ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

NEUMANN JÁNOS
FACULTY OF INFORMATICS

DEGREE PROJECT

OE-NIK Student's name: **KŐHEGYI, MARCELL**
2022 Student's registration number: **T/005523/FI12904/N**

Óbudai University
John von Neumann Faculty of Informatics
Institute for Cyber-physical Systems

THESIS WORK SPECIFICATION

Name of the student:	**Marcell Kőhegyi**
Identification number of the student:	T/005523/FI12904/N
Neptun's code:	[redacted]

Title of the thesis:

Comparative analysis of data services provided by the European Open Science Cloud
Európai Nyílt Tudományos Felhő (EOSC) adat szolgáltatásainak összehasonlító vizsgálata

Internal supervisor	Dr. habil. Róbert Lovas
External supervisor:	Dr. József Kovács (SZTAKI)
Deadline:	15 December 2020
Subjects of the final exam:	Computer Architectures Cloud service technologies and IT security specialization

The task:

The European Open Science Cloud initiative (EOSC) aims to build a large infrastructure to support and develop open science in Europe and beyond. Open science is defined as a new approach to scientific progress based on sharing all available knowledge using new collaborative tools and digital technologies. The EOSC evolves towards a public funded science to be more accessible, transparent, collaborative and closer to citizens. The EOSC infrastructure contains numerous data handling related services for staging, sharing, transferring, archiving and so on.
The task of the student is to perform an in-depth analysis of the available and accessible data services of EOSC in order to give a detailed comparative study. Beyond the analyses, the data services should be investigated from different approaches (functionality, useability, accessibility, performance, etc.). The student should design and develop an environment for demonstrating data services, which are selected to be potentially utilized by SZTAKI during the development of services in the NEANIAS European Project.

The thesis must contain:

- Introduction and overview of the European Open Science Cloud
- Short overview of concept, purpose and technological background of data handling in general in EOSC
- Categorization of available data services from different approaches
- Detailed comparative study of the data services
- Design plan of the demonstration environment for testing data services
- Demonstration and measurement of selected data service(s)
- Conclusions based on collected information on EOSC data services

Place of stamp

.....
Dr. habil. Róbert Lovas
head of institute

This specification is valid until: **15 December 2022**
(According to OE TVSz 55.§)

I consider the thesis suitable for submission:

.....
external supervisor

.....
internal supervisor

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Neumann János Faculty of Informatics

STUDENT'S DECLARATION

I the undersigned student hereby declare that this degree project is the result of my own work; I have disclosed the references and tools used in an identifiable manner. The results indicated in my degree project completed may be used by the university and the institution announcing the task for their own purposes free of charge.

Dated: Budapest, 2022. 05. 15.

.....

student's signature

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

John von Neumann Faculty of Informatics

Log of consultations

Name of the student: MARCELL KŐHEGYI

Neptun code: [redacted]

Program: COMPUTER ENGINEERING (BSc)

Phone number: [redacted]

Address: [redacted]

Title of the thesis work:

COMPARATIVE ANALYSIS OF DATA SERVICES PROVIDED BY THE EUROPEAN OPEN SCIENCE CLOUD

Internal supervisor: DR. HABIL. RÓBERT LOVAS

External supervisor: DR. JÓZSEF KOVÁCS

Please use BLOCK CAPITAL LETTERS to fill in this form!

Ocassion	Date	Topics discussed	Signature
1.	2020.04.08	DISCUSSION ABOUT THE TOPIC OF THE THESIS, TASK DESCRIPTION AND THE FIRST MILESTONE	JL
2.	2020.04.20	DISCUSSION ABOUT THE FIRST MILE-STONE (CATEGORIZATION)	JL
3.	2020.05.15	OVERVIEW OF CONTENT AND FORMAL REQUIREMENTS	JL
e4.	2020.05.20	OVERVIEW OF CONTENT AND FORMAL REQUIREMENTS	JL

This log should be signed on each consultation (by a supervisor), 4 times altogether.

The student fulfilled the requirements of the Thesis work subject, I allow his/her participation on the presentation/defence[1].

JL

.....

Internal supervisor

Budapest, 2020.05.20

[1] Underline the appropriate one.

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Neumann János Faculty of Informatics

CONSULTATION LOG

Student's name:	Neptun code:	Specialty:
MARCELL KŐHEGYI	[redacted]	COMPUTER ENGINEERING(Bsc)

Phone number: Mailing address (e.g. domicile):

[redacted]

Degree project / thesis[1] title in Hungarian:

EURÓPAI NYÍLT TUDOMÁNYOS FELSŐ (EOSC) ADAT SZOLGÁLTATÁSAINAK ÖSSZEHASONLÍTÓ VIZSGÁLATA

Degree project / thesis[2] title in English:

COMPARATIVE ANALYSIS OF DATA SERVICES PROVIDED BY THE EUROPEAN OPEN SCIENCE CLOUD

Internal supervisor:	External supervisor:
DR. HABIL. RÓBERT LOVAS	DR. JÓZSEF KOVÁCS

Please write the data in printed capital letters!

Occ.	Date	Content	Signature
1.	2022.04.06.	DISCUSSION OF EXISTING CONTENT, GOALS	
2.	2022.04.14.	DISCUSSION OF NEW CONTENTS	
3.	2022.04.22.	DISCUSSION OF CONTENTS, REQUIREMENTS	
4.	2022.05.06.	DISCUSSION OF CONTENTS, REQUIREMENTS	

The Consultation log is required to be countersigned by either supervisor on a total of 4 occasions of consultation.

The student has complied with the requirements of the subject titled Degree project I / Degree project II (BSc) or Thesis 1 / Thesis 2 / Thesis 3 / Thesis 4[3], and is allowed to proceed for reporting / defense[4].

Dated: Budapest, 2022. 05. 15.

.....
Internal supervisor

[1] Underline as appropriate
[2] Underline as appropriate
[3] Underline as appropriate
[4] Underline as appropriate

Abstract

In this paper I introduce EOSC, the European Open Science Cloud, a gateway to find information and resources that helps in carrying out scientific research. We also explore how Open Science and FAIR data principles can work together to help scientific research for both funded researchers, and citizen researchers.

All currently available EOSC data services have been individually categorized and studied, in order to gain an overview of the functionality, usage and intended targets of the services, which are potentially utilized in aiding SZTAKI (Institute for Computer Science and Control) to contribute to the NEANIAS (Novel EOSC Services for Emerging Atmosphere, Underwater & Space Challenges) [1] project.

After categorization, I have reviewed some services from the FAIR data principle's standpoint, then I've chosen services for which I created a python application in order to demonstrate how a data service can be used, that are found in EOSC and adheres to Open Science and FAIR data principles. The application provides functions which can be extended by other developers either by building on top of the application's functions, or by integrating its functions into an application of their own.

Table of contents

1 Introduction and overview of the European Open Science Cloud

In the age of the 21st century information boom, both the scientific community and a general public are on the path from leaving a data-sparse existence to having a data-saturated one. This comes with the excellent opportunity for creation of new inventions and for having new discoveries. However, the handling and consumption of all of this new, immense amount of data needs to be not forgotten. In order to continue and expand research with a higher rate of efficiency, new systems and tools are needed to be developed.

This does not apply exclusively to the already established scientific community, but the untapped resources of the "civilian" researchers as well. These new systems and tools are ought to enable any citizen, be it European or outside of Europe, to support ongoing research activities, and apply their own input to scientific advancements.

1.1 EOSC

The European Open Science Cloud (further on, referred to as: EOSC)is an environment for hosting and processing research data to support EU science. Open science is defined as a new approach to scientific progress based on sharing all available knowledge using new collaborative tools and digital technologies. The EOSC evolves towards a public funded science to be more accessible, transparent, collaborative and closer to citizens.

The European Open Science Cloud (EOSC) aims to accelerate and support the current transition to more effective Open Science and Open Innovation in the Digital Single Market by removing:

- legislative
- technical
- and human barriers

of re-using research data and tools, via the support of access to services, systems and the flow of data across disciplinary, social and geographical borders [2].

A shortage of data experts can be found both worldwide and in the European Union. Despite Open Science practices being successful, the fragmentation of communities still may produce repetitive and isolated solutions. The short and dispersed funding cycles of core research and e-infrastructures are not fit for the purpose of regulating and making effective use of global scientific data [2].

While there are initiatives inside the European Union to carry out an effort for supporting open science, there does not yet exist any mandated and dedicated effort in order to help the member states focus their efforts towards creating such an EOSC-like activity on the continent.

1.2 Factors to be considered

- A new way of scholarly communication needs to be established and implemented, which would be supported by machine actionability
- New recognition and reward practices needs to be introduced that is fit to the modern world
- The career perspective of data experts shall be improved
- Innovation in funding schemes needs to be developed for a sustainable infrastructure
- Shall support interdisciplinary collaboration
- A dedicated support is needed for the transition of of scientific insights towards innovation
- The possibility for exploitation of machine automation in data processing for the benefit of a more fluent production needs to be available
- A lightweight but effective guiding governance shall be established
- EOSC performance indicators need to be developed [2].

1.3 Open Science

The European Open Science Cloud would have no use, if it weren't for the open science initiative and open science practices. In order to understand part of the fundamentals of the European Science Cloud initiative, open science itself as a concept has also got to be discussed further.

In a nutshell, Open Science describes an on-going movement in the way research is performed, researchers collaborate, knowledge is shared, and science is organised. It affects the whole research cycle and its stakeholders, enhances science by facilitating more transparency, openness, networking and collaboration.

Open Science opens up scientific processes and products from all levels to everyone. As such it includes Open Access, Open Research Data, Open Methodology, Open Evaluation, Citizen Science[3].

1.4 The benefits of open science by actor type

Researchers Researchers are granted with greater visibility and reach as well. This gives researchers a better overview, and allows them to discover greater possibilities. Their working efficiency is increased and are granted with being discovered by funders and being granted with more funding. Moreover, collaboration and networking among researchers is also enchanced.

Funders Funders enjoy increased visiblity, and experience greater funding impact as they can have their funded research being re-used by the scientific community. This increases their return on investment, and incentivizes competition among funders to be even more active.

General public The general public benefits by an even faster exchange of knowledge. They are more prone to gain an increased expertise in their own interests and gain new ones. Reaching out to the general public will greatly support promoting engagement in science and research.

NGO's Non-governmental organizations benefit from enchanced access to research, as it allows for acquiring better technologies at a faster rate. These organization also gain benefit through more effective advocacy or lobbying.

Governments National government experience improvements via evidence-formed policies. Open science practices help promote human rights and democracy for a more stable and justified governing body [3].

1.5 EOSC: Modern concepts and tools for open science

New modes of scholarly communication Scholarly communication and delivery had been dominated by narrative and verbal means for centuries. The modern way should would improve communication by re-using formats that also better suit our main research assistants: the data generating and processing machines. [2]

Modern rewards and recognition Assessment, selection, funding and reward systems in research have to be urgently modernized. For instance, little support and incentive is provided by the current systems for data publishing and tool sharing. The development and data related expertise is lacking the appropriate acknowledgement. The encouragement of data discovery is a vital part of conducting science: it shall be supported as such. New forms of output publication for data and software are emerging that need to be given credit in research assessment. [2]

Cross-disciplinary collaboration Cross-disciplinary collaboration is critically needed, as scientists increasingly data and tools from disciplines other than their own. However, currently, other people's data is notoriously difficult to discover even within one's own discipline. With the current absence of proper meta-data standards and the lack of data and tool search engines, this continues to be a challenge. Use of text and data mining techniques are essential for the EOSC, research analysis and support of cross-disciplinary use. However too often there are legal barriers, which need to be discussed in an appropriate, democratic manner. [2]

Fostering transition from science to innovation Science is not without those, who use it. Societal innovation and productivity, which takes place predominantly

in collaboration between research institutes and the private sector needs not to be underestimated and is in need of support. The immense amount of discoveries and advancements shall be exploited by society and the private industry as well, and this possibility should also be publicly acoknowledged and supported. [2]

A complex eco-system of infrastructures Data increases at a much faster pace than we are able to build infrastructure to support it. This issue can not be solved by solely relying on building larger and larger infrastructure that are based on the same concepts and principles as before. Computer science advancements have allowed for a giant leap in processing large amounts of data, bringing innovative computing, storage, and new advanced algorithms for knowledge extraction from data.

