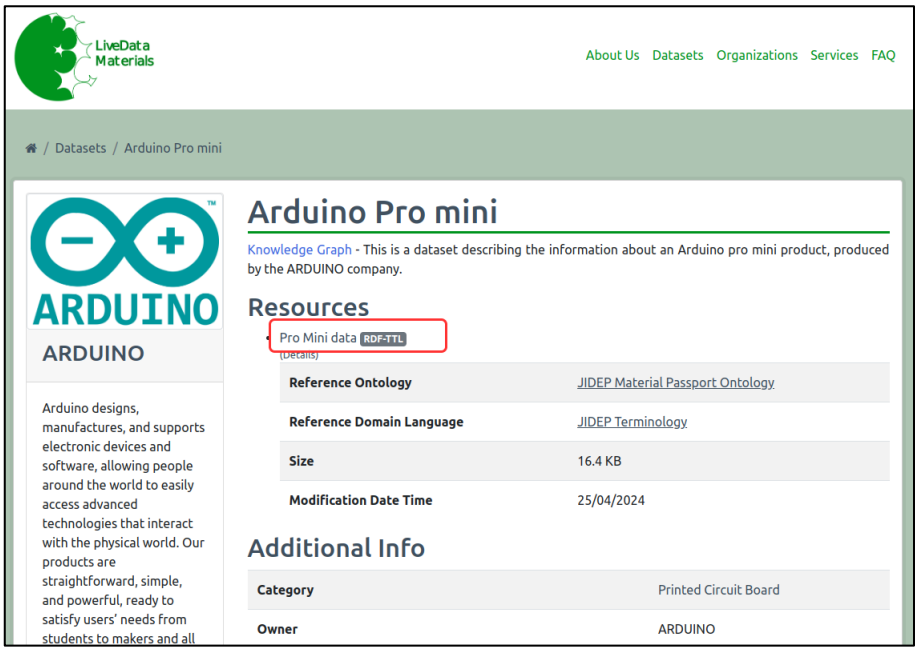

Figure 8 - JIDEP Catalogue resource download link

4.4 Catalogue API and Platform Integration

Managing and sharing data is essential for businesses and organisations in today's dynamic digital landscape. The JIDEP Platform has recognised this need and has developed a user-friendly interface that seamlessly empowers Material Passport (MP) owners (Products and Components owners, intended as users of the JIDEP platform) to publish their metadata to the Catalogue Platform (developed by the UNITN). This integration is made possible through JIDEP's Catalogue API (developed by TVS), which is intricately connected to the Catalogue Platform. The in-depth into the details of this process:

1. **User Interface:** The JIDEP Platform offers an intuitive interface to simplify publishing MP metadata. MP owners can access the dashboard (Material Passport) to control their passport data and take action with just a few clicks.
2. **Publish Catalogue Button**: At the heart of this integration is the "Publish Catalogue" button. When a Passport owner is ready to share their product information, they simply click this button to initiate the process.
3. **Catalogue API:** Behind the scenes, the Catalogue API is the bridge that connects the Passport owner's data with the Catalogue Platform. This API is responsible for securely transmitting the metadata to the Catalogue Platform.
4. **ETL Process:** Upon receiving the metadata, the API takes charge of the ETL (Extract, Transform, Load) process. This process involves:
    a. **Extract:** The API retrieves the Passport metadata, extracting all relevant information.
    b. **Transform:** The extracted metadata is then transformed into the format the Catalogue Platform requires. This step may involve metadata cleaning, formatting, and other adjustments to ensure the metadata is compatible with the destination system.
    c. **Load:** Finally, the transformed metadata is loaded into the Catalogue Platform in its new destination format.
5. **Efficiency and Seamlessness:** This entire process is designed for efficiency and seamlessness. Passport owners can be free of data integration or formatting technical details. The "Publish Catalogue" button abstracts these complexities, enabling smooth and timely publication of their product information.
6. **Enhanced User Experience:** Passport owners can enjoy an enhanced user experience with this integration. They can easily manage and share their data confidently, knowing that it will be handled professionally and integrated seamlessly into the Catalogue Platform.

5. Conclusions

This deliverable describes the data-sharing approach defined by the iTelos methodology and adopted by the JIDEP project. Moreover, it describes which are the resources that the JIDEP projects aim to distribute, and highlights their features. More in detail, the deliverable describes how such resources are described by specific metadata following a metadata schema defined for the resources produced by the JIDEP project. The deliverables then describe the data catalogue application, based on the JAKN framework, that is able to publish the metadata for the JIDEP resources, thus allowing external users to search, visualise and download the resources they are looking for. The instructions are provided, about how to set up the catalogue and how to make updates on it, as well as how to update the existing metadata or upload new ones. The JIDEP catalogue described in this deliverable, together with the iTelos data-sharing approach, defines the strategy, adopted by the JIDEP project, for the development of data sharing and data search services.

