Aristotle University of Thessaloniki
Faculty of Engineering
School of Electrical and Computer Engineering
Department of Electronics & Computers

Creating an Open Archival Information System compliant archive for CERN

Diploma Thesis
of
Chelakis Konstantinos Marios
8944

Supervisors: Andreas L. Symeonidis
Professor A.U.Th
Themistoklis Diamantopoulos
Postgraduate Researcher
CERN Supervisors: Jean-Yves Le Meur
Antonio Vivace

December 5, 2022

Abstract

Nowadays, we constantly produce data in an unprecedented scale at various domains. In the context of research data, large organizations, like CERN, produce information which is of significant importance and which cannot be reproduced in the future. It is therefore our responsibility to make sure that this information is preserved in a way that it will be available to the future generations.

This challenge, which is broadly referred to as digital preservation, has drawn the attention of several researchers and led to the design of a standard for long-term digital data storage, known as the Open Archival Information System (OAIS) standard. Several systems have been developed towards this direction, however these solutions were either not fully-compliant with the OAIS standard, some were short term projects that have been decommissioned or they were not open-source and available to the research community. In this context, CERN proposed the Digital Memory project, a digital archiving initiative that should allow researchers to archive their data in a way that it will be accessible in the future.

In this thesis, which is part of the Digital Memory project, we confront the aforementioned challenges by proposing an architecture that is fully OAIS-compliant, is integrated with CERN repositories and supports transparency, as the user can easily manage and monitor the actions performed on archival packages. Initially, we implement a tool that can be used to harvest data from various CERN sources like CDS, Indico, CERN Open Data, Gitlab and CodiMD in an OAIS-compliant format called Submission Information Package (SIP). This package can be supplied to the platform in order to create the actual archival packages that can be stored for long term preservation. These packages contain additional metadata and normalization of content that will guarantee long term survival of the information content. Additionally, we show how easy it is for a user to create, monitor and group their archives by using the User Interface. The platform can be easily deployed by anyone on Openshift with the use of Helm charts. Concerning our evaluation we discuss how the the performance of the platform can be improved and we show that the resulting packages as well as the platform as a whole is fully OAIS-compliant.

Special Thanks

I would like to thank Antonio and Jean-Yves from giving me the opportunity to work at CERN and to such an interesting project. I am very grateful for working in the past 14 months in the Digital Memory Project and for all the things I learned and the experience I gained that for sure changed my life.

I also want to thank my professors Andreas Symeonidis and Themistoklis Diamantopoulos for their guidance and assistance. I would like to say a special thanks to Chrysa who supported me all these months and to all my friends both the new and old with whom we created unforgettable memories. Last but not least, the biggest thanks I would like to give to my family, without whom I could not be here.

Contents

Chapter 1

Introduction

The European Organization for Nuclear Research (CERN) is the biggest particle physics laboratory in the world and the largest research center in Europe. It consists of many experiments which use particle accelerators to conduct high-energy physics research. CERN was founded in 1953 to promote nuclear research for peaceful purposes and to promote the cooperation of European states to achieve this goal. Since its creation, CERN has become an international collaboration that aims to answer to fundamental questions like what the universe is made of and how it works. According to CERN's founding convention,[1] its purpose is to promote nuclear research of a purely scientific character, to build particle accelerators for the purpose of carrying out experiments, to promote cooperation between scientists and the dissemination of the information that is obtained to the rest of the world.

The Large Hadron Collider (LHC) is the world's largest and most powerful particle accelerator.[2] Inside the accelerator, two high-energy particle beams are made to collide at four different locations where the particle detectors ATLAS, CMS, Alice, and LHCb are. The data produced in these experiments is enormous. In CMS alone, more than 1 billion interactions take place every second[3] so in order to filter the useful data, a triggering system is implemented. In the first level of the triggering 100.000 most significant events are kept and out of this number, the 100 most significant are selected while the rest are thrown out. As of 2018,the event size is calculated at 7.4 MB per event and the trigger rate goes up to 750 kHz. Even if we select 100 of the events of this size for storage, this means that we need to store 740MB of data per second.

In the latest years, we can see that the amount of information produced every second has grown rapidly. Based on the information above, at this moment, CERN stores more than 340 petabytes of data.[4] This figure will increase at it is expected that CERN will produce exabytes of data by 2030. [1]

[1]https://council.web.cern.ch/en/content/convention-establishment-european-organization-nuclear-research

[2]`https://home.cern/science/accelerators/large-hadron-collider`

[3]`https://cms.cern/detector/triggering-and-data-acquisition`

[4]`https://home.cern/news/news/computing/cern-data-storage-gets-ready-run-3`

1.1 The challenge of Data Archiving

Ever since the creation of CERN, great importance was given to the storage of data but despite of that, there are cases of data loss. For instance, the Large Electron-Positron Collider had produced around 500 terabytes of data by 2000 when it was disassembled for the creation of the LHC. At that point, in cooperation with the IT department, the data was copied on tapes for preservation. Unfortunately, two of the tapes were lost [2].

Problems such as loss of data or hardware errors are not the only threats. Many programs in the past were coded in languages that cannot be compiled anymore and there are file types that cannot be opened because there are no associated programs to open these files. Technology evolves and with it the way and means we store data. Sometimes, data is not well preserved because the value at that time cannot be foreseen. An example like this is the first web page created at CERN which was deleted after some updates. Later when the significance of the first website was found, there were initiatives to restore it[5]. Data archiving seems not important until we find out we need this data and we don't have it or we lack the means and the tools to decode it. It is our duty to preserve important data such as papers and data from experiments as is it very likely that these will be important for future generations.

1.2 Creating a Digital Archiving Platform for CERN

The awareness of creating a digital archiving platform is increasing lately. The amount of data produced is unprecedented so the need of archiving valuable data is more relevant than ever. In many organizations like CERN where a lot of knowledge and data are produced, the lack of a digital archiving service leads to the loss of a fraction of this knowledge that inevitably will happen. Also at CERN, there are people with different contracts and with the passage of so many people, inevitably data is lost. For the reasons stated above, CERN created the Digital Memory project with three main goals[6].

First of all, the project aims to digitize content that exists in an analog format such as slides, video and audio tapes, photographs, documents, etc. The project's goal is to digitize these multimedia files that are susceptible to damage and corruption. Also if these files remain untreated, the hardware needed to view and process them may be difficult to find in the future or it may not exist at all. The digitization process is carried out by external partners and the resulting files are stored in CERN.

The second goal of the project is the creation of an OAIS-compliant digital archive by making digital archiving an integral part of all CERN's processes and culture. CERN currently has many sources of data and stores these data in many different digital repositories such as CERN Document Server (CDS), CERN Open Data (COD) etc. All repositories should by default integrate long-term preservation policies in the way they store data. Archiving is cheaper and more efficient when it takes place at the same time when the data is produced. The more this is postponed, the more

[5]https://home.web.cern.ch/science/computing/birth-web
[6]https://digital-memory-project.web.cern.ch

expensive will the preservation process be and the more likely it is that a lot of content will be lost.

The third and final goal is the creation of the Digital Memory platform. This platform will be an application for researchers to preserve digital content. Through the platform, the users will be able to archive their data and create OAIS-compliant information packages. This platform will not be just a backup for the other repositories but a real digital archive where researchers will be able to easily preserve their data.

1.3 Diploma thesis scope

Building a CERN-wide platform that is compliant with the OAIS protocol is a complex task. The amount of data that needs to be stored is huge. As of April 2019, CERN stored 339 Petabytes of data on tapes, and out of them, 115 petabytes were "new" data created in 2018 only. The amount of data we need to archive is enormous and it increases exponentially. This poses a great challenge to the creation of the archival system. Fortunately, preservation policies are enforced on raw experimental data so the use of the platform should mainly focus on other types of data such as documents, multimedia, etc. A big challenge is the platform integration with existing data repositories at CERN and how the platform can retrieve and create submission archival packages from these repositories in a consistent way. The platform must be composed of a user-friendly user interface that will make easy for the archiver, to guide through the available repositories and files and easily create the archives they need and to follow the archival process.[7]

The scope of this diploma thesis is the implementation and explanation of a CERN-wide web platform for content archiving compliant with the OAIS standard that fulfills all the prerequisites stated above. This is going to be a full-stack web application containing the backend/server which will handle the creation of the packages and the archival process and a user interface/frontend deployed within the CERN network. The application should be able initially to retrieve data from the most commonly used repositories and applications used within CERN such as the CERN Document Server (CDS), Zenodo, InvenioRDM, CERN Open Data (COD), Indico, CodiMD, and Gitlab. The application and the archives produced should be fully compliant with the OAIS standard which will be presented in the next chapter. In the UI, the user should be able to search and select for archiving all his own records and any other publicly available record and also to get any information related to the archival process of the selected records.

1.3.1 Diploma Thesis Overview

In chapter 2, we will explain why digital preservation is very important, we will explain the basic components of the OAIS standard and we will show some examples of digital preservation systems implemented in the past. In chapter 3, we will explain the basic features of the platform and the user interface as well as the way the

[7]`https://information-technology.web.cern.ch/sites/default/files/CERNDataCentre_KeyInformation_July2019V1.pdf`

platform is deployed. In chapter 4, we will evaluate and improve the performance of the deployed application and we will assess the created packages. In chapter 5, we will make a synopsis of this diploma thesis and we will suggest some improvements for the future.

Chapter 2

Theoretical Background

2.1 Digital Preservation

Digital preservation is the process that aims to ensure the usability and accessibility of digital information over time. According to Harrod's Librarian Glossary, digital preservation is the method of keeping digital material alive so that they remain usable as technological advances render original hardware and software specifications obsolete [3]. Digital preservation is much more than just saving a file to a storage medium. The archiver should define policies to make sure that the archived content remains accessible after many years. These policies aim to protect and ensure the preservation of digital content. Most digital information is now stored in hard drives which can become unusable after many years for various reasons (damaged spindle motor, erosion, humidity, etc). Flash memory can also start losing data if not used for a long time. Although there are technologies that promise they can store data for longer periods like ArchivalDisc and M-DISC, which are designed to store data for 50 and 1000 years accordingly, these are proprietary formats in the first case and in the second case writing and reading data requires special equipment that no one guarantees it will be available after many years. Thus the archiver should design the archiving storage system in such a way that data remains unchanged and readable over time.

Digital preservation is not something absolute. All the policies that are taken target at reducing the possibility of information getting lost or corrupted. The goal is to perform specific sets of actions that together mitigate the risks associated with digital objects. Digital preservation is a part of the digital archiving process. It is a formal task that ensures that digital objects remain findable, accessible, readable and usable. It also covers a spectrum of processes and operations involved in ensuring the technical and intellectual survival of authentic digital objects through time.

2.2 Threats of digital obsolescence

Some of the hazards that a digital object may face are described below: [4]

- **Bit rot:** The change of a single bit of the stored file on RAM.

- **Technical Obsolescence:** There is no software or player to allow the file or media to be opened or played.
- **Format Obsolescence:** The format of the file has become unsupported, fonts and figures can no longer be interpreted correctly.
- **Migrations and transitions:** Software and hardware that is used to store and access to information changes regularly. These changes pose a risk of data loss.
- **Dissipation:** Content poorly described becomes unavailable.
- **Ambiguous intellectual property:** Lack of source information makes it impossible to know if the original or an altered version of the original data is being processed.
- **Human errors and computer attacks:** Can cause great loss of data.

We can see that in longer periods of time the chance of losing data increases as one of the following hazards may happen at some point. That's why the creation of a digital archive is necessary for every institution or entity.

2.3 Digital Dark Ages

In the 80s the world was inside a digital dark age where the amount of data produced back then has mostly been lost. This is the result of outdated file formats, different software and hardware of the time that it is no longer used, and the versatility of the storage mediums of that era. The same danger befalls also today's archives. Historians of the future may not be able to learn about our lives from the digital content we are producing because as the way we store information develops, data that use older storage technologies are becoming harder to access [5].

The first initiative for digital archiving took place in the 90s when the amount of data produced increased exponentially because of the widespread use of the general scope personal computers (PC). Many archivists warned that we were still in the middle of an archive dark age because many different systems were used and some valuable information was already not available to them. In 1990 the Consultative Committee for Space Data Systems (CCSDS) started developing a standard for long-term digital data storage generated initially from space missions. This resulted in the creation of the Open Archival Information System (OAIS) Reference Model the first draft of which was released in 1997 followed by two updates in 2002 and 2012. [6]

2.4 OAIS standard

The first version of a model for an archival system was developed in 2005 by the Consultative Committee for Space Data Systems (CCSDS). [7] As computer power and increasing, networking speeds were becoming available, CCSDS recognized that the volume of information that the number of information these organizations are making increased exponentially. The unrestricted and non-standardized way of producing and storing these data will create problems in keeping these data in the future. There are various issues that need to be considered which if ignored, will lead to the loss of information. The model introduces a series of requirements that set some common concepts and a common framework to face these issues. The main goal is to provide standardization so organizations have a common reference point in order to ensure long-term information preservation.

The OAIS reference model goals are:

1. to provide a framework and raise awareness of the need for long-term preservation
2. to formalize the concepts and the terms used by different organizations in order to have a standard way of development.
3. to provide a framework for comparing different long-term preservation strategies and data models.
4. to provide the foundation for the development of long-term preservation
5. to guide the identification and production of further OAIS-related standards.

The OAIS archive intends to preserve information for access and use by a designated community. They have to keep up with constant streams of information or with aperiodic inputs.

According to the OAIS standard, information is defined as «any type of knowledge that can be exchanged and is always expressed by some type of data in exchange». This information object is critical to have bit-level preservation capability so the final package is exactly identical to the submitted one.

Outside the OAIS archive there are three entities: Producers, Consumers and Management. The producer is the person or the systems that provide the information to be archived. Management are the persons that set the overall archiving policy and the consumers are the persons or systems that interact with OAIS to find and acquire preserved information of interest.

When the producer submits some information to an OAIS and when OAIS disseminates information to the consumer, then this occurs as one or more discrete transmissions, and the information exchanged in this case forms an Information Package. The information package contains the content information and the Preservation Description Information (PDI) which contains associated metadata. The PDI is divided into five types of information called provenance (describes the source of information), context (why the package was created and its association with other packages), reference (unique identifier to refer to the package), fixity (checksums) and access rights (who has access to that package).

In OAIS we can find three types of information packages: the Submission Information Package (SIP), the Archival Information Package (AIP), and the Dissemination Information Package (DIP).

- **SIP:** is the package that is created by the original repository of the object. It must contain the original content of the object and some descriptive metadata.
- **AIP:** is a self-sufficient package that is created to guarantee the survival of the package in the future. It must contain the object in a preservation format, de-

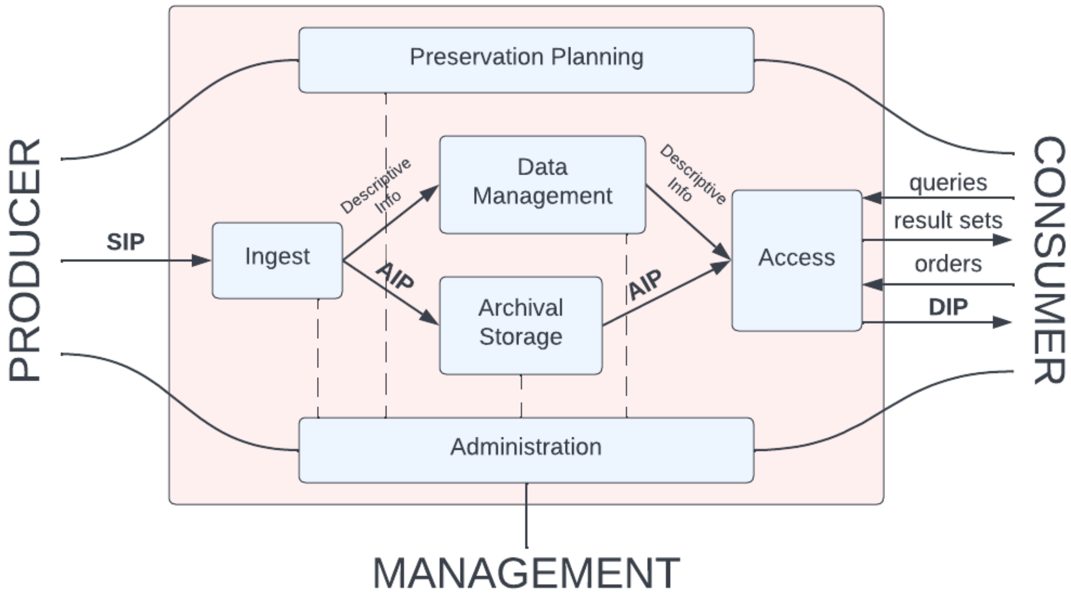


Figure 2.1: OAIS functional entities

scriptive metadata and all the context information such as provenance, rights, etc. as well as information explaining the way this package was created.

- **DIP:** is a package derived from the AIP on request of a user. It contains data derived from the preservation format and additionally simple metadata that allow easier consumption.

Initially, the SIP is submitted by the producer for archival which contains the actual submitted material together with some additional information in the form of metadata which will help to index and recreate the package. The platform then transforms the SIP into AIPs. The AIP contains a complete set of content information and the PDI. The consumer can in the end upon request retrieve the whole or a part of the AIP in the form of a DIP.

The OAIS has some mandatory responsibilities that the organization must take into account in order to create an OAIS archive. The OAIS should:

1. Negotiate for the type and accept appropriate information from the Producers.
2. Have the necessary control over the information so it can ensure long term preservation
3. Define which is the designated community and define the knowledge base and ensure the information is understandable.
4. Ensure the information is preserved against all possible contigencies and ensure the archive is never deleted unles there is a policy change.
5. Make sure the information is available to the designated community and it can be traceable to the original data objects with authenticity evidence.

To perform all these tasks, the OAIS standard has six distinct parts.

1. Ingest: Provides functionality to accept SIPs from the producer. Prepares the contents for storage, extracts metadata from the SIPs and transforms to AIPs.
2. Archival Storage: Stores the AIPs. It provides services to maintain and retrieve the AIPs.
3. Data Management: Manages and queries the database of the archive. Provides all the services for managing information and administrative data.
4. Administration: Provides services to monitor the archive as well as to create and maintain archive standards and policies
5. Preservation Planning: Provides services to ensure the archive information remains accessible.
6. Access: Supports consumers who request and receive information. Creation of DIPs.

2.5 Similar Projects

2.5.1 LOCKSS

LOCKSS program was created by Stanford University in 1995 and the name derives from Lots of Copies Keep Stuff Safe. It provides open-source services for secure digital preservation and it was initially created to store e-journals. Because it was created for this reason, the software architecture is very explicit to handle e-journals. LOCKSS uses a series of boxes that create a peer-to-peer network with each other. [8] The boxes communicate with each other and they locate copies of the same file which they regularly compare. If a node finds out that a copy disagrees with others, it could ask for repair based on the content of the other nodes. LOCKSS advertises itself as providing an OAIS-compliant infrastructure. However, as systems evolve, this compliance is referred to papers before 2013 and there is no such reference to the current website and documentation.

2.5.2 DAITSS

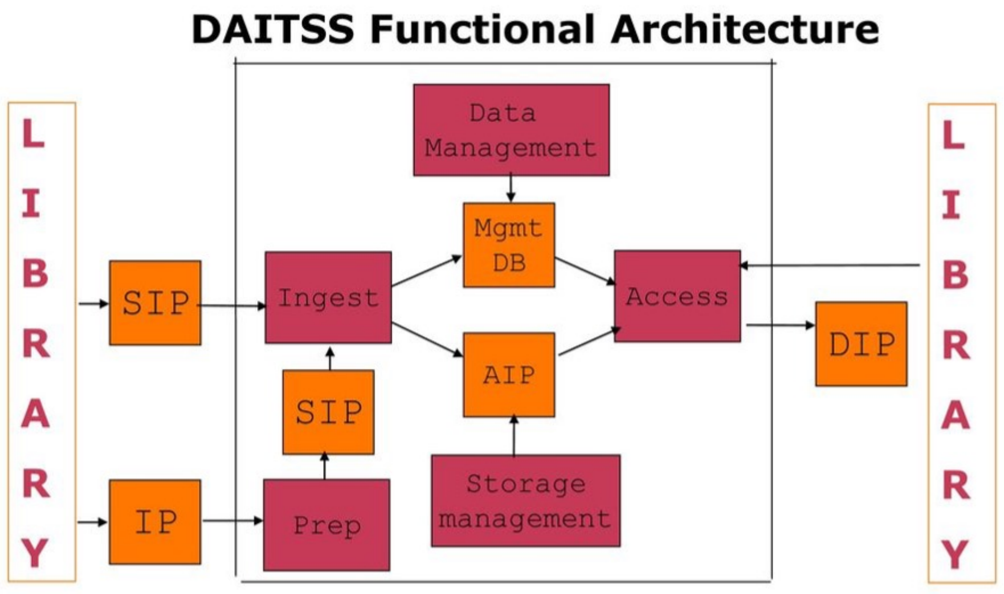


Figure 2.2: DAITSS Architecture [9]

DAITSS (Dark Archives In The Sunshine State) is a digital preservation software application developed by the Florida Center for Library Automation (FCLA). [10] It is an open-source application that provides automated support for the functions of submission, ingest, archival storage, access, withdrawal, and repository management, and according to Florida Academic Library Services Cooperative (FALSC) is made for materials in text, document, image, audio, and video. DAITSS supports a workflow for digital archiving and also provides functions for ingesting, data management, archival storage, and access. DAITSS uses FCLA Digital Archive (FDA) as a backend tool. Moreover, it is a “dark archive”, which means that there is no public access to the archives. Depositors can only request their own archives on request, while those who have not deposited anything, cannot access it at all. To access the archive, the user has to make a dissemination request and if access is approved, then a Dissemination Information Package (DIP) for that user is created. Another disadvantage is that the users can only submit zipped SIP files so they should use another application for the creation of the SIP before using DAITSS to deposit their package. FCLA has decommissioned DAITSS since 2018 and the last contribution to the code was in 2019. [11]

2.5.3 SPAR

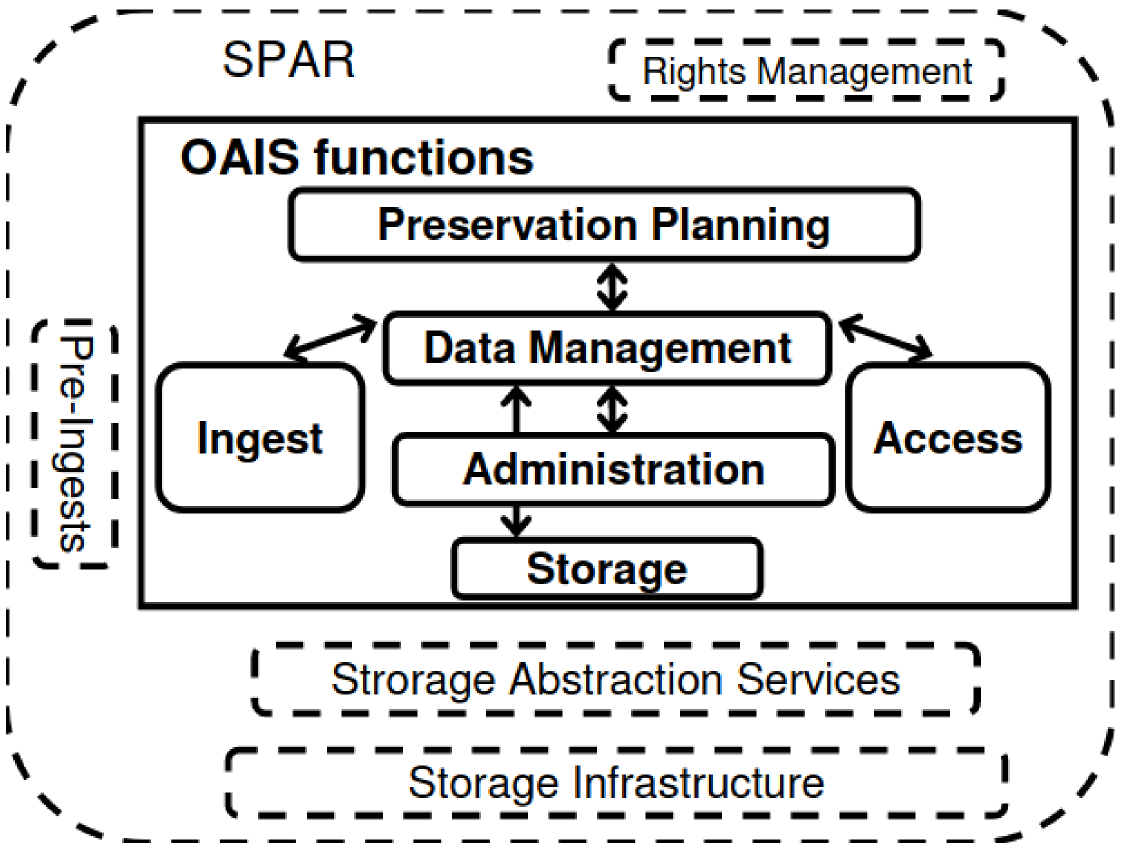


Figure 2.3: SPAR Architecture [12]

SPAR stands for Scalable Preservation and Archiving Repository and it was developed in France by the National Library. SPAR sets as a requirement to be OAIS compliant. The implementation of the SPAR system is divided into two sub-projects: The “SPAR Infrastructure” sub-project aims to acquire and implement the technical infrastructure required by the system. The “SPAR Realization” sub-project aims to build the system in such a way as to store, preserve for the long term and communicate the documents of the library from the available infrastructure. The main concept of SPAR is the track which has a collection of objects with the same preservation policy and homogeneous material. Although this program was created by the National French Library, it is not available as an open-source program to the public. [12]

2.5.4 eARK & eArchiving

eARK stands for European Archival Records and Knowledge Preservation and it was a 3-year project funded by the European Commission from 2014 to 2017. E-Ark is an end-to-end OAIS-compliant archival platform. The objective is to build a robust, single, and scalable approach to meet the needs of different organizations and to support complex data types. The specifications for the Information packages (SIPs, AIPs, DIPs) have been set and tools for each archiving stage were developed such as a SIP Creator tool and a SIP to AIP converter. In order to test and promote the implementation of the platform many use case pilots together with different National Archive Institutions took place. [13] Although the results were rated as excellent by the European Commission, there is no further update on this project and no maintenance.