This however is not sufficient for a self-sustaining system in the long run. The challenges of data size, rights to privacy and privacy sensitivity, legislation have got to be considered. The combination of relatively ligthweight, distributed workflows combined with parallelized analytic algorithms allows for a more sustainable environment. High throughput compute (HTC) capacity will continue to grow, but centralised supercomputing locations that are crucial for solving high capacity HPC scientific challenges alone will not adequately bear all of the challenges ahead. Complementary infrastructures are needed. [2]

Machine understanding Enermous amounts of data can only be processed by powerful computers that are fit for the task. These machines are able to reveal such patterns that would have taken humans for years. This automation is machine understanding. Machines have become essential research assistants for data generation, data processing and analytics as well. Data mappings and provenance must therefore be optimally organised in order to support machine processing as well as the human-mind knowledge extraction. [2]

Research integrity Reproducibility is alarmly lacking in contemporary research, which when combined with scientific fraud, inflicts large amount of damage on the reputation of science. Earyl decetions of fraud-signals and optimal re-use of already established assets would support data processing, and thus research and science as a whole. [2]

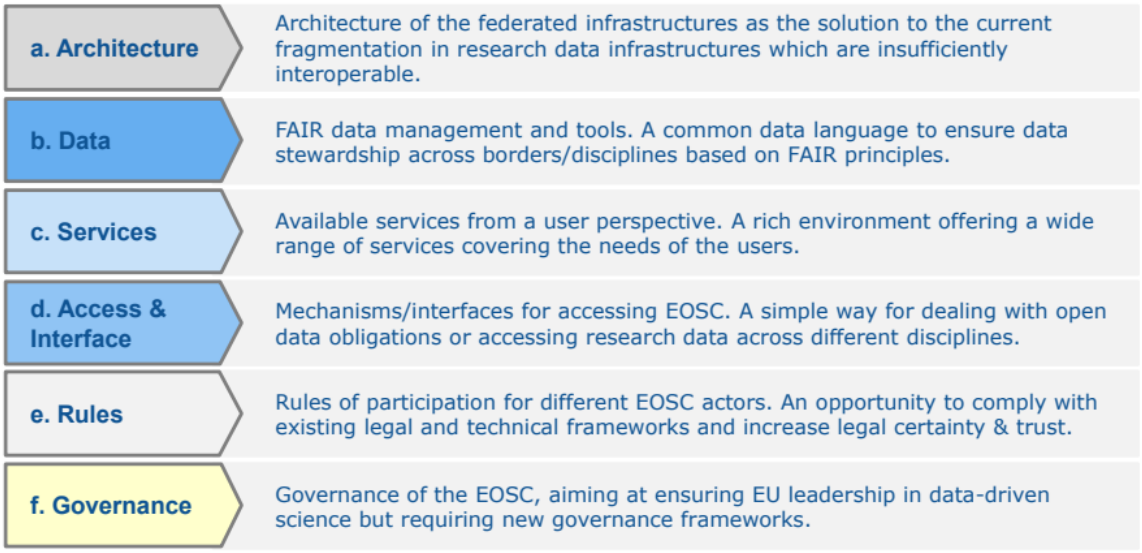

Figure 1.1: The 6 pillars of the EOSC model

The 6 pillars of the EOSC model are a foundation for the entire EOSC project: architecture, data, services, access, rules and governance (Figure 1.1) [4]. These six pillars bear the basis upon which the concepts are developed and enriched.

1.6 NEANIAS

As I have mentioned that the findings of this paper can be potentially used for SZTAKI (Institute for Computer Science and Control) to aid in the NEANIAS (Novel EOSC Services for Emerging Atmosphere, Underwater & Space Challenges) [1] project, let us explore NEANIAS in a brief overview.

NEANIAS promotes Open Science practices and plays an active role in materializing the EOSC ecosystem by efficiently engaging large scientific and professional communities; actively contributing to the technological, procedural, strategic and business development of EOSC [5].

Technology

From a technological perspective, NEANIAS will provide services that are flexible and open to accommodate the needs of communities beyond their original definition, and able to adapt to neighbouring cases, fostering reproducibility and re-usability.

Business

From a business perspective, NEANIAS aims to lift the barriers that may hinder industry / research engagement (data availability and access, big data handling and analytics challenges, research reproducibility, regulations) and lay a viable business plan for the services in question, identifying new actors and business opportunities [6].

NEANIAS Services

Neanias delivers four service groups. The first three being thematic ones that address cross-cutting application field in Underwater, Space and Atmosphere Research Sectors. The fourt one being the core service, which delivers reusable cross-sector services that are more generic.

- NEANIAS Underwater: offers service groups for underwater-related studies and engineering tasks for user communities linked to the underwater environment including archaeologists, geologists, oil and gas energy engineers, marine robotics, submarine geohazards, and insurance.

- NEANIAS Atmosphere: The target users of this service group are the sectors of the atmosphere field include meteorologists, industrial air pollutant emitters, ecologists, rural urban planners and air quality authorities, geohazards, civil protection, insurance or health agencies,

- NEANIAS Space: Provides service for space communities, including astrophysicists, planetary scientists, planetary mining engineers, planetary robotics, mobile telecommunication and space weather in virtual observatories with a variety of visual discovery scenarios.

- NEANIAS Core: Establishes generic, cross-community services that enhances thematic services, enabling Open Science, and facilitating the migration to the EOSC concepts by providing access to cloud resources.

2 Data handling in EOSC

The EOSC implementation roadmap contains 6 major lines of model. In order to get a glimpse into data handling, the second major line of model, data has to be elaborated.

2.1 Purpose

Shared resources have to be developed throughout the entire EOSC system. This includes:

- FAIR data tools
- Specifications
- Catalogues
- Standards
- and Services.

These shared resources are to be used by any researcher and be implemented by any infrastructure, mandated by research funders: all while covering all aspects of the FAIR concept (discussed in the next section). This process of defining shared resources across any discipline and country should be staged, iterative and flexible. Current tools and practices need to be evaluated, analized and streamiled, agreed upon, catalogued, certified and implemented as part of the other available EOSC shared resources that satisfy EOSC standards.

2.2 The FAIR concept

The success of developing the share of resources is influenced by the culture of research data management among scientists and innovators and the demand for FAIR data and FAIR incentives by funders and institutions.

The FAIR data handling concept stands for: Findable, Accessible, Interoperable, Reusable [7].

Findable:

- Data and metadata should have globally unique and persistent identifiers assigned to them.
- Information contained in metadata should be rich.
- Metadata shall explicitly include identifiers of data they are describing
- Data and metadata need to be registered or indexed on a searchable resource.

Accessible:

- Data and metadata shall be retrievable by an identifier using a standardized protocol. This protocol shall remail open and free, that allows for authentication and authorization.
- Metadata shall remain accessible, even remaining after data is no longer available.

Interoperable:

- Knowledge representation has to be established with (meta)data that use a formal, accessible, shared and broadly applicable language.
- FAIR principles need to be followed by (meta)data vocabularies.
- (meta)data include references to other (meta)data that are qualified.

Reusable:

- (Meta)data are richly described with relevant and accurate attributes.
- These data are to be released with an accessible, clear license.
- They shall be associated with detailed provenance,
- and need to domain relevant standards of the community.

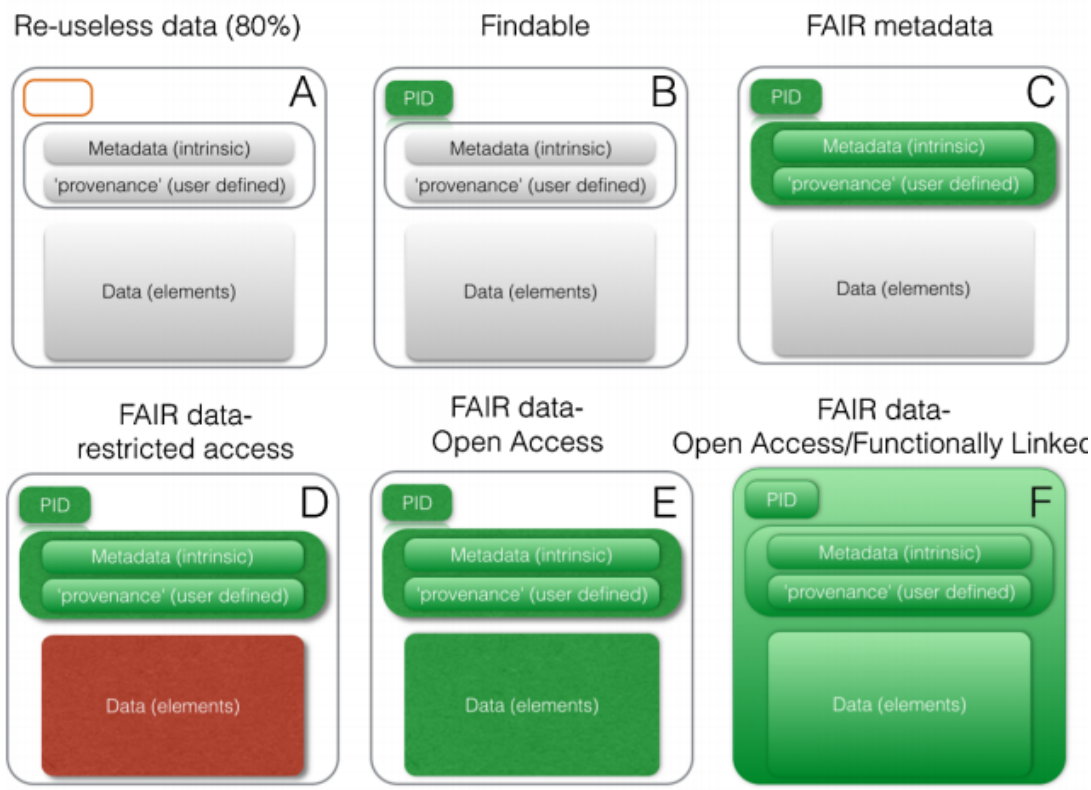

Figure 2.1: Varying degrees of FAIRness

As elements become coloured, they have become FAIR. For example, adding a persistent identifier (PID) increases the fairness of that component (Figure 2.1) [8].

2.3 Technological background

Access to services

Three types of access were recognized when categorizing EOSC data services: not needing registration, needing registration and needing authentication that the user is a researcher. Services that can be accessed through application programming interfaces (API's) need an extra layer of authentication by registering for the API and exchanging tokens using the developed programs.

Using services

Most services can be used via a GUI on the web. Even more support API's where Communications are carried out using application programming interfaces. These interfaces are established on endpoints, on which posting and getting data is possible most commonly via HTTP REST API's. Whenever a request is invalid, response data is sent in order for the programmer to be able to troubleshoot the problem.

In order to standardize the communication, schemas have to be followed for each data format, which are set by the scientific communities. When acquiring data, response files come in a variety of dataformats: csv's, json's or xml's. Requesting

data is possible via parameterized URL requests , or by posting request files (json, xml) for which the programmer needs to obey schemas set by the service's provider or scientific community. Uploading is done via posting with payload data using standard data formats such as json, xml or csv.

Almost all services provide training materials and development specification, documentation for implementing systems that are able to utilize their services. These documentations enable programmers to increase efficiency of data extraction, handling and publishing via machine automation.

3 Categorization and study

3.1 Specification

The current state of categorization in the EOSC systems is in its very early stage. Majority of the data services were not categorized, or were categorized incorrectly, or have been assigned to all of the available categories.

There were services that had descriptions that did not fully encompass the scope of that service. All of the currently available data services needed to be re-categorized manually, as (along with mis-categorization), the categories on EOSC were not sufficient to separate data services by functionality and target audience.

Services, which for there exists a website that can be used to access the service, have been provided with embedded links in the PDF document: the name of the service is written with blue text, when clicked redirects to the browser and uses the link of the service.

