



# FNS – Cloud

Food Nutrition Security

## Food Nutrition Security Cloud

### Deliverable 2.3

## Integration of general document and (meta)data repositories

<b>Due Date:</b>	30.09.2020
<b>Submission Date:</b>	30.09.2020
<b>Revision Date:</b>	04.08.2021
<b>Dissemination Level:</b>	Public (PU)
<b>Lead beneficiary:</b>	HYVE
<b>Main contact:</b>	Dr. Julia Kurps, <a href="mailto:julia@thehyve.nl">julia@thehyve.nl</a>

**Project acronym:** FNS-Cloud

**Project Number:** 863059

**Start date of project:** 01.10.2019

**Project duration:** October 2019 – September 2023



Food Nutrition Security Cloud (FNS-Cloud) has received funding from the European Union's Horizon 2020 Research and Innovation programme (H2020-EU.3.2.2.3. – A sustainable and competitive agri-food industry) under Grant Agreement No. 863059 – [www.fns-cloud.eu](http://www.fns-cloud.eu)

Information and views set out across this website are those of the Consortium and do not necessarily reflect the official opinion or position of the European Union. Neither European Union institutions and bodies nor any person acting on their behalf may be held responsible for the use that may be made of the information contained herein.

**D2.3 Integration of general document and (meta)data repositories**

Document Control Information			
<b>Title</b>	<i>D2.3 Integration of general document and (meta)data repositories</i>		
<b>Editor</b>	<i>Julia Kurps (HYVE), Adela Nacu (RTDS)</i>		
<b>Contributors</b>	<i>Tess Korthout (HYVE), Agnes Matuszczak (PMT), Wiktor Kapela (PMT), Karl Presser (PMT)</i>		
<b>Reviewer(s)</b>	<i>Irina Stoyanova (SF)</i>		
<b>Dissemination Level</b>	<input type="checkbox"/> <b>CO</b> Confidential <input checked="" type="checkbox"/> <b>PU</b> Public		
<b>Approved by</b>	<input checked="" type="checkbox"/> RTDS (COO) <input checked="" type="checkbox"/> QIB (SCO) <input checked="" type="checkbox"/> JSI <input checked="" type="checkbox"/> UCD <input checked="" type="checkbox"/> PMT <input checked="" type="checkbox"/> JDLC <input checked="" type="checkbox"/> EuroFIR <input checked="" type="checkbox"/> UWTSD <input checked="" type="checkbox"/> DTU <input checked="" type="checkbox"/> ENEA <input checked="" type="checkbox"/> HYVE <input checked="" type="checkbox"/> HYLO	<input checked="" type="checkbox"/> UM <input checked="" type="checkbox"/> NUTRIS <input checked="" type="checkbox"/> RIVM <input checked="" type="checkbox"/> WUR <input checked="" type="checkbox"/> UGent <input checked="" type="checkbox"/> IMDEA <input checked="" type="checkbox"/> HUA <input checked="" type="checkbox"/> TUM <input checked="" type="checkbox"/> GS1 <input checked="" type="checkbox"/> SF <input checked="" type="checkbox"/> UoR <input checked="" type="checkbox"/> IFA	<input checked="" type="checkbox"/> ILSI <input checked="" type="checkbox"/> BfR <input checked="" type="checkbox"/> AUTH <input checked="" type="checkbox"/> FEM <input checked="" type="checkbox"/> CNR <input checked="" type="checkbox"/> APRE <input checked="" type="checkbox"/> CAP <input checked="" type="checkbox"/> UNIFI <input checked="" type="checkbox"/> LIFE <input checked="" type="checkbox"/> Nutritics <input checked="" type="checkbox"/> EFF
<b>Relevant IPRs</b>	Not applicable		
<b>Underlying Datasets</b>	Not applicable		

Version/Date	Change/Comment
<i>0.1_2020-08-17</i>	<i>First version prepared by HYVE</i>
<i>0.2_2020-08-30</i>	<i>Second version including approach for usability evaluation</i>
<i>0.3_2020-09-15</i>	<i>Version for final review including feedback from SCO and input from PMT</i>
<i>0.4_2020-09-30</i>	<i>Final version created by HYVE and PMT</i>
<i>1.0_2020-09-30</i>	<i>COO edits, final version submitted to EC</i>
<i>2.0_2021-08-04</i>	<i>Updated deliverable as per experts request for revision</i>



## Table of Contents

<b>Publishable Summary</b>	4
<b>Introduction</b>	5
1.1 Background	5
1.2 Objectives	5
1.3 Target audience	5
<b>2. Repository requirements</b>	6
<b>2.1 Background</b>	6
2.2 Repository Requirements	6
<b>3. Data repository evaluation</b>	12
3.1 Introduction	12
3.2 Classifications of data repositories	12
3.3 Selection of data repository tools	13
3.4 Comparison of common features	22
3.5 Conclusions	26
<b>4. Fairspace Development and Implementation</b>	26
4.1 Background	26
4.2 Related work	26
4.3 FNS-Cloud methodology	27
4.4 Technical specifications	27
4.5 Fairspace technology readiness level (TRL)	40
<b>5. Usability testing</b>	41
5.1 Process	41
5.2 Results	41
<b>6. Conclusions</b>	43



## **Publishable Summary**

In the framework of the FNS-Cloud project a wide range of data sets from the food, nutrition and security fields will be integrated and analysed to gain new insights. Therefore, a number of data storage solutions have been tested and evaluated against criteria such as usage cost, open-source license, popularity in scientific European organisations and hosting options. Additionally, different types of data need to have appropriate repositories. In the best-case scenario, datasets for each area would have a standard format to be stored in structured databases (software-managed data), if such storage is not possible and data is stored in files, they should be stored in repositories allowing for rich metadata like Fairspace (file-managed data). If no standard metadata schema is available, the final option would be storing the data files in Zenodo and if that does not fit the needs of the data owner, FNS Cloud can provide file storage using NextCloud. The solution using NextCloud has been deployed for general file storage in the PMT infrastructure and has been made available to all consortium members.

Besides the general evaluation of existing (meta)data repositories, we also describe the Fairspace application which is envisioned to be leveraged for data storage and sharing, metadata annotation and collaborative data analysis for studies executed in WP4. Based on key user feedback we identified the microbiome study as the first use case for Fairspace in the FNS-Cloud project.



## 1. Introduction

This deliverable document is structured as follows: First, we shortly describe the background of the work, we state the main objectives of this deliverable and explain the target audience. After that we describe the requirements for the FNS-Cloud project, the evaluation of multiple existing (meta)data repositories including advantages and disadvantages for the use in the FNS-Cloud project. Next, we focus on the description of the FairSpace application, which is a solution for data storage, metadata assignment, data sharing and analysis. We will provide a high-level overview of the solution and highlight the main components, functionalities and technical implementation. Furthermore, we present an evaluation of the usability of FairSpace conducted through a demonstration to potential key users from WP4 and WP5. We conclude this document with the description of the approach for the usability evaluation of FairSpace.

### 1.1 Background

In the framework of the FNS-Cloud project a wide range of data sets from the food, nutrition and security fields will be integrated and analysed to gain new insights. Therefore, existing data sets as well as data created in research studies that are being executed as part of the project, need to be stored and managed in a way that allows consortium members to share data sets and closely collaborate in data analysis. Therefore, this deliverable aims at identification of solutions to store FNS Cloud data sets and make them available to all consortium members.

### 1.2 Objectives

The main objectives of this document are to

- Gather repository requirements
- Evaluate available data storage tools
- Select tool(s) to be used in the FNS Cloud
- Describe functionality and technical development of FairSpace application
- Describe the usability evaluation of FairSpace

### 1.3 Target audience

The target audience includes

- developers of the FNS Cloud and
- end users of the FNS Cloud:
  - data providers
  - data users



## 2. Repository requirements

### 2.1 Background

The FNS-Cloud project requires an application that aims at helping researchers to better comply with the FAIR principles. Those principles or guidelines have been created to support secondary use of research data by increasing **Findability, Accessibility, Reusability and Interoperability** of data (see box below: DOI: 10.1038/sdata.2016.18).

#### Box 2 | The FAIR Guiding Principles

**To be Findable:**

- F1. (meta)data are assigned a globally unique and persistent identifier
- F2. data are described with rich metadata (defined by R1 below)
- F3. metadata clearly and explicitly include the identifier of the data it describes
- F4. (meta)data are registered or indexed in a searchable resource

**To be Accessible:**

- A1. (meta)data are retrievable by their identifier using a standardized communications protocol
  - A1.1 the protocol is open, free, and universally implementable
  - A1.2 the protocol allows for an authentication and authorization procedure, where necessary
- A2. metadata are accessible, even when the data are no longer available

**To be Interoperable:**

- I1. (meta)data use a formal, accessible, shared, and broadly applicable language for knowledge representation.
- I2. (meta)data use vocabularies that follow FAIR principles
- I3. (meta)data include qualified references to other (meta)data

**To be Reusable:**

- R1. meta(data) are richly described with a plurality of accurate and relevant attributes
  - R1.1. (meta)data are released with a clear and accessible data usage license
  - R1.2. (meta)data are associated with detailed provenance
  - R1.3. (meta)data meet domain-relevant community standards

The core propositions of such an application are:

- Know which data you have
- Collaborate on data analysis
- Comply to regulations and best practices
- Manage the full data lifecycle

### 2.2 Repository Requirements

The following paragraphs describe in detail which requirements need to be met by the FNS Cloud repository. This initial set of repository requirements has been clarified in more detail through input about specific use cases and demonstrators that have been provided by expected key users from WP4 and WP5.

#### **Data**

Data that is currently stored in many places often does not adhere to standard codes or formats, which makes it difficult to combine data sets. Having a common data model (see D2.1 and D2.2) for categorising and indexing the data would already make it much easier to connect



## **D2.3 Integration of general document and (meta)data repositories**

data from different sources. We see a wide variety of types of data that will need to be integrated in the FNS Cloud. It could be databases, spreadsheets, survey data, lab measurements, images, sequencing data or others.

For the purpose of collaborating and sharing the data, the data needs to be stored as (directories containing) files. The contents of the files are not considered, only the descriptive metadata.

### ***Processes***

Different processes are relevant for research data management in the FNS Cloud repository, such as data production and data sharing. We identify several processes and phases, and the different data management needs in those different phases:

- *Data production*: Capture raw data (surveys, lab data, etc.), Store, Annotate, Share/Publish.
- *Analysis*: Explore/select available data within a project, Run analysis in a restricted environment, Download data, Store results, Annotate, Share/Publish.
- *Reuse existing data*: Find, Request access, Retrieve/collect source data, Analyse/use, Annotate, Share/Publish results, Discard.

### ***Research data management challenges***

Researchers might want to find food/nutrition data that is currently distributed over different data storages that are publicly available (e.g. ENA database, MGnify, etc) or owned by different research groups.

It is difficult to relate data in diverse data storages, because there is no connection to data entities, e.g., research participant records or sample databases. This makes it hard to find out if some research or specific analysis has already been done in another research organisation.

### ***Data access policies***

Data access is not restricted for users of the FNS Cloud that have been granted access.

### ***Security, data protection, scalability requirements***

#### ***Relevant regulations and standards***

Audit logging is required, so administrators can produce lists of who is accessing what files, who is changing files, who is changing metadata entities.