6. Updates since the last version

This is the second and last version of the current deliverable. The changes that have been applied respect to the previous version are list here below:

- The section 4.2.2 reports the updates that have been done over the metadata schema applied to each metadata set published in the JIDEP catalogue. The descriptions of the two metadata linking to the JIDEP ontology and JIDEP terminology, have been added to that section.
- The section 4.3 has been updated by adding a description for the search functionalities defined through the organisation and category filters. Moreover, the section 4.3.3 has been updated by providing a more detailed description of the download functionality offered by the JIDEP catalogue.

Appendix A - Customise and Update a JKAN Catalogue

In this appendix are reported the instructions to be followed in order to customise an already existing catalogue (or even a newly created one). The idea is to provide the administrator, the possibility to develop locally (in development) new features for their catalogue, and then push remotely such modifications in the catalogue's repository to be available online (in production). The catalogue web application (website + business logic) is developed in node js, therefore, the administrator needs to be sure to have *node* and *npm* tools [4] properly installed on theirlocal machine. The instructions below require a Unix operating system.

1. Clone the JKAN catalogue repository template to create your new catalogue (or clone the existing catalogue repository, if that is not a new one).
2. Through the Webpack integrated tool, install the JKAN framework components required to build the catalogue project locally. To this end, open a terminal in the /jkan directory and execute the following commands:
    a. # npm install webpack
    b. # npm run build
3. This last command will verify that you are able to build locally the catalogue source code.
4. Install the Jekyll web app server that will locally execute your catalogue.
    a. Inside the /jkan directory open a terminal and run the following:
        i. # sudo gem install jekyll bundler
        ii. Some conflicts may appear during the execution of this command. Follow the instructions you will receive from the terminal directly, to solve such conflicts.
    b. Inside the /jkan directory open a terminal and run the following commands:
        i. # sudo bundle install
    c. Then run the server executing, within a terminal, in the /jkan directory the following command:
        i. # sudo bundle exec jekyll serve
    d. If everything worked well the Jekyll server is running at (localhost) http://127.0.0.1:4000/jkan/

Once the instructions above have been properly executed, the administrator can execute the following step every time she wants to make modifications locally, and then push remotely such modifications in the relative catalogue repository.

1. Do the modifications/changes/updates needed in the catalogue source code.
2. Make a new build (thus generating a new bundle.js file that will be executed online) by running the following command, in a terminal, inside the /jkan directory:
    a. # npm run build
3. Run the Jekyll server (see step 4. c), or close (Ctrl+c in the running terminal) and restart it if it is already running.
4. Open the http://127.0.0.1:4000/jkan/ where the catalogue will be shown with the last updates performed.
5. Verify that the expected behaviour is correct, then commit and push the modification on the catalogue repository by the ordinary git commit-push process.

Appendix B - JKAN Update Metadata and Search Functionality

One of the most important catalogue functionalities (if not the most important) is the research of metadata-set describing the resources published in the catalogue. This appendix reports the instructions to be followed in order to update the metadata structure of the JKAN catalogue, and how to update the catalogue search functionality to enable the search over new (custom) metadata. To this end, the administrator has to execute, locally, the instructions below.

1. Add the new metadata by adding a new element in the file:
    a. _data/schemas/default.yml
2. Update the file datasets.json to upgrade the dataset schema used by the search functionality.
3. Update, as follows, the function "_createSearchFunction (datasets)" in the file: scripts/src/components/datasets-list.js
    a. Modify the second line of the function, by adding the name of the new metadata (those to be used in the search functionality) in the list defined in "const keys = ['title', 'notes', 'maintainer', 'tags']"
4. Make a new build of the catalogue source code, in order to produce a new bundle.js file. Push the last build outcome in the remote GitHub catalogue's repository to make the updates available online.

References

- [1] https://www.go-fair.org/fair-principles/
- [2] https://github.com/timwis/jkan
- [3] https://jekyllrb.com/
- [4] https://docs.npmjs.com/downloading-and-installing-node-js-and-npm

Field Code Changed

Field Code Changed

Field Code Changed

Acronyms and Abbreviations

ADL	ALMAS Partecipazioni Industriali S.P.A.
ADS	Adscensus, MB
AVO	Arteevo Technologies Ltd
BUL	Brunel University London
CRF	Centro Ricerche Fiat Scpa
FHV	Fachhochschule Vorarlberg GMBH
PVI	Precision Varionic International Limited
TPI	TPI Composites
TVS	Technovative Solutions Ltd
UCAM	The Chancellor Masters and Scholars of the University Of Cambridge
UNITN	University Degli Studi Di Trento
UPCE	Univerzita of Pardubice
ZORE N	Zorlu Enerji Elektrik Uretim As

CFRP	Carbon Fibre Reinforced Plastic
CO2	Carbon Dioxide
DLT	Distributed Ledger Technology
EC	The European Commission
ELV	End-of-Life-Vehicle
EOL	End-of-Life
GW	Giga-Watt
IC	Integrated Circuit
Mt	Mega-tons
NMF	Non-Metallic Fraction
PCB	Printed Circuit Board
R&D	Research & Development
RSD	Requirements Specification Document
SME	Small-Medium Enterprise
WEEE	Waste Electrical and Electronic Equipment