When a service does not have a direct link, only a reference, that service is most likely only reachable when registered as a researcher. Each data-service's description is referenced to the appropriate EOSC catalogue page. The introductory overview of data services is organized in this document by:

- Numbered subsection (3.2, 3.3, ...): each is one major research area
- Numbered sub-subsection (3.2.1, 3.2.2, ...): each numbered sub-subsection is either:
  - a functionality scope
  - a research sub-area
  - or multiple services presented under a single subsection (eg. same provider, interconnected services)
- Non-numbered sub-subsection: each sub-subsection (non-numbered) is a data service (that belongs below a numbered sub-subsection), or seemingly multiple services that are actually just parts of a single service (such as the OpenAire helper services, under section 4.6)).

3.2 Interdisciplinary

These services are not restricted to a specific research area, but can be used to support any or multiple research areas in general.

3.2.1 Research & collaboration supporting services

Each of these services aid in conducting research, the coordination of employees and scientists/researchers and collaboration between multiple parties.

OpenAIRE Graph Access API

The OpenAIRE Graph Access API allows developers to access the metadata information space of OpenAIRE programmatically [9].

Collaboration Framework (BlueBRIDGE)

The Collaboration Framework is a set of social tools to share data, updates and messages with others and to keep abreast of new data, prospects, services, users [10].

DMPonline

DMPOnline helps to create, review, and share data management plans that meet institutional and funder requirements. Institutions can customise the tool to provide local guidance and example answers to assist researchers in developing plans [11].

ARGOS (OpenAIRE & EUDAT)

ARGOS is an online tool in support of automated processes to creating, managing, sharing and linking DMPs with research artifacts they correspond to [12].

OpenAIRE Research Community Dashboard (Open Science)

The OpenAIRE Research Community Dashboard on demand Community Gateways that communities can customize to fit their sharing, publishing and discovery needs [13].

3.2.2 Research storing & publishing/sharing services

These services aid in the publishing/sharing of conducted research and short-term or long-term storage of already conducted research as well.

Zenodo

Zenodo is a general purpose repository that enables researchers, scientists, projects and institutions to share, preserve and showcase multidisciplinary research results (data, software and publications) that are not part of the existing institutional

or subject-based repositories of the research communities. It is founded in the trustowrthy CERN data centre. Enables everyone to participate in Open Science.Used by more than 50K researchers and 3K communities all over the world. [14].

B2SHARE

Store and publish research data. A user-friendly, reliable and trustworthy way for researchers, scientific communities and citizen scientists to store and publish small-scale research data from diverse contexts [15].

Figshare for Institutions

Figshare is a repository where users can make all of their research outputs available in a citable, shareable and discoverable manner [16].

Open Access Research Data Archive (MIDAS)

MIDAS helps to Manage Your Research Data in a Safe Data Repository. To collect, process, store and analyse scientific research data and other relevant information in all subject areas enabling free, easy and convenient access to it via the Internet for any user without infringing copyright and intellectual property rights [17].

OpenAIRE Content Provider Dashboard

The OpenAIRE Content Provider Dashboard is a one-stop-shop web service where data providers (repository, data archive, journal, aggregator, CRIS system) interact with OpenAIRE. It provides the front-end access to many of OpenAIRE's backend services [18].

3.2.3 Data processing & handling/managing services

These services help in processing and modifying data according to the researchers' needs, or are able to handle/manage said specific data.

V-Seem Clowder

Clowder is a research data management system designed to support any data format and multiple research domains. Contains three major extension points: preprocessing, processing and previewing [19].

Access to Open Data platforms (BlueBRIDGE)

Access to Open Data platforms is a group of services to enable standardized access to data [20].

AMNESIA (dataset anonymization)

Amnesia allows end users to anonymize sensitive data in order to share them with a broad audience. The service allows the user to guide the anonymization process and decide on a flexible trade-off between privacy guaranty and data utility [21].

BiOnym (BlueRidge)

Online environment to support online and onsite courses. BiOnym is a domain-specific framework and an environment to compare high volumes of names between registries.

- Software Developers that need to reconcile lists and sets of names with spelling hierarchical errors. The framework offers a flexible set of algorithms to define a software-driven approach to compute lexical distances
- IT Managers are provided with a cost-effective and performant solution where an online service is needed to compute similarities between lists and sets, and where IT decision support is needed based on computed similarities
- Data Managers have a flexible solution when the need is to harmonize and standardize (reference) data, and face issues with spelling mistakes, different vocabularies and hierarchical mistakes. BiOnym can support and automate some of the steps to reconcile differences between taxonomies
- Individual scientists are assisted in identifying the correct name(s) of individual or groups of species names on-line, with a flexible user interface where preferences can be set
- e-infrastructure experts/representatives are supported in performing data-sets linking and capturing the results in a formalized way as a computed distance between two sets, with information of the method(s) and weights applied as integral part of the computational results [22].

B2HANDLE

B2HANDLE is the distributed service for minting, storing, managing and accessing persistent identifiers (PIDs) and essential metadata (PID records) as well as managing PID namespaces. The implementation of the service relies on the DONA/Handle persistent identifier solution [23].

EGI Data Transfer

Transfer large sets of data from one place to another. Data Transfer allows you to move any type of data files asynchronously from one place to another. The service includes dedicated interfaces to display statistics of on-going transfers and manage network resources [24].

B2NOTE

Data annotation service to create annotations on research data. Allows to easily create, search and manage annotations. B2NOTE is a standalone service for annotating data content hosted within the EUDAT CDI [25].

Scalable Data Mining (BlueBridge)

Scalable Data Mining is a comprehensive and collaborative environment to perform statistical data analysis. Individual scientists have access to a wide spectrum of available data analytical models and can easily test, validate, and run algorithms developed in a large spectrum of software languages (e.g. R, Java, Python, Fortran) [26].

Dynamic Reporting

Dynamic Reporting is a services-driven environment to enable interactive analysis and presentation of data analysis and scientific findings. Individual scientists simplify the organization and presentation of the scientific findings resulting from the research activities [27].

NYM Framework

NYM Framework allows comparison, harmonisation and validation of large datasets against a tailored controlled vocabulary. Data Managers can customize first and then harmonize and standardize (reference) data, and face issues with spelling mistakes, different vocabularies and hierarchical mistakes [28].

3.2.4 Data storing

B2SAFE

Allows repositories maintainers to implement data management policies on their research data across multiple administrative domains.
An abstraction layer is provided of a large scale, heterogeneous data storages and guards against data loss in long-term archiving and optimizes user data access [29].

EGI Archive storage

Archive Storage allows you to store large amounts of data in a secure environment freeing up your usual online storage resources. The service is optimised for infrequent access [30].

3.2.5 Data resources & data acquisition

These services aid in gathering and acquiring already existing data that is available through the service.

Snap4City (IOT as a Service, SCIaaS)

Demo is provided immediately upon reaching the site's link. Snap4City provides a flexible method and solution to quickly create a large range of smart city applications exploiting heterogeneous data, performing data analytics, and enabling services for stakeholders by IOT/IOE, data analytics and big data technologies [31].

Snap4City is for organizations/communities interested in Smart City and IOT, which would like to perform experiments on smart city data, that can upload or reuse from those available, for research purpose and validation [31].

B2FIND (metadata discovery)

B2FIND is the EUDAT metadata service and provides a discovery portal which allows users to find data collections within an international and inter-disciplinary scope. It is based on a comprehensive metadata catalogue of research data collections stored in EUDAT data centres and other repositories [32].

LEMONADE

LEMONADE Visual Data Analytics Tool. Live Exploration and Mining Of a Non-trivial Amount of Data from Everywhere. Support the following flows of work: Importing, exporting or managing datasets and visualize data [33] [34].

FAIRsharing

FAIRsharing is a FAIR-supporting resource that includes: an informative and educational registry on data standards, databases, repositories and policy, alongside search and visualization tools and services that interoperate with other FAIR-enabling resources [35].

For researchers: Identify and cite the standards, databases or repositories that exist for your discipline when creating a data management plan, releasing data or submitting a manuscript to a journal [36].

Identifiers.org

The Identifiers.org system provides a range of services to generate, resolve and validate persistent (PID) and Compact Identifiers, to enable the referencing of scientific data and to promote the citability of individual data provider, enabling integration across e-infrastructures [37].

eTDR

eTDR is a service provided to ensure that digital scientific information remains findable, accessible, interoperable and reusable over time [38].

3.2.6 Other

These individual services were group together, as they would only be able to present alone in their own category.

Open Telekom Cloud (PaaS, SaaS)

Open Telekom Cloud: highly scalable public cloud services based on OpenStack, with granular services for compute, storage, connectivity & network, security & management, databases, data analysis, PaaS and SaaS functions [39] [40].

OpenAIRE helper services

The items down below were all listed as an individual data service among all the other EOSC services.

- Technical support towards OpenAIRE compliance [41].
- The OpenAIRE validator [42].
- The OpenAIRE Catch-all Notification Broker Service [43].

PerfSONAR

PerfSONAR enables Real-time, multi-domain performance monitoring. Can be used by GÉANT, national research and education networking (NREN) organisations, campuses and major projects for quick and easy performance troubleshooting.

TSD

This service provides researchers with a desktop with secure storage and software to run your collection and analysis of sensitive data [44].

Jelastic Multi-Cloud (DevOps PaaS)

Jelastic is a Multi-Cloud DevOps PaaS for: ISVs, telcos, service providers, enterprises. The platform automates: creation, scaling, clustering and security updates of cloud-native and traditional applications [45]. Requires a verification from a scientific affiliation to use this service.

3.3 Engineering and technology

Services aid in research of engineering and technology.

NOMAD repository

NOMAD repository is the world's largest collection of materials science data, that provides the base of related computations [46] [47].

AiiDA lab

Reproducible turn-key workflows for materials science. AiiDA is a workflow manager for computational science with a strong focus on provenance and performance [48].

3.4 Agricultural sciences

These services are closely related to agricultural and food sciences.

AGINFRA+ services

AGINFRA+ aims to exploit core e-infrastructures such as EGI.eu, OpenAIRE, EUDAT and D4Science, towards the evolution of the AGINFRA data infrastructure, so as to provide a sustainable channel addressing adjacent but not fully connected user communities around Agriculture and Food [49].

Geoanalytics Visualization

AGINFRA+ Geoanalytics Visualization is a simple yet efficient GIS system that facilitates analysts and scientists to visualize, analyze and manage geospatial information [50].

AGRIS Elastic Index

AGRIS Elastic Index is an API for accessing and searching the AGRIS database [51].

Semantic Linking Service

AGINFRA+ Semantic Linking Service is a service for defining arbitrary links between entities of two or more semantic specifications pertaining to the agri-food sector [52].

Ontology Engineering Service

AGINFRA+ Ontology Engineering Service is a service for collaborative development, maintenance and sharing of semantic resources (ontologies, vocabularies, thesauri, etc.) related to agriculture and food sector [53].

Data Transformation Service

AGINFRA+ Data Transformation Service allows the import of tabular data in CSN or MS Excel format and their manipulation via the application of rules over rows, columns and cells [54].

Chart Visualization

AGINFRA+ Chart Visualization provides to users generic capabilities to visualize bar charts, scatter plots and other basic charting graphs [55].

3.5 Earth sciences, Earth Data

The services here are targeted at researchers who wish to acquire or handle data on Earth sciences

3.5.1 CREODIAS

Earth observation data management in the system of CREODIAS.

CloudFerro Data Collections Catalog

The CloudFerro Data Collections Catalog is a data publication and subscription service specialized in Earth Observation [56]. The services include most of the Copernicus, Sentinel, Landsat other various satellite services aiding in Earth Observation.

Cloudferro Data related Services - EO browser

The EO Browser allows visualization and basic processing of selected CREODIAS data collections [57].

Cloudferro Data related Services - EO Finder

The EO finder tool allows finding data products stored in the repository, obtained or processed at selected times with selected cloud coverage levels and with other selection criteria [58].

3.5.2 Naval services & fishery

The services found here aid in acquiring and handling data on fishery and various naval services.

Aquaculture Atlas Generation Spatial Data Catalog

Connect spatial information communities and their data using a modern architecture, which is at the same time powerful and low cost on the GeoNetwork [59].

Aquaculture Atlas Spatial Data Repository

A repository of datasets hosted by D4Science. Datasets include species distribution maps on the GeoServer [60].