Any information related to research participant's consent should be captured in the FNS Cloud environment (at data collecting sites).

#### ***Metadata management***

We describe the purpose and characteristics of the metadata in the catalogue and propose a number of models for access control on the metadata. An essential aspect is the possible presence of sensitive metadata in the catalogue, e.g., business sensitive information about collaboration with competing companies or privacy sensitive information such as shared research participants or sample pseudonyms.



## D2.3 Integration of general document and (meta)data repositories

### **Metadata**

Metadata is data about data. Metadata is used to describe data assets, e.g., for making it easier to find or use certain data. Because metadata is data itself, it can be difficult to make a proper distinction between data and metadata in a system.

#### *Types of metadata*

In a digital archive, *technical metadata* is linked to data assets, like file type, location, size, creation or modification dates, checksums for checking data integrity, ownership. Such metadata is essential for a system to store and retrieve data files. Technical metadata can also include data format specific properties, like encoding, data layout, resolution, etc., required to correctly read the data.

With most publications, *bibliographic metadata* is associated, such as author, title, abstract, publication details, keywords and subject categories. Such metadata makes it possible to find relevant publications. This is the kind of metadata used by libraries and archives and numerous standards exist for such data, such as Dublin Core<sup>1</sup> and METS<sup>2</sup>.

More detailed *descriptive metadata* provides information about the contents of the data, e.g., description of rows and columns, summary statistics, project information, geographical information, results, study design, methods, materials or equipment. In the extreme case, the entire content of the file is captured in descriptive metadata.

We can distinguish different kinds of descriptive metadata, such as:

- Description of the *contents* (rows, columns, values, summary statistics)
- Description of the *subject*, what the data is about (subject, topic, project, study design, object of study, time, location)
- Description of *data sources* (for derived or processed data)
- Description of the *methods* or technology used to produce or capture the data, such as scripts and versions.

In the context of food and nutrition data, it can be needed to link data to research samples or participants (if available, which is often not the case, see D2.1). The values of the metadata can be of any type (numerical, free text, date, etc.), conform to a controlled vocabulary or reference to a typed entity within the database or external entities. Likewise, the data the metadata describes can be of any type: a file system, a tabular file, image, genomic data, a relational database, etc.

---

<sup>1</sup> <https://www.dublincore.org/>

<sup>2</sup> <https://www.loc.gov/standards/mets/>





## D2.3 Integration of general document and (meta)data repositories

### *Purpose of metadata*

Metadata is used for several purposes:

- Descriptors to enable use of the data (file type, file format, encoding, how it was created/generated). The metadata may be used by users or scripts to read or interpret a particular file or data set.
- Finding relevant data for analysis:
  - Metadata may be used to organise data within a data set that a researcher is working on, by using (study specific) categories linked to individual files.
  - Metadata may be used in search queries or navigation to find out if data is available that meets certain selection criteria (e.g. data types, categories, cohort characteristics), for inclusion in a new analysis.
  - Metadata may be used to identify data that is linked to a specific entity, such as a research participant or a sample, to determine if such data has already been analysed, in order to avoid duplicate analysis.

It is important to identify for which purpose metadata is collected and used, as it may affect which types of metadata are collected, how they are navigated and if access control on metadata is desired or required.

### *Use cases*

A *use case* expresses a goal and the steps it involves to achieve that goal. Steps are formulated as interactions between actors (the stakeholders) and system(s).

A use case consists at least of a *Title* (the goal), the *Primary actor* and the *Steps*.

#### **I. Find data (Researcher)**

Find available deposited data linked to a particular research area or project:

- The system provides overviews of configured entities (e.g. research samples, projects)
- The researcher selects a research sample or project of interest
- The system provides metadata about the selected entity (research sample or project)
- The system provides an overview of data sets and files linked to the selected topic of interest
- The researcher downloads the data sets or files

#### **II. Deposit data set (Researcher)**

Store and publish a data set, so that it can be found and used by other researchers

- The researcher starts a data set deposit action in the context of a project (e.g., via a 'deposit data set' action in the collections overview of a project)
- The system provides a dialog for uploading and metadata entry
- The researcher uploads files
- The researcher enters metadata for the data set and individual files
- The system validates the metadata



**D2.3 Integration of general document and (meta)data repositories**

- The system stores and publishes the data set and metadata, assigns the current project as the owner of the data set, and assures the data is persisted and read-only, and grants read access to the owner project.

**III. Automated data ingestion**

Periodically upload bulks of data from source systems and annotating files with metadata.

- A script is triggered, which checks if new data is available
- The script sends new data accompanied with metadata to the system
- The system validates the metadata
- The system stores and publishes the data set and metadata and assures the data is persisted and read-only

**IV. Add project or data set description (Researcher)**

Set a description on project and/or data set to describe the purpose, structure or contents to users with access.

**V. Run an analysis (Data scientist)**

Access data in a secure environment, run analysis and store the results.

- The system provides an interactive analysis environment where the data sets accessible by the data scientist are readable.
- The data scientist opens or creates an analysis script
- The system stores the script in a working directory
- The data scientist tells the system to execute the script
- The system executes the script, possibly reading input files from available data sets, storing created files in a working directory
- The data scientist tells the system to move the scripts and results from the working directory to a project storage
- The system moves the scripts and results to a project storage

**VI. Add project (Management office)**

Add a new project to the systems, assign project manager.

**VII. Archive a project (Project manager)**

Make project data read-only.

**VIII. Close a project (Project manager)**

Make a project inaccessible.

**IX. Reactivate a project (Management office)**

Make a project accessible and project data writable.

**X. Delete a project (Management office)**

Delete a project, only in exceptional cases.

**XI. Manage access to a project (Project manager)**

Grant or revoke access to project data in the data repository to users

- The system provides an overview of users with access to the project
- The project manager adds or removes a user from the list
- The system grants or revokes access to the project data in the repository for the added or removed user

**XII. Add project data (Researcher)**

Upload files to a project in the project data storage.

**XIII. Organise project data (Researcher)**

Organise files in directories, add metadata annotations to a project, data set or file. Create, rename, delete directories, rename, move or delete files.



**D2.3 Integration of general document and (meta)data repositories**

**XIV. Find project data (Researcher)**

Find data within a project in the project data storage based on file name and location and annotations.

**XV. Edit project data (Researcher)**

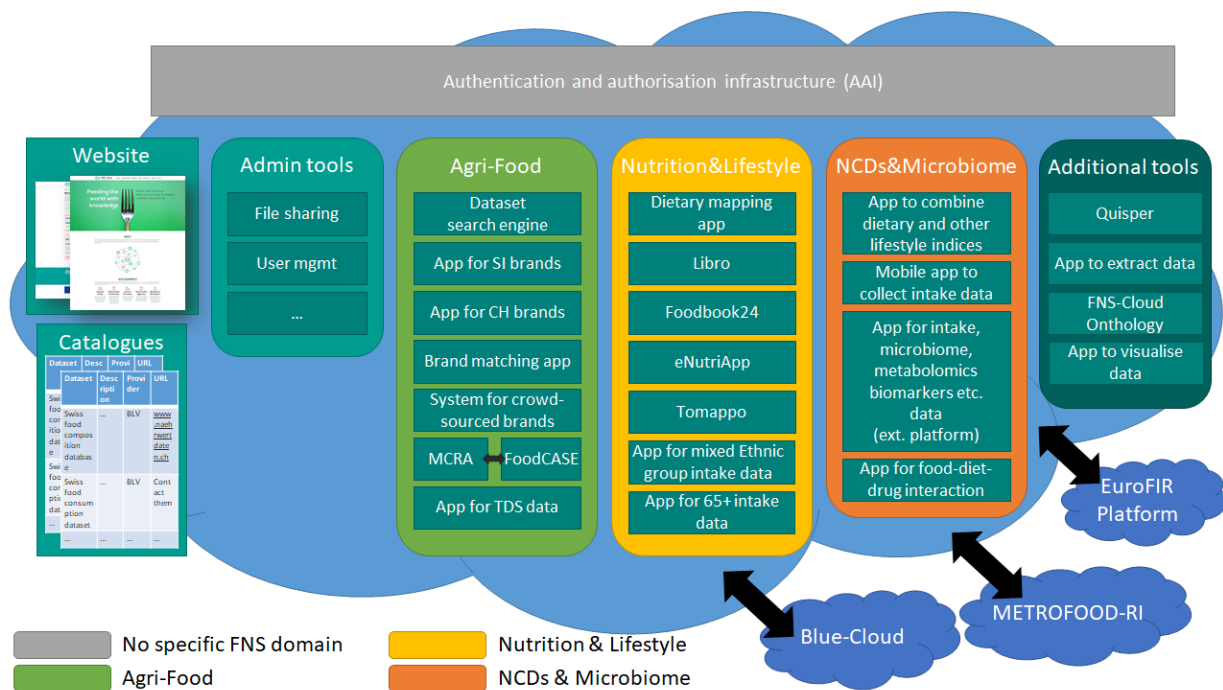
Open files for editing, reupload files, overwrite files via filesystem access. Locking files to avoid data corruption.

**XVI. Share project data (Project manager)**

Share project files with members of another project for reading.

**Specific demonstrator requirements**

While for most demonstrators the details are still being clarified, most of them will be implementing or improving dedicated apps for the data storage from their area (e.g., Libro, Foodbook24, eNutriApp, Tomappo for the Nutrition & Lifestyle demonstrator, FoodCASE for food composition and branded food data, etc.). The exception would be the NCDs and in particular Microbiome study, where different types of data will be collected, not all of which already have a dedicated repository.



This specific use case (DEMO3 - Microbiome demonstrator) has been selected for the Fairspace application demonstration. The technical objective of this demonstrator is to show how diet and microbiome study data can be accessed and analysed through the FNS-Cloud. This demonstrator should result in fully accessible study data with linked diet, biomarker, anthropometric, glucose CGMs, wearables (activity, sleep) and microbiome data. A potential research objective that the demonstrator should serve would be to establish the effect of a complex diet on the diversity of gut microbiota.

The Fairspace team, together with nutritional researchers from WP5 and other subject matter experts, spent a few sessions to determine and assess scenarios for the DEMO3 demonstrator wrt Fairspace. First, the scientists in charge of the DIME study detailed possible cases, including searching for all DIME study data, retrieving this from Fairspace and/or other data sources, and



## **D2.3 Integration of general document and (meta)data repositories**

using the data for analysis or visualization, and optionally submitting any analysis results back to FairSpace. Based on this, the FairSpace team formulated and presented a few user workflows to validate assumptions and acquire feedback. During these sessions, it became clear that there were still some uncertainties to resolve. For instance, participants differed in their view of what was meant by (study) metadata and to what extent it needs to be stored in FairSpace. Also, the criteria choosing FairSpace versus other established data repositories to deposit study data were not clear. Furthermore, some possible overlap in functionality between FairSpace and the catalog component were spotted that need to be resolved. To resolve these issues and aid the discussion, a prototype instance of FairSpace will be deployed, loaded with preliminary DIME study data and sample data from a few nutritional studies. While these sessions mainly focused on requirements gathering for human users, this prototype will also serve to gather requirements for the “machine” part of FAIR for FNS-Cloud, for instance, findability of datasets by software agents such as other FNS-Cloud components.