2.5.5 EPrints

EPrints is an open-source institutional repository platform created by Southampton University. It uses the OAI protocol which means it is a repository to store data but does nothing about the long-term preservation of these data. An effort was made from 2009 to 2011 to create an OAIS-compliant digital preservation tool integrated with EPrints called the JISC KeepIt project. The scope is to enable digital preservation for some institutional repositories (IRs) serving institutions of higher institutions. After a paper issued in 2011, there is no further research on this topic or news about this. [14]

2.5.6 DSpace

DSpace is an open-source archival platform that enables users to build open digital repositories. DSpace was first released in November 2002, and it was a joint effort between MIT and HP Labs. It uses the METS (Metadata Encoding and Transmission Standard) method of content packaging to implement metadata in digital library applications. DSpace's use of the OAIS reference model is patchy but increasing. Since 2003 there were efforts to make DSpace an OAIS-compliant platform. Although there was no significant result or any further commitment to this project, it is not yet OAIS compliant. Same as EPrints, it is used for institutional repositories. Although there are attempts to make DSpace fully OAIS-Compliant, these projects had short-term funding and did not continue. [15]

2.5.7 Dataverse

The Dataverse project started developing at Harvard's Institute for Quantitative Social Science (IQSS). It is mainly used to store, share and preserve valuable research data. The main goal is to make data available to others and make it easier to replicate and analyze each others' work. Dataverse mostly targets the archiving of large datasets. It is an open-source project that when installed, hosts many virtual archives called Dataverse collections. Each of them has datasets and each dataset contains metadata and files. The goal of this project is to provide reliable data to datasets so researchers can give and receive credits for the creation of the datasets. [16]

2.5.8 Archivematica

Archivematica is a web-based, open-source, and OAIS-compliant digital preservation system that was created in 2010. Archivematica was created to reduce the complexity, and the price and to simplify digital curation. This is achieved by the implementation of some microservices which perform a specific archiving task. In this case, the microservices are grouped together in the OAIS workflow categories which correspond to a different type of OAIS package (SIP, AIP, DIP).

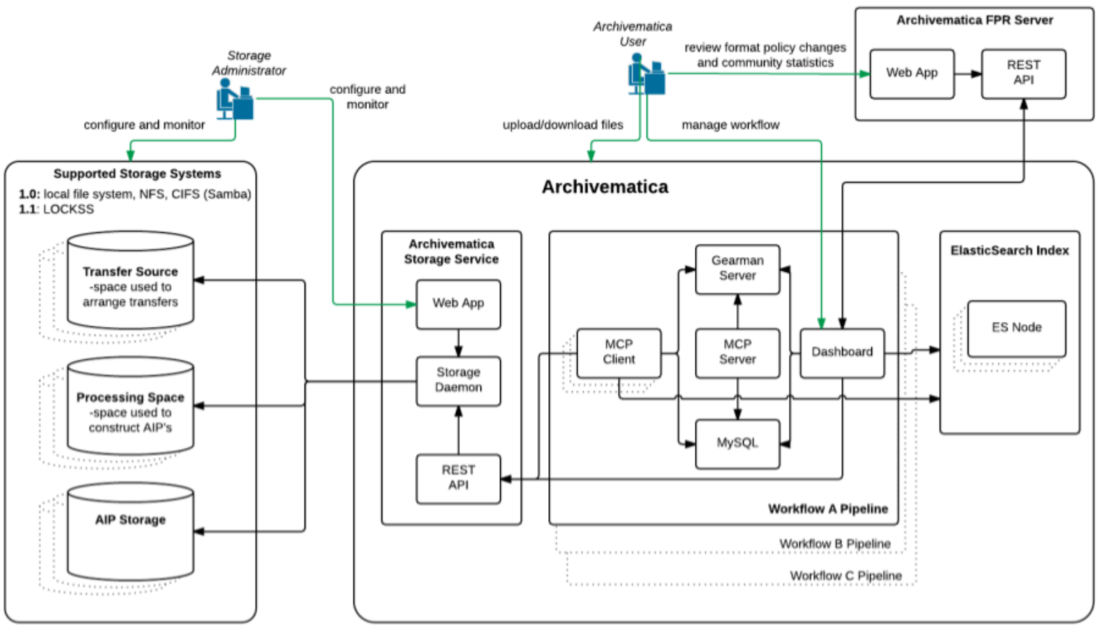
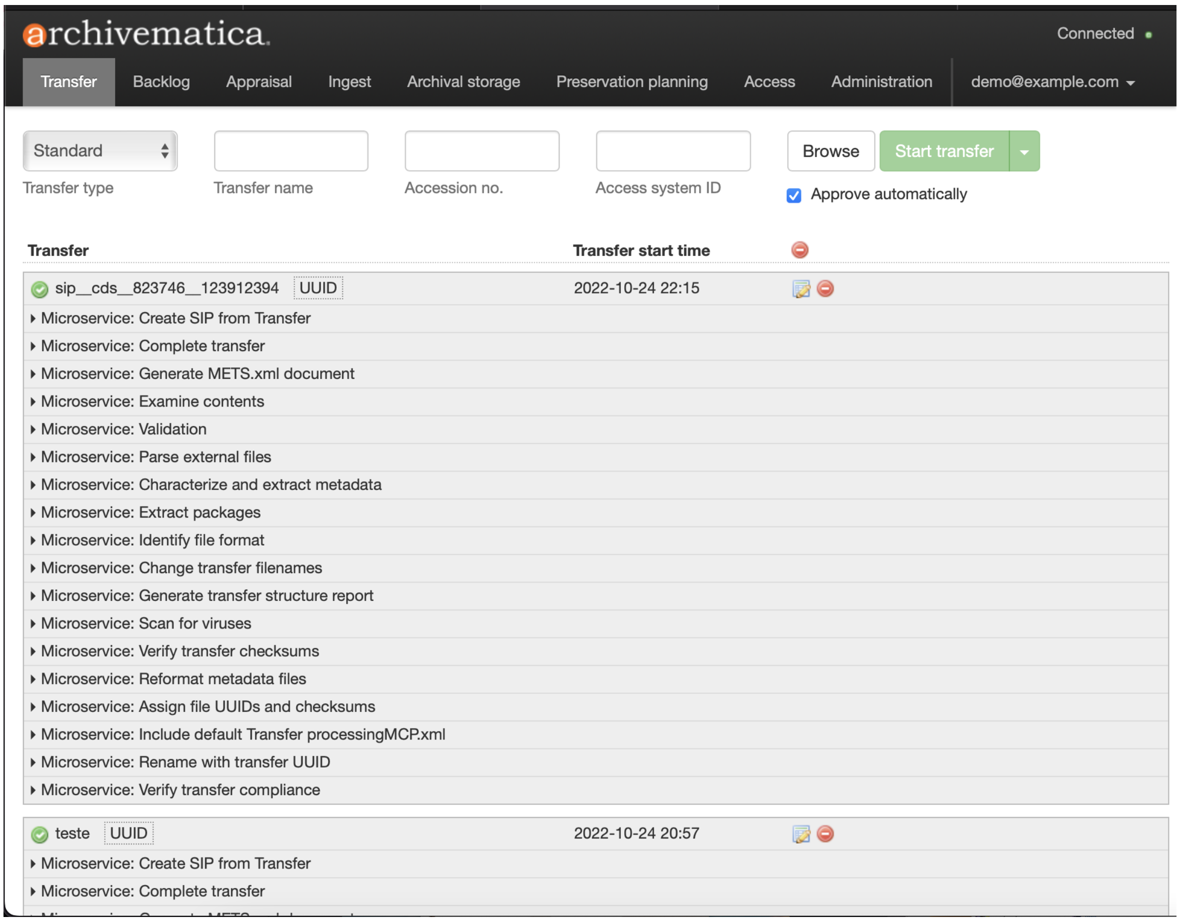


Figure 2.4: Archivematica technical overview [17]

Figure 2.4 shows the main components of Archivematica and their tasks are the following:

- **Dashboard** The Archivematica dashboard contains the user interface2.5 and the main interaction endpoint of the whole system. It can be used through the Rest API or the web interface to start transfers, check the state, add metadata and change the configuration.
- **Storage Service** The Storage Service is an additional layer that handles the storage of the system regardless of the file system and the location.
- **MCP Server** MCP Server is part of the Master Control Program (MCP) which handles the microservice scheduling and assigns the tasks to be executed. It uses file hooks and callbacks to keep track of the state of each microservice and distributes them using Gearman which is a framework for jobs distribution.
- **MCP Client** MCP Client gets the tasks from the MCP server and runs the microservices as an isolated process. MCP client contains a series of scripts and external tools in order to execute the microservices.
- **Database (MySQL)** Archivematica uses at least two databases: one is for the MCP service and the dashboard and the other is for the storage service.
- **Elasticsearch** Elasticsearch is an open-source library that indexes the packages and allows to make queries and search for specific archives.
- **Fits** Fits (File Information Tool Set) is an open source tool developed by Harvard University which validates data formats, extracts useful metadata embedded within files, and outputs the metadata in various formats.

- **ClamAV** ClamAV is an open source antivirus used to scan files for potential viruses.



Transfer type: Standard

Transfer name:

Accession no.:

Access system ID:

Transfer	Transfer start time
sip__cds__823746__123912394 UUID	2022-10-24 22:15
Microservice: Create SIP from Transfer	
Microservice: Complete transfer	
Microservice: Generate METS.xml document	
Microservice: Examine contents	
Microservice: Validation	
Microservice: Parse external files	
Microservice: Characterize and extract metadata	
Microservice: Extract packages	
Microservice: Identify file format	
Microservice: Change transfer filenames	
Microservice: Generate transfer structure report	
Microservice: Scan for viruses	
Microservice: Verify transfer checksums	
Microservice: Reformat metadata files	
Microservice: Assign file UUIDs and checksums	
Microservice: Include default Transfer processingMCP.xml	
Microservice: Rename with transfer UUID	
Microservice: Verify transfer compliance	
teste UUID	2022-10-24 20:57
Microservice: Create SIP from Transfer	
Microservice: Complete transfer	

Figure 2.5: Screenshot from Archivematica dashboard homepage

Archivematica can be deployed either using packages in CentOS, Red Hat and Ubuntu or using Docker although artefactual states that Docker is not officially supported for production deployments.

Archivematica can be easily scaled both vertically and horizontally. Vertical scaling can be achieved by increasing the number of MCPClient instances. In this way, the MCPServer has more clients available to distribute the execution of microservices and more of them can be executed at the same time thus increasing the speed and the throughput of the system. Horizontal scaling can be achieved by increasing the Archivematica instances that run at the same time. In this way, most of the services are duplicated apart from Elasticsearch and the Storage Service which enables to index and keeping track of all the packages in all the instances. [18]

2.5.9 Summary

As we can see, there have been many initiatives to build a long-term digital archiving system in the past. Although there are multiple examples, there is yet no complete an OAIS complied solution to long-term digital archiving. The most common issues about this are the following:

- Complexity and abstraction of OAIS standard which makes the creation of such an archive a complex task.
- Many projects were short term which is inappropriate for a long-term archiving project.
- Lack of continuous funding and continuous development is a basic requirement for this type of project.
- The problem remains of minor importance to organizations and institutes.

Interestingly, we can see that Archivematica already satisfies many of the basic requirements of the platform an observation also made by Jorik van Kemenade [19]. The main reasons why Archivematica should be integrated with the platform is that it can provide the creation of the AIP and the DIP out of the box, it is open source and well maintained, can be easily integrated with the current infrastructure, it is scalable and adjustable.

2.6 Digital Preservation at CERN

2.6.1 Early initiatives for a long-term archive

In the late 90s, CERN found out that most of the data were stored on private web servers, and individual machines and rarely in structured public access systems such as institutional repositories like CDS and mail. The amount of valuable information stored on individual computers that have been lost is unknown. For this reason, in 1997 a committee was created with the purpose to produce a set of implementable solutions for long-term electronic archiving (LTEA) [20]. This committee found out that the most important factor in this loss of data is the lack of a common document handling CERN policy. Also, they recommended e-mail archiving for selected important users and folders. Furthermore, they noted the possibility of potential loss of access due to a person leaving CERN without backing up their data or due to changes in the data format. Last but not least the committee decided to postpone a large-scale long-term archiving of data as long as the measures above are taken. This is because the implementation of a system like this was very expensive at that time and cheaper solutions were expected in the future.

2.6.2 Integrated Project Support Study

As CERN Document Server (CDS) and Engineering & Equipment Data Management Service (EDMS) were becoming the main repositories for CERN Documents and depositing documents there was considered safe, most of the effort was moved to the archiving of e-mails. In parallel to these conclusions, in 2006 the International Linear Collider Project demanded a study group to analyze the data storage systems developed at CERN for the need of the Large Hadron Collider. The reason for multiple repository systems was investigated and some policies for long-term archiving were proposed like a common archival policy across the different document management systems, a coherent high-level metadata search across all the systems, and a common high-level repository to store the users, groups, authentications, and authorizations so that a single CERN account is used to access all these systems.

2.6.3 HEP Data Preservation project

In 2009 the HEP Data Preservation project (DEHEP) was created. The principle of this collaboration was that some experiments cannot be carried out again so it is crucial to saving the data of these experiments. Those data should be accessible, find-able, and reproducible. Out of this initiative, there were created CERN Open Data (for publishing large experimental data sets), CERN Analysis Preservation (for physics data and analysis preservation), and REANA (for reusable research data analysis workflows creation).

2.6.4 Digital Memory Project

In 2016, the Digital Memory Project was created after repeated observation that a large part of CERN's heritage was in danger of being lost. The goal of the Digital Memory Project is to create a reliable and robust solution for long term content preservation of current repositories as well as to collect content that is endangered to be lost which may be valuable to future generations. The main activities of the CERN Digital Memory Project are i) the digitization of content that is only available in analog format (like papers, videotapes, floppy disks etc), ii) the creation of a digital archiving tool compliant with the OAIS reference model and iii) the design of outreaching actions in order to provide additional value to the significance of the digital archive. [21]

2.7 Differentiation of the Implemented System

We are living in a digital world and every day we create and store petabytes of data. CERN as an international research organization produces a large amount and variety of research data. Preserving and accessing this data is a responsibility for every institution as CERN. Just storing data means nothing if there is no efficient and secure way to get this data back when you need it. CERN has various open-source institutional repositories which store a lot of data and each one has different functionality. Some of them are CDS, zenodo, inspire, and CERN Open data. OAIS Platform is not just another institutional repository. The goal of the platform is not to store data, but to make sure that the data stored will be still accessible after many years. Technology is always evolving and data formats are always changing. In order to guarantee that the files we store will be readable and understandable after 50 or more years, we have to make sure that we store it in a way that the future user will have a high probability to be able to read. One example is the audio files. Today there are many audio file formats that represent the same thing after all for example mp3, aac, flac, bwf, wav, aiff, au, pcm etc. Of course there are differences between these formats in terms of file compression and lossless or lossy compression but after all, what we want to store is the audio information. The question we have to solve is whether will all these audio formats be available after many years and how we make sure we save this data more efficiently so they remain readable. How can we provide enough metadata so we can describe any file and find all the relevant information in the future?

The review of current systems indicates that these metadata are not always available, especially since several of them do not fully comply to the OAIS standard.

Furthermore, we note that most of these digital archiving platforms were actually short-term projects that have been decommissioned and are thus not longer maintained. And, finally, some of these systems are not open-source, therefore their code is not available for safety and maintenance reasons, and they are not open to the research community.

2.8 Requirements

As archivematica will be used for the creation of the AIP and the DIP which includes also the necessary transformation of files in order to comply with the OAIS standard, the main focus of the platform is the integration with current CERN repositories, the creation of a user-friendly interface, the creation of the SIPs that will be transferred to archivematica, the tracking and indexing of the archives. The platform should provide long-term access to digital resources to its designated community. It should also expect hardware failures and it should be able to continuously migrate and upgrade. Archiving should be easy from the users and the process must be automatized as much as possible. It must support as many data sources as possible and it should allow also the upload of local files or folders from the user. The platform must provide metrics and information about the archives and the archival process. Finally, it should be able to verify the transferred content and be scalable.

Chapter 3

CERN Digital Memory Platform

3.1 System Overview

The following diagram shows a high level overview of the system. The users can use the system to create an SIP from the upstream supported repositories or their local system using BagIt Create. BagIt Create is responsible for creating these packages according to the CERN SIP Specification. Users can also upload their already created SIPs using the upload tool.

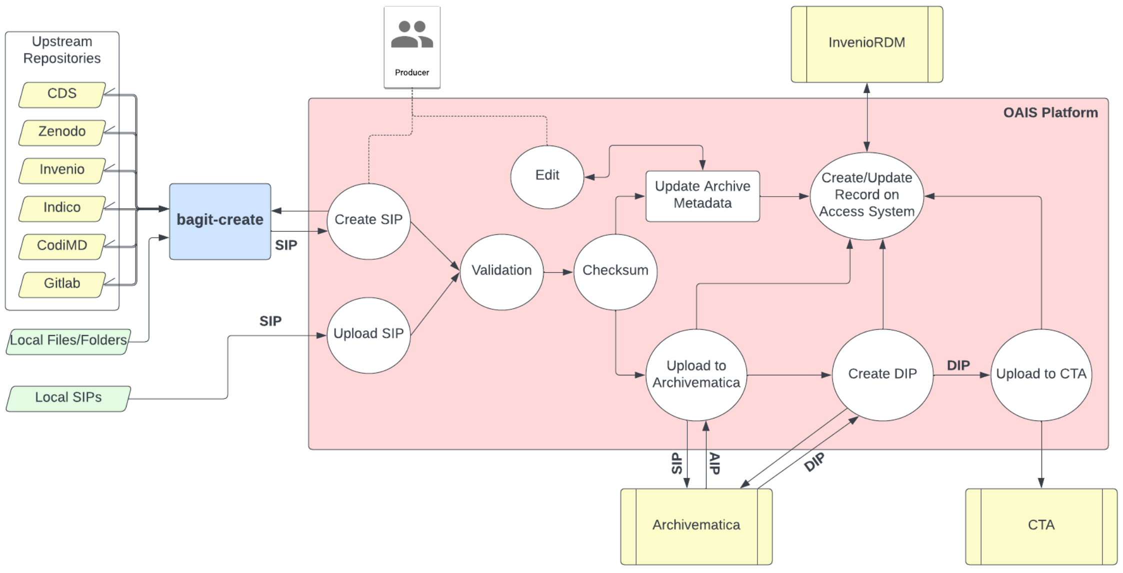


Figure 3.1: OAIS Platform overview

Then the system performs some additional steps like validation of the format and validation of the checksums and in the end calls Archivematica which is an additional software that provides out-of-the box functionalities like file type identification, metadata extraction, virus check, file normalization and the creation of the AIP according to the OAIS Standards.

An external Invenio[1] instance can be used to index, search for archives and to provide versioning. Finally, important AIP packages can be uploaded to a long term preservation storage medium like CERN Tape Archive (CTA)[2].

3.2 CERN SIP Specification

In the context of the creation of an OAIS archive at CERN, we have created a Submission Information Package (SIP) standard which is compliant with the OAIS Reference Model[3]. The scope of this standard is to provide to producers a formal specification of the structure of the submitted packages. The SIP consists of the original content and information that is necessary to ensure data integrity, while it also includes indexing information so that the archive can be easily retrieved in the future.

According to the OAIS standard 2.4, the SIP should provide:

Content Information which contains the original data to be archived and is composed of:

- the *Content Data Object* which has the original data to be stored and
- the *Representation Information* which contains the information to understand the context of the archive.

The Content Information is further described by the PDI - Preservation Description Information:

The **PDI Preservation Description Information** (with its Representation Information). The PDI must include all the information necessary to preserve and associate the archive:

a. *Reference Information* - identifies the Content Data Object allowing outside systems to refer to this particular Content Data Object.

b. *Context Information* - documents the relationships of the Content Data Object with its environment.

c. *Provenance Information* - keeps track of the history of the Content Data Object including the origin, the properties to be preserved, any changes that have taken place, and the person who has the ownership since it was originated, providing an audit trail for the Content Data Object.

d. *Fixity Information* - includes data integrity checks to guarantee that the data is the same as the original submitted one.

e. *Access Right Information* - identifies the access restrictions including the legal framework, licensing terms, and access control.

According to the OAIS specification, the SIP should contain some of the elements specified in the PDI while the AIP should contain all of them.