Argo floats data discovery

Argo floats data discovery. The broad-scale global array of temperature/salinity profiling floats, known as Argo, has already grown to be a major component of the ocean observing system [61] [62].

Data-driven Atlas Production

Data-driven Atlas Production is a group of services to generate atlases of coastal and marine sites based on analyses of hosted geospatial layers that can be derived from interoperable external infrastructures that provide EO based products [63].

BlueBRIDGE Atlas services

Multiple services are listed on EOSC separately, which are actually part of a unified service system [64] [65]:

- French Tuna Atlas Spatial Data Catalog
- French Tuna Atlas Spatial Data Repository
- Global Tuna Atlas Spatial Data Catalog
- Global Tuna Atlas Spatial Data Repository
- Indian Ocean Tuna Commission Spatial Data Catalog

Global Record of Stocks and Fishery

Global Record of Stocks and Fishery is a domain-specific combined registry of information on the status and trend in fisheries. The service is used to summarize fisheries and exploitation reports [66].

Regional Database for Fishery Stock Management

Regional Database for Fishery Stock Management is a domain-specific group of services to collect and collate data and perform analysis through tools and reports [67].

3.5.3 Multiple SeaDataNet services

These services have been categorized as their own category as they are closely tied in an service ecosystem. The following services have identical descriptions: The SeaDataNet infrastructure links already more than 100 national oceanographic data centres and marine data centres from 35 countries riparian to all European seas [68].

- SeaDataNet Common Data Index (CDI)
- SeaDataNet DOI minting service

- SeaDataNet European Directory of Marine Environmental Data (EDMED)
- SeaDataNet European Directory of Marine Environmental Research Projects (EDMERP)
- SeaDataNet European Directory of the Cruise Summary Reports (CSR)
- SeaDataNet European Directory of the Initial Ocean-Observing Systems (EDIOS)
- SeaDataNet Vocabulary Services - underpinned by the NERC Vocabulary Server (NVS).

Spatial Planning

Spatial Planning (e.g. monitoring of Aquaculture, MPA, Integrated Coastal Zone Management - ICZM) is a group of services to generate holistic maps to support spatial planning by deep analysis of the relationships between hosted layers (intersection and interpolation analysis, proximity analysis, merging and editing of features) [69].

Stock Assessment Support

Stock Assessment Support is a domain-specific group of services to upload and harmonize (or have access to) tabular data, analyse using state-of-the-art tools (DLM Toolkit, FLR, SS3, CMSY-as-a-service, etc.), to generate dynamic and interactive reports and maps and to generate standardized and harmonized fisheries data sets [70].

Western Central Atlantic Fishery Commission Spatial Data Catalog

This service Connect spatial information communities and their data using a modern architecture, which is at the same time powerful and low cost, based on International and Open Standards for services and protocols (a.o. from ISO/TC211 and OGC) [71].

Western Central Atlantic Fishery Commission Spatial Data Repository

A repository of datasets hosted by D4Science. Datasets include species distribution maps [72].

Aquaculture Farming

Aquaculture Farming is a comprehensive and collaborative environment to analyse and forecast aqualculture farm performances [73].

3.6 Biology

Biodiversity (BlueBDRIGE)

Biodiversity is a domain-specific group of services to i) provide access to biodiversity repository on taxonomic names and species occurrences and ii) project and perform analysis [74].

Species Modeling

Species Modeling is a domain-specific group of services to model species distribution based on observational records or other information, and predict current and future distributions based on the interaction between environmental and biological parameters [75].

3.6.1 Spatial data

D4Science Spatial Data Catalog (GeoNetwork)

Connect spatial information communities and their data using a modern architecture, which is at the same time powerful and low cost, based on International and Open Standards for services and protocols (a.o. from ISO/TC211 and OGC) [76].

D4Science Spatial Data Repository

D4Science Spatial Data Repository is a repository of datasets hosted by D4Science. Datasets include species distribution maps [77].

D4Science Protected Area Impact Maps Spatial Data Repository

Protected Area Impact Maps Spatial Data Repository provides a comprehensive overview of spőatial info and processes. Scientists access spatial data and processing on demand [78].

Spatial Data Infrastructure Laboratory Catalog

Spatial Data Infrastructure Laboratory Catalog. Catalog application to manage spatially referenced resources [79].

3.6.2 Other

EODC Data Catalogue Service

For Earth Observation Data. The EODC Data Catalogue service allows querying the Copernicus Sentinel satellite data hosted at EODC. The service is available through a simple Web GUI, eomEX+, as well as an API [80].

Sentinel Hub

Sentinel Hub is a multi- spectral and multi-temporal big data satellite imagery service, capable of fully automated archiving, real-time processing and distribution of remote sensing data and related EO products [81].

3.7 Medical and biochemical sciences

3DBIONOTES

3dBIONOTES s a web application designed to automatically annotate biochemical and biomedical information onto structural models. Current sources of information include post-translational modifications, genomic variations associated to diseases, short linear motifs, immune epitopes sites, disordered regions and domain families [82].

Dipcheck

DipCheck is a validation tool for the evaluation of protein backbone geometry[83].

MetalPDB

MetalPDB collects and allows easy access to the knowledge on metal sites in biological macromolecules, starting from the structural information contained in the Protein Data Bank (PDB) [84].

3.8 Humanities

3.8.1 Culture, heritage

Europeana APIs

Large-Scale Data Discovery, Acquisition and Management of Digital Cultural Heritage in Research. Accessing this service requires a verification from a scientific affiliation.

The Europeana REST API is a set of APIs which provides a service calibrated to the needs of professionals interested in the reuse of digitised or digital-born cultural heritage in education, research and the creative industries [85].

Digital Humanities and Cultural Heritage OpenAIRE Community Gateway

Growing a single entry point for Digital Humanities and Cultural Heritage Researchers in Digital Humanities and Cultural Heritage [86].

3.8.2 Social sciences

Data Service Portal Aila

A data service of the Finnish social science data archive. Provides related Finnish social science data [87].

da—ra

da—ra assigns persisitent identifiers to digital objects by using the DOI system [88].

Transkribus

Transkribus is a platform for the digitisation, transcription, recognition and searching of historical documents [89].

3.9 Other

DataFurn (furniture businesses)

Datafurn collects and analyzes various data, including social media platforms in order increase efficiency of business for furniture companies [90].

4 Open Science and FAIR data adhering services

4.1 The syngergy of open science and FAIR principles

The advancement of digital science thrives on the timely sharing and accessbility of digital data. Data, as a highly abundant resource it might be, is also highly precious as well, but this value does not come solely by the contents of such data, but the manner in which these data are organized, maintained, shared, used, and collaborated on with as well. In the 21st century, information is the most valuble resource, and with this value comes a well-defined and highly efficient handling of such data as well.

The Open Science initiative and movement can be highly enhanced by such services that are FAIR (Findable, Accessible, Interoperable, Reusable) compliant. FAIR principles emphasize machine-actionability (i.e., the capacity of computational systems to find, access, interoperate, and reuse data with none or minimal human intervention) because humans increasingly rely on computational support to deal with data as a result of the increase in volume, complexity, and creation speed of data.

More promising, the non-established, "citizen researcher", as an untapped opportunity, has finally been uncovered. Tools based on Open Science practices enable ordinary citizens to contribute to scientific research, initiate new research and access valuable results which they can implement and use for innovation.

The demand of research institutions, funder and publishers has significantly grown in terms of research on data management, and opening up research data for reuse. FAIR data principles encourage funded and citizen researchers to ensure that their data is soundly managed and subsequently shared. The four principles helps citizen researchers to find the data they need: the principles help them become accustomed to how data is shared and managed, and can more easily find data in a more rapid manner.

Funded researchers and citizen researchers may also create and/or utilize automated systems, computing systems to gather, save and publish such data. These advancements are promising for a more free and open-source data economy, in which every participants benefits, from large institutions to investors, onto each citizen researcher that participates as well. Let us explore some of the services that are promising candidates for use from a FAIR data management standpoint, in the next section.

4.2 Data service candidates

The services provided ought to aid in handling data of a certain scientific project, from the currently available data services provided by the EOSC system, some should be found which provide either one or multiple the following features:

- Ability to provide data, datasets as the basis for computation
- Provide storage: for archival purposes, or more frequent retrievals

- Provide a working directory for an ongoing project.

Such services are proposed in the following section.

4.3 Figshare

Figshare [91] having launched on January 2012 has since been one of the data sharing platforms for researchers to share their files and findings to the public. Figshare is also one of the data services found in EOSC that shares the goal of implementing FAIR principles to their digital services. Figshare is FAIR compliant in the following ways:

Findable

Fighsare creates DOI's (Digital object identifiers) for each dataset uploaded. DOIs in may also be reserved. The DOIs created are persistent and permanent. Rich metadata may also be assigned to data (: title, authors, description, keywords, categories, licence, item type, external references and funding). Metadata are registered or indexed in searchable resources, and metadata assigned specify the data identifier.

Accessible

Metadata are retrievable by their identifier, using the standard HTTPS. This protocol is open, free and universally implementable which adheres to the "A" principle of "FAIR". Allows for authentication and authorization. Assigned metadata is accessible even when data no longer available.

Interoperable

Metadata can be set via multiple languages via Dublic core, Datacite, RDF and Cerif XML.

Re-usable

Metadata have plurality of accurate attributes, with a clear ans accessible data usage license. The data are associated with their provenance and meet domain-relevant community standards.

4.4 Zenodo

Zenodo is a general purpose repository that enables researchers, scientists, projects and institutions to share, preserve and showcase multidisciplinary research results. Zenodo also provides application programming interfaces and a webpage for ease of use to save, share and maintain research data in a FAIR manner. Zenodo is the repository in relation with the NEANIAS project described in the NEANIAS chapter, described earlier in this paper.
Let us see how Zenodo implement FAIR compliance [92] in the following sections.

Findable

- A DOI is issued to every published record on Zenodo.
- Zenodo's metadata is compliant with DataCite's Metadata Schema minimum and recommended terms, with a few additional enrichements.
- The DOI is a top-level and a mandatory field in the metadata of each record.
- Metadata of each record is indexed and searchable directly in Zenodo's search engine immediately after publishing, and sent to DataCite servers during DOI registration and indexed there.

Accessible

- Metadata for individual records as well as record collections are harvestable using the OAI-PMH protocol by the record identifier and the collection name, as well as through the public REST API.
- Data and metadata will be retained for the lifetime of the repository. This is currently the lifetime of the host laboratory CERN, which currently has an experimental programme defined for the next 20 years at least.
- Metadata are stored in high-availability database servers at CERN, which are separate to the data itself.

Interoperable

- Zenodo uses JSON Schema as internal representation of metadata and offers export to other popular formats such as Dublin Core or MARCXML.
- For certain terms we refer to open, external vocabularies, e.g.: license (Open DefiniEach referrenced external piece of metadata is qualified by a resolvable URL.tion), funders (FundRef) and grants (OpenAIRE).
- Each referrenced external piece of metadata is qualified by a resolvable URL

Reusable

- Each record contains a minimum of DataCite's mandatory terms, with optionally additional DataCite recommended terms and Zenodo's enrichments.
- Data downloaded by the users is subject to the license specified in the metadata by the uploader.
- All data and metadata uploaded is tracable to a registered Zenodo user. Metadata can optionally describe the original authors of the published work.
- Zenodo is not a domain-specific repository, yet through compliance with DataCite's Metadata Schema, metadata meets one of the broadest cross-domain standards available.

4.5 Concorda

CONCORDA [93] "Concentrated Cooperation on Research Data", is a Hungarian research data repository system. The research data repository serves as a data store and repository solution both for international and national uses, controlled data sharing platform in a definite peer-to-peer form. The system builds on the secure data storage and well defined access rights existing in the MTA Cloud services, which is jointly maintained by SZTAKI and Wigner research institutes [94]. The project has put emphasis on COVID-19 pandemic related research, research group and institute as when the repository project has started launching.