The requirements described in this section are focused on the implementation of the FAIR guidelines in the FNS Cloud. In the following chapter, an overview of publicly available tools as well as the FairSpace application can be found

### **3. Data repository evaluation**

#### **3.1 Introduction**

The purpose of this task was to evaluate selected existing storage solutions for documents and datasets based on the requirements described in section 2, and to investigate interesting tools around data storage that can be useful for the FNS-Cloud.

As a first step, two classifications for data repositories are introduced and specified. The two classifications should help to understand purposes of data repositories and their limitations. Then, we describe how storage solutions and tools were selected. The selection of tools was executed in two steps, since publicly available repositories could be used directly from the initiation of the project, whereas applications under development (such as FairSpace) needed to reach a certain TRL before they could be leveraged as part of the FNS Cloud.

Each of the selected solutions and tools are then evaluated and a direct comparison is provided. Finally, the section closes with a conclusion.

#### **3.2 Classifications of data repositories**

The first classification differentiates file-managed datasets and software-managed datasets. File-managed datasets are normally stored in a file like an Excel worksheet or in a simple text file. Also, publications belong to this group if some data is presented. While raw data is mostly stored in separate files, a publication only contains a part of the raw data. File-managed datasets are typically smaller datasets and are typically outcomes of timely limited projects. When the project finishes, the datasets remain on an institute drive and making the datasets FAIR is often not considered. In contrast, software-managed datasets have their data stored in a data management system



## **D2.3 Integration of general document and (meta)data repositories**

and in addition software is available, like a web app, that allows users on the web to search and use data. Such systems are also called information systems in computer science while in food science the term database is often used. Information systems are normally built to manage bigger datasets where one or more organisations collaborate and maintain data over many years. Funding is typically coming from governments together with a mandate to maintain the database for the scientific community. Typical examples are national food composition databases, the UniProt database (a comprehensive, high-quality resource of protein sequence and functional information), the Chebi database (dictionary of molecular entities focused on 'small' chemical compounds) as well as Wikipedia. For these datasets, special software was developed to manage, modify, and search data.

Another classification for data repositories differentiates cloud storages and annotation systems.

Typical cloud storage services like Dropbox, Google Drive, or Microsoft OneDrive are designed to allow collaborating users to have common file shares. Because collaborating users can be spread around the world, common file shares must be in the cloud, accessible from everywhere, and the main goal is to replace the local file drive. The functionality of these tools is focused on creating, modifying and sharing files and the graphical user interface is similar to what users are familiar with in Windows Explorer. A cloud storage can be open for any user and can be used as a dataset repository. The folder and file structure make it difficult to organise a big number of datasets and search functionality is mostly limited. In contrast, annotation repositories put the main focus on the findability of datasets by utilising an annotation approach. Meta-data, called tags, are used to annotate files which describe them. Ontologies are used to create a solid structure for the meta-data schema that is used to tag documents. Tags can be used for search offering a flexible and advanced functionality. Twitter works similarly. Tweets can be assigned with hashtags and these hashtags can be used to re-find tweets. These tags or hashtags provide semantical meaning to database entries and allows users to search in a more natural manner, and also allow more sophisticated search than in cloud storage services

### **3.3 Selection of data repository tools**

The focus of this evaluation is only on file-managed datasets and not on software-managed datasets. Software-managed datasets are covered in the demonstrators in WP4 and WP5 where they will be extended and implemented. File-managed datasets are numerous and data formats are so diverse that it most probably is not possible to bring them into a database management system without modifying the format. Software-managed datasets mostly comply with the FAIR principles or comply with many of the FAIR aspects while file-managed datasets do not.

Cloud storage services provided by big commercial companies were excluded from our investigations because most users know how they work, many description and comparison reports are online available. Instead, cloud storage services that can be hosted on their own premises were selected for investigations because there are some interesting European open-source projects. The advantage of storage on own premises is the available capacities of several European organisations and the independence of commercial providers. In addition, annotation systems are also investigated with a focus on Fairspace in later sections.



**D2.3 Integration of general document and (meta)data repositories**

Therefore, the following requirements were used to select the solutions that were evaluated:

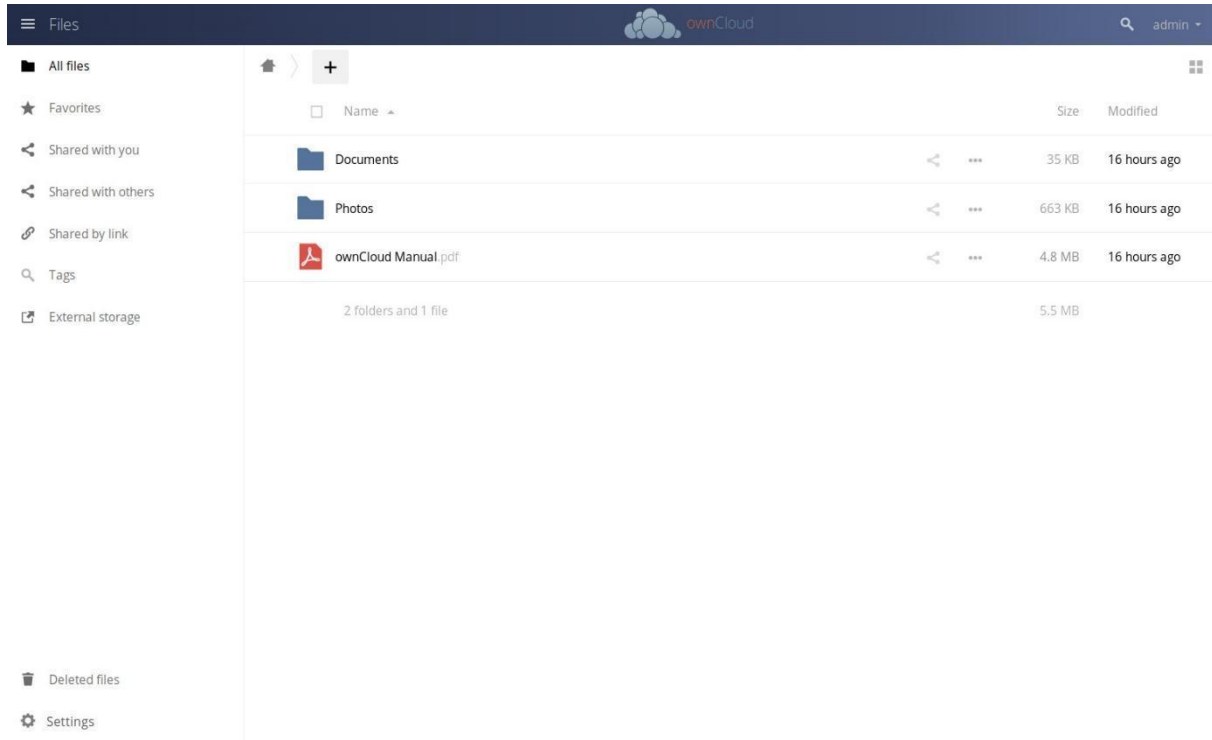
- Free to use, not being constrained by hardware or license model
- Open source
- Supported by well-established scientific organisation(s) in Europe
- Possibility to self-host data or hosted by European research infrastructures

The following tools have been identified to match the requirements and were selected for detailed evaluation:

1. OwnCloud
2. NextCloud
3. Zenodo
4. EUDAT CDI
5. Jupyter applications



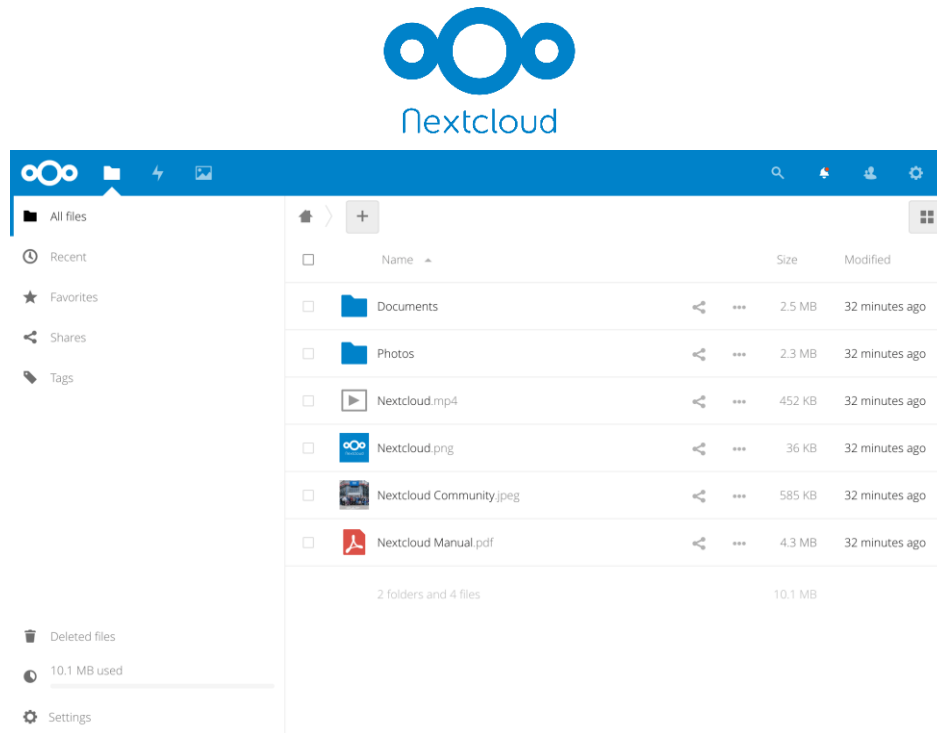
**OwnCloud / NextCloud**



OwnCloud is a file sharing platform that can be hosted on own premises or externally. OwnCloud can host file-managed datasets and is a cloud storage solution. Annotation and search are not the primary goals of this system. It has a web app which allows users to upload and download files and share them with other users. It is also possible to share files with the persons not having an account. OwnCloud has also a desktop client, similar to Dropbox or GoogleDrive. The desktop client allows integration in the system file explorer and keeps the file synchronised with the central storage. Meaning that files are stored locally allowing offline working. OwnCloud also offers mobile clients for Android and iOS to synchronise files with mobile devices. In this case, files are only stored locally if the user opens the file and no general file synchronisation is implemented.



## D2.3 Integration of general document and (meta)data repositories



NextCloud is a fork of the OwnCloud project created by the original OwnCloud author Frank Karlitschek. Hence the two platforms look similar and offer very similar features, mostly being different only in their readiness level (TRL), see section comparison. The big difference however lies in the licencing model of both applications: OwnCloud has a separate licencing model for enterprise version, that is closed-source, while NextCloud is open-source in both versions.