[1] `https://inveniordm.web.cern.ch`
[2] `https://cta.web.cern.ch/cta/`
[3] CERN SIP Specification: `https://gitlab.cern.ch/digitalmemory/sip-spec`

The CERN SIP Specification aims to implement the Submission Information Package (SIP) according to the OAIS Reference Model. SIPs are packages that will be sent by the producers to the OAIS platform which will be processed by the OAIS services to produce an Archival Information Package (AIP).

3.2.1 Specification

CERN SIP contents are included in a folder which conforms to to the BagIt File Packaging Format specification [4]. All the contents of this file are structured according to this standard plus some additional configuration. The BagIt format is in reality a file system structure that contains the original files and data together with checksums to guarantee data integrity and metadata to provide further information about the related content. As it can be seen in 3.2, BagIt is the baseline upon which further specs are added that create the CERN SIP format.

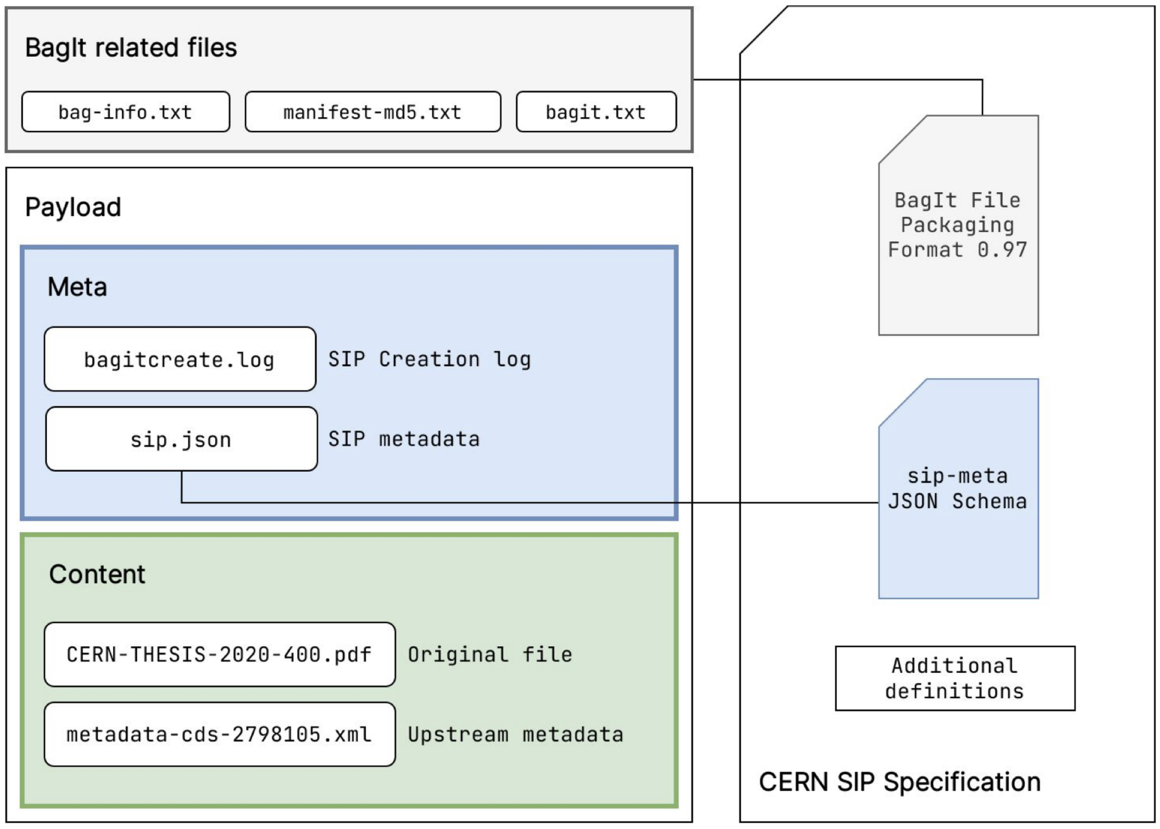


Figure 3.2: SIP data layers

This makes the BagIt bag a mechanism for serialization and data transport consistency while at the same time all the contents of the folder (that follow the SIP specification) provide integrity, identification, annotations, provenance, metadata as well as the content of the resource.

[4]https://tools.ietf.org/id/draft-kunze-bagit-09.html

3.2.2 Elements of a bag

The elements that constitute a bag are the following:

- `data/` folder which contains 2 subfolders.
  - `content` which contains the original data, in the original data structure.
  - `meta` which contains the following metadata files:
    - `sip.json` which provides high-level metadata about the resource target of the archival
    - *upstream metadata object* which contains metadata with additional information about the content and can be more than one file
- `bagit.txt` which includes information about the bagit version and the character encoding

  ```
  BagIt-Version: 1.0
  Tag-File-Character-Encoding: UTF-8
  ```

- `bag-info.txt` which includes more information about the tool used for the generation of the SIP. The exact same values are also specified in the *audit* field of the `sip.json` file.

  ```
  Bag-Software-Agent: bagit-create 0.1.5
  Bagging-Date: 2022-06-10
  Payload-Oxum: 5512.3
  ```

- One or more additional manifest file that contain the file names of all the files that are present in the *data* directory together with their checksums. The checksum algorithm that is used is appended at the manifest file name (e.g. `manifest-md5.txt` `manifest-sha1.txt` etc.)

  ```
  1859fa2f63abcbdb15 data/content/document.doc
  43a3ac7de8fbd34ac3 data/content/image.png
  1a12b00410a05b22b5 data/meta/bagitcreate.log
  4d5487ba5d1fcb277b data/meta/sip.json
  ```

An example of the structure of the SIP folder is shown below:

```
│ bag-info.txt
├─bagit.txt
├─data
│ ├─content
│ │ ├─document.doc
│ │ ├─image.png
│ │ └─metadata.xml
│ └─meta
│   ├─bagit-create.log
│   └─sip.json
└─manifest-md5.txt
```

3.2.3 The sip.json file

The `sip.json` file provides some additional high-level metadata that is used for the description of the archival process and more detailed information about the resource target. Fields that must be included inside the `sip.json` file are the following:

- `source` The name of the upstream repository where the SIP originated.
- `resource_id` The original identifier of this archive at the original resource.
- `metadataFile_upstream` The API endpoint that these data come from.
- `contentFiles` Includes details about files as well metadata for each file.
- `sip_creation_timestamp` The creation timestamp of the SIP
- `audit` Contains information about the SIP creation process.

3.2.4 Versioning

Versioning can be achieved by submitting subsequent SIP(s) with the same `recid` and `source` but with a different `timestamp`. This can be applied to SIPs created by harvesting records from upstream repositories. In case the user uploads or deposits a new version of a SIP from their local computer, they have to manually designate if it is a version of an existing SIP.

3.2.5 Serialisation

According to Bagit Specification Draft 14 Section 4 [5], a SIP can be serialized as a ZIP or TAR file although the following rules must be applied:

a. At the top-level directory there must be only one bag.

b. The serialized file should have the same name as the bag's base directory plus an extension to identify the format. For example, a bag with base directory `my_bag` should be serialized into `my_bag.zip`.

c. The bag must not be serialized from inside the base directory but from the parent of that directory. For example, when the bag .`my_bag.zip` is deserialized in an empty directory it should create a directory called `my_bag` which contains all the payload.

d. When a bag is deserialized, it should produce a single base directory bag with the top-level directory complied with this specification without the need for additional deserialisation step.

3.2.6 Compliance to OAIS Standard

According to the OAIS Standard which is analyzed in 2.4, the Submission Information file (SIP) should contain the content information and some Preservation Description Information (PDI). CERN SIP Packages store the *content information* inside the `data/content` folder in the original data structure. The *representation information* which contains the context information is stored in the `bagit.txt` file.

[5] `https://datatracker.ietf.org/doc/html/draft-kunze-bagit-14`

The Preservation Description Information can be found in many layers of the bag as the OAIS standard does not specify how this information should be delivered.

1. *Reference Information* can be found in the bag folder name

   e.g. `sip::cds::547698::1630483649`.
2. *Context Information* - Can be found in the *sip.json* (e.g. the url to the original source)
3. *Provenance Information* - Can be found in many layers of the bag:
   a. *Origin* is in Source and Record ID and also in the `sip.json`
   b. *Information Properties*
   c. *Changes that have taken place* can be found in the `sip.json` in the audit trail
   d. *Custody* - Can be found in the `creator` field.
4. *Fixity Information* - Checksums are available in the `sip.json` and in the manifest file.
5. *Access Right Information* - This is one of the biggest challenges that we have to face. Right now the creator of the archive can choose if the archive will be entirely public or entirely private. The main issue is that we need to know beforehand who has access rights to each archive. In order to achieve this we need to have administrator status in the upstream repository so we can know exactly the permissions on each record. The second issue is that we must have an authentication mechanism that will certify if the person that asks for access to the archive, is the one who claims to be.

3.2.7 OAIS Utils library

To be able to check if a folder meets the SIP CERN requirements criteria, the `oais_utils` library was implemented[6]. This library contains tools that given a specified folder, implement the following steps:

- Verifies that the directory structure is the one specified by the CERN SIP standard (Chapter 3.1.2)
- Validates the `sip.json` file against the desired JSON schema so we can check that all the fields needed are there.
- Validates that the folder is a valid BagIt package according to the BagIt specification.
- Checks that each file mentioned in the checksum manifest is actually present in the content folder.

[6] `https://gitlab.cern.ch/digitalmemory/utils`

3.3 BagIt Create

BagIt Create[7] is a tool that allows the user to create a SIP package according to the CERN Submission Information Package specification. The records can be exported either from a digital repository or from a local folder. The supported repositories at this moment are Invenio v1, Invenio v3, Invenio RDM, CERN Open Data, CodiMD, Gitlab and Indico. BagIt Create can be easily modified to support other repositories that use the framework of the currently supported ones like ilcdoc, ilcagenda, IKEE AUTh etc. BagIt create is a python module that is available to install through pip.

BagiIt Create can be called from the user by passing the `source` of the upstream repository (or local in case of a folder or file found on the local filesystem) and the `recid` (record ID) of the file on the upstream repository. In the case of local mode, the path of the folder is needed instead of the record ID. Other parameters that affect the execution of the program are the following:

- `source` (mandatory) The upstream repository (or local) to be harvested.
- `recid` The original identifier of this archive at the resource.
- `url` Instead of passing the source + recid of a resource, the user can give only the url of the upstream record to identify it.
- `loglevel` The logging level of the program execution (0 = None, 1= Verbose, 2= Very Verbose).
- `source_path` In the case of local mode, the field to pass the path of the folder or file to be processed.
- `dry_run` When enabled only the metadata of the resource are harvested and the resulting package contains no content data. Additionally a `fetch.txt` file is created that stores the url path for each file.
- `bibdoc` Set to true to get metadata for a CDS record from the bibdocfile utility.
- `bd_ssh_host` Specify SSH host to run bibdocfile. Access must be promptless.
- `skipssl` Skip SSL authentication in HTTP requests. Useful for misconfigured or deprecated instances.
- `token` Authentication token, depending on the chosen source pipeline.
- `cert` Full path to the certificate to use to authenticate in firewalled CERN services.

[7]`https://gitlab.cern.ch/digitalmemory/bagit-create`

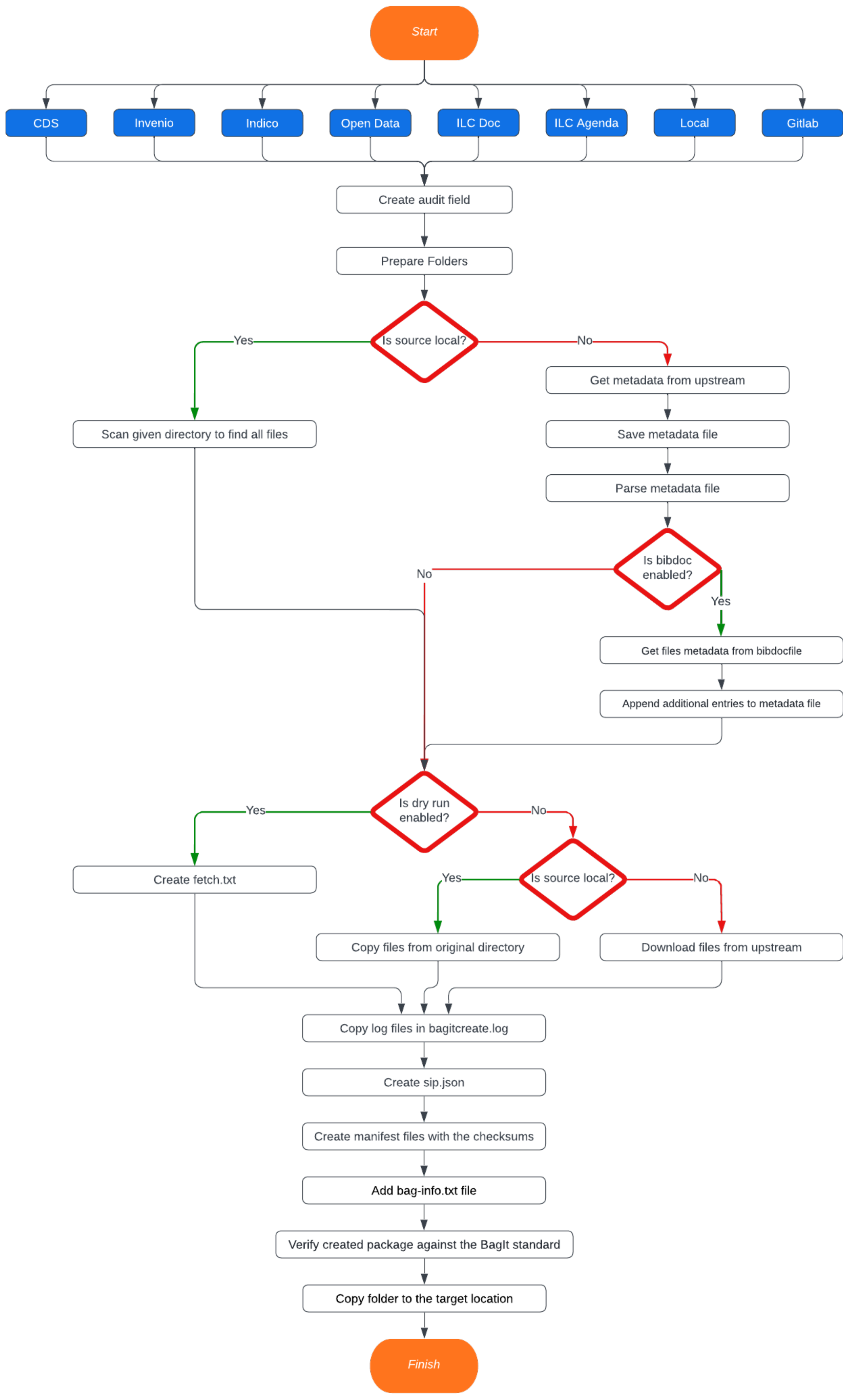

Start
CDS
Invenio
Indico
Open Data
ILC Doc
ILC Agenda
Local
Gitlab
Create audit field
Prepare Folders
Is source local?
Yes
No
Scan given directory to find all files
Get metadata from upstream
Save metadata file
Parse metadata file
Is bibdoc enabled?
No
Yes
Get files metadata from bibdocfile
Append additional entries to metadata file
Is dry run enabled?
Yes
No
Create fetch.txt
Is source local?
Yes
No
Copy files from original directory
Download files from upstream
Copy log files in bagitcreate.log
Create sip.json
Create manifest files with the checksums
Add bag-info.txt file
Verify created package against the BagIt standard
Copy folder to the target location
Finish

Figure 3.3: BagIt Create Flowchart

The execution steps when BagIt Create is called are summarized below and in 3.3:

1. Initially it stores the parameters with which this program was called in the audit field of the sip.json file.
2. Requests and stores the metadata from the upstream repository including the file structure of the resource
3. If the dry run mode is not enabled, then the files are downloaded and stored in the content folder.
4. If the checksums are available from upstream, they are downloaded and validated. Otherwise, the checksums are computed and saved in the manifest file.
5. Creates and populates the bag-info.txt file with information about the payload size and the number of files.
6. Verifies that the bag complies with the BagIt specification.

3.3.1 Installation and Execution

Bagit-Create can be easily installed and executed by using the Python Package Index (pip) or by cloning and installing the repository from the CERN gitlab page: `https://gitlab.cern.ch/digitalmemory/bagit-create`.

After installation, you need to specify the location of the record you want to create the package for by using the source and the record ID of the upstream repository.

For example, to create a SIP package for the record 2728246 from CDS, you need to run the following command:

bic -recid 2728246 -source cds

Additional parameters specified in 3.3, can be given by using the corresponding flag.

The result of this execution will be the creation of a folder in the same directory (unless a target directory is specified using the *target* option), with the following name: *sip::cds::2728246::timestamp.* This directory has the following structure:

```
│ bag-info.txt
├─bagit.txt
├─data
│ ├─content
│ │ ├─CERN-THESIS-2020-092.pdf
│ │ └─metadata-cds-2728246.xml
│ └─meta
│   ├─bagitcreate.log
│   └─sip.json
└─manifest-md5.txt
```

which is fully compliant with the SIP CERN specification.

3.4 OAIS Platform

The CERN Digital Memory OAIS platform[8] manages the archive workflow and harvesting. It also exposes resources to access systems. It consists of a Web API built on Django [9] and the main goals are:

1. To provide an easy way for users to harvest resources and produce SIPs (using the BagIt-create tool).
2. To allow users and services to deposit SIPs they have in possession and ingest them in the platform.
3. To trigger SIP validation and checksumming tasks
4. To trigger preservation workflows (like interfacing with a distributed version of Archivematica).
5. To send SIPs and prepared AIPs to other systems like InvenioRDM and CERN Tape Archive (CTA) for storage.
6. To maintain a database registry of successfully harvested and processed resources as well as of the available metadata.

All these operations are different tasks that the platform should support. These tasks should work asynchronously because they may need some time to execute and they should not keep the entire process on hold. For this reason, a task queue manager called Celery[10] is used to run these tasks. Celery can asynchronously execute tasks outside of the HTTP request-response cycle resulting in better server response times. Integrating Celery with Django needs the use of a data store and message broker called Redis[11]. Redis is an in-memory data structure store and is used to support the communication between Celery and Django.

For storing data in our platform, we have a relational database which uses PostgreSQL[12] as management system. The main element of the database is the archive which stores the basic archive information. Steps signify and allows us to keep track of operations on the archive (e.g. upload, validation, etc.). Archives can also be categorized and organized using Tags. Users have profiles and roles which are used to define the permissions they will have. Resources table is used to keep track of the uploaded instances of the archives in repositories such as Invenio. The main elements of the schema are also described in the following paragraphs and in 3.4.

[8] `https://gitlab.cern.ch/digitalmemory/oais-platform`
[9] `https://www.djangoproject.com`
[10] `https://docs.celeryq.dev/en/stable/`
[11] `https://redis.io`
[12] `https://www.postgresql.org`

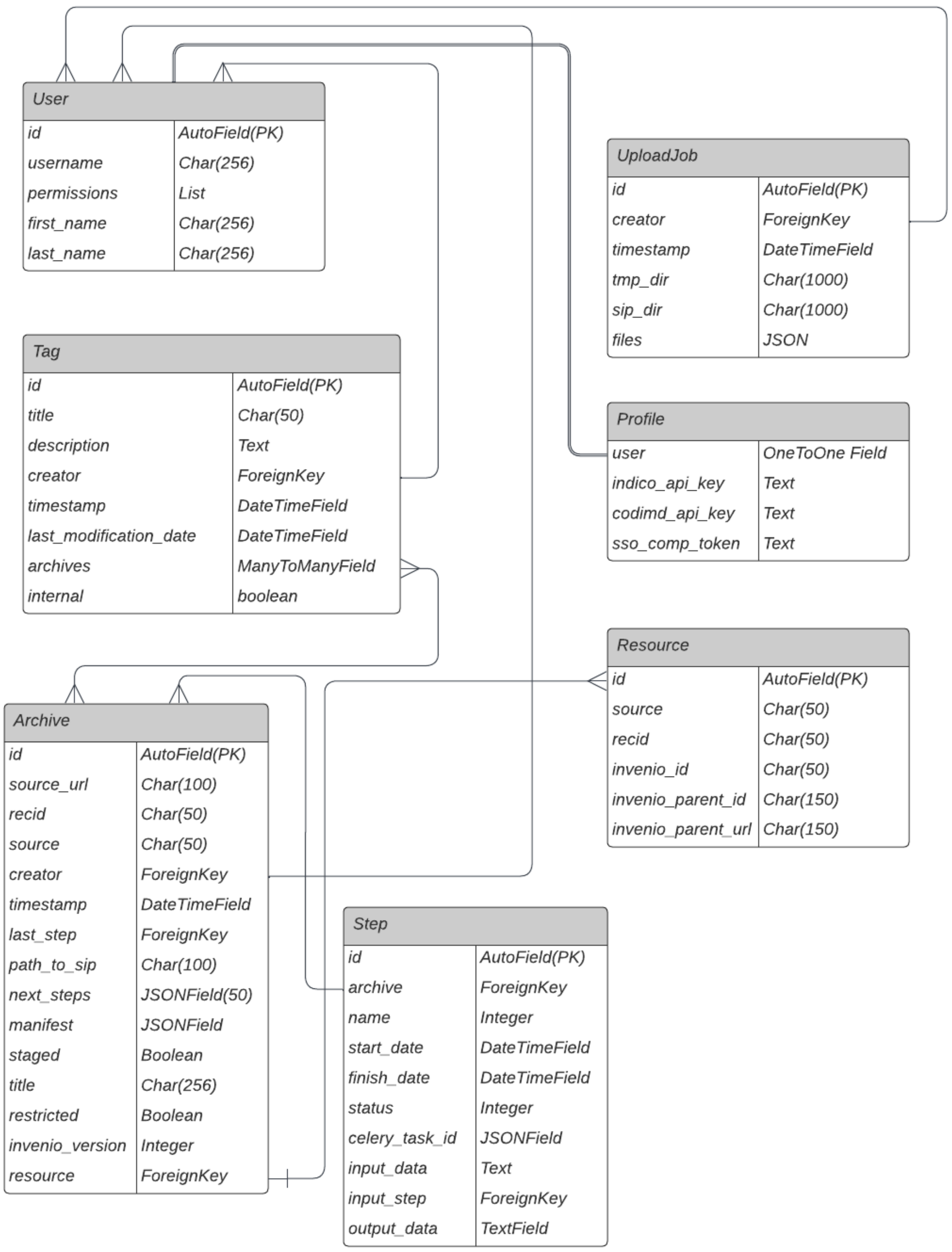


Figure 3.4: OAIS Platform database diagram

3.4.1 Archive

The core element of the OAIS Platform is the Archive. An Archive represents the whole history of a resource we want to preserve, starting from the form in which it was originally submitted to the platform till the final package we want to archive.

An Archive can go through a number of different operations, which can transform the data, add, modify or validate data or metadata in the Archive.

Archive information is stored in the Archive table in the database.

An archive can be "staged". A staged archive is displayed in the staged area which is a preliminary area where the user can check which records they have selected to archive, see if they have archived these records in the past or add tags to them. Then all the staged records which are sent for archiving are organized together in a job. There the user can see the status of the archival process and keep track of all the archives.

3.4.2 Step

Each step is a specific operation on the archive. This operation can be the harvest of the contents from the original source, the validation of the package or the validation of the checksum. Each step has to be retriable and independent from each other and tied to a specific operation.