Findable

- Metadata have persistent identifiers: DOI, ORCID, URN
- Data description is rich with metadata
- Metadata precisely define the identifier of the described data
- Metadata are searchable in service

Accessible

- Downloadable via open protocols, no additional software or financial capital needed
- Metadata remain in the repository even after the data described are no longer public

Interoperable

- Open source formats are used to describe metadata: JSON, XML, RDF
- The metadata publishing schemas adhere to the FAIR principles
- Metadata verifiably refer to other metadata

Reusable

- Description of data contain multiple precise and relevant information
- Available licenses for usage are precise and open
- Conception and source of data are well documented (provenance)
- Description of data adhere to the expectations and practices of the professional community

5 Design

The application will contain a library of methods and a client that invokes these methods as well.

5.1 Goal

The goal is to create a shared API (Application Programming Interface) for data repositories that are available through EOSC. This API will be used to create implementations for communicating with Zenodo and Concorda. The implementations are a demonstration for how the shared API's functions can be implemented for communication with data repositories, as shown in figure 5.1.

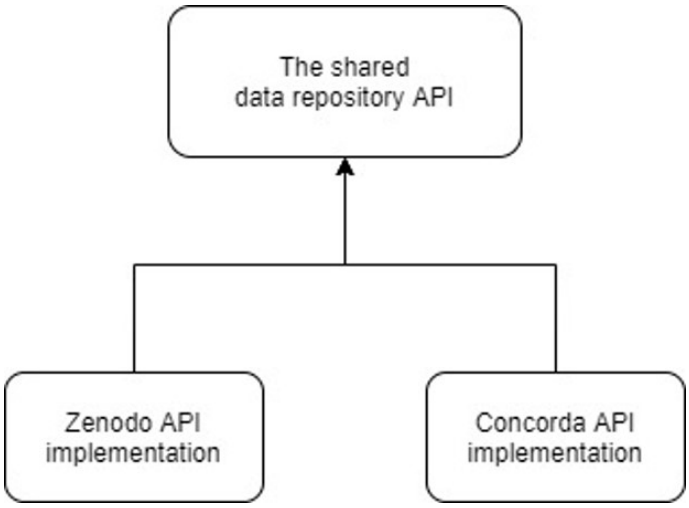

Figure 5.1: Implementations referencing the shared API

5.2 Workflow

The shared API and the implementations are done via Python 3. Python was chosen, as it's a widely used language in the scientific community, which is well supported by its developers and users worldwide.

The application will contain the shared API and its two implementations (Zenodo, Concorda). The application will upload, read, edit and delete data via invoking methods in the application. The application will communicate via HTTP with the two repositories and will exchange data in a common format, such as JSON or XML.

Authentication and authorization will be conducted via tokens, which are generated hashes given by the service provider which we use in our application to prove our identity and access rights to the services.

6 Implementation

6.1 Zenodo as a FAIR data service

Zenodo is one of the chosen candidate for use as a FAIR data service. Its services are used via a client developed in Python.

6.1.1 Services via web application

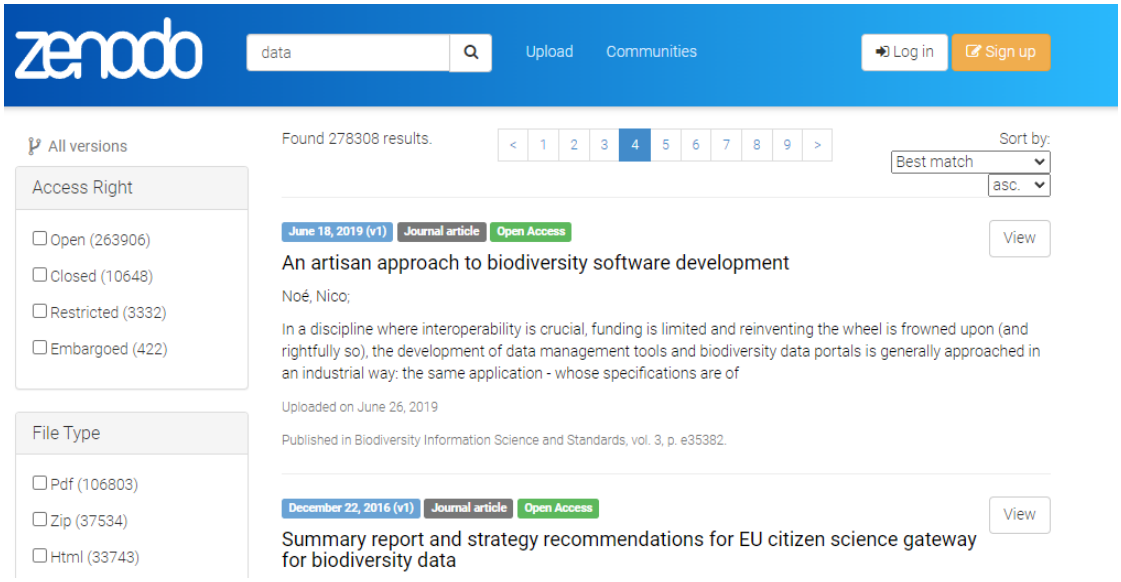

Figure 6.1: Zenodo web app UI

Figure 6.1 shows the landing page of the Zenodo web application, with the user not yet logged in.

The web application supports basic functionalities, such as:

- Uploading and publishing of research outputs
- Searching among published records
- Download and upload files of/to records
- Search and engage with communities, funders, grants
- Search for and use licenses.

6.1.2 Services via REST API

The Zenodo REST API currently supports [95]:

- Deposit: upload and publishing of research outputs (identical to functionality available in the user interface)

- search published records
- download/upload of files

The following services are currently in testing before Zenodo launches them:

- Communities: search communities.
- Funders: search for funders
- Grants: search for grants
- Licenses: search for licenses

6.1.3 Python application for Zenodo

Zenodo has been chosen as a demonstration for using a FAIR data services of the EOSC catalog. Ease of use and an extensive, well-detailed developer documention has been the main aspects when choosing among the other FAIR data services of EOSC. With well established service providers supporting Zenodo, it was chosen as the service used for the developed application.

Python has been chosen as the programming language for its ease of use, its familiarity in the scientific community, and developers alike. The functions and logic of this application can be further used and developed on top of, to completely enchance an easy use of the Zenodo services through their REST services. The application is able to upload, search and download with all three of the available services of Zenodo: deposits, records, files.

The application source code is publicly available on Zenodo specified with DOI 10.5281/zenodo.5779118 [96].

Architecture

The main logic of the application is separated into 2 sections: depositioncontroller and metadatahelper. Depositioncontroller is where the following logic are contained. Metadatahelper is where the metadata setting logic is contained.

Depositioncontroller:

create_new_deposition():

Responsible for creating a new deposition. A deposition is the container which we deem to publish. We upload our filed and metadata to a deposition. Returns the REST response which contains all relevant data, such as the ID in order to interact with the deposition further. To invoke this function, Metadatahelper is used to build the json used for the transaction.

get depositions(q, status):

Retrieves and returns depositions based on text search query (Elastic search) and status of the deposition.

upload files to deposition(dir path, deposition id):

Able to upload files to a specific deposition.

```
def upload_files_to_deposition(dir_path, deposition_id):
    input_directory_path = dir_path + "/files/input/"
    processed_directory_path = dir_path + "/files/processed/"

    url = full_url_depositions + "/" + str(deposition_id) + "/files"

    print(url)

    filenames = FileHelper.get_filenames(input_directory_path)

    if len(filenames) < 1:
        print("[WARN] Input folder empty.")

    for file_name in filenames:
        path = os.path.join(input_directory_path, file_name)

        file_upload_headers = {"Content-Type": "multipart/form-data"}

        data = {'name': file_name}
        files = {'file': open(path, 'rb')}
        r = requests.post(url, data=data, files=files, params=params)
        print(r.text)

    print("Files processed for deposition id: ", deposition_id)
```

Figure 6.2: File upload logic

The logic for uploading files is the following as shown in figure 6.2. It returns JSON response regarding all new information of all of the uploaded files.

put required metadata(...)

Using this logic we are able to set any of the mentioned metadata of a deposition for which we have files, which we wish to have published.

The input parameters are the required metadata we can set for the deposition: deposition id, upload type, optional upload subtype, publication date, title, authors, description, access right, optional access conditions, optional license, optional embargo date.

Returns the REST response which contains all relevant information.

Metadatahelper:

The metadatahelper contains most of the logic for assembling a JSON to be sent. The request json is assembled in the following way, this technique is used for the other jsons as well:

```
data = {}



if upload_type == "publication":
    data['publication_type'] = optional_upload_subtype
elif upload_type == "image":
    data['image_type'] = optional_upload_subtype


data['title'] = title
data['upload_type'] = upload_type
data['publication_date'] = publication_date
data['creators'] = authors
data['description'] = description
data['access_right'] = access_right

if access_right == "embargoed":
    data['embargo_date'] = optional_embargo_date
    data['license'] = optional_license
elif access_right == "restricted":
    data['access_conditions'] = optional_access_conditions
elif access_right == "open":
    data['license'] = optional_license

return data
```

Figure 6.3: Metadata building logic

build_required_metadata(...)

This logic is used the build the metadata as shown in figure 6.3.

The inputs parameters are the following: deposition id, upload type, optional upload subtype, publication date, title, authors, description, access right, optional access conditions, optional license, optional embargo date.

Example communication

Authentication is done via issued "token" which are string literals. We assign these tokens in the HTTP header field, in order to verify our identity and gain authorization to modify the data we have in the Zenodo services. This token is used for every transaction.

Creating a new deposition:

By invoking create new deposition() , we send a post request to the

`/api/deposit/depositions"`

endpoint, from which we get the following response:

```
"conceptrecid": "981626",
"created": "2021-11-11T15:44:47.819738+00:00",
"files": [],
"id": 981627,
"links": {
        //...contains links with id's we can use the edit
            data...
},
"metadata": {
        "prereserve_doi": {
                "doi": "10.5072/zenodo.981627",
                "recid": 981627
        }
},
"modified": "2021-11-11T15:44:47.819738+00:00",
"owner": 98504,
"record_id": 981627,
"state": "unsubmitted",
"submitted": false,
"title": ""
```

Uploading files to a deposition:

The python app has a folder called files in which a folder named "input" exists. We we invoke the upload files function, it uploads each file in the folder to the ID specified deposition. The response for such a communication reads as:

```
"checksum": "2eed4cad3e37665155f597fb861f642b",
"filename": "testfile2_-_Copy_2.json",
"filesize": 38,
"id": "4cdee8b5-5d11-43ef-8d17-6e9bd7cc046e",
"links": {
        "download": "https://sandbox.zenodo.org/api/files
            /9c848276...",
        "self": "https://sandbox.zenodo.org/api/deposit/
            depositions/981627/..."
        }
```

Editing metadata for a deposition:

By invoking the put required metadata() function, we have set the following data:

- Files
- Upload type (+optional subtype)

- Publication date
- Title
- Authors
- Description
- Access rights and conditions (licensing)

The response is the following, which contains the metadata we have set for our deposition:


```
"conceptrecid": "981626",
"created": "2021-11-13T15:44:47.819738+00:00",
"doi": "",
"doi_url": "https://doi.org/",
"files": [{
                "checksum": "2eed4cad3e37665155f597fb861f
                   642b",
                "filename": "testfile2_-_Copy_2.json",
                "filesize": 38,
                "id": "4cdee8b5-5d11-43ef-8d17-6e9bd7cc04
                   6e",
                "links": {...}}]
"id": 981627,
"links": {...},
"metadata": {
        "access_conditions": "msg me to access",
        "access_right": "restricted",
        "creators": [{
                        "affiliation": "SZTAKI",
                        "name": "John Doe"}],
        "description": "rest description",
        "doi": "",
        "prereserve_doi": {
                "doi": "10.5072/zenodo.981627",
                "recid": 981627},
        "publication_date": "2021-11-11",
        "title": "My title rest",
        "upload_type": "presentation"},
"modified": "2021-11-13T15:59:00.061450+00:00",
"owner": 98504,
"record_id": 981627,
"state": "unsubmitted",
"submitted": false,
"title": "My title rest"
```

The end result on the web UI

We are able to visit this deposition that we have created and modified purely via our REST application. The deposition was created and its data is visible on the Zenodo Web's UI as well.

Figure 6.4: The newly created deposition

Figure 6.4 shows the title and date of the newly created deposition.
Figure 6.5 shows the files that we have uploaded via our program.
Then, figure 6.6 shows the data that we have set via our program.