Pros:

- Can be self-hosted for free
- Many file access control options, access for groups, solo users, ability to share password protected files/documents
- Plugin system in place, that allows to extend the installation with more functionalities easily
- Ability to easily store, share and access any files (desktop/mobile applications)
- Federation of sharing, allowing to connect other installations of NextCloud/OpenCloud and easily share files between them
- As opposed to some commercial cloud systems (for ex. Synology Drive) does not require specialized hardware to run



## D2.3 Integration of general document and (meta)data repositories

Cons:

- No built-in backup/replication system
- Desktop/mobile clients could have more options, for example sharing files requires using the web panel

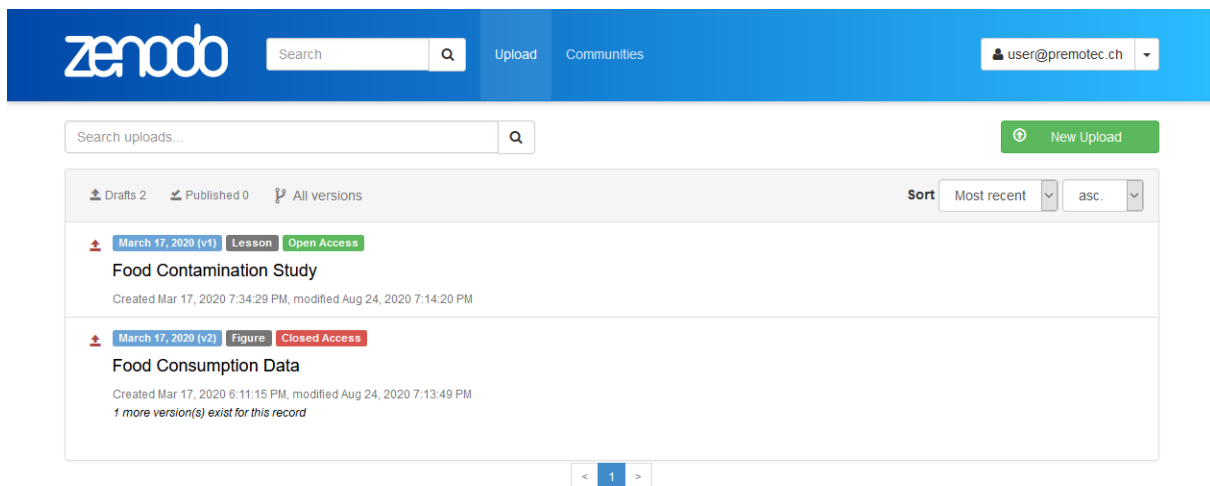
### Zenodo



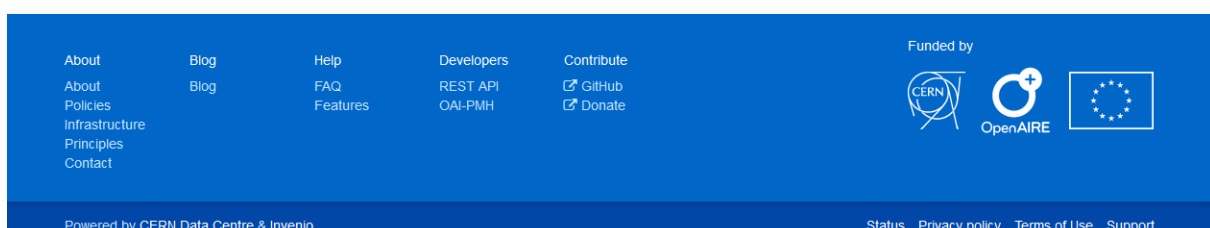
Zenodo is a web-based general purpose open-access research repository created by OpenAIRE and hosted at CERN. Zenodo can host file-managed datasets and is an annotation system. The software comes in two variants:

- Cloud based – available on [zenodo.org](https://zenodo.org)
- Self-hosted – can be downloaded from the [GitHub](https://github.com)

When using the Cloud based version, the data is hosted at CERN Data Centre and the storage size is limited to 50GB per publication (can be expanded on a case-by-case basis by directly contacting support). While self-hosting there are no such limitations (apart from the available storage space on own servers). It's also worth noting that the documentation of Zenodo states that the items in the cloud version will be retained for the lifetime of a repository, which is expected to be at least another 20 years.



The screenshot shows the Zenodo web interface. At the top, there is a navigation bar with the Zenodo logo, a search bar, and links for 'Upload' and 'Communities'. A user profile dropdown shows 'user@premotec.ch'. Below the navigation bar, there is a search bar for uploads and a 'New Upload' button. The main content area displays a list of uploads with filters for 'Drafts 2', 'Published 0', and 'All versions'. The list is sorted by 'Most recent' in ascending order. Two uploads are visible: 'Food Contamination Study' (Lesson, Open Access) and 'Food Consumption Data' (Figure, Closed Access). The 'Food Consumption Data' entry indicates that 1 more version exists for this record. A pagination bar at the bottom shows page 1 of 1.



The footer of the Zenodo website is blue and contains several sections. On the left, there is a 'About' section with links to 'About', 'Policies', 'Infrastructure', 'Principles', and 'Contact'. In the center, there are links for 'Blog', 'Help', 'Developers', and 'Contribute'. On the right, there is a 'Funded by' section with logos for CERN, OpenAIRE, and the European Union. At the bottom, there is a 'Powered by CERN Data Centre & Invenio' logo and a row of links for 'Status', 'Privacy policy', 'Terms of Use', and 'Support'.



**D2.3 Integration of general document and (meta)data repositories**

The goal of Zenodo platform is to provide researchers with a place to upload and store research artifacts regardless of scientific branch, which can be in any format or size, and then tag them with rich metadata. Metadata tags allow the publications to be searchable using an integrated search engine. The resources are also easily citable due to usage of common identifiers (DOI).

**Pros:**

- Ability to self-host, but the cloud version also offers data safety (CERN Data Centre, replication over multiple servers, regular backups<sup>3</sup>),
- Based on fairly known frameworks which are popular in industry, e.g. Invenio, which is also developed by CERN,
- Provides 4 access control levels: public, embargoed (available after a certain date), restricted (manually grant access to selected users) and closed (only the metadata is visible).

**Cons:**


- Strictly defined metadata schema,
- Forbids charging users for access to data stored on it (in the cloud version, not specified for self-hosted version).

**EUDAT CDI**



EUDAT CDI (Collaborative Data Infrastructure) is an infrastructure of integrated data services and resources with a goal of supporting research. EUDAT CDI can host file-managed datasets and is an annotation system. It's currently sustained by a network of over 20 European research organisations and data and computing centres. The main feature of this infrastructure is the ability to preserve, find, access and process data, all inside one trusted environment.









The EUDAT CDI consists of 9 separate services:

	Service	Underlying Technology
	B2ACCESS – federated cross-infrastructure authorisation and authentication infrastructure for other services.	Unity IDM

<sup>3</sup> <https://about.zenodo.org/infrastructure/>



**D2.3 Integration of general document and (meta)data repositories**

	B2DROP – cloud file storage and data exchange service for researchers and scientists to keep their data synchronised.	NextCloud
	B2SHARE – repository for storing and publishing research data sets, with functionality similar to Zenodo.	Invenio (same as Zenodo)
	B2NOTE – service for creating, storing and managing annotations (additional information) about online resources with a model based on W3C web annotation data model.	-
	B2HANDLE – distributed service for managing persistent identifiers for data hosted in EUDAT CDI.	Handle.Net Registry
	B2SAFE – service for secure long-term preservation of research data, while replicating them over several servers.	iRODS
	B2FIND – discovery / search engine service based on metadata harvested from research data collections, in particular B2SHARE and B2SAFE services and community repositories.	-
	B2STAGE – service that takes care of data staging – transferring data into and out of EUDAT High-Performance Computing workspaces.	-
	EasyDMP – simplifies the task of creating data management plans for researchers by providing plan templates and helping fill them.	-

**Pros:**

- Allows tailored metadata per community in the repository (B2SHARE) in opposite to Zenodo,
- Utilises a lot of open source/free software instead of “reinventing the wheel” (NextCloud, Invenio, Unity IDM) and connects them into one system.

**Cons:**



**D2.3 Integration of general document and (meta)data repositories**

- Limit of 20GB of storage per user or publication,
- Does not support “restricted” access in comparison to Zenodo (only community-restricted access),
- Self-hosting depends on service and requires some setup and EUDAT support,
- Visible performance issues during testing, not very user-friendly UI,
- Forbids charging users for access to data stored on it (in the cloud version, not specified for self-hosted version).



**D2.3 Integration of general document and (meta)data repositories****Fairspace**

Fairspace is a platform to support researchers in storing, sharing and analyzing data from multiple sources and to comply with the FAIR guidelines (DOI: 10.1038/sdata.2016.18). Use cases, functionalities and usability evaluation are described in detail in paragraph 3 Fairspace Development and Implementation.

**Pros:**

- Ability to self-host on premise or to be hosted in the cloud
- Allows metadata model customization and can be highly tailored to use cases and research questions
- Great flexibility in access permissions and user management (based on user role ((e.g. admin, user, etc), based on workspace)

**Cons:**

- Global metadata model might result in irrelevant metadata field for certain data sets (use case dependent)
- Forbids charging users for access to data stored on it



**D2.3 Integration of general document and (meta)data repositories**
**3.4 Comparison of common features**

	OwnCloud	NextCloud	Zenodo	EUDAT CDI	FairSpace	Jupyter
Organisation behind	ownCloud GmbH	Nextcloud GmbH	OpenAIRE / CERN	EUDAT Ltd	The Hyve	Project Jupyter
Licence	AGPLv3 / OwnCloud Commercial (enterprise)	AGPLv3+	GPL-2.0	Depending on service	will be available under open-source license	MIT
Open source	Yes / No for enterprise	Yes	Yes		WIP (planning phase)	Yes
Popularity (GIT stars)	7.2k	11.6k	454	~10-100	n.a. (since not publically available yet)	7.5k / 10.2k
Primary use	Data storage		Publication repository	Data storage, publication repository, data processing	Data storage, Data search, Metadata assignment, Access management	Managing interactive computing documents
Cloud Hosting	Paid, hosted on private servers		Free, CERN Data Center, 50GB per publication limit (can be expanded)	Free, Hosted at EUDAT partners servers, 20GB per user or publication limit	Possible	None
Self-hosting	Allowed		Allowed	Depending on service, complex and requires EUDAT support	Allowed	Allowed



**D2.3 Integration of general document and (meta)data repositories**

Versioning of files	Has a version control plugin		Files attached to publications are versioned	Files attached to publications are versioned	Is a standard feature	Versioning is possible using git integration
Access control	Per groups, per users, password protected files		public, closed, per user, embargoed	public, closed, per community, embargoed	access to collections per user or per workspace (group); read or write access per user or read access per workspace	password protected, public, private
Can charge users for access to data	Yes		No (at least not in cloud version)	No (at least not in cloud version)	No	Yes
TRL	8-9	8-9	8-9	8-9	6-7*	8-9

\*to reach level 8-9 Fairspace needs to be successfully deployed for the demonstrator and tested by key-users