Steps currently supported by the OAIS platform are the following:

- **Harvest:** Harvests data from the supported repositories using *bagit-create* and creates the Submission Information Package (SIP).
- **Upload:** Supports the upload of a local SIP by the user (supports only SIPs created using *bagit-create*).
- **Validate:** Validates that the archive is a valid SIP using *oais-utils* library.
- **Checksum:** Calculates and verifies the checksums of all the files inside the SIP.
- **Archivematica Upload:** Uploads a SIP to archivematica (see section 3.5) and verifies that the submission and the creation of the AIP have been completed successfully.
- **Edit Manifest:** Supports the edit of the manifest file from the user.
- **Invenio RDM Push:** Pushes the package (SIP or AIP) to the Invenio RDM instance.

Each step has at any given moment a status that determines if the step is being executed at that moment or it has finished. Step status can be one of the following:

- **NOT RUN:** The step has been spawned but no task has started yet.
- **IN PROGRESS:** The operation that corresponds to that step has started and is currently in progress.
- **FAILED:** Step operation finished but failed. (e.g. package is not a valid SIP, checksum validation failed)
- **COMPLETED:** Step operation finished successfully.
- **WAITING APPROVAL:** Operation is waiting for approval by the administrator or an authorized person.

- **REJECTED:** Step execution has been rejected.
- **WAITING:** Step has been spawned and a Celery task has been assigned but the execution has not started.

Step sequence

As mentioned above steps are independent of each other and retriable. There are some logical restrictions, for example, validation before harvest or upload will fail in any case because the SIP is not there yet, however that doesn't mean that this sequence is not allowed. The user can trigger a validation or checksum step at any time to verify that the SIP is consistent. Moreover triggering multiple Push to Invenio will create revisions of the same archive on the Invenio instance. An example pipeline of the archive object in the OAIS platform is the following one:

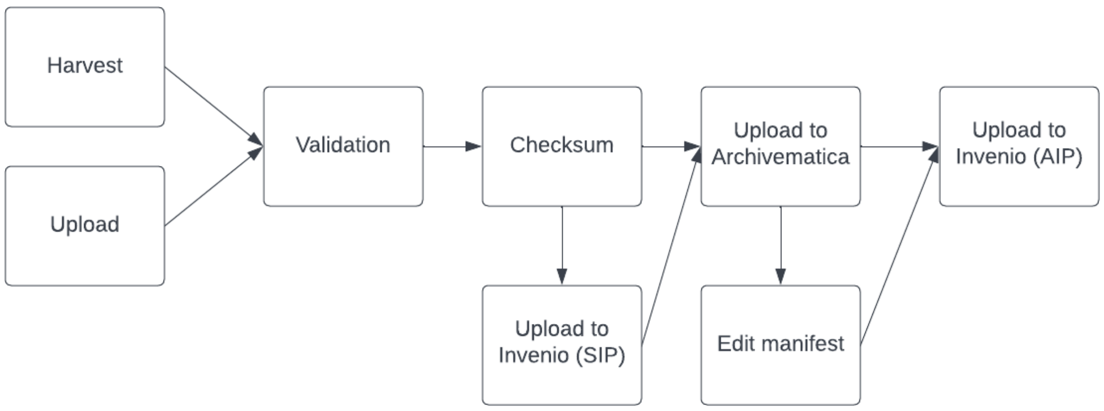


Figure 3.5: An example of an OAIS platform step sequence

3.4.3 Tags

Tags are used to categorize archives into different "groups". A user can create a tag, give it a title and a description and add archives under this tag. An archive can have one or multiple tags. Using tags is an easy way to group similar archives so they can be indexed easily. Tags can also be used by the system to categorize archives internally. This happens when the user selects some archives to harvest from the staged area. First the user selects the archives they want to harvest, then all these selected archives are grouped together in a job which is a special type of internal tag. These jobs help the user keep track of the archival process of this group of selected records.

This happens when the user selects some archives to harvest from upstream. Initially these archives go to the staged area which is a preliminary area where the user can check which archives they want to harvest, check if the record has already been harvested or add tags to these staged archives.

3.4.4 User

OAIS platform initially targets CERN personnel mainly because the repositories supported for harvesting are repositories and services hosted at CERN. At this moment, the platform supports two kind of users: the normal user who can log in with their CERN SSO and the administrator who is a privileged user. The normal user can harvest records from the supported repositories (either public records or private records using his tokens), can create or edit tags, add/remove archives to/from tags and set their tokens. The administrator can do all these actions and they can also approve/disapprove a harvest request by a normal user and change the platform system settings. These privileges are handled by Django and REST framework permissions.

Each User object in the database stores the basic user information such as name, e-mail and permissions. It also has a one-to-one connection with a Profile object where additional information is stored such as the API tokens that are needed in order to retrieve private records in some repositories.

3.5 Archivematica

As shown in 2.5.8 Archivematica has been chosen as the application which will provide the main core of functionality for the OAIS platform. The reasons for this choice is that it is open-source and uses open standards, it is used by many institutions and services and has an active community, it is OAIS compliant and offers many functionalities out of the box which are the main functionalities of the platform (like AIP, DIP creation). Archivematica is integrated with the platform by using the API that is available. Archivematica can be accessed also through an existent web dashboard which allows the user to control, view and process the workflows.

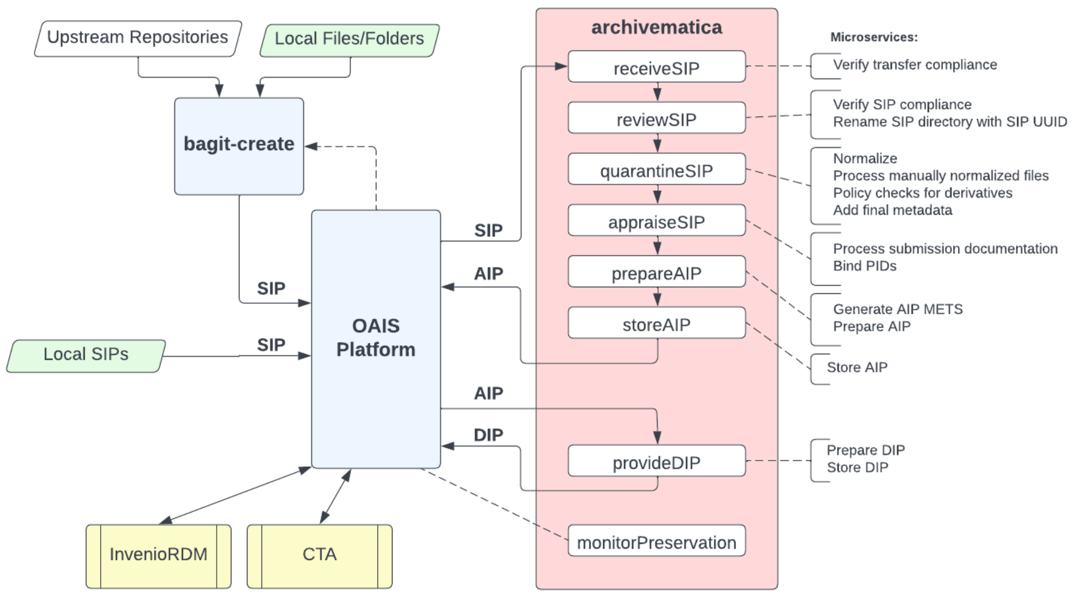


Figure 3.6: Archivematica's microservices in the OAIS Platform context

Archivematica is an open source digital preservation system that includes a set of micro services and each of them performs a very specific task on the workflow. The different workflows that exist in archivematica are used for the creation of the different information packages (SIP, AIP, DIP). Since we already produce SIPs using bagit-create, we are interested on using archivematica for the creation of the AIP and DIP. As we can see in Figure 3.3, Archivematica is used to create AIPs out of SIPs and to export DIPs out of AIPs when needed. Various microservices make the necessary transformations and operations so the resulting files are also OAIS complant.

During Ingest, digital objects which are packaged into SIPs run through various microservices which result in an AIP package. These microservices include the following:

- **Verify SIP compliance:** verifies that the given SIP is compliant with the OAIS and archivematica standard.
- **Rename SIP directory with SIP UUID:** Associates the SIP with its metadata by appending a unique identifier to the SIP directory name.
- **Normalize:** converts ingested objects to the preferred preservation formats according to the user preference. Choices about the preservation can be made by the user when the service is run or it can be automated.
- **Process manually normalized files:** looks and processes already normalized files.
- **Policy checks for derivatives:** checks access and preservation derivatives created during normalization against the registry.
- **Add final metadata:** allows user to add final metadata if desired.
- **Transcribe SIP contents:** runs Tesseract OCR tool on JPG or TIFF images in the SIP.
- **Process submission documentation:** processes any submission documentation in the SIP and adds it to the *object* directory
- **Bind PIDs:** creates persistent PID identifiers.
- **Generate AIP METS:** generates Archivematica AIP METS.xml file which lists all of the digital objects in the AIP, describes their relationships, and links digital objects to their descriptive, technical, provenance, and rights metadata.
- **Prepare AIP:** creates an AIP package, the pointer file, indexes it and losslessly compresses it.
- **Store AIP:** verifies and moves the AIP to the designated AIP storage directory

After all these microservice are executed successfully then the created AIP is stored in the selected storage space and the corresponding Archivematica Step status is successful. The user then can push this AIP to InvenioRDM, or they can request for a Dissemination Information Package (DIP). For the creation of the DIP a new request is made to Archivematica which will use the two following microservices:

- **Prepare DIP:** creates a DIP containing copies of objects, thumbnails and a copy of the METS file.

- **Store DIP:** stores the DIP in the designated DIP storage directory.

After this operation, the user has the final dissemination package which they can make sure it is put in a secure place that can be preserved for a long time. There is also the possibility for Archivematica to directly upload the DIP to a designated upstream repository if the access options are provided.

3.5.1 Interacting with Archivematica

Interacting with Archivematica is feasible through Archivematica API[13]. To make transfers from the platform to Archivematica we need to first make sure that both applications have read/write access to the same file system. As common file system, we will use eos which is a disk storage system at CERN and which can be easily mounted to the applications. For Archivematica, during the deployment we need to make sure that it has access to eos and after the initial deployment, the user must change the Transfer Source in the Storage Service to target eos.

When an API call to archivematica is made, information such as the package path, the transfer type and the connection configuration is passed but not the actual payload of the file. That's why it is crucial to make sure that both applications can read and write on the same path.

3.5.2 Archivematica Management

As mentioned in 2.5.8, Archivematica's throughput can be scaled easily by using more MCP Client instances. By making this change, Archivematica can handle more archives at the same time. Although there is always the danger of reaching the transfer capacity which will result in Archivematica dropping packages automatically. This is caused by the way Archivematica handles package transferring. [19] Archivematica by default accepts and schedules every incoming transfer. As Gearman schedules the tasks in a round-robin format, every new transfer increases the processing time of all the active transfers.

3.5.3 Preservation Planning

Preservation Planning[14] is the set of actions defined by the user to control Archivematica operations on particular file formats - for example how Archivematica should normalise a PNG photo for long term preservation.

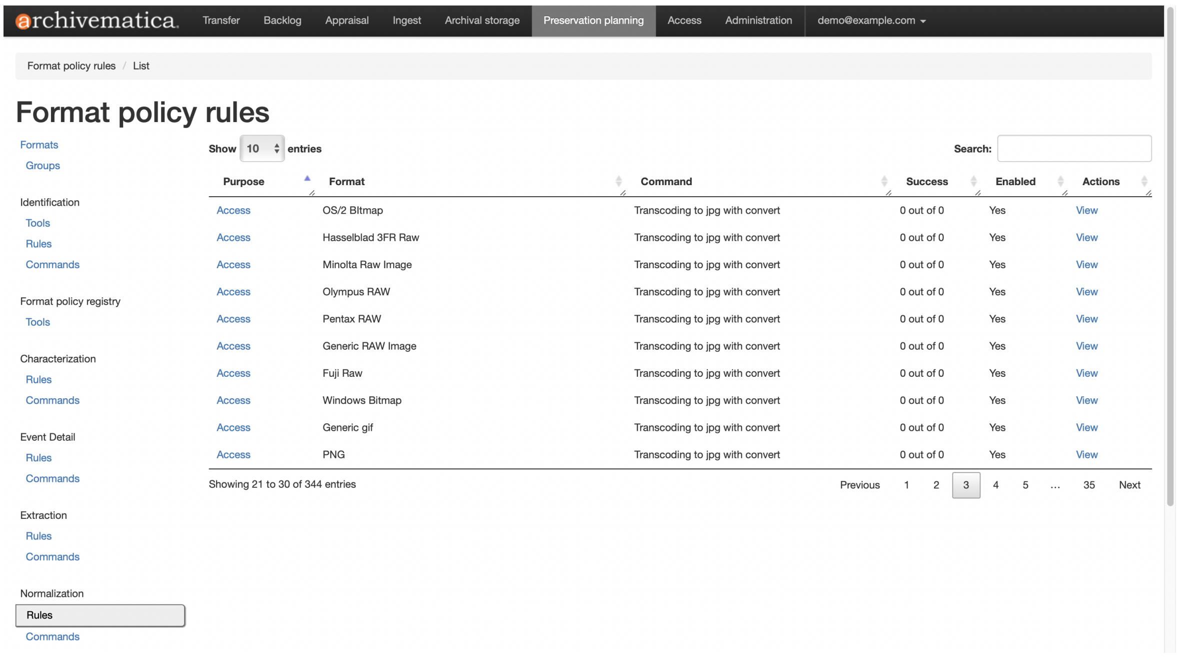
Preservation actions include normalisation, package extraction, characterization, validation, identification, verification, and transcription and they can be managed in the Preservation Planning tab in Archivematica dashboard.

Archivematica uses the FITS2.5.8 software for file identification. The user can create rules and use integrated tools to normalise files from one format to another. For example in figure 3.6, we have specified that all image type files will be normalised

[13]https://wiki.archivematica.org/Archivematica_API

[14]https://www.archivematica.org/en/docs/archivematica-1.13/user-manual/preservation/preservation-planning/

to *.jpg* format because we assume that this format (JPG) is widely used now so it will also be readable after many years.



Format policy rules

Purpose	Format	Command	Success	Enabled	Actions
Access	OS/2 Bitmap	Transcoding to jpg with convert	0 out of 0	Yes	View
Access	Hasselblad 3FR Raw	Transcoding to jpg with convert	0 out of 0	Yes	View
Access	Minolta Raw Image	Transcoding to jpg with convert	0 out of 0	Yes	View
Access	Olympus RAW	Transcoding to jpg with convert	0 out of 0	Yes	View
Access	Pentax RAW	Transcoding to jpg with convert	0 out of 0	Yes	View
Access	Generic RAW Image	Transcoding to jpg with convert	0 out of 0	Yes	View
Access	Fuji Raw	Transcoding to jpg with convert	0 out of 0	Yes	View
Access	Windows Bitmap	Transcoding to jpg with convert	0 out of 0	Yes	View
Access	Generic gif	Transcoding to jpg with convert	0 out of 0	Yes	View
Access	PNG	Transcoding to jpg with convert	0 out of 0	Yes	View

Showing 21 to 30 of 344 entries

Figure 3.7: Screenshot from Archivematica's preservation planning tab. Note that all image type files in this section are normalised to .jpg

Preservation planning includes also Characterisation which is the process of producing technical metadata for a specific object. Characterisation rules can be created and modified from the dashboard. Preservation planning policies should be reviewed thoroughly from time to time in order to make sure that the preservation packages contain as much useful and retrievable information as possible.

3.6 External Repositories

After the creation of the DIP, the created package will be stored in an external repository for long term. At this point, the platform will not be responsible for storing the data but only to keep track of the location so the users in the future can use it to search for and retrieve the archives they want to.

3.6.1 Push to CTA

The CERN Tape Archive (CTA) is a tape backend for file systems [22] and together with eos filesystem is managing the data from the LHC experiments. The advantages of a magnetic tape is that it can provide the lower cost per storage unit and can deliver a bandwidth of 360 MB/s per device. On the other there is high latency especially if trying to access non adjacent files. CTA tapes can store data for around 30 years and this makes the use of tapes the ideal medium for long term storage.

3.6.2 Invenio

Invenio[15] is an open source digital repository which is mainly developed by CERN and provides tools for data management and indexing. Invenio can provide out of the box functionality for the platform such as a search tool where the user can search for the archive they are looking for and additionally versioning as archives related to the same original source and record id, can be linked to each other.

3.7 OAIS User interface

The CERN Digital Memory web interface[16] provides access to the features exposed by OAIS platform. It allows the visualization of the archiving process, the triggering of new jobs and to provide information about the archival process. The goal of the UI is to provide to the user an easy way to archive their data and to monitor the archival process.

For the development of the User Interface, we use ReactJS which is a popular, free and open source JavaScript library for building user interfaces.[17] It was developed by Facebook (now Meta) and it has an active community of many developers and companies.

React is the most popular JavaScript library at the moment. The innovation about React is the use of component based, functional and declarative programming for creating interactive user interfaces. React pages are rendered faster by making use of the 'DOM' which renders only the components that change instead of the whole page. (https://www.lambdatest.com/blog/best-javascript-framework-2020/) Also the use of reusable components makes programming faster and more reliable as components can be used many times inside the page.

Also many of the IT department and the IT-CA section services, use ReactJS for the user interfaces of the applications (such as InvenioRDM, Indico). This is another advantage of using React as it will enable the interoperability between the services as people will be able to work from the one service to the other easily while using the same library. Also there is an internal community at CERN for knowledge sharing and advice.

In order to simplify the website development and make it easier to build the application, we use the Semantic UI framework for React which is a framework that provides many templates and components that make the development easier.

[15] `https://inveniosoftware.org`
[16] `https://gitlab.cern.ch/digitalmemory/oais-web`
[17] `https://reactjs.org/`

3.7.1 User authentication

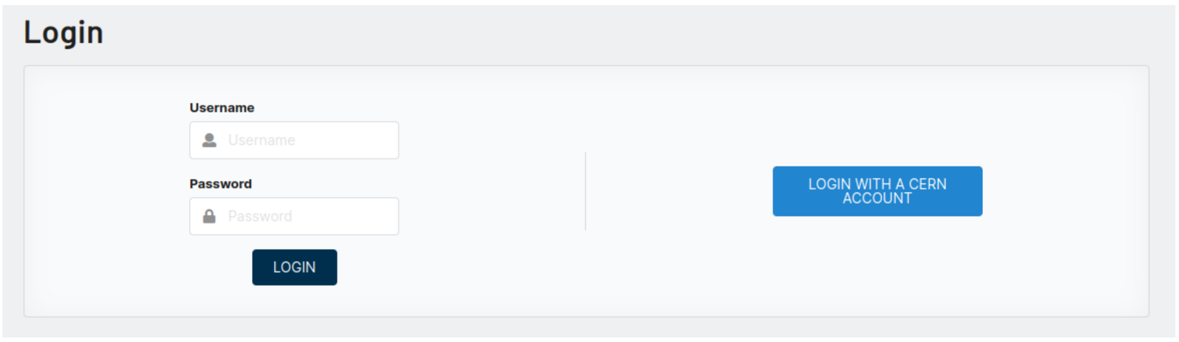
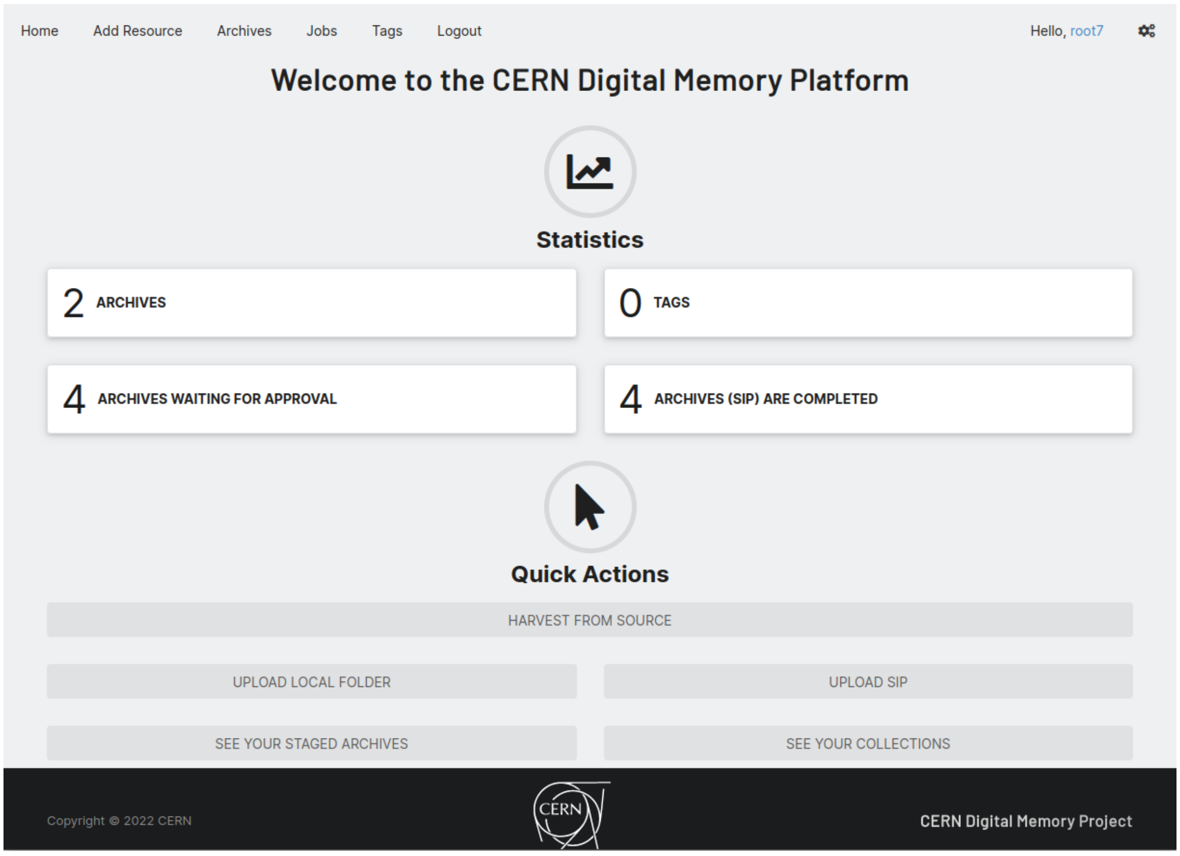


Figure 3.8: OAIS User Interface - Login page

User authentication is achieved by using the Django authentication system. There are two ways to authenticate a user. The first one is by using the classic username/password combination. The backend then gets the credentials and authenticates the user against the authentication backend. If the credentials are correct then the user login is successful and the user is redirected to the homepage. If the credentials are wrong then the backend raises a `PermissionDenied` error and the authentication is unsuccessful.

The second way of authenticating is by using the CERN Single Sign-On (SSO). In this way the user is redirected to the CERN Authentication which is a service that provides a centralized authentication and authorization infrastructure.