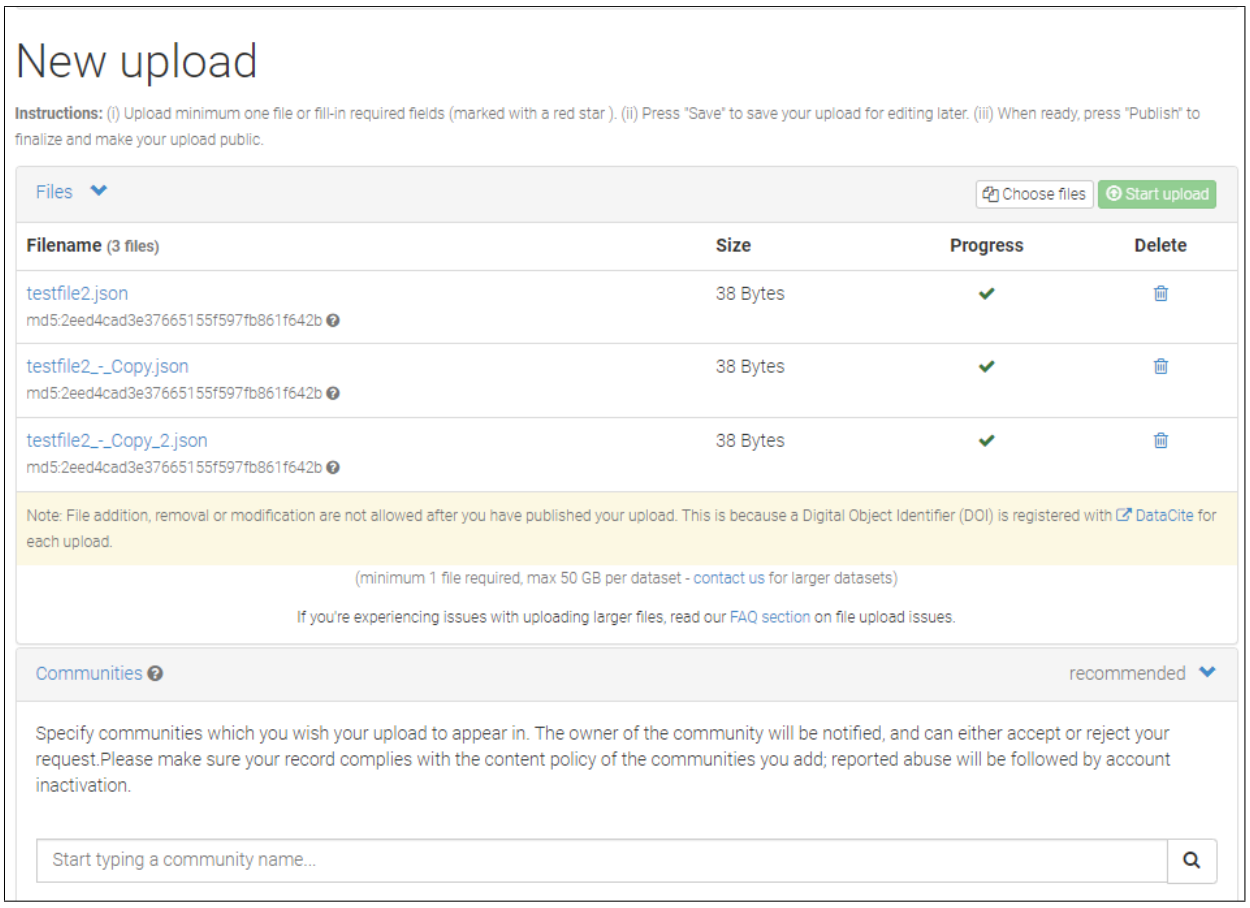

Figure 6.5: Files uploaded

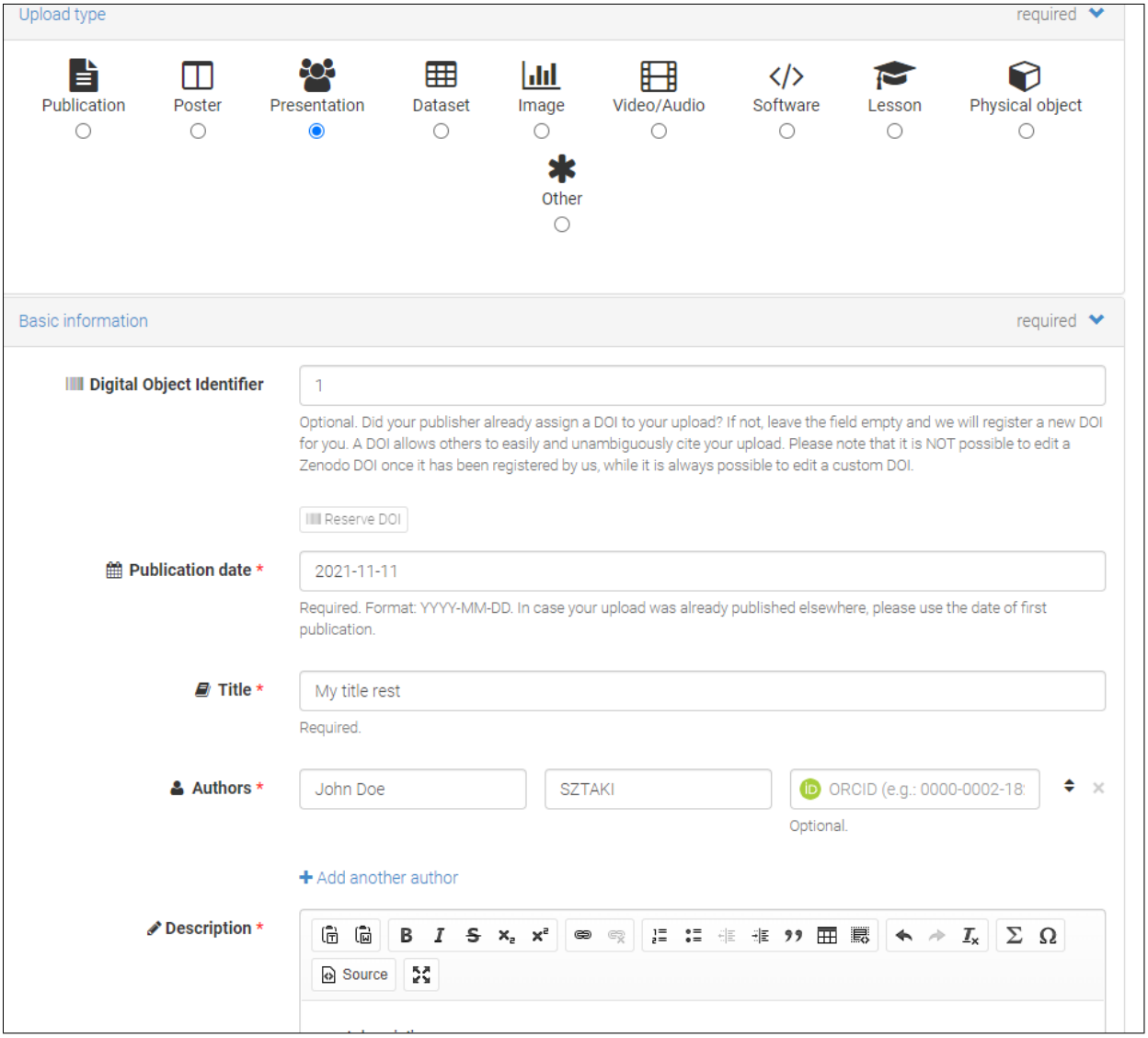

Figure 6.6: The data we have set

6.2 Concorda as a FAIR data service

Concorda is the second potential candidate for use as a FAIR data repository system.

6.2.1 Services via web application

Figure 6.7: The main page of concorda

The figure 6.7 shows the landing page of Concorda. The web application provides all functionality through a human interface, that would be needed for a repository management system:

- Account creation and management
- Finding data by basic and advanced search queries
- View metadata and download datasets
- Data collection viewing
- Data collection management: create, upload and edit data collections

6.2.2 Services via REST API

The functionality available through the web interface are available through programming interfaces as well, for a more automated use [97]. The API supports

authentication and authorization via tokens. These tokens are created and assigned to accounts created via the Concorda web application. The API consists of 4 main theme:

- Search
- Data Access
- Native (data collection management)
- Metrics

Search allows for data search in the repository. The search API can be used for basic search with parameters such as:

- q: basic string for general search, ie. Using "title:data" will search only the "title" field.
- sort: The sort field. Supported values include "name" and "date". See example under "order".
- order: The order in which to sort. Can either be "asc" or "desc".
- per_page: The number of results to return per request. The default is 10. The max is 1000
- fq: A filter query on the search term. Multiple "fq" parameters can be used. This is used for advanced search queries.
- metadata_fields: used for metadata queries. Multiple of this parameter can be used for advanced search techniques.

The Data Access API provides programmatic download access to the files stored in a Dataverse installation. Basic download by dataset can be done by prodividing the parameters:

- the API token
- the server URL we wish to download from (production or sandbox environment)
- and the persisten DOI which we identify the dataset we wish to download.

Example: "demo.dataverse.org/api/access/dataset/:persistentId/?persistentId=doi:10.70122"

Additional parameters can be used such as dataset version, if the dataset is versioned by it's manager. Individual files can be accessed via the /api/access/-datafile/$id URI, which is related to the dataset it's published to.

The Dataverse Software exposes most of its GUI functionality via a REST-based API, called the Native API. A Dataverse collection is a container for datasets and other Dataverse collections. The steps for creating a Dataverse collection are [97]:

- Prepare a JSON file containing the name, description, etc, of the Dataverse collection you'd like to create.
- Figure out the alias or database id of the "parent" Dataverse collection into which you will be creating your new Dataverse collection.
- Execute a curl command or equivalent.

For our case, instead of executing a curl command we would be using a client application. The native API exposes many functionalities, including but not limited to:

- Dataverse collection creation, editing and uploading
- Dataset exploration, viewing, creating, editing and uploading
- File management of datasets
- Metadata management and editing
- Manage available, standard licenses

The Metrics API provides counts of downloads, datasets created, files uploaded, and many more.

- Access and read metrics of multiple categories: By month, type and time intervals
- Specify the return format, filter by parameters such as geolocation
- Access the endpoint table of currently available, up to date metric endpoints.

6.2.3 Python application for Concorda

Architecture

The implementation of the Concorda API is able to conduct the following actions:

- Create a new collection
- Search and read via queries of collections
- Upload files to a collection
- Delete a collection

The Concorda API implements functions of the AbstractController. Due to this, the ConcordaController can be used to initialize the RepositoryInterface for communicating with Concorda.

Example communication

The following chapters explain some example communication in detail.

Creating a collection

```
def create_new_deposition(self):
    url = self.base_url + "/dataverses/1"
    print(url)
    data = {}
    data['name'] = str(uuid.uuid4())
    data['alias'] = str(uuid.uuid4())
    data['dataverseContacts'] = [{"contactEmail": "unnamed@example.edu"}]
    response = requests.post(url, params={'key': self.access_token},
                             data=json.dumps(data),
                             headers=self.headers)
    return response
```

Figure 6.8: Implementation code of creating the deposition

The implementation of the collection creation can be seen in figure 6.8. Creating a new collection is executed with no parameters. The application automatically sets required fields for creating a collection. These fields are:

- name
- alias
- dataverseContacts

Once a new collection is created the following response is returned by the web server:

```
{
"status": "OK",
"data": {
        "id": 1923414,
        "alias": "d35e5887-4537-4554-8fb4-997a8802b939",
        "name": "d35e5887-4537-4554-8fb4-997a8802b939",
        "dataverseContacts": [
                {
                        "displayOrder": 0,
                        "contactEmail": "unnamed@example.
                           edu"
                }
        ],
        "permissionRoot": true,
        "dataverseType": "UNCATEGORIZED",
        "ownerId": 1,
        "creationDate": "2022-04-29T10:56:41Z"
    }
}
```

Status refers to the communication status. Data contains all the fields that were set during the creation. Name and alias are used to refer to the collection by name. The dataverseContacts field contains information about the collection's owner. Displayorder s a field we can use to order the contacts. PermissionRoot is used to set the permission setting to be root. DataverseType can have various publication types, according to the category of our collection. Ownerid is generated by the server, and creation date is as-is.