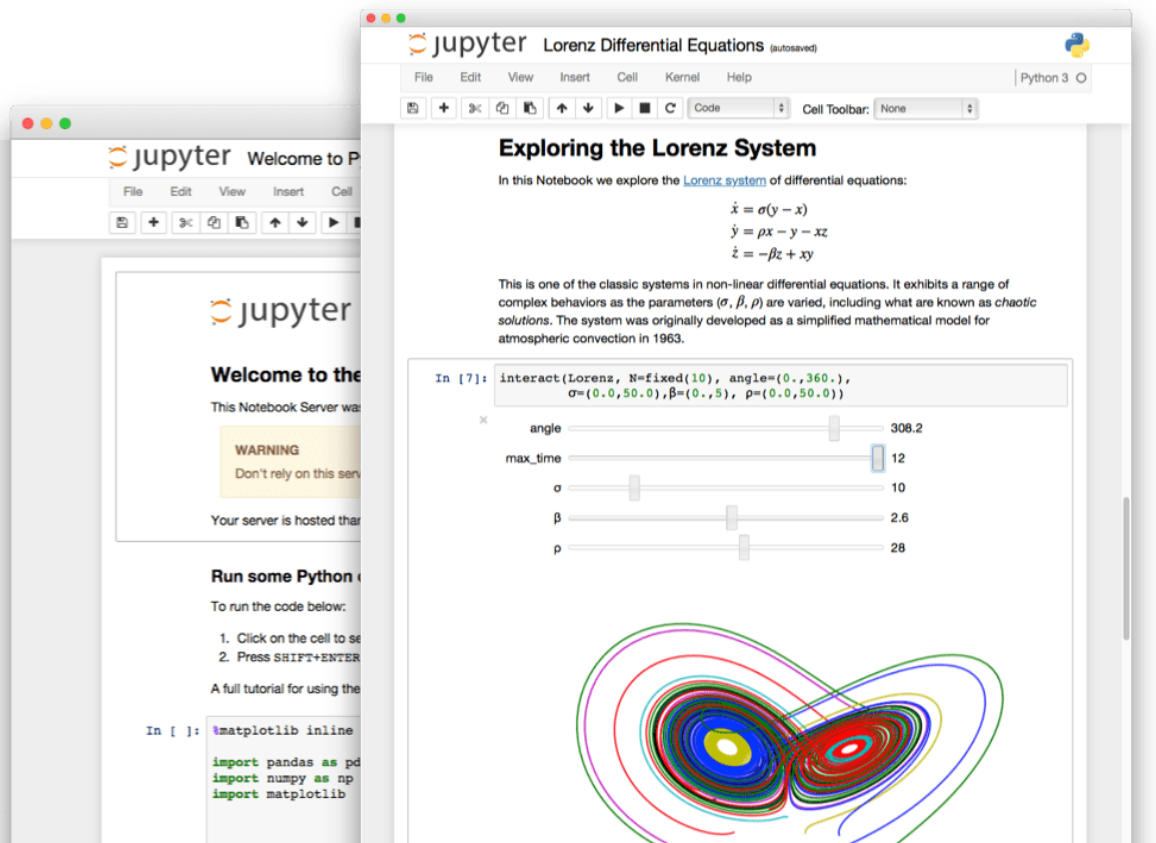
Jupyter is listed in the last column. Jupyter's primary goal is to produce documents with integrated graphics. It is therefore strictly speaking not a cloud storage solution but as side-effect to generate files, the tool also offers to store files. As Jupyter could be used to store files, it was selected into the list of tools to investigate.

**Jupyter apps**



Jupyter applications are developed by Project Jupyter, a non-profit organisation that aims to develop open-source software, standards and services for interactive computing. There are three main products that are part of those apps:

- Jupyter Notebook – an interactive computational environment for creating Jupyter notebook documents, which are advanced text files that can execute code written in many programming languages and create visualisations of results. The software also offers a directory viewer with a simple file manager, but with a very limited access control.

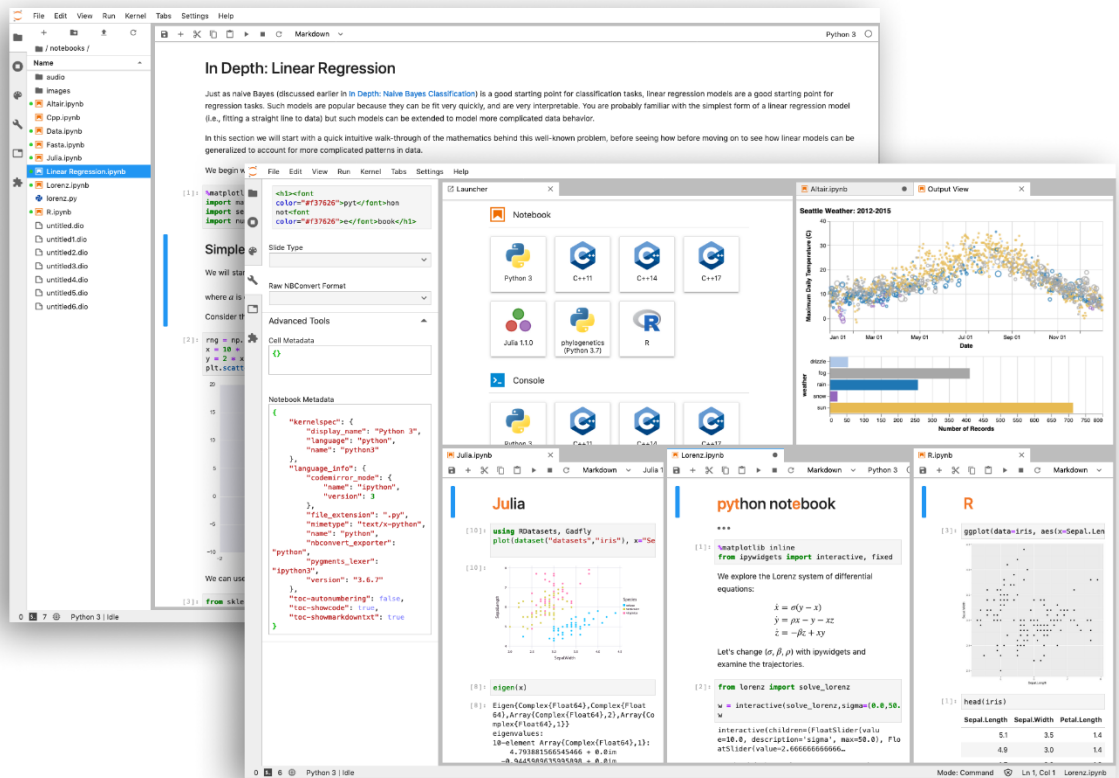


- JupyterLab – next-generation version of Jupyter Notebooks, that extends the previous version with new functionalities and a more powerful user interface.





## D2.3 Integration of general document and (meta)data repositories



- JupyterHub – multi-user server version of the Jupyter Notebook. Allows to run a shared installation for a group of users (for example a company or research institute).

### Pros:

- Interactive computing inside Notebook documents,
- Open format for Notebook documents,
- Can work together with other tools (for external cloud storage tools)

### Cons:

- Common functions (i.e., code) cannot be shared between multiple Notebooks, each Notebook is a standalone document,
- Very limited file-storage features, not suited for a storage solution.



### **3.5 Conclusions**

Through the evaluation, a lot of knowledge could be gained. The substantial contributions are the requirement gathering, the system classification and the technical specification that were developed. It gives an understanding and an overview of the different kinds of tools and helps to plan the FNS Cloud infrastructure.

For the FNS-Cloud we decided not to use EUDAT services since many of the infrastructure parts are not needed and we have our own solutions in the consortium.

Different types of data relevant to the FNS Cloud need to have appropriate repositories. In the best-case scenario, datasets for each area would have a standard format to be stored in structured databases (software-managed data), if such storage is not possible and data is stored in files, they should be stored in repositories allowing for rich metadata like FairSpace (file-managed data). If no standard metadata schema is available, the final option would be storing the data files in Zenodo, as it is well established and developed and maintained by a reliable organisation - CERN. If for any reason, the data owner does not want to use Zenodo, FNS Cloud can provide file storage using NextCloud. The decision to use the self-hosted NextCloud platform, was based on the fact that it is more popular at the moment and more actively developed than OwnCloud and also the licencing of the solutions is better in the case of NextCloud. Additionally, NextCloud is open source in both editions. We installed NextCloud as a Docker container on the PMT infrastructure and made it available for the consortium.

For file-managed datasets and annotation systems, we developed FairSpace and it is planned in the project proposal to integrate it into the FNS-Cloud. The technical specifications of FairSpace will be laid out in detail in the following chapters. Zenodo is widely supported and is currently the most used system for file-managed datasets and annotation.

## **4. FairSpace Development and Implementation**

### **4.1 Background**

After the exploration and classification of available repository options, it became clear that an additional application is needed to help improve the compliance of the FNS Cloud data with the FAIR guidelines. Therefore, we further developed FairSpace to meet the requirements described above and to facilitate the use cases identified in WP4 (with special focus on the microbiome study).

### **4.2 Related work**

The main effort for this task is software development to improve the TRL of the FairSpace application to comply with the requirements specified above. A detailed description of the technical specifications can be found in the following sections.



### 4.3 FNS-Cloud methodology

After the finalization of the Fairspace development and the implementation of the application as part of the FNS-Cloud infrastructure, we will leverage the data set map developed as part of task 2.1. High priority data sets will be selected for integration into the Fairspace application and will be loaded via an iterative approach. In task 2.2 data catalogue options have been explored and a catalogue with dataset metadata has been developed.

Furthermore, we will collaborate with WP2, WP3 and WP4 to discuss options to further integrate the solution with other tools and services developed under the framework of FNS Cloud. We will also need strong alignment with WP4 to develop processes to integrate newly created data.

### 4.4 Technical specifications

Here we specify what the Fairspace application delivers to the users, and how this supports the FNS consortium members to perform their data management tasks (see use case description in Background section).

We first describe the high-level *purpose* of the solution. This specifies how far the data management improvements are covered by the solution. Then we describe the *structure and terminology*, the *workflow and access modes* and *roles and permissions* that will be used in the remainder of the solution specification. Then we detail the *functionality*, *data models*, and *system requirements* of the proposed solution.

#### 4.4.1. Purpose

The proposal includes the central data catalogue solution (for a detailed description, see D2.2), an integrated overview of data from multiple environments, and a data repository where researchers can collaboratively create, modify and share files.

Although the different features are integrated in a single solution, we first describe its purpose as clearly distinct features. The different features require different data constraints and business rules, which need to be combined in the proposed implementation.

#### *Central data archive and catalogue*

The solution provides a *central data archive and catalogue* for depositing data, annotated with metadata, for improving the findability, accessibility and reuse of data. Data is organised in collections, collections belong to a workspace, access is managed on collection level. Collections can be shared with other workspaces or individual users. This allows projects to have their own (protected) workspace, but also enables sharing of data with other projects and users.

A central data archive and catalogue provides:

#### *Data archive and catalogue with rich metadata capabilities*

to deposit (and version) your data (collections)

where data is organised in workspaces, where data for a project can be stored, and access can be managed;



### **D2.3 Integration of general document and (meta)data repositories**

where data can be annotated with metadata that conforms to well defined vocabularies which are suited for the research domains or data types for which the catalogue is used;

where data is easy to find, either by browsing metadata (categories) linked to the data, or by their organisation in collections;

where data is easily accessible through a convenient user interface and standard APIs;

where data can be easily shared with other projects, by sharing collections with multiple workspaces, or with individual users;

where data access and changes are logged.

#### *Secure analysis working environment with access to the data archive*

where researchers can explore, visualise and analyse available data, e.g., Excel-like, or JupyterHub/RStudio like tools

where up-to-date packages and tools are available

where scripts and results can be stored and shared

#### *Publishing workflow from working environments to the data archive*

for annotating deposited files with mandatory metadata

for bulk publishing or single collection/file upload with metadata

for copying existing files and metadata from a working environment to a collection

for migrating collections in a project workspace to the data archive

#### *Project collaboration environment*

We want to provide researchers with a working environment, where researchers can securely store, edit and share data, for improving collaboration on work in progress.