3.7.2 Homepage



Welcome to the CERN Digital Memory Platform

Statistics

2 ARCHIVES

0 TAGS

4 ARCHIVES WAITING FOR APPROVAL

4 ARCHIVES (SIP) ARE COMPLETED

Quick Actions

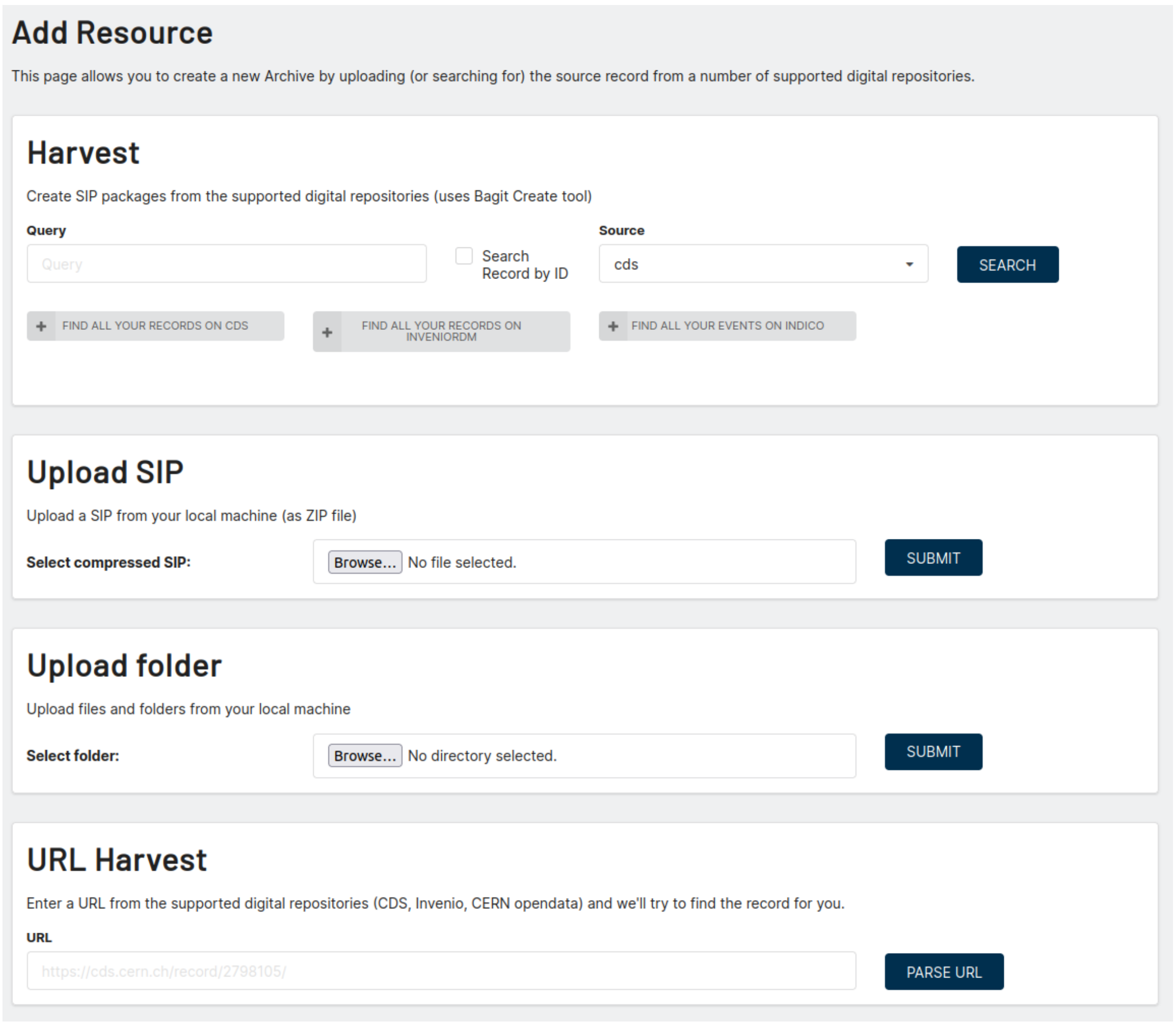
Copyright © 2022 CERN

CERN Digital Memory Project

Figure 3.9: OAIS User Interface - Homepage

When the users are successfully logged in, then they land at the homepage where they can quickly navigate through the various pages of the platform through the navigation menu at the top, see some statistics about the platform and the archives harvested and perform some quick actions by pressing the buttons at the bottom.

3.7.3 Add Resource


Add Resource

This page allows you to create a new Archive by uploading (or searching for) the source record from a number of supported digital repositories.

Harvest

Create SIP packages from the supported digital repositories (uses Bagit Create tool)

Query: Query

Search Record by ID

Source: cds

SEARCH
FIND ALL YOUR RECORDS ON CDS
FIND ALL YOUR RECORDS ON INVENIORDM
FIND ALL YOUR EVENTS ON INDICO

Upload SIP

Upload a SIP from your local machine (as ZIP file)

Select compressed SIP: Browse... No file selected.

Upload folder

Upload files and folders from your local machine

Select folder: Browse... No directory selected.

URL Harvest

Enter a URL from the supported digital repositories (CDS, Invenio, CERN opendata) and we'll try to find the record for you.

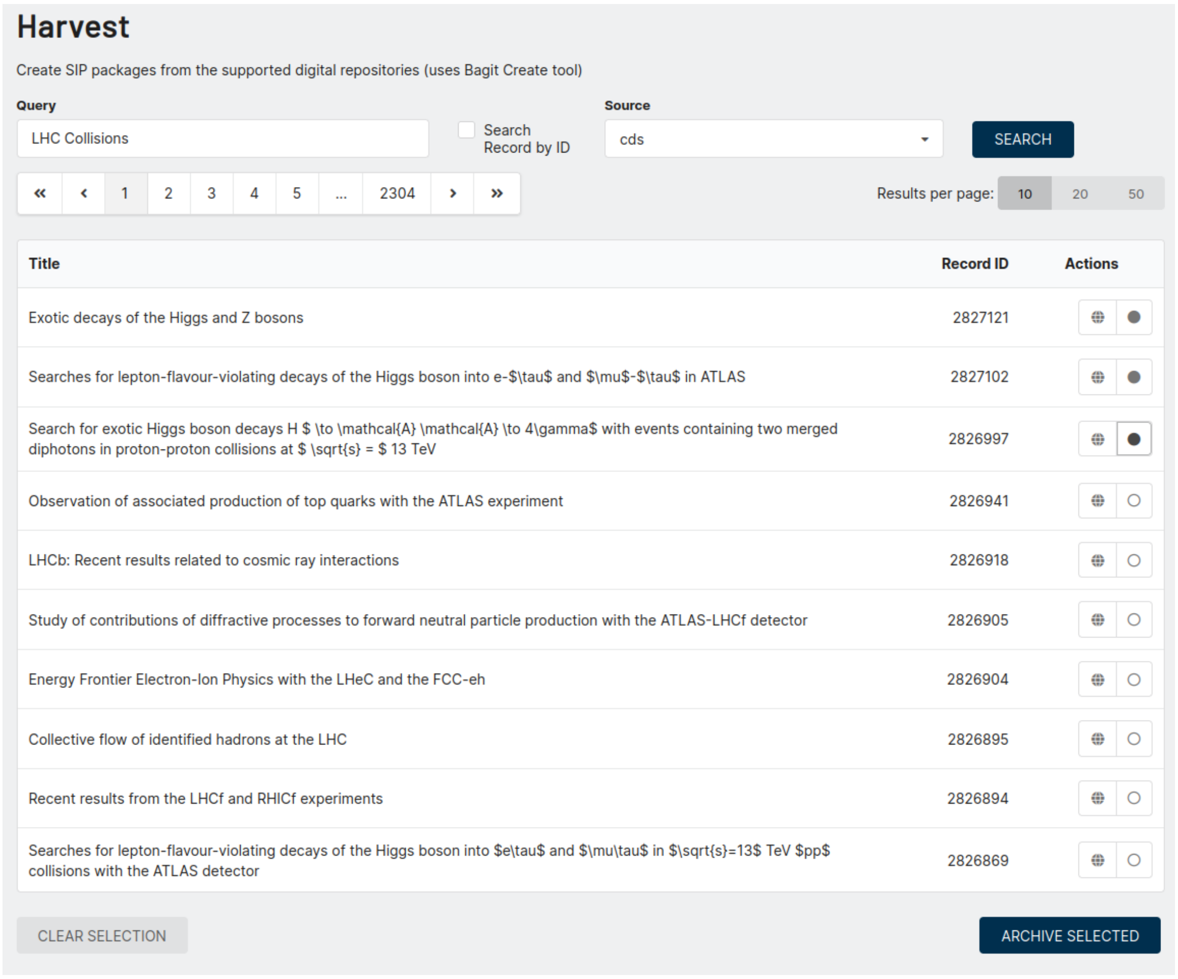
URL: https://cds.cern.ch/record/2798105/

Figure 3.10: OAIS User Interface - Add resource page

The Add Resource page is the main input page of the application. Here the user can search for records at the various repositories supported by the platform (CDS, Zenodo, Invenio RDM, CERN Open Data, Indico, CodiMD and Gitlab), either by performing a specific query, or by searching directly by using the Record ID of that document. If the user is logged in by using the CERN SSO service, then they can use the shortcut buttons available and directly get all their documents on CDS or InvenioRDM and all their events on Indico. The results are then displayed at the harvest page (see below).

Apart from the harvest mode, a user can upload an already created SIP from their local machine. In this case the compressed file is sent to the backend where it is extracted and is validated using the OAIS utils library (see ch. 3.1.4). If the validation succeeds then the upload step is successful and the archive is registered. The third option is the Upload folder/file, where the users can directly upload a folder or a file they have locally to the platform. Then the platform will initialize `bagit-create` using the local pipeline on the folder or file uploaded. If the creation of the SIP is successful then a new Archive object will be created in the database corresponding to the newly created SIP. The forth option is by directly typing the URL of the supported upstream source. The back-end will parse the URL and return the source and the Record ID that corresponds to that URL. These results will be used to populate the harvest page.

3.7.4 Harvest


Harvest

Create SIP packages from the supported digital repositories (uses Bagit Create tool)

Query: LHC Collisions

Search Record by ID

Source: cds

SEARCH

« ‹ 1 2 3 4 5 ... 2304 › »

Results per page: 10 20 50

Title	Record ID	Actions
Exotic decays of the Higgs and Z bosons	2827121	🌐 ●
Searches for lepton-flavour-violating decays of the Higgs boson into e-τ and μ-τ in ATLAS	2827102	🌐 ●
Search for exotic Higgs boson decays H $ \to \mathcal{A} \mathcal{A} \to 4\gamma$ with events containing two merged diphotons in proton-proton collisions at $ \sqrt{s} = $ 13 TeV	2826997	🌐 ●
Observation of associated production of top quarks with the ATLAS experiment	2826941	🌐 ○
LHCb: Recent results related to cosmic ray interactions	2826918	🌐 ○
Study of contributions of diffractive processes to forward neutral particle production with the ATLAS-LHCf detector	2826905	🌐 ○
Energy Frontier Electron-Ion Physics with the LHeC and the FCC-eh	2826904	🌐 ○
Collective flow of identified hadrons at the LHC	2826895	🌐 ○
Recent results from the LHCf and RHICf experiments	2826894	🌐 ○
Searches for lepton-flavour-violating decays of the Higgs boson into $e\tau$ and $\mu\tau$ in $\sqrt{s}=13$ TeV pp collisions with the ATLAS detector	2826869	🌐 ○

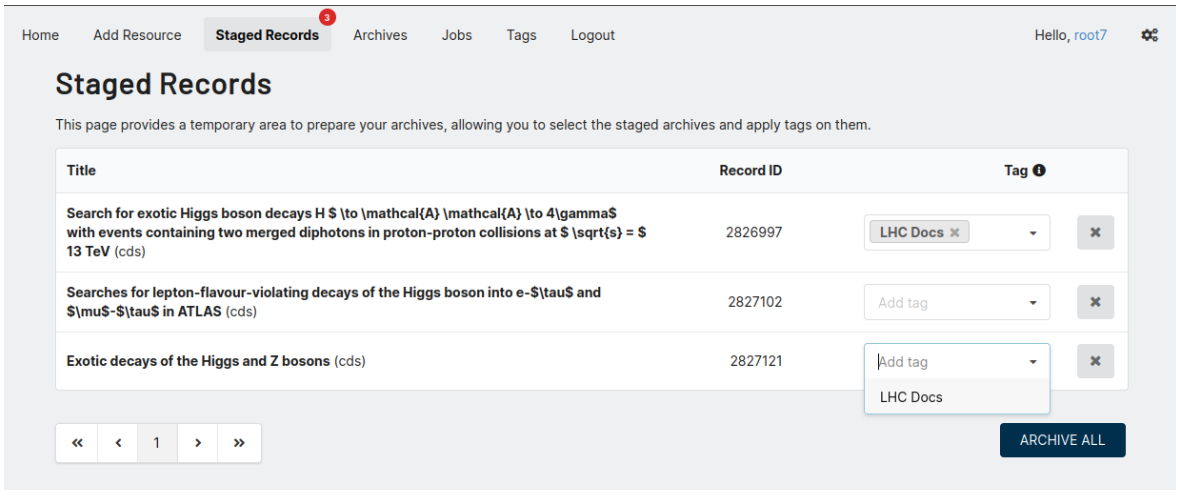
CLEAR SELECTION

ARCHIVE SELECTED

Figure 3.11: OAIS User Interface - Harvest results page

In the harvest page, the users can guide through the results of the query they have performed. When performing a query, the back-end sends the search term to the upstream repository and the repository replies with a list of paginated results for that query. The user then can guide through the different result pages and select which records they want to harvest. There is a quick action button in the bottom of the page so the users can add all the records for archiving. From the frontend side, the selected results are stored in a centralized application variable using the Redux library. In this way the user can move through different pages and continuously add/remove records to the list. In the end the user presses the "Archive Selected" button and these archives are moved to the Staging Area for further action.

3.7.5 Staging Area



Staged Records

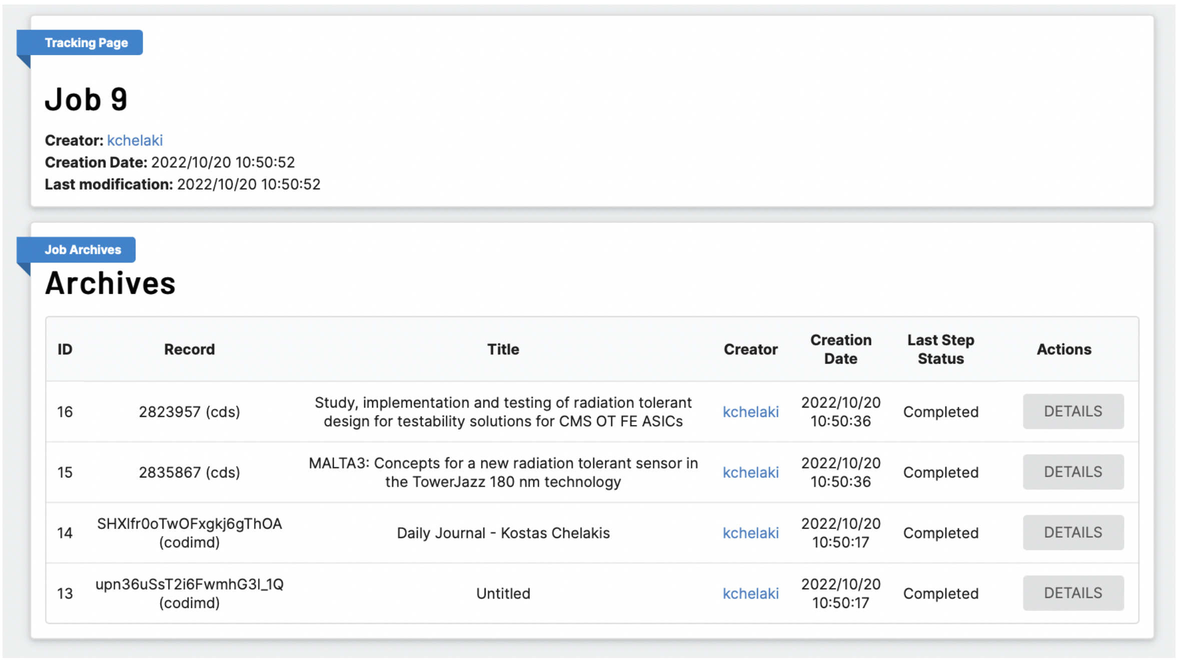
This page provides a temporary area to prepare your archives, allowing you to select the staged archives and apply tags on them.

Title	Record ID	Tag
Search for exotic Higgs boson decays H $ \to \mathcal{A} \mathcal{A} \to 4\gamma$ with events containing two merged diphotons in proton-proton collisions at $ \sqrt{s} = $ 13 TeV (cds)	2826997	LHC Docs
Searches for lepton-flavour-violating decays of the Higgs boson into e-τ and μ-τ in ATLAS (cds)	2827102	Add tag
Exotic decays of the Higgs and Z bosons (cds)	2827121	Add tag

Figure 3.12: OAIS User Interface - Staged area page

The staged page is a preliminary area where the user can get an overview of the records they have selected for archiving. Here the user can add tags to the record using the "Add Tag" dropdown menu. There is the possibility to add an existing tag on the archive, or to easily create a new one. There is also the possibility to remove a record from the staged area if the user no longer wants it. When the user has collected all the records they want for archiving, then they press the "Archive All" button which "unstages" the archive, thus if no approval is needed from the administrators, initiates the creation of an SIP from the upstream repository using `bagit-create`.

3.7.6 Jobs



Job 9

Creator: kchelaki
Creation Date: 2022/10/20 10:50:52
Last modification: 2022/10/20 10:50:52

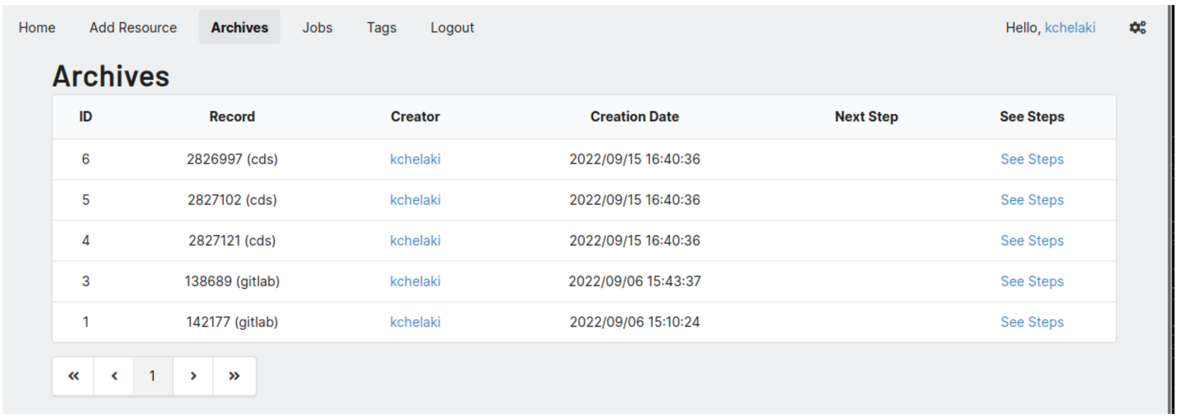
Archives

ID	Record	Title	Creator	Creation Date	Last Step Status	Actions
16	2823957 (cds)	Study, implementation and testing of radiation tolerant design for testability solutions for CMS OT FE ASICs	kchelaki	2022/10/20 10:50:36	Completed	DETAILS
15	2835867 (cds)	MALTA3: Concepts for a new radiation tolerant sensor in the TowerJazz 180 nm technology	kchelaki	2022/10/20 10:50:36	Completed	DETAILS
14	SHXlfr0oTwOFxgkj6gThOA (codimd)	Daily Journal - Kostas Chelakis	kchelaki	2022/10/20 10:50:17	Completed	DETAILS
13	upn36uSsT2i6FwmhG3l_1Q (codimd)	Untitled	kchelaki	2022/10/20 10:50:17	Completed	DETAILS

Figure 3.13: OAIS User Interface - Jobs detail page

When the user archives records from the staging area, they create a Job which contains the batch of the archives selected. These Jobs can be viewed and managed from the "Jobs" tab in the main menu. By choosing a specific job, the user can go to the Job detail page where they can check all the archives associated with that job and check their status as we can see in 3.13.

3.7.7 Archives list & Archives detail page

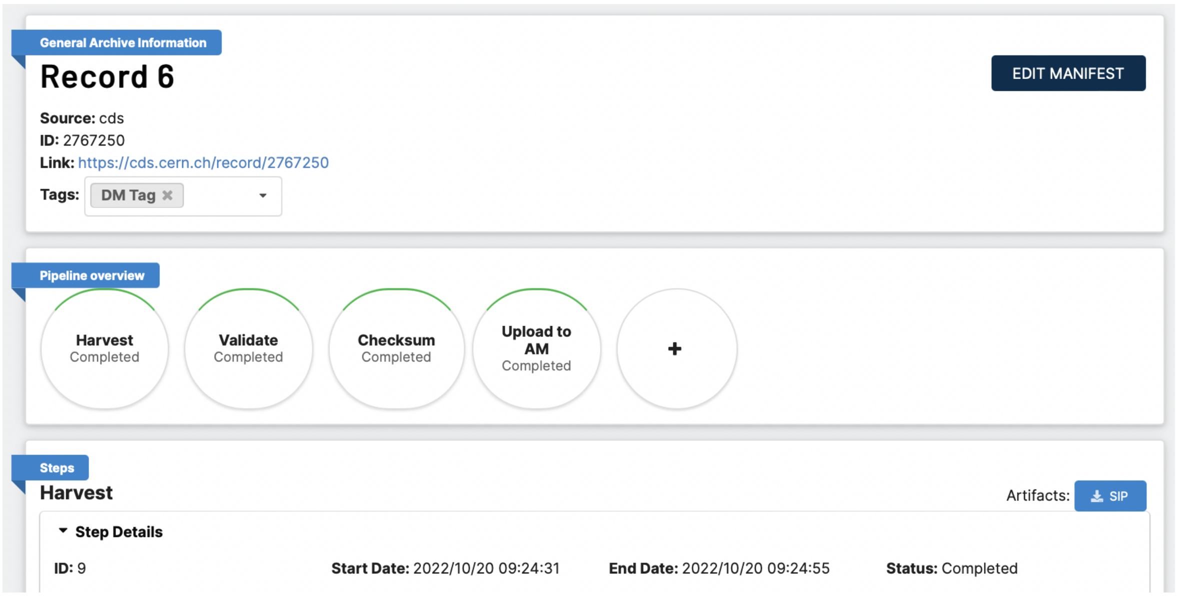


Archives

ID	Record	Creator	Creation Date	Next Step	See Steps
6	2826997 (cds)	kchelaki	2022/09/15 16:40:36		See Steps
5	2827102 (cds)	kchelaki	2022/09/15 16:40:36		See Steps
4	2827121 (cds)	kchelaki	2022/09/15 16:40:36		See Steps
3	138689 (gitlab)	kchelaki	2022/09/06 15:43:37		See Steps
1	142177 (gitlab)	kchelaki	2022/09/06 15:10:24		See Steps

Figure 3.14: OAIS User Interface - Archives List page

The archives page shows the list of the archives that have been harvested by the logged in user. In case of the administrators, they have access to all the archives from all the users. By clicking on the archive, the user will go to the archive detail page where additional information about the archives is displayed including Record ID, the source, the tags etc. Here the user can see the information of the archive, follow the archive steps and their status.