Searching via query

This is such an action that takes in a parameter as well. The parameter is the query text which is a single string. Executing this action invokes the API to the searching function of the collection, and returns an appropriate response of all the results the server has found. When we search for "test" we get multiple collections, which were created in the sandbox environment.

```
"status": "OK",
"data": {
        "q": "test",
        "total_count": 272,
        "start": 0,
        "spelling_alternatives": {},
        "items": [
                {
                "name": "Test Test Test Test Dataverse",
                "type": "dataverse",
                "url": "https://demo.dataverse.org/
                   dataverse/testtesttesttest",
                "identifier": "testtesttesttest",
                "published_at": "2022-04-29T23:46:05Z"
                },
```


The list goes on for the "items" field of this response, however one is shown here. The type can be seen here as well, which is named "dataverse". In the previous example, this type was set to be uncategorized.

6.3 The shared API

6.3.1 RepositoryInterface

The API is a shared abstract definition of how we can use the repositories in a common manner. The implementation of the Zenodo API and the Concorda API are derived from this shared API which in the project is called "AbstractController". This abstract controller is then used in the RepositoryInterface to conduct the actions we want to execute. These actions are:

- Setting the access token for communication
- Searching by general query in the repository
- Creating, reading deleting a new collection in the repository
- Uploading files to a collection

Any developer who implements the AbstractController functions may use the RepositoryInterface too for communication. The RepositoryInterface class is initialized first with an instance that implements the functions of the AbstractController. Once initialized, the RepositoryInterface's methods can be invoked. Some of the actions of the RepositoryInterface require input data, but the input data are shared across all implementations. The implementations, may use additional variables in their own code, but when the shared API is invoked, the inputs will always be the same.

This way, multiple repositories, such as Zenodo and Concorda can be used by applying the same architecture, meaning: given the same input data and invoking

the same functions, the outcome will also be the same. Using the create function of the RepositoryInterface, by providing the implementation for the repositories, we are able to execute the create functions through the API's of both Zenodo and Concorda for example. This is true for the other functions as well, such as deleting a collection, searching via queries or uploading files to one of the repositories.

6.3.2 Controllers

Any developer who wish to use this API must create a "controller" class, that implements the following functions of the AbstractController:

- set_access_token(self, token)
- get_access_token(self)
- create_new_deposition(self)
- upload_files_to_deposition(self, dir_path, deposition_id)
- def delete_deposition(self, deposition_id)

The controllers may also use additional functions as well, but must implement all the functions of the RepositoryInterface. If the developer uses an extension of the RepositoryInterface and the AbstractController, then this rule is not broken, as the functionality is only extended, and is not reduced in any shape or form.

7 Testing

Functional testing is done for both of the repositories implemented in Python app. Performance measurement is out of scope of testing, as this python application consist of wrapper classes and simple logic that do not add any measurable overhead to the communication towards the web-servers.

The two repositories tested are Zenodo and Concorda. The same functions are invoked in a "for cycle" of an array which has two options: Concorda and Zenodo. This is important to note, as this nicely demonstrates that an arbitrary amount of repository API's (for our case, Zenodo and Concorda) can be used in a unified way from one single interface. The implemented code for the test is shown in figure 7.1.

```
for repoStr in repositories:
    print("DEBUG: " + repoStr)
    repo = RepositoryInterface(repoStr)
    if (repoStr == "zenodo"):
        repo.set_access_token(ZENODO_ACCESS_TOKEN)
        data = json.loads(repo.create_new_collection())
        print(data)
        id = data['id']
    elif(repoStr == "concorda"):
        repo.set_access_token(CONCORDA_ACCESS_TOKEN)
        data = json.loads(repo.create_new_collection())
        print(data)
        data = data['data']
        id = data['name']
    print(repo.search_by_general_query_param("zenodo"))
    fileResponse = repo.upload_files_to_collection(path, id)
    print(fileResponse.status_code)
    print(fileResponse.text)
    deleteReponse = repo.delete_collection(id)
    print(deleteReponse.status_code)
    print(deleteReponse.text)
```

Figure 7.1: Code which defines the testing logic

First a new collection is created and its ID is fetched. Then, a regular search query is executed on the repository. After the text search, a file upload is attempted on the collection ID that was just fetched after creation. Having done the upload, this collection is then deleted as well.

For each step, prints are done to the standard output console, from which we can read the HTTP response codes, and if available, the response text as well delivered in the payload. Analysis of the response data are not part of this section, as they are explained in previous chapters of the Python implementation for Zenodo and Concorda.

The reason for having written an else-if for the first part is that the structure of the JSON in the response of creating a new collection is different for Zenodo and Concorda. For Zenodo, the ID is an immediate first-level field in the json response.

For Concorda, the ID is stored as a field of an other field called "data". First "data" field has to be fetched, and then we are able to fetch the "id" field from this one.

7.1 Zenodo

Console logs for Zenodo testing are shown in figure 7.2

```
DEBUG: zenodo
DEBUG: start create new collection
{'conceptrecid': '1059096', 'created': '2022-05-09T06:22
DEBUG: start search by query
[{"conceptdoi": "10.5072/zenodo.982321", "conceptrecid":
DEBUG: start upload files to collection
201
{"checksum": "77ff1ae4f09aa69bc3717d32113c8214", "filena
DEBUG: delete collection
204
```

Figure 7.2: Console logs of Zenodo test

The Zenodo collection was created successfully, and a query for the text "Zenodo" was also conducted successfully. New files were uploaded to the collection for which an HTTP 201 status code (which stands for "OK: Created") was returned along with additional data of the files uploaded in the response json. For the delete action, no JSON is returned, but an HTTP 204 (which stands for "OK: No content"), which means that no additional information is returned. The response for deletion will be different for Zenodo and Concorda in this manner.

7.2 Concorda

Console logs for Concorda testing are shown in figure 7.3

```
DEBUG: concorda
DEBUG: start create new collection
{'status': 'OK', 'data': {'id': 1963510, 'alias': '5c93cb63-f649-49d6-8a21-c12e4bc0e55f', '
DEBUG: start search by query
{"status":"OK","data":{"q":"zenodo","total_count":2,"start":0,"spelling_alternatives":{},"i
DEBUG: start upload files to collection
500
{}
DEBUG: delete collection
200
{"status":"OK","data":{"message":"Dataverse 5c93cb63-f649-49d6-8a21-c12e4bc0e55f deleted"}}
```

Figure 7.3: Console logs of Concorda test

Concorda collection creation is successful as well. For the "zenodo" text query, results are found on Concorda as well among the searchable collections. Note, that testing the file upload was not possible at the time of conducting tests, as internal

server issues were present at the sandbox environment of Dataverse.org, where the API tests were conducted.

Console logs show an HTTP 500 status code (Internal server error) in the response, with no additional text returned. However, the implementation of the file upload is correct: the server still validates the format of the request sent by the client.

For cases when the request format was incorrect, the server still returned different error codes and error messages that explained why the format was wrong in detail. These errors all fall under the HTTP 400 (Bad request) error code. Once these format errors were corrected, no such errors were returned by the web server. Unfortunately we are unable to circumvent internal server errors, as these errors can only be fixed by the server provider.

For deleting a collection, an HTTP 200 (OK) response is delivered, along with a message in a JSON that the deletion was successful (as compared to Zenodo, which only returns an HTTP status code with no additional message).

8 Future work

The two implementations (Zenodo and Concorda API) do not contain all the available functions that we are able to use on the web UI of these repositories. These implementations are sufficient enough to demonstrate the plan, and the workflow of how implementation for the shared API are to be created and used. These two serve as good bases for further development of these two repositories.

The implementations can be extended with additional functionality that are already server on the web UI of Zenodo and Concorda, such as role management, and advanced search queries. Additional unique metadata can be set for these as well, which can be added as part of a future work.

Implementations for new data repository services can be developed as well. As the shared API contains basic functions that all data repositories are capable of (uploading, reading, editing, deleting), additional implementations for data repositories other than Zenodo and Concorda are possible to be made. This way, the application is extended with support of additional data services.

9 Conclusion

Service providers are still in the process of releasing and establishing their systems on the European Open Science Cloud, some services do not have well-detailed descriptions, categorizations and information on required access rights. Due to this reason I have made a categorization and study, in order to have a general overview of the data services available in EOSC.

The categorization and study of the EOSC data services helped us in choosing data services that match our needs and interests. This need was to find a service which we can use to upload, store and publish scientific research data and metadata. The categorization helped us to select and to focus on data services that hold the potential for being used at SZTAKI (Institute for Computer Science and Control) for aiding the NEANIAS (Novel EOSC Services for Emerging Atmosphere, Underwater & Space Challenges) [1] project.

I have introduced the principles and benefits of Open Science and FAIR data. After understanding the Open Science and FAIR data principles, I have studied some services that are compliant to these principles, and that are eligible candidates for our needs. We are able to use these data services via application programming interfaces (APIs) through the HTTP, and exchange information using standard data formats, such as JSON or XML.

I created a unified repository API, for which I have also created two implementations: Zenodo and Concorda. The unified API is then invoked by a client I have created in the application. The goal of the API is to hide the difference of using the Zenodo and Concorda API implementations. This makes the application widely extensible, with an arbitrary number of API's implemented for the shared API.

10 Összefoglalás

A szolgáltatások fenntartói jelenleg is folyamatában vannak a rendszereik kiajánlásában és publikálásában az EOSC-on (Európai Nyílt Tudományos Felhő) Előfordult, hogy némelyik szolgáltatás leírása nem volt kellően részletes, vagy nem megfelelő kategóriába tartozott, vagy pedig nem volt elég egyértemű az információ az elérési jogok megszerzéséről. Ezen okok miatt végeztem saját kategorizálást és elemzést annak érdekében, hogy legyen egy általános rálátásunk arra, hogy milyen adatszolgáltatások érhetőek el az EOSC-on.

Ezen elemzés segített abban, hogy kiválaszthassunk olyan adatszolgáltatásokat, melyek megfelelőek az igényeink számára. A választott szolgáltatások képesek tudományos adatok és meta-adatok felöltésére, tárolására és publikálására. Sikerült választani olyan szolgáltatásokat, amelyek segítségére válhatnak a SZTAKI (Számítástechnikai és Automatizálási Kutatóintézet) számára, a NEANIAS (Novel EOSC Services for Emerging Atmosphere, Underwater & Space Challenges)[1] projekten való munkában.

Bemutattam a Nyílt Tudomány (Open Science) és a FAIR data irányelveit. Ezen alapok részletezése után, olyan szolgáltatásokat elemeztem, amelyek megfelelnek ezen irányelveknek, és megfelelnek a saját igényeinknek is. Ezeket az adatszolgáltatásokat alkalmazásprogramozási felületeteken (API-kon) keresztül használhatjuk, a HTTP-n keresztül. Az adatok cseréje standardizált formátumban történik meg, például JSON vagy XML formátumban.

Létrehoztam egy közös adattár API-t, amihez létrehoztam 2 implementációt is, ami a mi esetünkben Zenodo és Concorda. A közös adattár API-t egy klienssel használom, az applikáción belül. Az API célja, hogy elrejtse az implementációk közötti különbségeket. Az alkalmazás ezáltal jól bővíthető, bármely mennyiségű implementációval, melyek használják és bővítik a közös adattár API-t.

List of Figures

References

[1] Neanias. (https://www.neanias.eu/), accessed May 22, 2020.

[2] Paul Ayris, Jean-Yves Berthou, Rachel Bruce, Stefanie Lindstaedt, Anna Monreale, Barend Mons, Yasuhiro Murayama, Caj Södergård, Klaus Tochtermann, and Ross Wilkinson. Realising the european open science cloud. 2016.

[3] What is open science? https://www.openaire.eu/what-is-open-science, last updated 26 September 2018.

[4] Strategic roadmap. https://ec.europa.eu/research/openscience/pdf/eosc_strategic_implementation_roadmap_large.pdf, accessed 19 May, 2020.

[5] Neanias about. (https://www.neanias.eu/index.php/about-neanias-project), accessed Dec 14, 2021.