A *project collaboration environment* provides a dynamic working environment, where researchers can collaboratively edit data in a secure environment, share mutable files and organise work in progress in a central environment.

This provides a central storage for project data/project workspaces, both accessible via an API and via a user interface to navigate files, share files and edit files. The environment allows multiple users to edit the same file, therefore there needs to be measures to prevent data corruption caused by simultaneous editing, such as versioning of files and the possibility to undelete files.

The environment has a secure storage, i.e., include audit logging, versioning, and access control mechanisms. However, the environment allows files to be overwritten and deleted, which makes this purpose different from the data archiving purpose (mainly targeted at archiving data, to enable reproducibility).

#### *Integrated overview*

Users need an overview of data from different sources with different dynamics: versioned and annotated data in an archive that is available for all users, and dynamic project environments



## D2.3 Integration of general document and (meta)data repositories

with project-based access restrictions. With this combination, you could find relevant published data and ongoing projects related to a particular sample or research participant.

Such an overview requires a shared data model between the data repository and the project workspaces. This is achieved by storing all metadata in the central data catalogue.

Explicit references to external systems, data sets, directories or files should be added to the catalogue first. This reference may become out of sync, e.g., when the referenced file is changed or deleted. Whenever the catalogue notices that the file has changed (e.g., by the timestamp of last change) or is deleted, this may be added as a property to the reference.

The *integrated overview* provides:

Overviews of available data, either published in a central repository or in restricted project specific workspaces.

Data in the central data repository can be directly annotated

For data in external, possibly volatile, storages, explicit references should be added to the data catalogue first.

Browse and access data from heterogeneous source systems through a common WebDAV API, hence also available in the analysis environment and file browser user interface.

### 4.4.2 Structure and terminology

We define the roles and entities of the proposed solution. This sets the stage for the following topics and provides us with a common terminology to describe the functionality of the solution.

We define the following core entities of the data repository:

*Users*: individual users in the organisation, looking for data, contributing to data collections or managing data.

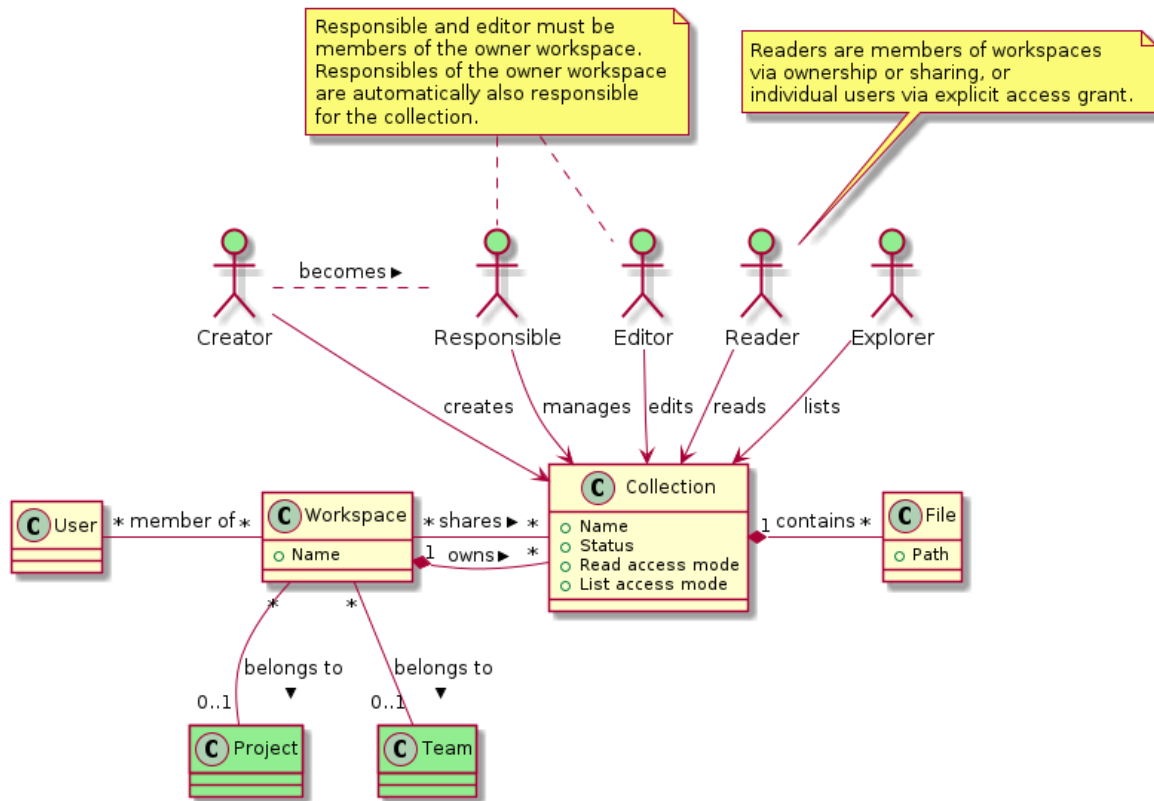
*Workspaces* (for projects, teams): entities in the system to organise data collections and data access.

*Collections*: entities in the system to group data files. These are the minimal units of data for data access and data modification rules.

*Files*: The smallest units of data that the system processes. Files always belong to a single collection. Files can be added, changed and deleted, but not in all collection states. Changing a file creates a new version. Access to a file is based on access to the collection the file belongs to. Files can be organised in *Directories*, which we will leave out of most descriptions for brevity.



**D2.3 Integration of general document and (meta)data repositories**



The diagram above sketches the relevant entities and actors. The basic structure consists of users, workspaces, collections and files as represented in the system. The organisational entities of projects and teams do not play a role in the access rules, but are important for understanding if the proposed model works in reality.

Collections are the basic units for data access management. A collection is owned by a workspace. The responsibility for a collection is organised via the owner workspace: members of the owner workspace can be assigned as editors or managers of the collection.

This reflects the situation where in an organisation, a data collection belongs to a project or a research team. This way the workspace represents the organisational unit that is responsible for a number of data collections.

Data can be shared with other workspaces or individual users (for reading) and ownership may be transferred to another workspace (e.g., in the case the workspace is temporary or the organisation changes).

The *data catalogue* contains all the metadata, which is visible for all users with catalogue access. Users with write access can add metadata to the catalogue. Data stewards can edit contributions.

Metadata on collection and file level is protected by the access policy of the collections.

*User administration* is organised in an external component: Keycloak.

A back-end application is responsible for storing the data and metadata, and for providing APIs for securely retrieving and adding data and metadata using standard data formats. A user interface application provides an interactive file manager and (meta)data browser and data entry forms based on the back-end APIs.



## D2.3 Integration of general document and (meta)data repositories

Besides the data storage and data management, the solution offers *analysis environments* using Jupyter Hub. In Jupyter Hub, the data repository is accessible. Every user has a private working directory.

We make no assumptions on the structure of the data or the permissions of the external file systems that are connected to the data repository and referenced in the data catalogue. The organisation structure may be replicated in the different systems in incompatible ways and the permissions may not be aligned.

### 4.4.3 Workflow and access modes

During the lifetime of a collection, different rules may be applicable for data modification and data access. We propose a workflow with the following statuses:

*Editing*: for the phase of data collection, data production and data processing;

*Archived*: for when the data set is complete and is available for reuse;

*Closed*: for when the data set should not be available for reading, but still needs to be preserved;

*Deleted*: for when the data set needs to be permanently made unavailable.

In these different statuses, different actions on the data are enabled or disabled. Also, visibility of the data and linked metadata depends partly on the collection status.

We also distinguish three access modes for reading and listing files in a collection (where listing also includes seeing the metadata):

*Restricted*: only access to a explicitly selected workspaces and users

*Public metadata*: the collection and its files are visible, metadata linked to them is visible for all users

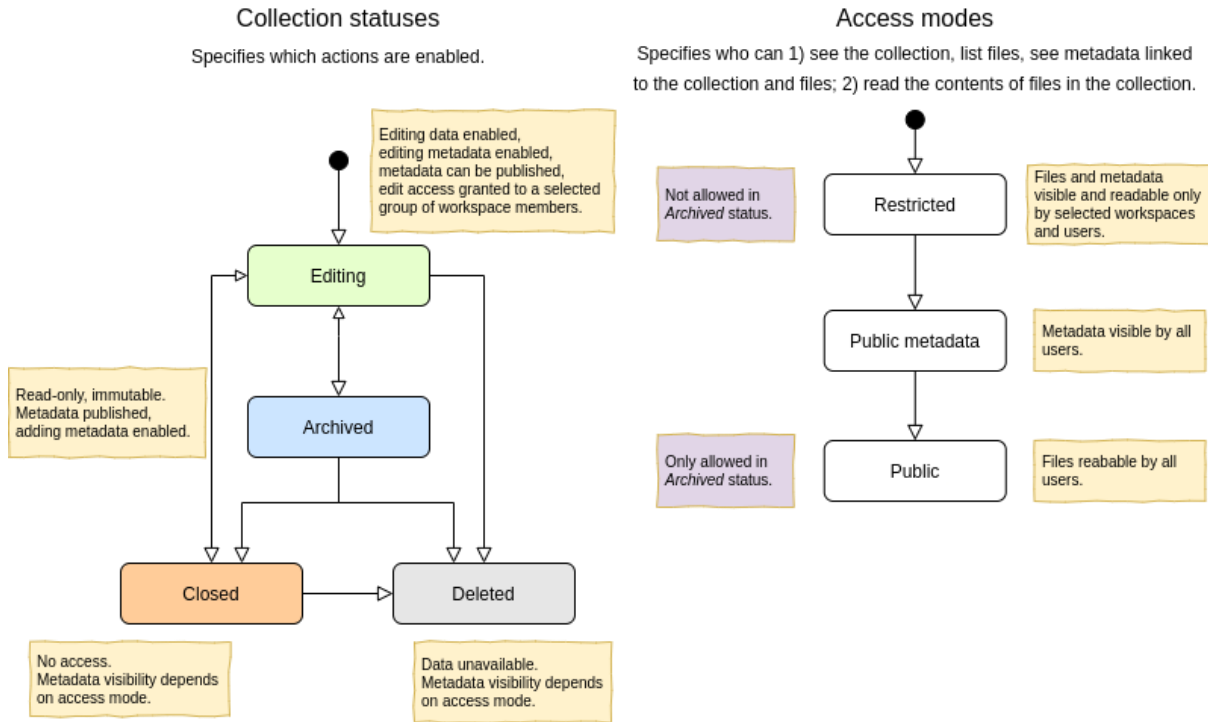
*Public*: the files in the collection are readable for all users



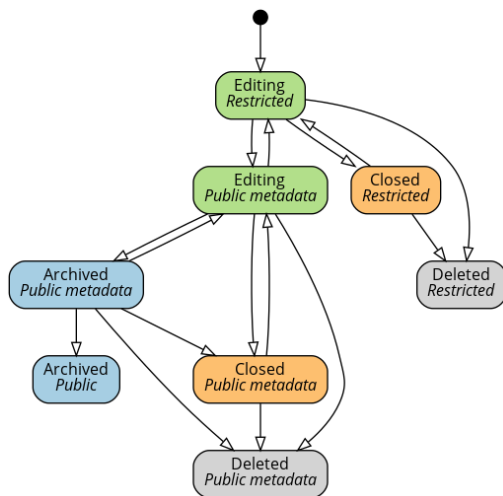


**D2.3 Integration of general document and (meta)data repositories**

*Collection editing and publication workflow*



Combined in one status diagram, that yields the following statuses and interactions:



**4.4.4. Roles and permissions**

We distinguish the following roles in the solution:

*Administrator:* can create workspaces, assign roles and permissions (Management office)

*Data steward:* can edit and delete metadata, overwrite and delete collections and files (Management office)

*Internal user:* any internal user can view public metadata, workspaces, collections and files (Researchers, data scientists, bioinformaticians).