Record 6

Source: cds
ID: 2767250
Link: https://cds.cern.ch/record/2767250
Tags: DM Tag

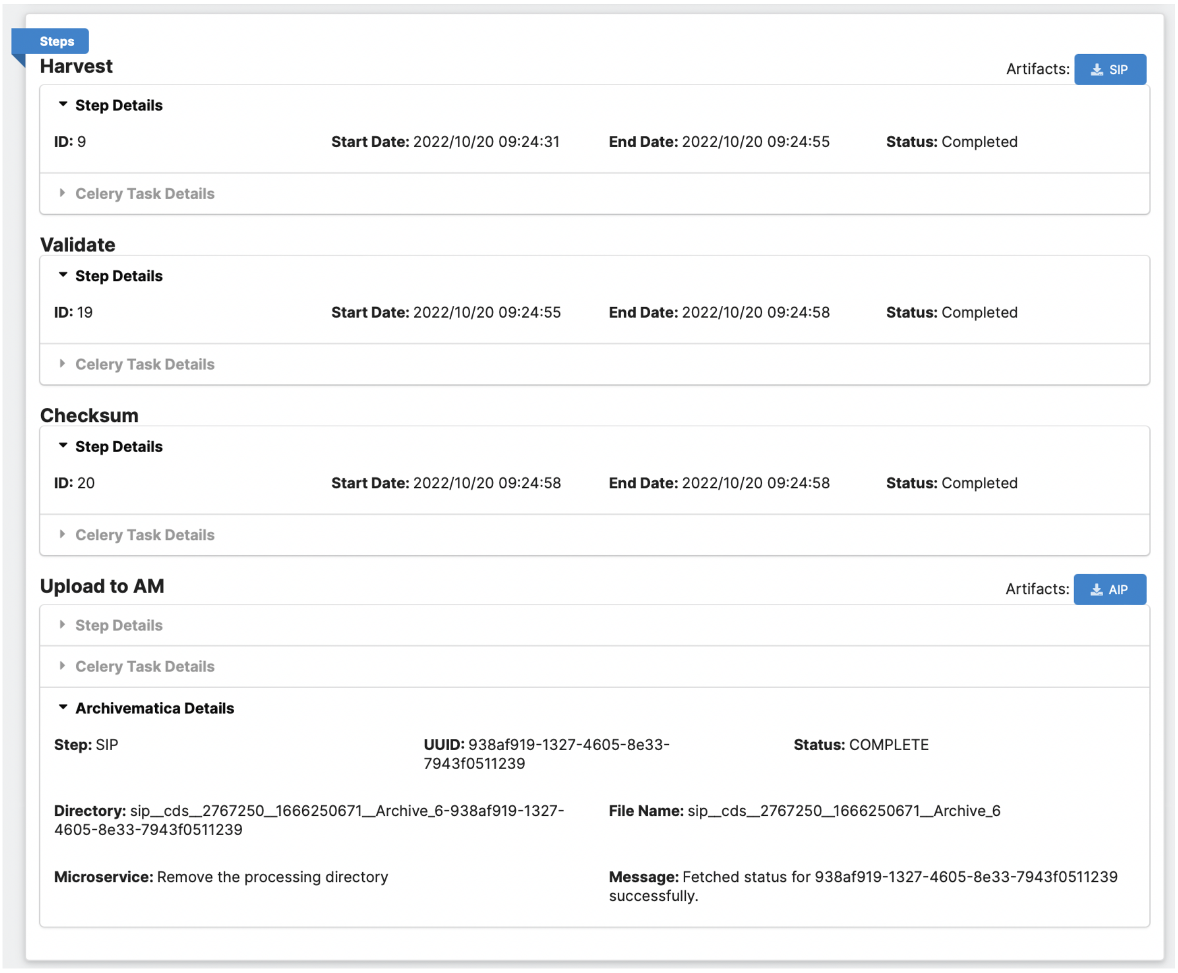
Harvest

ID: 9 **Start Date:** 2022/10/20 09:24:31 **End Date:** 2022/10/20 09:24:55 **Status:** Completed

Figure 3.15: OAIS User Interface - Archives Detail page

The archive detail page as shown in 3.15 is where the user can keep track of the archiving status. At the top there is information about the archive such as the internal ID, the source and the record ID, and moreover the original links and the tags of this archive. On the top right there is the "Edit Manifest" button where the user can add additional information about the archive.

In the pipeline section, the user can see the progress of each step the archive has gone through and to choose the next steps from the available next steps dropdown menu by pressing the "+" button.


Steps

Harvest Artifacts: SIP

Step Details

ID: 9 **Start Date:** 2022/10/20 09:24:31 **End Date:** 2022/10/20 09:24:55 **Status:** Completed

Celery Task Details

Validate

Step Details

ID: 19 **Start Date:** 2022/10/20 09:24:55 **End Date:** 2022/10/20 09:24:58 **Status:** Completed

Celery Task Details

Checksum

Step Details

ID: 20 **Start Date:** 2022/10/20 09:24:58 **End Date:** 2022/10/20 09:24:58 **Status:** Completed

Celery Task Details

Upload to AM Artifacts: AIP

Step Details

Celery Task Details

Archivematica Details

Step: SIP **UUID:** 938af919-1327-4605-8e33-7943f0511239 **Status:** COMPLETE

Directory: sip__cds__2767250__1666250671__Archive_6-938af919-1327-4605-8e33-7943f0511239 **File Name:** sip__cds__2767250__1666250671__Archive_6

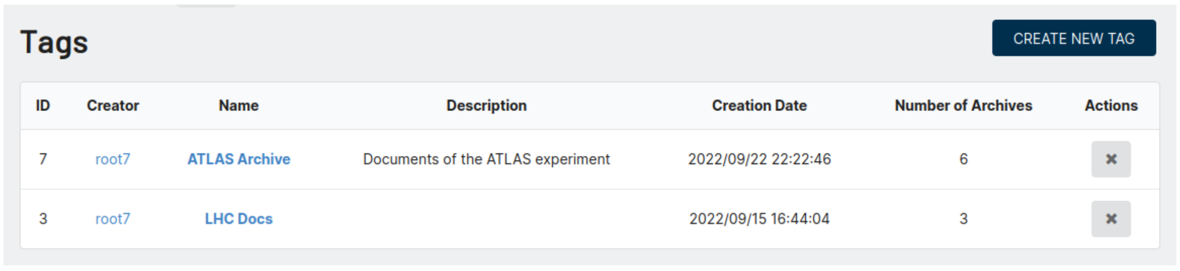
Microservice: Remove the processing directory **Message:** Fetched status for 938af919-1327-4605-8e33-7943f0511239 successfully.

Figure 3.16: OAIS User Interface - Steps Detail

In the bottom of the page, there is a detailed list of all the steps together with important information such as the start and the end date of each step, the step ID, the Celery task ID and the step status. If the step contains the placement of data in any form (either SIP or AIP) then a button to the link of the target is displayed as we can see in 3.16.

3.7.8 Tags page

The tags page is where the user can see and manage the tags they have created. Tags help the user to organize their archives in one place and to manage and monitor simultaneously similar archives.

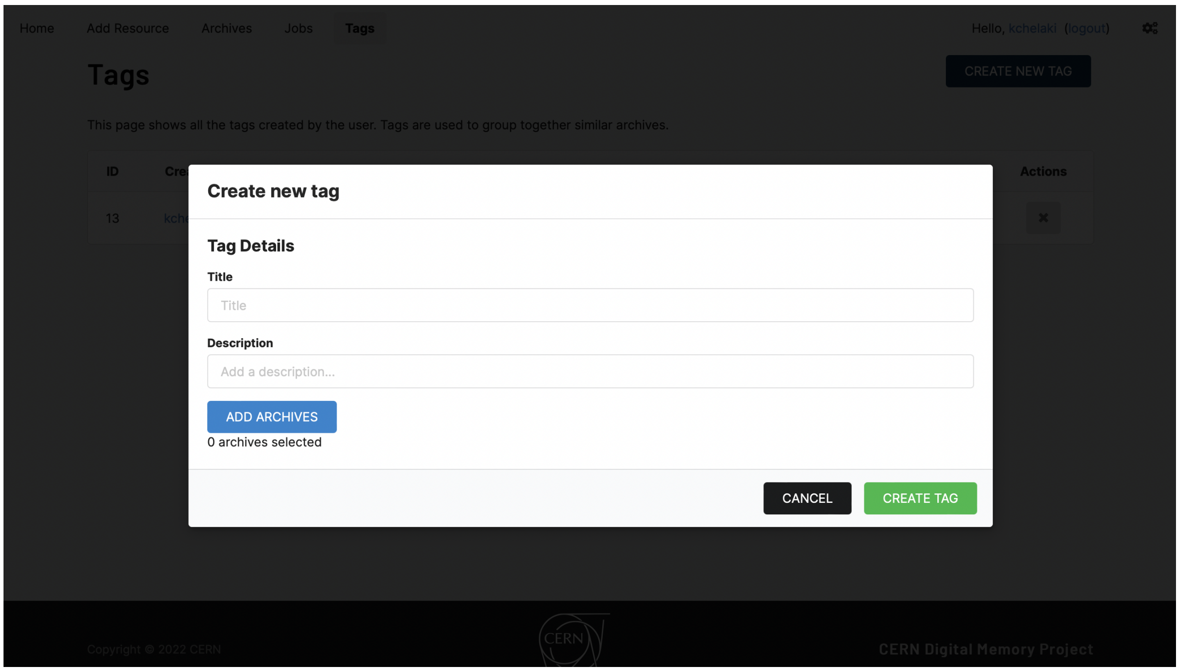

Tags

CREATE NEW TAG

ID	Creator	Name	Description	Creation Date	Number of Archives	Actions
7	root7	ATLAS Archive	Documents of the ATLAS experiment	2022/09/22 22:22:46	6	✖
3	root7	LHC Docs		2022/09/15 16:44:04	3	✖

Figure 3.17: OAIS User Interface - Tags list page

The creation of an archive can be either done through the dropdown menu that was presented in 3.7.5 or by pressing the `Create new Tag` button which opens the module 3.18 through which the user can create a new tag by inserting the desired title and description and selecting which archives they want to include as we can see in 3.19.



Tags

This page shows all the tags created by the user. Tags are used to group together similar archives.

Create new tag

Tag Details

Title: Title

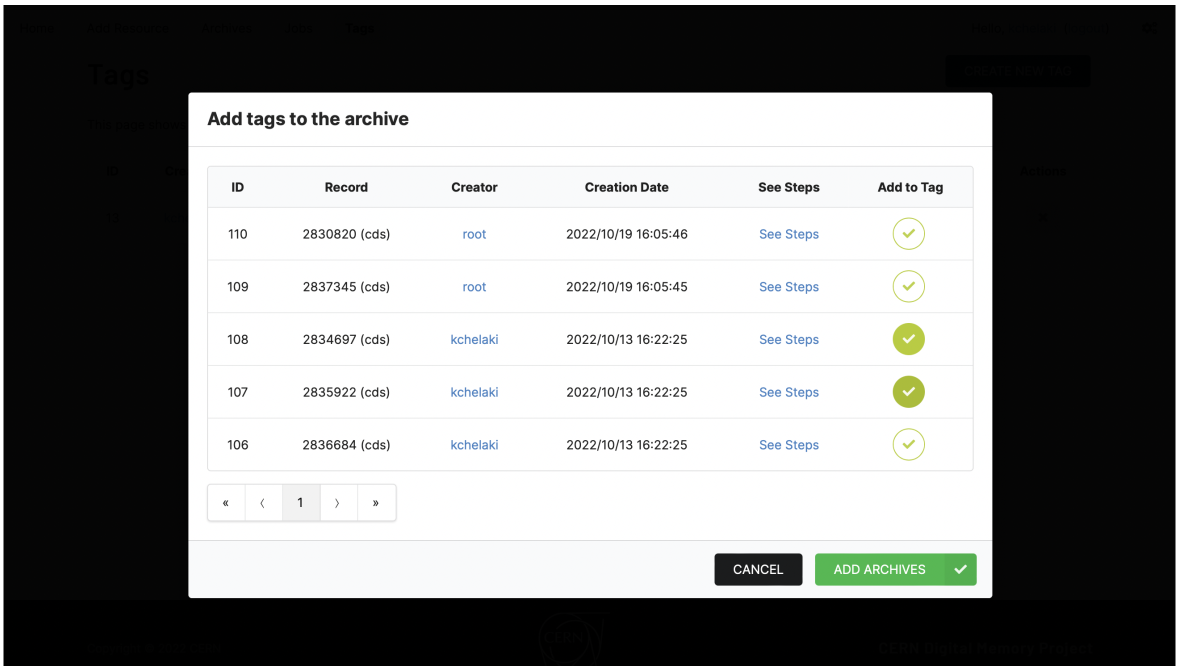
Description: Add a description...

ADD ARCHIVES

0 archives selected

CANCEL CREATE TAG

Figure 3.18: OAIS User Interface - Create Tag page

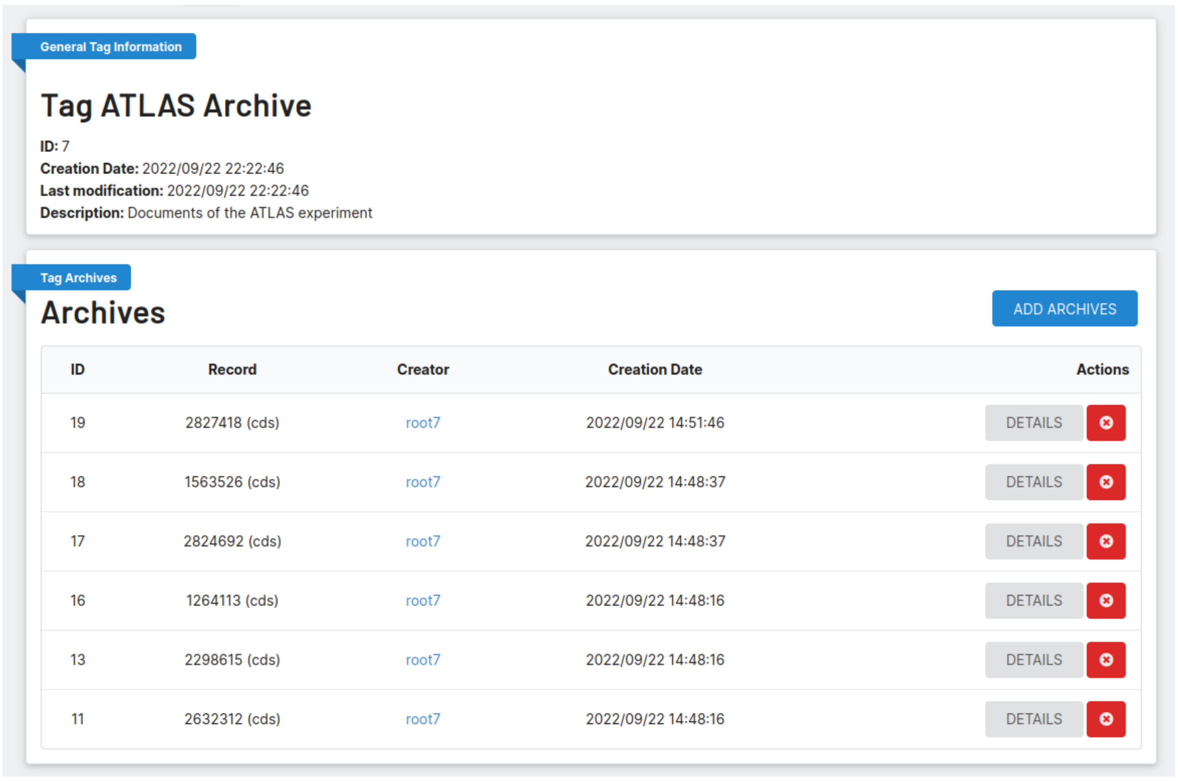


Add tags to the archive

ID	Record	Creator	Creation Date	See Steps	Add to Tag
110	2830820 (cds)	root	2022/10/19 16:05:46	See Steps	✓
109	2837345 (cds)	root	2022/10/19 16:05:45	See Steps	✓
108	2834697 (cds)	kchelaki	2022/10/13 16:22:25	See Steps	●
107	2835922 (cds)	kchelaki	2022/10/13 16:22:25	See Steps	●
106	2836684 (cds)	kchelaki	2022/10/13 16:22:25	See Steps	✓

Figure 3.19: OAIS User Interface - Add archives to existing tag module

By pressing the name of the archive, the user can go to the tags detailed view 3.20 where the user can see a detailed view of the tag including the name, the description and the list of archives in this tag. The user can remove/add archives, edit the description or the title and go to the detail page of each archive included.



General Tag Information

Tag ATLAS Archive

ID: 7
Creation Date: 2022/09/22 22:22:46
Last modification: 2022/09/22 22:22:46
Description: Documents of the ATLAS experiment

Tag Archives

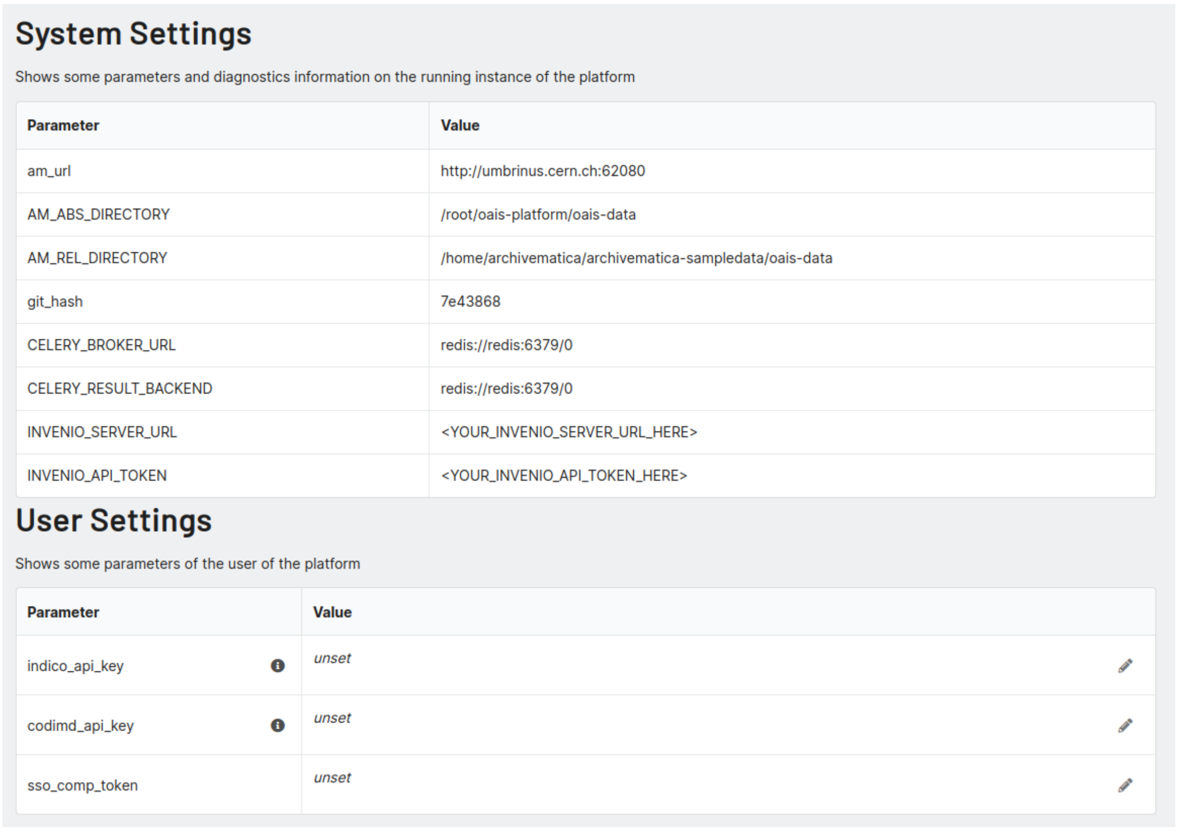
Archives

ID	Record	Creator	Creation Date	Actions
19	2827418 (cds)	root7	2022/09/22 14:51:46	DETAILS
18	1563526 (cds)	root7	2022/09/22 14:48:37	DETAILS
17	2824692 (cds)	root7	2022/09/22 14:48:37	DETAILS
16	1264113 (cds)	root7	2022/09/22 14:48:16	DETAILS
13	2298615 (cds)	root7	2022/09/22 14:48:16	DETAILS
11	2632312 (cds)	root7	2022/09/22 14:48:16	DETAILS

Figure 3.20: OAIS User Interface - Tags Detail Page

3.7.9 Settings

In the settings page, the users can see the current settings configuration and they can set or update their API tokens which can be used for the harvest of records from the repositories that require them (like Indico, CodiMD and Gitlab)



System Settings

Shows some parameters and diagnostics information on the running instance of the platform

Parameter	Value
am_url	http://umbrinus.cern.ch:62080
AM_ABS_DIRECTORY	/root/oais-platform/oais-data
AM_REL_DIRECTORY	/home/archivematica/archivematica-sampledata/oais-data
git_hash	7e43868
CELERY_BROKER_URL	redis://redis:6379/0
CELERY_RESULT_BACKEND	redis://redis:6379/0
INVENIO_SERVER_URL	<YOUR_INVENIO_SERVER_URL_HERE>
INVENIO_API_TOKEN	<YOUR_INVENIO_API_TOKEN_HERE>

User Settings

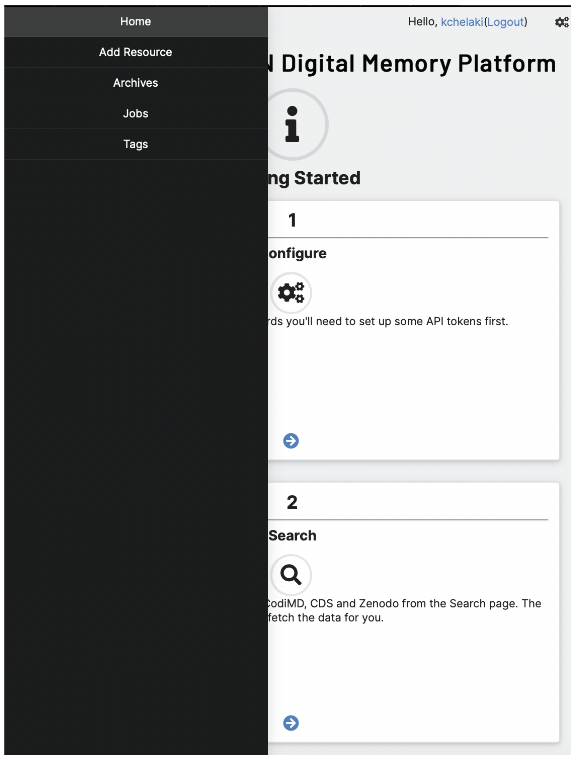
Shows some parameters of the user of the platform

Parameter	Value
indico_api_key	*unset*
codimd_api_key	*unset*
sso_comp_token	*unset*

Figure 3.21: OAIS User Interface - Settings page

3.7.10 Smartphone compatibility

The platform is fully responsive and compatible with mobile devices. All the components are optimized for smaller viewports so it can be used from a mobile device. The menu changes to a foldable one which makes it easier for the user to guide through the application. Figures 3.22, 3.23 and 3.24 depict screenshots from a mobile device.