[6] Neanias context. (https://www.neanias.eu/index.php/about-neanias-project/2-uncategorised/455-the-context-and-neanias-challenges), accessed Dec 14, 2021.

[7] Fair principles. https://www.go-fair.org/fair-principles/, accessed 20 May, 2020.

[8] Barend et al. Mons. Cloudy, increasingly fair; revisiting the fair data guiding principles for the european open science cloud. https://content.iospress.com/articles/information-services-and-use/isu824, Mar 7, 2017.

[9] Openaire graph access. https://marketplace.eosc-portal.eu/services/openaire-graph-access-api-d6c67a59-0a33-4ebf-b86c-726cda41d365, accessed 21 May, 2020.

[10] Collaboration framework. https://marketplace.eosc-portal.eu/services/collaboration-framework, accessed 21 May, 2020.

[11] Dmp. https://marketplace.eosc-portal.eu/services/dmponline?fromc=data-management, accessed 21 May, 2020.

[12] https://marketplace.eosc-portal.eu/services/argos?fromc=data-management, accessed 21 May, 2020.

[13] Openaire dashboard. https://marketplace.eosc-portal.eu/services/openaire-research-community-dashboard?fromc=data-management, accessed 21 May, 2020.

[14] Zenodo. (https://marketplace.eosc-portal.eu/services/zenodo), accessed Nov 12, 2021.

[15] B2share. https://marketplace.eosc-portal.eu/services/b2share, accessed 21 May, 2020.

[16] Figshare. https://marketplace.eosc-portal.eu/services/figshare-for-institutions, accessed 21 May, 2020.

[17] Midas. https://marketplace.eosc-portal.eu/services/the-national-open-access-research-data-archive-midas, accessed 21 May, 2020.

[18] Openaire content provider dashboard. https://marketplace.eosc-portal.eu/services/openaire-content-provider-dashboard, accessed 21 May, 2020.

[19] Clowder. https://marketplace.eosc-portal.eu/services/v-seem-clowder, accessed 21 May, 2020.

[20] Open data platform. https://marketplace.eosc-portal.eu/services/access-to-open-data-platforms, accessed 21 May, 2020.

[21] Amnesia. https://marketplace.eosc-portal.eu/services/amnesia-3d9b3e7a-e574-4626-9182-5451fb03eb2b, accessed 21 May, 2020.

[22] Bionym. https://marketplace.eosc-portal.eu/services/bionym, accessed 21 May, 2020.

[23] B2handle. https://marketplace.eosc-portal.eu/services/b2handle, accessed 21 May, 2020.

[24] Egidat. https://marketplace.eosc-portal.eu/services/egi-data-transfer, accessed 21 May, 2020.

[25] B2note. https://marketplace.eosc-portal.eu/services/b2note, accessed 21 May, 2020.

[26] Data mining. https://marketplace.eosc-portal.eu/services/scalable-data-mining, accessed 21 May, 2020.

[27] Dynamic report. https://marketplace.eosc-portal.eu/services/dynamic-reporting, accessed 21 May, 2020.

[28] Nym. https://marketplace.eosc-portal.eu/services/nym-framework, accessed 21 May, 2020.

[29] B2safe. https://marketplace.eosc-portal.eu/services/b2safe, accessed 21 May, 2020.

[30] Egi archive. https://marketplace.eosc-portal.eu/services/egi-archive-storage, accessed 21 May, 2020.

[31] Snap4city architecture, users. https://www.snap4city.org/drupal/node/57, accessed 21 May, 2020.

[32] B2find. https://marketplace.eosc-portal.eu/services/b2find, accessed 21 May, 2020.

[33] Lemonade. https://marketplace.eosc-portal.eu/services/lemonade-live-exploration-and-mining-of-a-non-trivial-amount-of-data-from-everywh accessed 21 May, 2020.

[34] About lemonade. https://www.lemonade.org.br/index-en.html#about, accessed 21 May, 2020.

[35] Fairsharing. https://marketplace.eosc-portal.eu/services/fairsharing, accessed 21 May, 2020.

[36] About fairsharing stakeholders. https://fairsharing.org/stakeholders/, accessed 21 May, 2020.

[37] Identifiers.org. https://marketplace.eosc-portal.eu/services/identifiers-org-resolution-services, accessed 21 May, 2020.

[38] etdr. https://marketplace.eosc-portal.eu/services/etdr-european-trusted-digital-repository, accessed 21 May, 2020.

[39] Open telekom services. https://open-telekom-cloud.com/en/products-services, accessed 21 May, 2020.

[40] Open telekom cloud. https://marketplace.eosc-portal.eu/services/open-telekom-cloud, accessed 21 May, 2020.

[41] Openaire support. https://guidelines.openaire.eu/en/latest/, accessed 21 May, 2020.

[42] Openaire validator. https://www.openaire.eu/validator/, accessed 21 May, 2020.

[43] Openaire catchall. https://provide.openaire.eu/home, accessed 21 May, 2020.

[44] Tds. https://marketplace.eosc-portal.eu/services/tds, accessed 21 May, 2020.

[45] Jelastic. https://marketplace.eosc-portal.eu/services/jelastic-platform-as-a-service, accessed 21 May, 2020.

[46] About nomad. https://www.nomad-coe.eu/index.php?page=nomad-repository, accessed 21 May, 2020.

[47] Nomad. https://marketplace.eosc-portal.eu/services/nomad-repository, accessed 21 May, 2020.

[48] Aiida. https://marketplace.eosc-portal.eu/services/aiida-lab, accessed 21 May, 2020.

[49] About aginfra+. http://plus.aginfra.eu/about, accessed 21 May, 2020.

[50] Aginfra+ geoanalytics. https://marketplace.eosc-portal.eu/services/aginfra-chart-visualization, accessed 21 May, 2020.

[51] Aginfra+ elastic. https://marketplace.eosc-portal.eu/services/aginfra-agris-elastic-index, accessed 21 May, 2020.

[52] Aginfra+ semantic. https://marketplace.eosc-portal.eu/services/aginfra-semantic-linking-service, accessed 21 May, 2020.

[53] Aginfra+ ontology. https://marketplace.eosc-portal.eu/services/aginfra-ontology-engineering-service, accessed 21 May, 2020.

[54] https://marketplace.eosc-portal.eu/services/aginfra-data-transformation-service.

[55] Aginfra+ chart. https://marketplace.eosc-portal.eu/services/aginfra-chart-visualization, accessed 21 May, 2020.

[56] Cloudferro data collections. https://marketplace.eosc-portal.eu/services/cloudferro-data-collections-catalog, accessed 21 May, 2020.

[57] Cloudferro eo browser. https://marketplace.eosc-portal.eu/services/cloudferro-data-related-services-eo-browser, accessed 21 May, 2020.

[58] Cloudferro eo finder. https://marketplace.eosc-portal.eu/services/cloudferro-data-related-services-eo-finder, accessed 21 May, 2020.

[59] Aquaculture spatial catalog. https://marketplace.eosc-portal.eu/services/aquaculture-atlas-generation-spatial-data-catalog, accessed 21 May, 2020.

[60] Aquaculture spatial data repository. https://marketplace.eosc-portal.eu/services/aquaculture-atlas-spatial-data-repository, accessed 21 May, 2020.

[61] Argo floats data discovery. https://marketplace.eosc-portal.eu/services/argo-floats-data-discovery, accessed 21 May, 2020.

[62] About argo floats data discovery. http://www.argo.ucsd.edu/, accessed 21 May, 2020.

[63] https://marketplace.eosc-portal.eu/services/data-driven-atlas-production.

[64] Tuna atlas spatial. https://marketplace.eosc-portal.eu/services?service_id=&anchor=&sort=_score&q=tuna+atlas+spatial, accessed 22 May, 2020.

[65] Indian ocean tuna spatial data catalog. https://marketplace.eosc-portal.eu/services/indian-ocean-tuna-commission-spatial-data-catalog, accessed 22 May, 2020.

[66] Global records of stocks and fishery. https://marketplace.eosc-portal.eu/services/global-record-of-stocks-and-fishery, accessed 21 May, 2020.

[67] Fishery stock management. https://marketplace.eosc-portal.eu/services/regional-database-for-fishery-stock-management, accessed 21 May, 2020.

[68] Seadatanet. https://www.seadatanet.org/Standards/, accessed 21 May, 2020.

[69] Spatial planning. https://marketplace.eosc-portal.eu/services/spatial-planning, accessed 21 May, 2020.

[70] Stock assessment support. https://marketplace.eosc-portal.eu/services/stock-assessment-support, accessed 21 May, 2020.

[71] Atlantic fishery data catalog. https://marketplace.eosc-portal.eu/services/western-central-atlantic-fishery-commission-spatial-data-catalog, accessed 21 May, 2020.

[72] Atlantic fishery data repository. https://marketplace.eosc-portal.eu/services/western-central-atlantic-fishery-commission-spatial-data-repository, accessed 21 May, 2020.

[73] Aquaculture farming. https://marketplace.eosc-portal.eu/services/aquaculture-farming, accessed 22 May, 2020.

[74] Biodiversity. https://marketplace.eosc-portal.eu/services/biodiversity, accessed 22 May, 2020.

[75] Species modeling. https://marketplace.eosc-portal.eu/services/species-modeling, accessed 22 May, 2020.

[76] D4science spatial catalog. https://marketplace.eosc-portal.eu/services/d4science-spatial-data-catalog, accessed 22 May, 2020.

[77] D4science spatial data repository. https://marketplace.eosc-portal.eu/services/d4science-spatial-data-repository, accessed 22 May, 2020.

[78] Protected area impacts. https://marketplace.eosc-portal.eu/services/protected-area-impact-maps-spatial-data-repository, accessed 22 May, 2020.

[79] Spatial data infrastructure laboratory. https://marketplace.eosc-portal.eu/services/spatial-data-infrastructure-laboratory-catalog, accessed 22 May, 2020.

[80] Eodc data catalogue. https://marketplace.eosc-portal.eu/services/eodc-data-catalogue-service, accessed 22 May, 2020.

[81] Sentinel hub. https://marketplace.eosc-portal.eu/services/sentinel-hub, accessed 22 May, 2020.

[82] 3dbionotes. https://marketplace.eosc-portal.eu/services/3dbionotes, accessed 22 May, 2020.

[83] Dipcheck. https://marketplace.eosc-portal.eu/services/dipcheck, accessed 22 May, 2020.

[84] Metalpdb. https://marketplace.eosc-portal.eu/services/metalpdb, accessed 22 May, 2020.

[85] Europana apis. https://marketplace.eosc-portal.eu/services/europeana-apis, accessed 22 May, 2020.

[86] Digital humanities. https://marketplace.eosc-portal.eu/services/digital-humanities-and-cultural-heritage-openaire-community-gateway, accessed 22 May, 2020.

[87] Aila. https://marketplace.eosc-portal.eu/services/data-service-portal-aila, accessed 22 May, 2020.

[88] da—ra. https://marketplace.eosc-portal.eu/services/da-ra-doi-registration-agency-for-social-and-economic-data, accessed 22 May, 2020.

[89] Transkribus. https://marketplace.eosc-portal.eu/services/transkribus, accessed 22 May, 2020.

[90] Datafurn. https://marketplace.eosc-portal.eu/services/datafurn-furniture-enterprise-analytics, accessed 22 May, 2020.

[91] Figshare fair. (https://figshare.com/articles/journal_contribution/Figshare_and_the_FAIR_data_principles/7476428), accessed Nov 12, 2021.

[92] Zenodo fair. (https://about.zenodo.org/principles/), accessed Nov 12, 2021.

[93] Concorda website. (https://science-data.hu), accessed March 27, 2022.

[94] Concorda sztaki. (https://www.sztaki.hu/en/innovation/projects/concorda), accessed March 27, 2022.

[95] Zenodo rest. (https://developers.zenodo.org/?python#rest-api), accessed Nov 12, 2021.

[96] Kőhegyi Marcell. Zenodo python application 1.0, DOI: 10.5281/zenodo.5779119. https://doi.org/10.5281/zenodo.5779119. published Dec 14, 2021.

[97] Concorda rest api. (https://guides.dataverse.org/en/latest/api/index.html), accessed March 27, 2022.