## D2.3 Integration of general document and (meta)data repositories

Workspaces are used to organise collections in a hierarchy. On workspace level there are two access levels:

*Manager*: can edit workspace details, manage workspace access and manage access to all collections that belong to the workspace.

*Member*: can create a collection in the workspace.

Access to collections and files is managed on collection level. We distinguish the following access levels on collections:

*List*: see collection, directory and file names and metadata properties/relations.

*Read*: read file contents.

*Edit*: add files, add new file versions, mark files as deleted.

*Manage*: grant, revoke access to the collection, change collection status and modes.

Access levels are hierarchical: the *Read* level includes the *List* level; the *Edit* level includes *Read* level; the *Manage* level includes *Edit* and *Read* level access. The user that creates the collection gets *Manage* access.

### 4.4.5 Functionality

Here we give a detailed description of the functionality the solution will provide to facilitate the efforts of the FNS-Cloud project.

*Data repository and catalogue with rich metadata capabilities*

#### a) Repository for editing and archiving collections

The repository consists of collections, where users can add, update and annotate files. A collection belongs to a workspace. A workspace manager can manage membership of the workspace and access to the collections that belong to the workspace. Access to collections can be granted to workspaces (workspace members) or individual users.

The repository has the following functions:

- Create a workspace
- Edit workspace description
- Manage membership of the workspace: a workspace manager can add users as manager or regular user to the workspace or remove members from the workspace.
- Add a new collection to a workspace
- Transfer ownership of a collection (to another workspace)
- Browse workspaces and collections

On a collection level the repository has the following functions:

- Edit collection description and metadata.
- Manage collection status and access level



### D2.3 Integration of general document and (meta)data repositories

- Manage access: grant or revoke access to workspaces (workspace members) or individual users, grant or revoke edit or manage access to members of the owner workspace.
- Create folders, upload files.
- Edit folder and file description and metadata.
- Versioning:
  - Add new file versions
  - List and download previous versions of a file
- Deletion:
  - Delete a file (mark as deleted)
  - List deleted files
  - Restore a deleted file
  - Permanently delete a deleted file: a permanently deleted file disappears from the list of deleted files and cannot be restored.
- Access collections via standard protocol (WebDAV)
- Persistent identifiers for archived data: A file gets an identifier that can be referenced from other entities and externally. This also means that when a file is deleted in archive mode, its identifier and associated metadata remain.
- File viewers and editors are not included in the proposed functionality.
- Note that not all actions are possible in all collection statuses or access modes.
- For archived collections that are available for referencing (in documentation, scripts or elsewhere), it is essential that no changes are made to them (identifier, structure, files, metadata) unless strictly necessary (e.g., for legal reasons).
- External file storage proxy
- Proxy for external storage systems, which is also available via WebDAV.

#### Catalogue

- The catalogue contains metadata entities, conforming to a data model that is maintained by the data stewards. The workspaces, collections and files of the repository are automatically represented by metadata entities in the catalogue and can be linked to other metadata.

The catalogue has the following functions:

- Create and edit metadata entities
- Manual metadata entry through the user interface
- Uploading metadata via an API
- Annotate collections and files with suitable metadata values and link to metadata entities
- Link collections or files to other versions or related content
- Ensure consistency and usability of the metadata:
  - Automatic validation of metadata (types, cardinalities, mandatory fields, coding system, etc.; based on constraints in the data model)
  - Enable safe evolution of the data model (only adding new features, not changing or removing types of metadata)
  - Restrict changes to the data model to data stewards or administrators



## **D2.3 Integration of general document and (meta)data repositories**

- Possibly restrict loading of certain entities to automated pipelines, data stewards or administrators (e.g., automatically populated research participant or sample identifiers or externally managed entities such as projects)
- Unique persistent identifiers: prevent dead links, data duplication (same sample represented with different identifiers) and identifier collisions (using the same identifier for different entities).  
Besides these unique identifiers, it should be possible to link to other identifiers, e.g., from different source systems.
- Search and explore metadata and files via convenient overviews
- Integration with existing data sources via explicit references to external collections, directories and files. The identifier of the external storage should match the identifier used in the data repository for the data proxy.

### **Search and overviews**

Researchers should be able to find samples similar to the samples they are working on or find research participants similar to research participants that participate in a study. To explore the data in a convenient way, most relevant properties need to be displayed in overviews and it should be possible to filter on those properties.

Basically, the overviews and search will provide faceted search, providing summaries of the available data and enabling easy and fast filtering based on a selection of relevant properties.

To guide the users in their exploration, it is good to provide separate overviews for the different entity types, such as research participants, samples, collections, etc. The collections can be browsed as a directory structure. Other entities are displayed in a tabular form.

### **Configure overviews and search**

Because the data model is flexible, also the entities and properties that appear in overviews and can be used for filtering and search are flexible.

The overviews will have the following configuration options:

- Select entity types for inclusion in the overviews
- Specify which properties, references and properties of referenced entities are included and how they can be used in filters (e.g., using autocomplete for selecting values).

This allows the organisation to configure project, research participant, sample and analysis overviews, including filtering (search), in the user interface. A file browser to navigate collections and files is available by default.

It is advisable to configure the search behaviour as part of the system configuration, to enable testing if the configured setup works as intended, both functionally and performance-wise. This means that the overviews and available data types are not configurable at run time by data stewards.

However, users can choose to hide or reveal certain columns in the overviews, i.e., to include columns that are only relevant for certain groups or exclude irrelevant fields.

### **Editing metadata**



## D2.3 Integration of general document and (meta)data repositories

For editing entities that can be linked to data, there needs to be an easy-to-use editor that enables editing all properties that are specified in the vocabulary and validates all the data in the form. For properties that link to other entities, the form should offer a way to select and search existing entities.

For the metadata to be usable by others, it is important to prevent users to accidentally edit entities that are not supposed to be changed after importing (e.g., research participant or sample characteristics).

### Security

(meta)data access restrictions, auditing and logging

Data loss prevention

Measures to prevent data loss by accidental user action:

Metadata can be added by any user with write permission but modifying or deleting can only be done by data stewards.

For collections:

Overwriting an existing file creates a new version, the history of previous versions is maintained

Previous versions can be restored

Deleted files can be viewed and restored in a list of deleted files

Permanently deleting files requires two actions: deleting it and then permanently deleting it from the list of deleted files

Files can only be added, modified or deleted in collections in *Editing* mode

Besides these restrictions to user actions, data is secured by executing actions in a transactional way, committing changes only when all steps are successful. Data is always appended to the underlying storage and transaction logs are stored.

*Jupyter Hub (see 2.9 Jupyter apps)*

Interactive environment (incl. Jupyter Hub) for running and storing scripts (data analysis) or bulk uploads.

The environment provides at least support for R and Python and has a range of analysis packages pre-installed.

Each user has a private working directory in the environment.

The data repository is available via the file system.

### *Data ingestion*

Large amounts of food and nutrition data (in a number of files, and their volume) need to be added to the system, as well as metadata. These (meta)data is expected to originate from different source systems. To make it easier to add data, we propose a combination of APIs, software libraries and command line tools that are easy to integrate in a data loading pipeline and easy to use interactively.

Features:



**D2.3 Integration of general document and (meta)data repositories**

Creating a collection

Adding files to a collection

Annotating uploaded files with metadata using an easy to edit format

*Access rules*

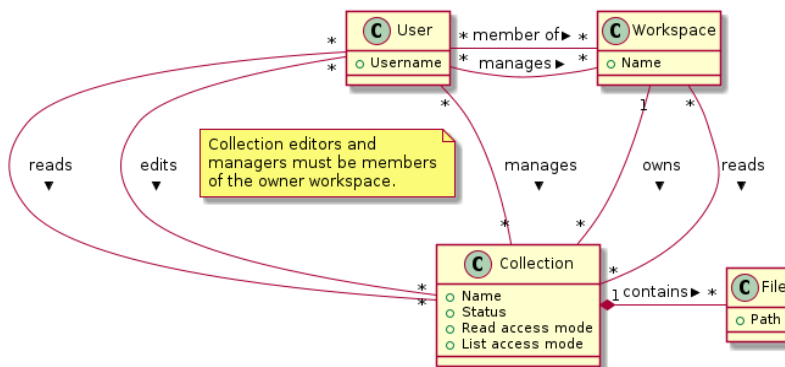
How the roles permissions affect what users can do in the different collection statuses, and what metadata they can see, is determined by the access rules of the solution.

**Workspace level access with collection level groups**

Within a project, different groups of users may require different levels of access to collections that belong to the project workspace.

E.g, some collections may be generated by pipelines and should not be edited by project members at all. Some collections may be edited by subgroups formed around the type of data (sequencing data, clinical data) or different phases (data collection, data analysis).

To enable granting specific subgroups editing access to collections, the access levels need to be assigned to individual users, rather than to an entire workspace. Similarly for managing access.



Therefore, *manage* and *edit* access is granted explicitly to individual members of the owner workspace.

*Read* access is granted to workspaces or individual users. This way, collections can be shared for reading with arbitrary groups of users, but possibly destructive actions are limited to the owner workspace.

By default, the owner workspace has no read access to the collection, but this can be easily granted, just as for other workspaces.

The workspace manager is automatically also the manager of all collections in the workspace. This prevents having unmanaged collections when the only collection manager leaves the owner workspace.

**Collection access rules**

On a collection *C*, owned by workspace *W*, a user *U* has access level:

*Manage* if:

*U* has *manage* permission on *W*; or

*U* is member of *W* and has *manage* permission on *C*;

*Edit* if: *C* is in status *Editing* and *U* is member of *W* and has *edit* permission on *C*;



### D2.3 Integration of general document and (meta)data repositories

*Read if:*

*C* is in status *Archived* and *C* has access mode *Public*; or

*C* is in status *Archived* or *Editing* and *U* has *read* permission on *C*; or

*C* is in status *Archived* or *Editing* and *U* is member of a workspace *W'* with *read* permission on *C*.

*List if:*

*C* has access mode *Public*.

If the user has no access level based on these rules, the user has no access to the collection.

Access rule

The user can only see properties of workspace, collection and file entities, belonging to or shared with workspaces that the user is a member of.

*Data models*

Here we briefly describe the data elements in the data model of the solution.

There are a limited number of core entity types in the solution:

*Workspace:* A workspace consists of a number of (shared) collections and has members with a particular level of access to the workspace.

*Collection:* A collection consists of files and belongs to a workspace but can be shared with other workspaces.

*File:* Files belong to a collection.

*User:* A user can be a member of a workspace with an access level that determines if the user can only read or also add data in a workspace.