N Digital Memory Platform

ng Started

1

onfigure

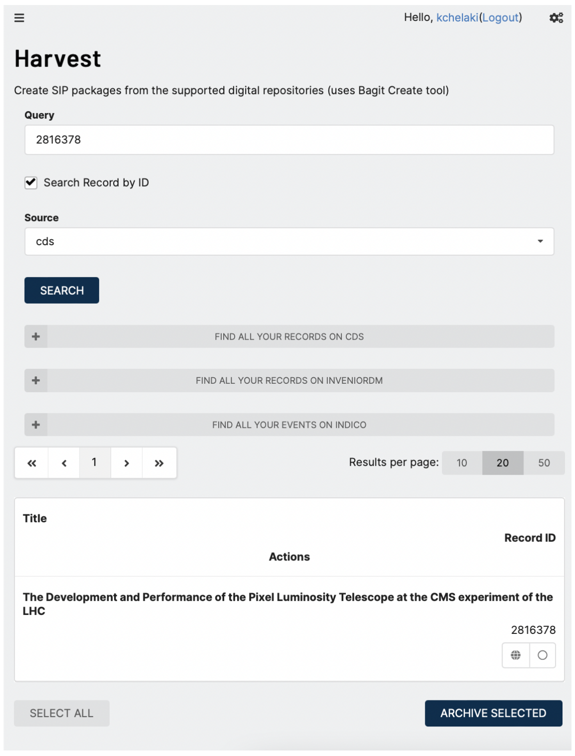
rds you'll need to set up some API tokens first.

2

Search

CodiMD, CDS and Zenodo from the Search page. The fetch the data for you.

Figure 3.22: OAIS User Interface - Home page mobile view with the foldable menu



Harvest

Create SIP packages from the supported digital repositories (uses Bagit Create tool)

Query: 2816378

[✓] Search Record by ID

Source: cds

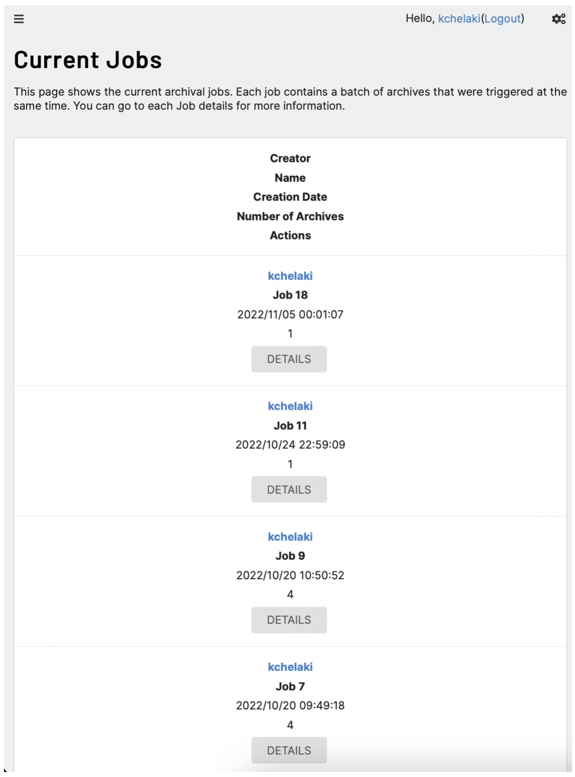
SEARCH
FIND ALL YOUR RECORDS ON CDS
FIND ALL YOUR RECORDS ON INVENIORDM
FIND ALL YOUR EVENTS ON INDICO
«
‹
1
›
»

Results per page: 10 [✓ 20] 50

Title	Record ID	Actions
The Development and Performance of the Pixel Luminosity Telescope at the CMS experiment of the LHC	2816378	

SELECT ALL
ARCHIVE SELECTED

Figure 3.23: OAIS User Interface - Harvest resources page mobile view



Current Jobs

This page shows the current archival jobs. Each job contains a batch of archives that were triggered at the same time. You can go to each Job details for more information.

Creator	Name	Creation Date	Number of Archives	Actions
kchelaki	Job 18	2022/11/05 00:01:07	1	DETAILS
kchelaki	Job 11	2022/10/24 22:59:09	1	DETAILS
kchelaki	Job 9	2022/10/20 10:50:52	4	DETAILS
kchelaki	Job 7	2022/10/20 09:49:18	4	DETAILS

Figure 3.24: OAIS User Interface - Jobs list page from a mobile view

3.8 Deployment

The deployment of the platform is a complex process as it contains many different tools and services that must work together properly. From the one hand there is the user interface and the OAIS Platform which is comprised of the Django application, Celery, the Database, pgadmin, Redis and nginx. From the other hand there is also Archivematica which is comprised of many components such as the Dashboard, the Storage Service, the MCP Server and MCP Client as well as the database and also external tools like ClamAV antivirus, Elasticsearch, Gearman and Redis. This makes the synchronization and the smooth operation of all the components difficult as there may be dependency issues, connection failures or a service can stop working and break the entire application.

All these can be solved by containerizing the whole application. In this way we can package each component separately and isolate it in its own container. Using containers has many benefits[18]:

- **Fast and easy to create:** Applications can be easily configured and set up by using YAML files and environment variables. In this way we can control how the container behaves.
- **Re-usability:** Imagine that a MySQL database container is needed.There are many docker containers available that perform exactly this, so we don't have to implement the whole MySQL application from scratch. We can take an existing MySQL docker image, change the configuration using the YAML file and environment variables and integrate it with our application. Docker has also an online library of docker images called Docker which help developers create, manage and deliver containerized applications.
- **Isolation:** Containers run all isolated from each other even if they are urn on the same server. This means that there is no reason to care about dependency issues between the different applications.
- **Security:** Containers communicate with each other by creating and using a local network. This creates an additional security network as important modules such as the database can only be accessible only from inside this network and it can prevent attackers from having access to the system.
- **Scalability:** An application using containers can be easily scaled by adding multiple identical containers. Also when a container crashes other containers running the same application can continue running making our whole system more reliable.

3.8.1 Openshift / Kubernetes

To use the benefits of containers to the maximum, like smart scaling, better resource management and container handling, a container orchestration can be used. Containers alone are just standalone isolated processes independent of the operating system they run and the infrastructure they run on. Container orchestration

[18]https://www.aquasec.com/cloud-native-academy/docker-container/container-advantages/

is the automation of the operations such as deployment, management, scaling and networking of containers. Openshift[19] is a hybrid cloud platform as a service orchestration management application which can be managed by the web application portal or through the CLI. Openshift uses Kubernetes as a kernel. Kubernetes[20] is an open source containerization system created by Google developers which allows users to manage services and workloads and helps the deployment, scaling and operations.

Some of the basic components of Kubernetes are shown below:

A **pod** is the smallest unit that can be deployed into Kubernetes and it is made up of one or more containers. Here lies the application code and logic. A pod is ephemeral which means that Pods are may be killed and re-spawned many times on one deployment. Each pod is assigned with an IP which is used to communicate with the other components. Each pod has a distinctive **deployment** which manages its performance.

As the pods are created and destroyed continuously depending on their state, a new IP is assigned every time a pod is re-spawned. This is problematic as the other services should keep track of the IP address so they can connect to the pod. This is solved with the **service**[21] which is an abstraction which contains a set of pods and the policy to access them. Each service can contain multiple pods of the same application which allows an easy way to scale the app.

Kubernetes places containers into Pods that run on **Nodes**[22]. A node is a virtual or physical machine, that is managed by the control pane and has all the services that are needed to run the Pods. The node name should be unique because kubernetes thinks that two nodes with the same name are the same object.

As pods are ephemeral, data storage is not persistent. Kubernetes user can request **Persistent Volume Claim (PVC)** to ask for storage. This PVC can request specific size and access modes. The piece of storage that is given from the cluster to the user is called **Persistent Volume (PV)** and is used to store data permanently.

Kubernetes manages the life cycle by using a control mechanism. It continuously checks the configuration passed and checks it against to the current status. If there is a difference on these values, then Kubernetes will act on it so it will automatically make fixes. For example if a Pod crashes for some reason, then Kubernetes will check its status against the expected value and if it is not the expected one it may stop that pod and create a new one.

3.8.2 Helm charts

Deploying the application on Openshift is done by using Helm charts. A helm chart is a files collection that describe a set of Kubernetes resources. These files are yaml components which are used by helm and deployed by Kubernetes. By configuring a helm chart it is very easy to use production ready applications like MySQL in

[19]https://docs.openshift.com
[20]https://kubernetes.io
[21]https://kubernetes.io/docs/concepts/services-networking/service/
[22]https://kubernetes.io/docs/concepts/architecture/nodes/

different environments. This simplifies a lot the development as using helm charts, we can override default values and configure any application according to our needs. Also it reduces complexity as a chart can be reused over and over again in many environments. Also it boosts productivity as there is no need to spend much time on the deployment of most of the services.

The basic structure of a Helm chart is the following one:

- **charts.yaml** It is a metadata file that contains information about the chart version.
- **values.yaml** It is the default configuration file which contains the basic configuration details for the helm chart.
- **templates/** This folder contains the template which combined with the values file creates the Kubernetes manifest file.

The helm charts provide in principal the instructions and the configuration on how the applications should be deployed. These instructions contain information of whether a pod, a route or a service will be created, which docker image will it use and what environment variables or further configuration is needed. One helm chart is sufficient to create all the components needed to run the whole application.

3.8.3 OAIS Platform Deployment

The OAIS platform is comprised of four different services which are: i) OAIS Platform, which contains the django application, the nginx server and the eos mount. ii) Celery iii) Redis and iv) the Postgres Database. As mentioned in 3.4, Celery is used for asynchronous task execution and management while Redis is an in-memory data store with which Celery and Django communicate. For these four services, four pods are needed respectively which will contain the images for each application and for the four Pods, four distinctive deployments which manage the Pods.

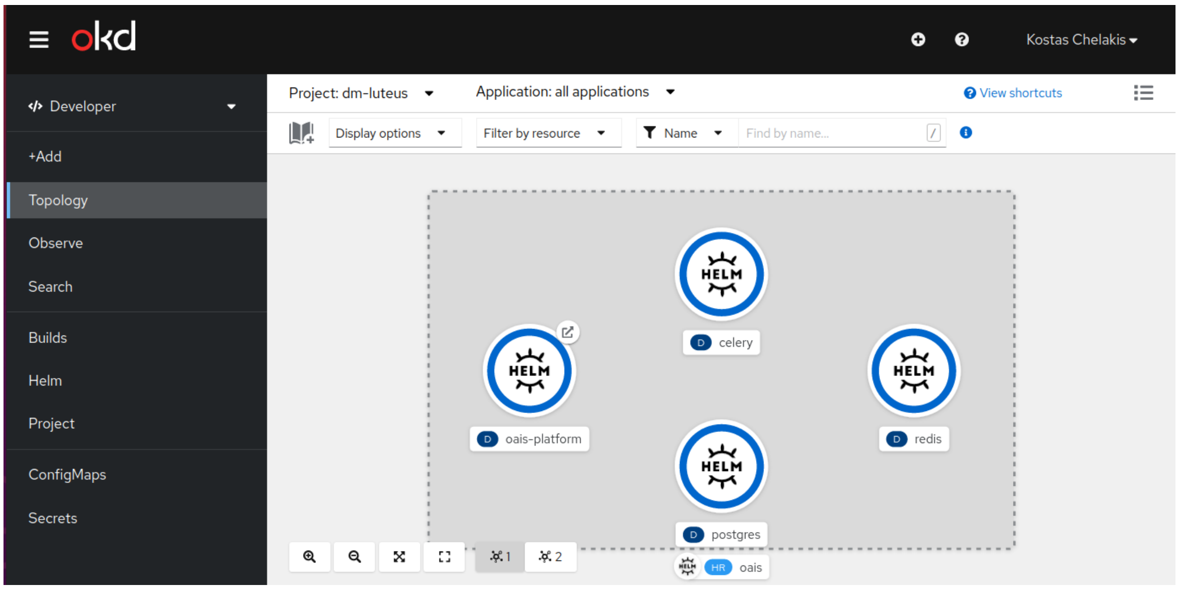


Figure 3.25: OAIS Platform deployment components

The deployments that need a Service to be exposed are the OAIS Platform, Postgres and Redis. In this way it is possible to communicate between the pods as the service assign a persistent IP to them. All pods need a service apart from Celery because it reads and writes back to Redis using Redis IP to connect.

In the Openshift deployment, eos can be added as a volume giving access to the disk storage system at CERN. In this way the platform can read and write directly on the eos filesystem. Openshift gives the ability to the user to create and expose a port using a specified hostname and domain name `hostname.web.cern.ch`.

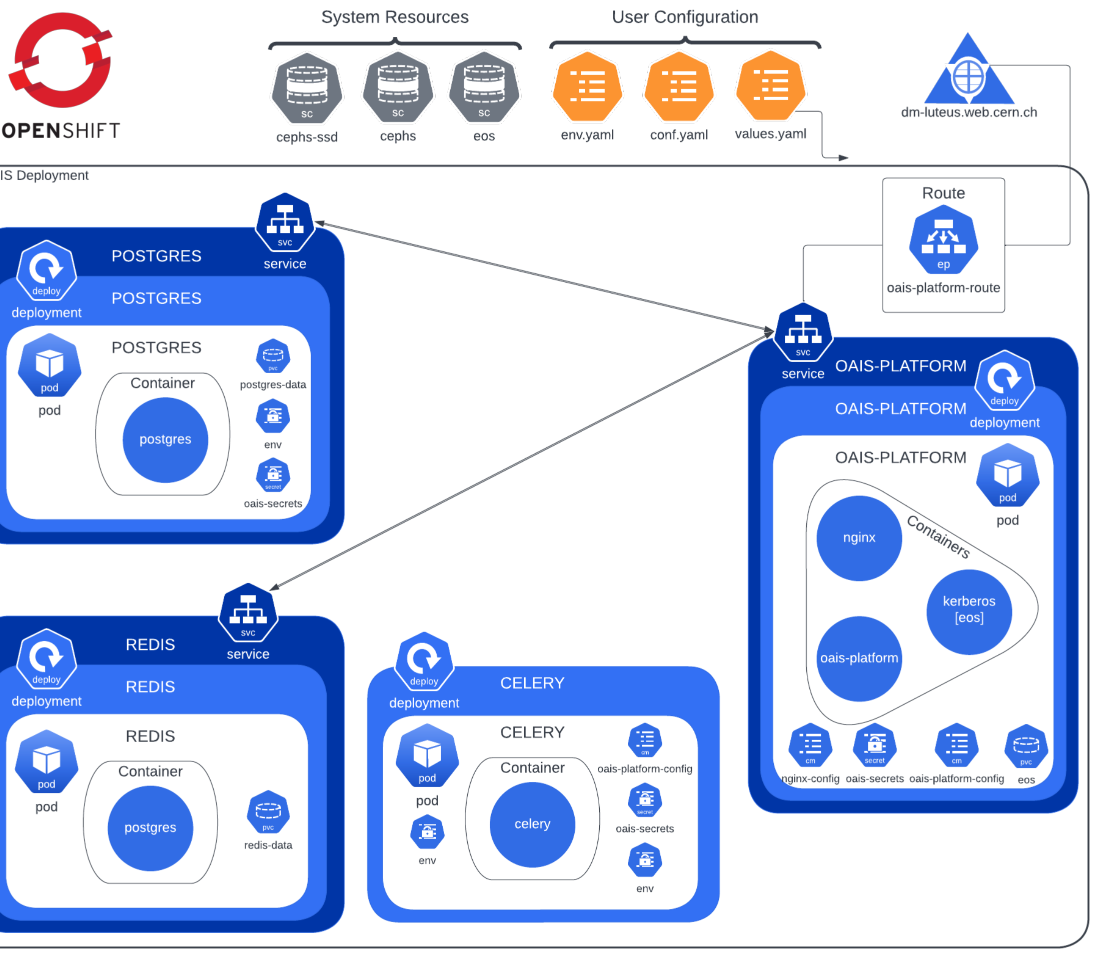


Figure 3.26: A detailed overview of the OAIS platform deployment.

3.8.4 Archivematica Deployment

Archivematica deployment is more complex than the platform because of the multiple components that are needed. As mentioned in 2.5.8, Archivematica is comprised of 10 different services that need to work together. These are the following:

- Dashboard
- Storage Service
- MCP Server

- MCP Client
- Gearman
- Elasticsearch
- Fits
- MySQL
- ClamAV
- Redis

The challenging part of deploying archivematica is that there is not enough documentation for the deployment using helm charts. There are published images to create the containers but there is no documentation to know how to control and set the variables for these containers. One attempt has been made by a company called Piql[23] which specializes in digital archiving but their solution was tailored for use with Amazon Web Services (AWS) platform and not for Openshift. The good thing is that their code is available on public and it can provide the basis to create the deployment. AWS can easily provide some services like the database and redis out of the box so these services have to be manually added. Moreover, images for the dashboard, the storage service, the MCP server and client can be created by using the containerised version of Archivematica while for the others there are many images and helm charts available.

[23] https://github.com/piql/PiqlConnect-AM

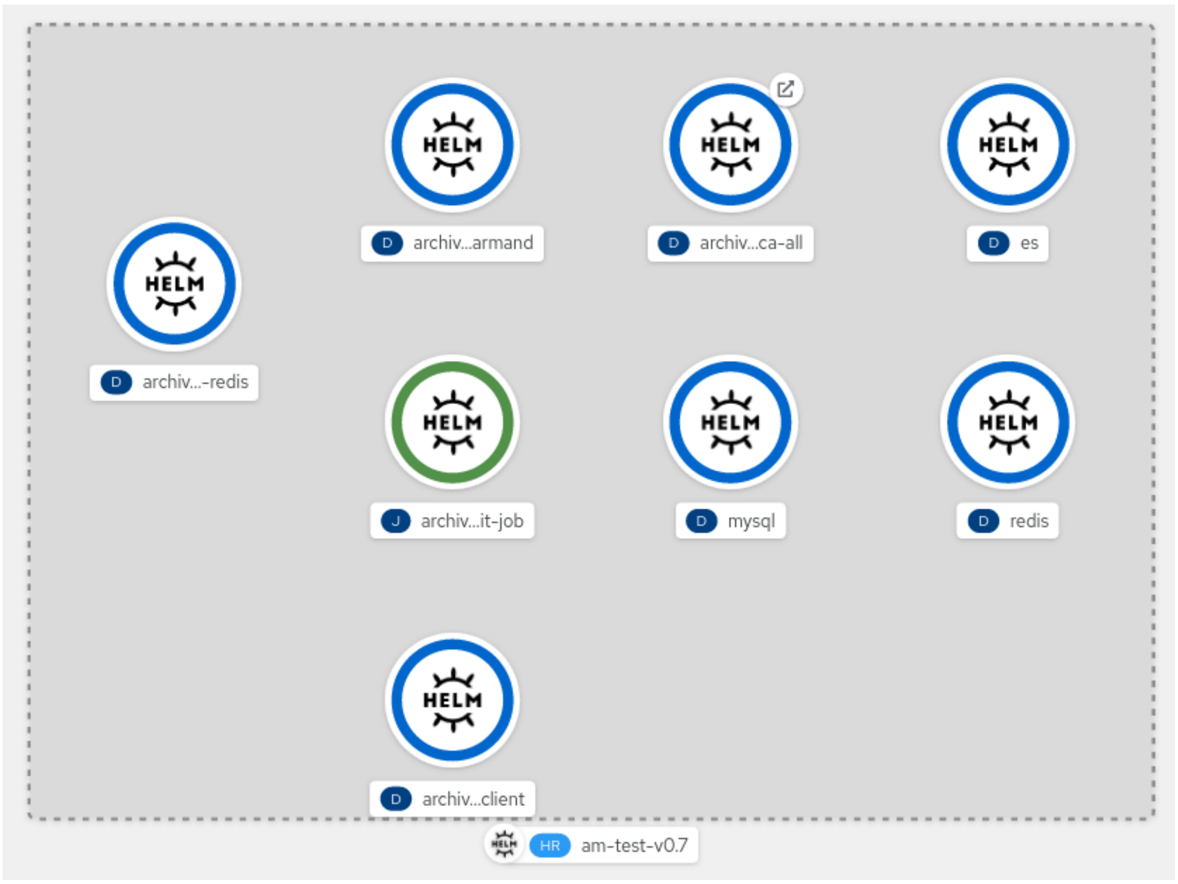


Figure 3.27: Archivematica deployment on Openshift

The core of archivematica are the dashboard, the storage service, the MCP server, the MCP Client, ClamAV, fits which can be added in a single pod. The rest of the containers can be added in separate pods.

At the deployment phase, we have to make sure that when deploying for the first time some initialization steps are executed. These initialization steps are needed to execute the python migrations and also to create an administrator user for the Dashboard and the Storage Service respectively. These tasks can be done by using the `initContainers` option in the helm chart. In this way we can specify some specific actions to be executed on the container creation. Here, we need to specify that there is a difference between the task of applying migrations when a container is initialized and the task of creating the user. In the first case, applying migrations will be needed after the initial deployment in case there is an update in the source code of the dashboard or the storage service.

On the other hand, the creation of the user must be done only once in the original deployment. For this reason, for the creation of the user a kubernetes job will be created. According to kubernetes docs, a job creates a pod which continues to run till some criteria are met. When this pod is successfully complete, the job tracks the successful completions and when the specified number of successful completions has been reached, then the job finished and does not run again. According to the above, a job is the appropriate solution for creating a user which will terminate when one user has successfully been created.

3.8.5 Gitlab CI/CD

Gitlab CI/CD is a tool for continuous deployment. It is used to find bugs and errors in the deployment as it can run tests and check if the code succeeds on them and if so, then it allows to merge to the master branch. Openshift deployment can be triggered directly by the gitlab pipelines. When the user pushes a new commit to the `master` branch then this can trigger an automatic update of the openshift deployment making sure that the deployment is always up to date with the latest code in the main repository. For testing purposes a conditional triggering was implemented which keeps track both of the changes on the `master` branch which contains the production code but also the changes on the `develop` branch which is a branch used for testing the latest features.