Besides these core types, other types can be defined by vocabularies, enabling users to add properties and links to other entities to workspaces, collections and files. This is the flexible part of the data model that enables organisations to tailor the data model to the relevant data domains, while enforcing consistency on the allowed metadata.

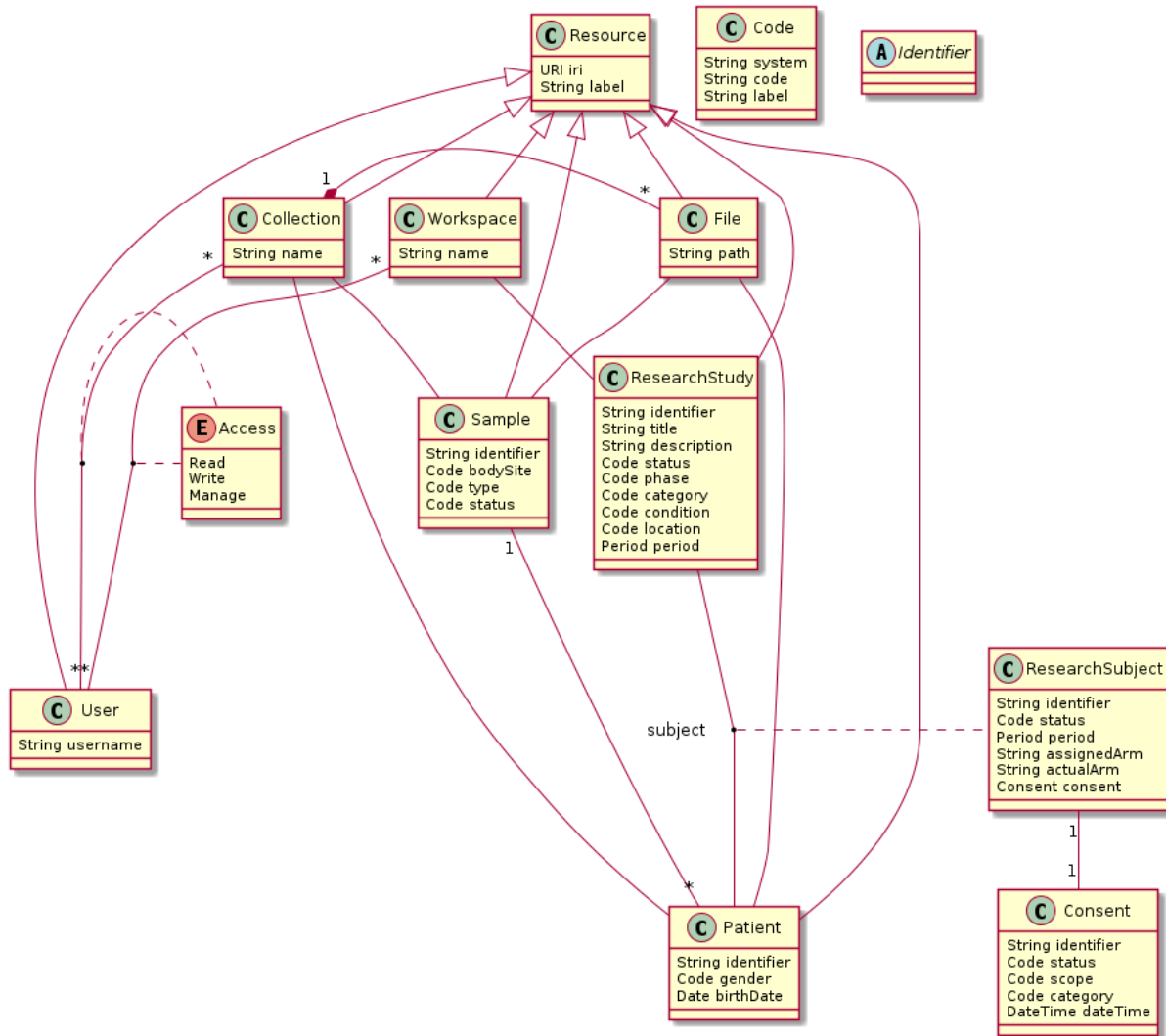
This flexible part could also include a generic way to add key-value or tag annotations to data, to enable project specific annotations without changing vocabularies.

We show an example instantiated data model with custom entities: research participant, Sample, ResearchStudy, research participant and Consent. These classes are simplified versions of the respective resources in FHIR.



**D2.3 Integration of general document and (meta)data repositories**

Example instantiated data model for Fairspace



*System requirements*

**Security, reliability**

Audit logging

Data loss prevention (caused by accidental or malicious deletion of data or hardware or software failures)

Security

**Logging**

Audit logging

**Deployment and delivery**

Setup and documentation

**User interface tests**

The User interface tests are not specified in detail yet. The test plan will be finalized during the project.



## 4.5 Fairspace technology readiness level (TRL)

When the FNS Cloud project started, a prototype version of Fairspace had been developed by The Hyve and was available to the consortium. This version of Fairspace was rated with a TRL between 5 and 6. During the start phase of the FNS-Cloud project, the application has been re-evaluated with respect to initial requirements and functionalities, focusing on the ones which are critical to the success of the application and the impact for the FNS-Cloud project. This process resulted in the identification of several conflicting requirements and technical challenges.

The following functionalities have been improved to reach a TRL of 6-7 with an improved application architecture:

### **Permission model:**

The permission model for access rights needed to be improved to allow more granular distinction between the permission of individual roles concerning both metadata and actual data. The earlier version of Fairspace did not allow this granular definition of roles. Especially the assignment of read and/or write permissions concerning metadata elements was an important aspect that needed improvement.

### **Workspaces:**

In the earlier version of Fairspace, we identified a conflict in the setup of the environments. The main conflict is due to the requirement of a) a controlled environment in which static (non-changing) files can be searched for and worked with and b) a mutable environment in which files can be constantly updated and processed. This was a very complex problem, which required a well thought through technical solution. Here we also implemented version control features.

### **Collaboration with external partners:**

To ensure potential addition of new partners in the FNS Cloud ecosystem and to secure sustainability of services and applications, we believe that collaboration with external partners is a crucial element. Therefore, we implemented options to allow easy and secure collaborations.

The developments described above were absolutely necessary to bring the Fairspace application to higher TRL and to greatly enhance its value to the consortium. Since the development of such an application is highly complex, a big fraction of the hours has been spent on technical design and architecture.

We are convinced that the deployment of Fairspace will help the consortium in at least two ways:

- Deployment of Fairspace at partner organisations that do not have a dedicated solution for data management
- Support for specific use cases (e.g. microbiome (FL4-DIME) study in WP5, in which research data from multiple streams will be integrated)

To fully reach TRL 8-9 Fairspace needs to be successfully deployed for the demonstrator and tested by key-users.





## 5. Usability testing

This section describes the process and results of the Fairspace usability testing.

### 5.1 Process

The Hyve provided a remote Fairspace demonstration during which we collected user feedback from WP4 and WP5 task leaders. The following aspects have been demonstrated and user feedback was collected to evaluate usability:

#### Workspaces

- Create a new workspace
- Manage members of a workspace

#### Collections

- Create a new collection
- Upload files to the collection
- Create a directory in the collection
- Annotate collection, directories and files with metadata
- Share a collection with another user
- Delete a collection
- Add metadata to collections

### 5.2 Results

#### General feedback:

The main idea is to use Fairspace for one (or more) studies in WP4. A good example would be the microbiome study executed by Maria Traka (QIB), since this study contains a vast variety of different samples and therefore different data sets in different formats (e.g., data on diet, glucose levels, physical activity, etc). This is an optimal use case for Fairspace since the underlying global metadata model will allow researchers to integrate data from multiple sources and allow collaborative analysis.

The question arose whether Fairspace can also be used for existing data (as opposed to data that will be gathered in future studies), such as food consumption or food composition data. Since most of the existing data is stored in structured databases, it was decided that Fairspace will be leveraged only for studies that will be executed in the framework of the FNS Cloud project.

It was also pointed out that there are several layers of data storage and queriability. Most of the food and nutrition data sets is currently stored in structured databases (e.g., composition data) and the main tools with online apps (such as foodEXplorer) leveraged in WP4 make use



### **D2.3 Integration of general document and (meta)data repositories**

of this to allow querying on record basis. FairSpace operates on a higher level, where multiple data sets can be integrated and users can search for a record based on the metadata labels. Since FairSpace focuses more on integration of data from multiple sources, the query function focuses on the metadata layer rather than the individual data record.

A concern was raised about the fact that FairSpace supports one global metadata model. Advantages of this approach are a reduced redundancy (e.g. descriptions of properties) and the possibility to integrate data from multiple sources through the adding metadata labels to collections of data sets. However, as seen in D2.1, the data sets in the fields of food, nutrition and security can be diverse and do not necessarily have common ontologies or metadata standards. In this case the global metadata model can be less optimal, since there can be irrelevant data fields which are not applicable for all data sets. WP3 is exploring options for the creation of a specific FNS Cloud model, the FNS Ontology.

#### **Feedback on workspaces/collections and access permission management:**

The data access and data sharing options provided through the implementation of specific workspaces and collections fit the requirements of the consortium well. Interviews with data owners as part of T4.1 have made it clear that flexibility and customization options for access permissions based on users as well as workspaces are seen as essential. Having a clear system in place to manage accessibility will ultimately ensure a higher level of compliance to the FAIR guidelines, since data owners oversee who can assess the data.

#### **Feedback on data preservation options:**

It became clear that it would be desirable to connect FairSpace with publishing tools such as Zenodo, where Zenodo can be used to publish research articles and support open science, while the underlying data sets can be stored in FairSpace, where the access permissions can be managed much more granularly (also see D3.2).

#### **Feedback on Analysis using Jupyter notebooks:**

Jupyter notebooks offer great opportunities for shared data analysis and visualisation. Multiple organisations linked to the EOSC see a lot of potential in this application. Currently it seems that most researchers in the fields of food, nutrition and security prefer using SPSS and R which also provide a wide range of graphics and visualisations.



## 6. Conclusions

This deliverable covers two important aspects of the FNS Cloud infrastructure. First, we evaluated several existing and publically available (meta)data repositories based on multiple requirements which have been identified as important for the consortium. Second, we described the development of the FairSpace application and we gathered feedback on FairSpace usability.

The first evaluation led to the selection of the self-hosted NextCloud platform as general file storage. A NextCloud instance is currently hosted in PMT infrastructure and has been made available for the FNS-Cloud project. For file-managed datasets, metadata annotations and integration of multiple datasets, FairSpace can be used by consortium partners. Based on feedback collected from potential key users from WP4 and, we identified a number of studies for which FairSpace can be a suitable option to store data, annotate data and allow shared data analysis. The most obvious example is the microbiome study in WP5. The requirements gathering efforts for this study are described in 2.2. We are currently implementing FairSpace for this demonstrator to undergo a second round of usability testing and to explore the optimal use of the FairSpace application that will bring the biggest benefit for the FNS-Cloud project.

In the coming months, we will focus on the following aspects: (A) Clarification of use cases and studies for different (meta)data repositories and applications, (B) Implementation and deployment of repository solutions (centrally and/ or on organisation level), (C) further development of FairSpace application and publication under open-source license.