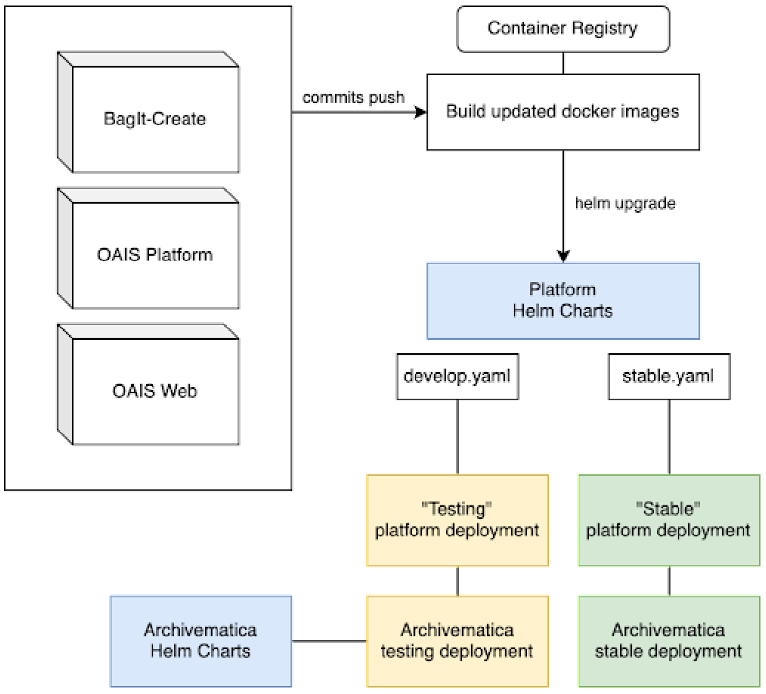
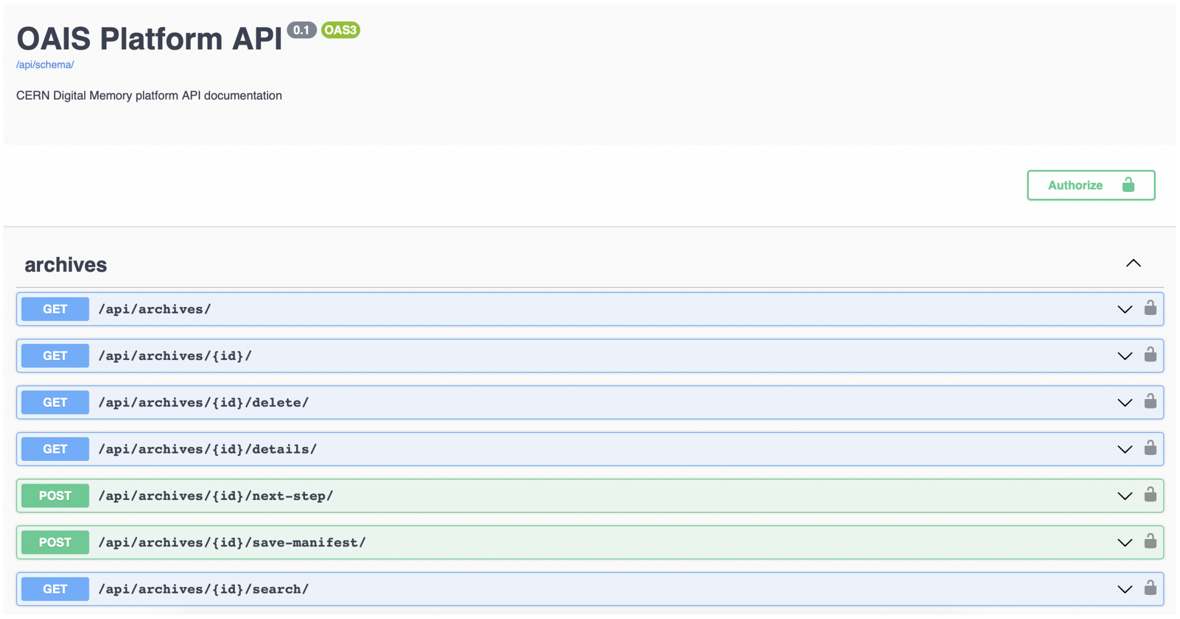

BagIt-Create
OAIS Platform
OAIS Web
commits push
Container Registry
Build updated docker images
helm upgrade
Platform Helm Charts
develop.yaml
stable.yaml
"Testing" platform deployment
"Stable" platform deployment
Archivematica Helm Charts
Archivematica testing deployment
Archivematica stable deployment

Figure 3.28: Gitlab pipeline flow

For this reason there are actually two OAIS Platform deployments, one for production and one for testing purposes which are exposed to different URLs. This conditional deployment every time new code is merged in one of these two branches, checks which branch triggered the pipeline and deploys either on the testing or the production deployment respectively. This helps to test the application directly from the test deployment and identify bugs and missing features more quickly. Also it gives the possibility to test the whole deployment directly on Openshift before merging to the production branch.

3.9 Rest API

The Rest API is the way to interact with the platform either by using the user interface provided or through the command line interface. API is developed using Django Rest Framework (DRF) and it is based on the OpenAPI specifications. More details about the API routes can be found in the Appendix.



CERN Digital Memory platform API documentation

archives

```
GET   /api/archives/
GET   /api/archives/{id}/
GET   /api/archives/{id}/delete/
GET   /api/archives/{id}/details/
POST  /api/archives/{id}/next-step/
POST  /api/archives/{id}/save-manifest/
GET   /api/archives/{id}/search/
```

Figure 3.29: Screenshot from Swagger UI Homepage

Researchers and archivists, can easily consult the Swagger UI web page for the documentation of the API routes. They can also see examples, demos and make API calls directly.

Chapter 4

Results & Optimization

In this chapter we will evaluate the performance of the platform and we will check how we can modify the deployments in order to reduce the processing time. Moreover we will check the created packages and we will compare it with the OAIS standard specifications explained in chapter 2.4. Finally we will check how the system as a whole is compliant with the standard.

4.1 Deployment Performance

Deployment on Openshift is the first step to set the application in production but because we are using shared resources, we need to make sure that all the services use adequate resources to run efficiently and at the same time that we allocate as much resources as we need and not more. Openshift allocates a specific amount of resources available for projects not in production with a maximum of 4GB of RAM and 2 CPU cores. In the beginning a performance measurement should be made, to see how the deployment is performing in terms of processing speed, memory management, package transmission etc and then check if there are ways to increase the performance. Since there are two deployments one for the platform and one for Archivematica, we will make study these two deployments separately. For testing the configuration we are going to use a sample data set of archives containing all different type of information such as videos, photos, documents etc. and modify the resources and the replicas of the services in deployment so we can compare them and find out which configuration results in the optimal processing time together with fair resource usage. The size of this batch of archives is 810 MB and includes 2 records with large videos from CDS, 2 events with various files like presentations, photos and documents uploaded from Indico and 2 publications from CDS.

4.2 Digital Memory Deployment

4.2.1 Digital Memory Performance

OAIS Platform deployment mentioned in 3.8.3 has four services running: OAIS Platform, Celery, Redis and Postgres. All the tasks run on Celery so in order to improve overall performance, we need to focus on optimizing Celery. In the default pipeline execution, the steps that are executed in the pipeline are the Upload, Harvest, Validate and Checksum. Even for large files, these steps are executed very quickly and because of Celery we can have parallel execution. The only step that may take more time to execute, is the harvest step but not because of the resources available, but because of the network speed. This happens because on this step, data from the upstream repositories are fetched.

4.3 Archivematica Deployment

4.3.1 Archivematica Performance

Archivematica as mentioned in 3.8.4, is a more complex deployment and the various applications may affect the overall performance considerably. Concerning the first execution of the first batch of archives, the MCP Client which is responsible for the execution of the microservices as an isolated process (see subsection 2.5.8), crashes due to memory limitations that occur from the default configuration. As a result these, the large videos included in the first batch hadn't been completed after 24 hours and the Pod running MCPClient had crashed multiple times with an OutOfMemory error.

4.3.2 MCPClient Quota

By default, Openshift assigns specific resources to each Pod but we can assign less or more resources till we can reach the optimal configuration. The first step is to increase the memory available to MCPClient to 1GB per pod and check check the performance again while extracting certain metrics. For metrics extraction, we use a functional query language called PromQL (Prometheus Query Language)[1] which is integrated with Openshift.

With these changes, archives are treated with an average time of 31 minutes and 12 seconds.

Below there are two useful charts about the CPU and memory usage for the configuration with 1 MCP Client with increased available memory.

[1] `https://prometheus.io`

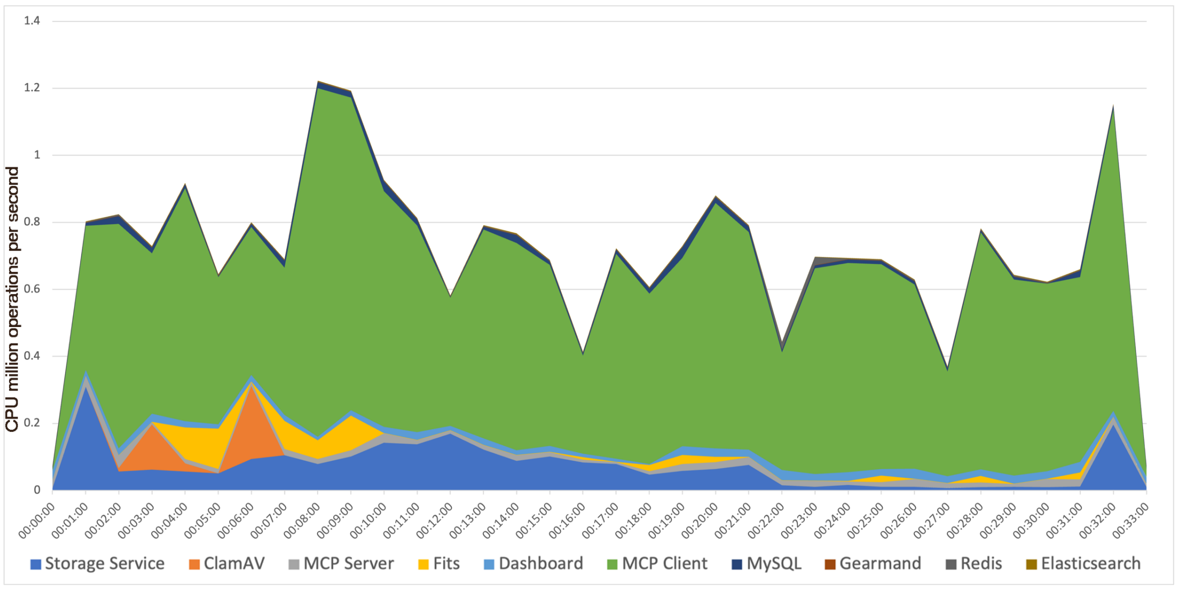


Figure 4.1: CPU Usage with 1 MCP Client

In figure 4.1, we can see that almost all CPU power is consumed on the MCP Client, while the Storage Service use more processing power in the beginning and the end of the archiving because data retrieval and storage takes place at that time.

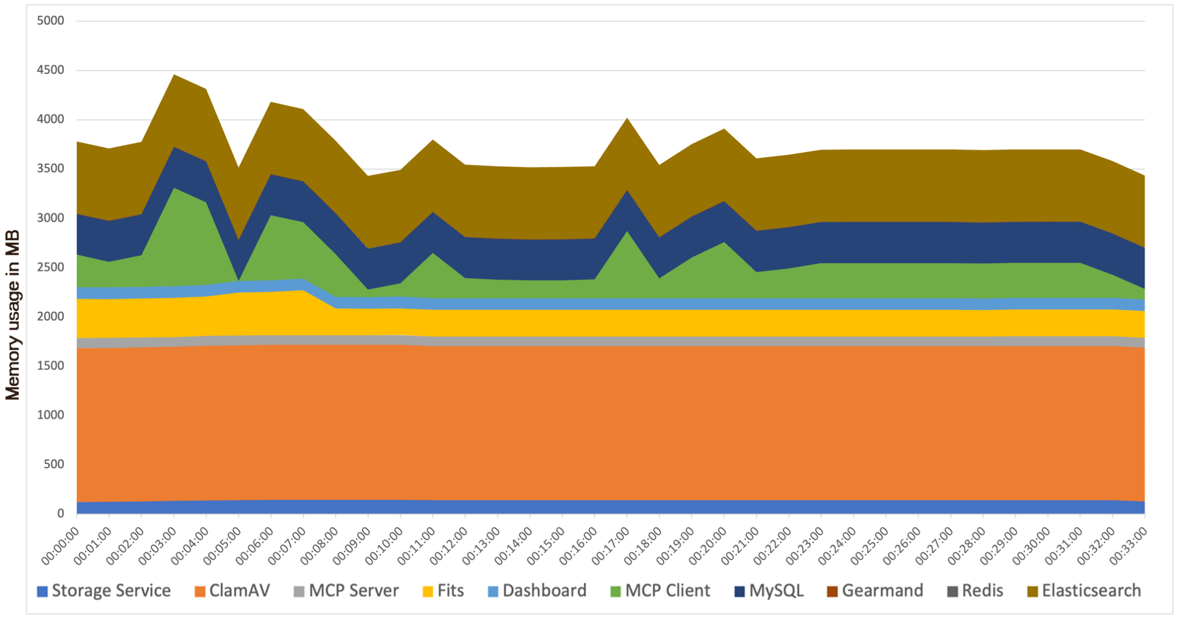


Figure 4.2: Memory Usage with 1 MCP Client

Memory usage is evenly distributed with MCP Client only requiring more memory usage when necessary, reaching also the 1GB use in a specific point.

4.3.3 MCPClient Replicas

In this case, with the suitable configuration, we can increase the replicas of MCP Clients to 2. With this configuration the goal is to execute more tasks in parallel, using multiple MCP Client instances. In this case execution time for the same amount of archives was reduced to 10 minutes and 55 seconds.

In figure 4.3 there are the charts about the CPU and memory usage for the configuration with 2 MCP Client with 1GB available memory each.

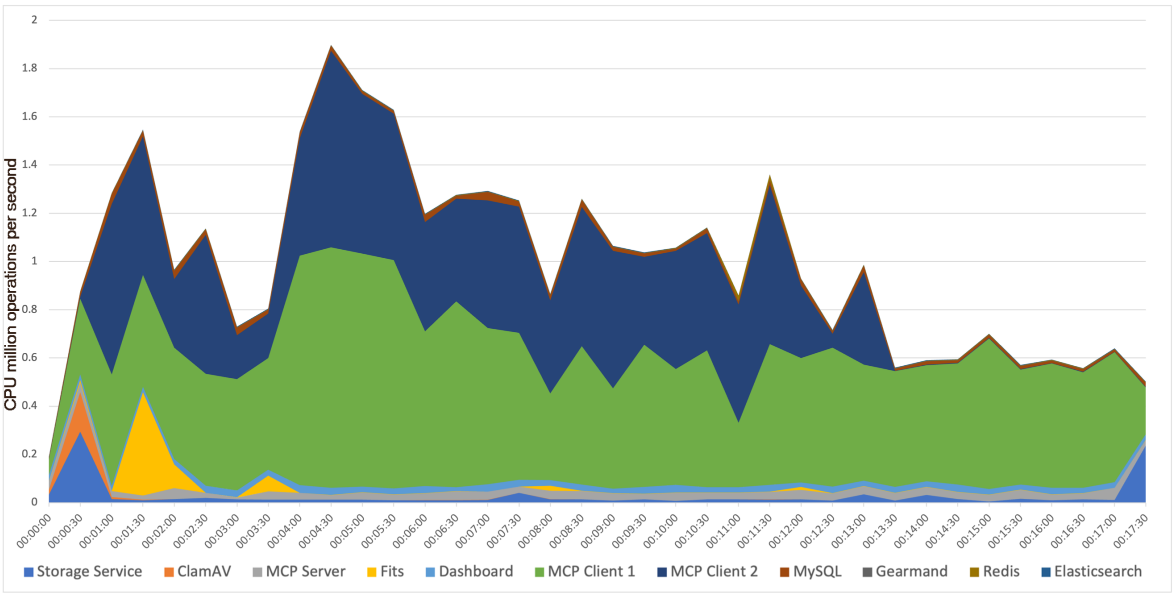


Figure 4.3: CPU Usage with 2 MCP Clients

In figure 4.3, we can see again that most of the CPU processing power is consumed by the MCP Clients, but in this case because of parallel execution, there is a significant decrease in time needed. Note that in the last minutes of the execution only one MCP Client is being used. This can be attributed to the processing of a big file such as a video that needed more time.

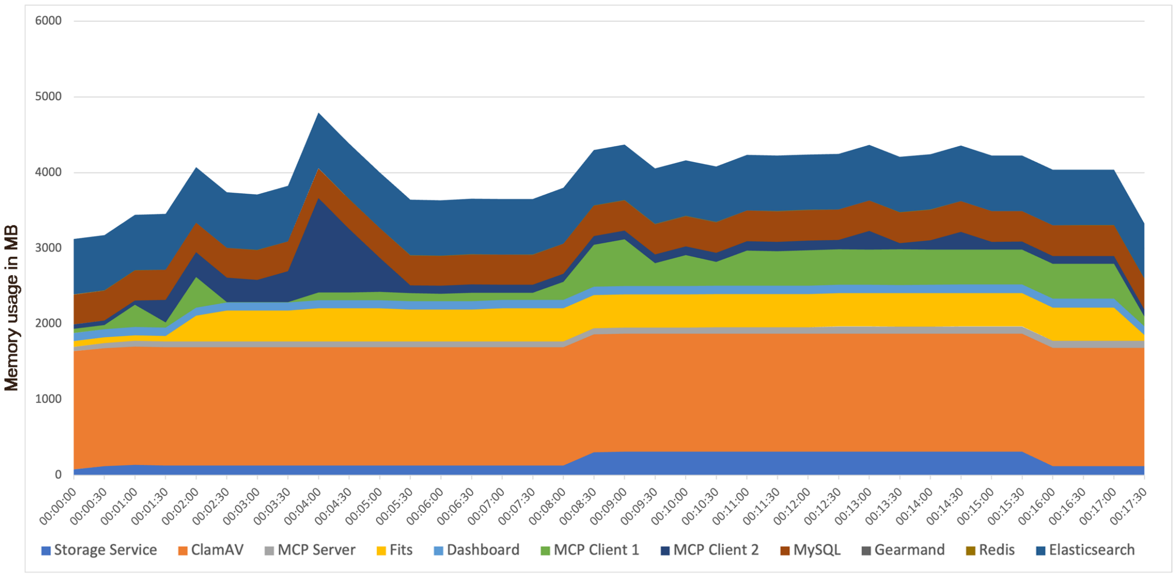


Figure 4.4: Memory Usage with 2 MCP Clients

4.3.4 MCPClient CPU

In the final configuration, we increase the CPUs available to the MCP Clients to the maximum available. This has indeed an increase in the final performance and the archives need on average 6 minutes and 50 seconds to be processed which is a much improved result.

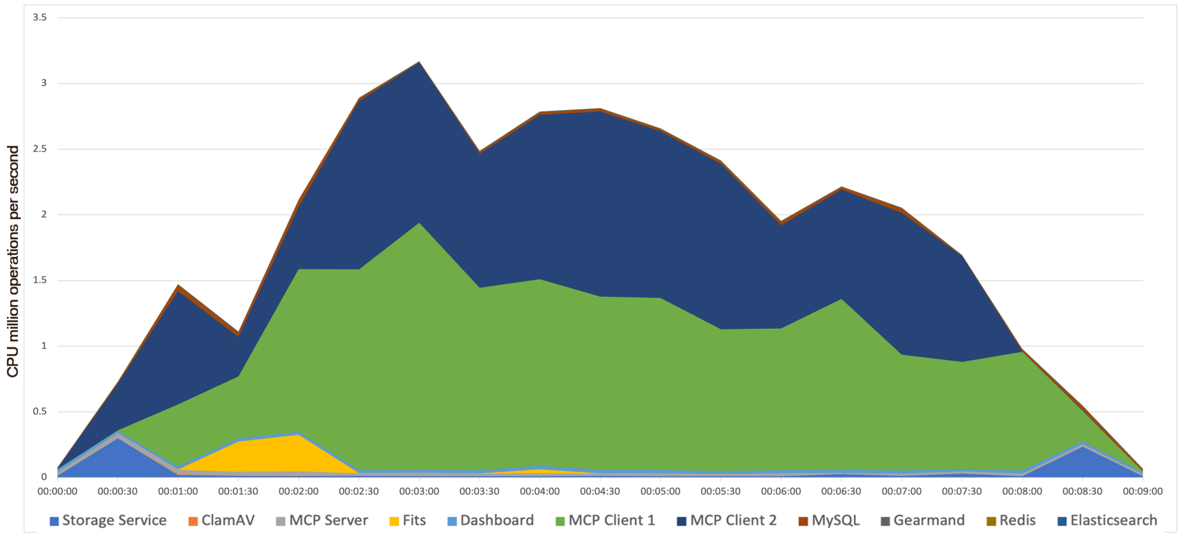


Figure 4.5: CPU Usage with 2 MCP Clients and increased CPU availability

In figure 4.5, we can see again that most of the CPU processing power is consumed by the MCP Clients, but in this case because of parallel execution, there is a significant decrease in time needed.

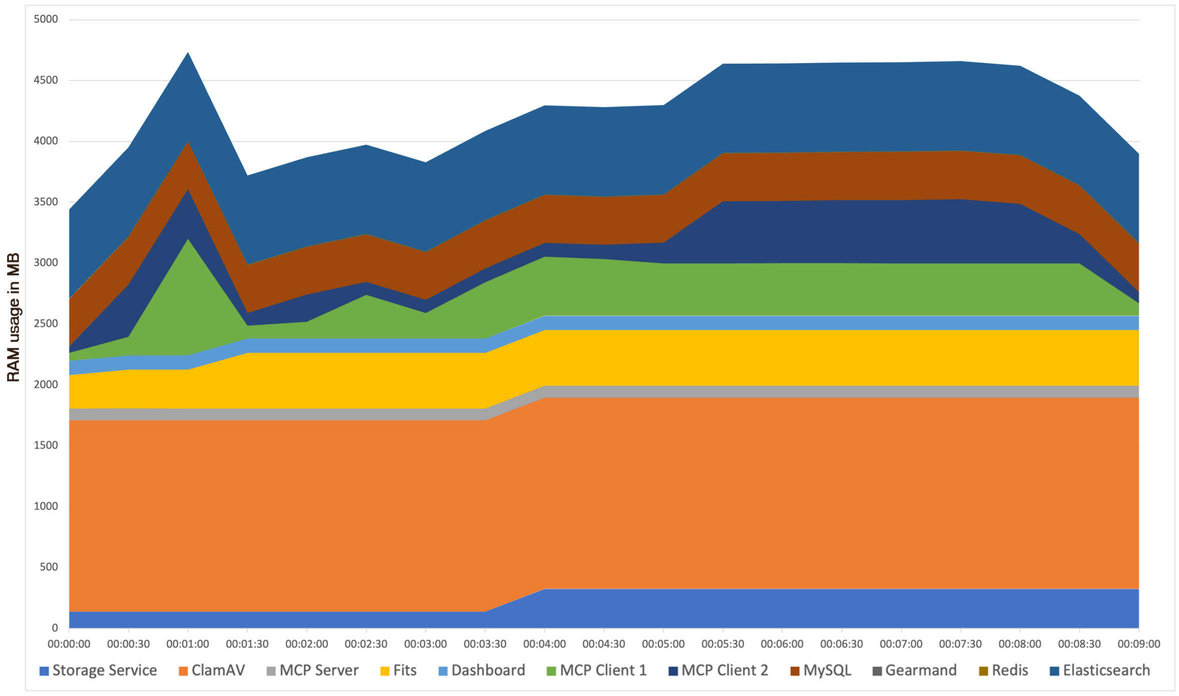


Figure 4.6: Memory Usage with 2 MCP Clients and increased CPU availability

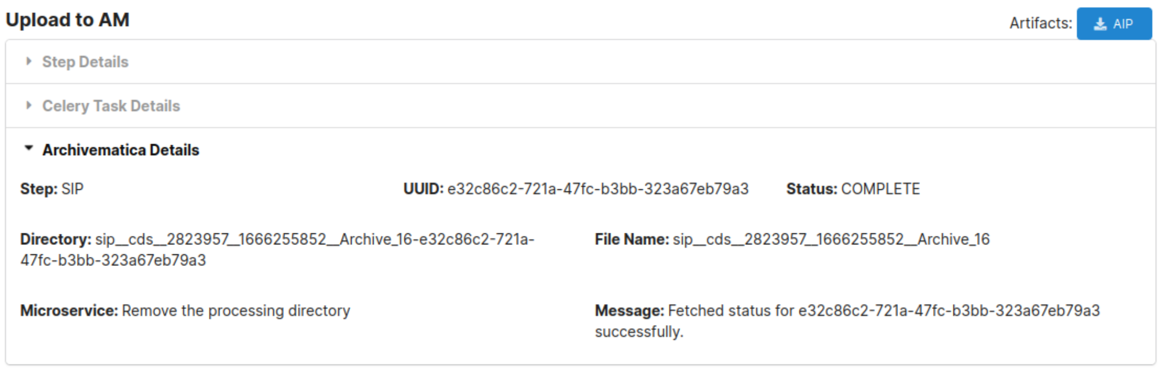
4.3.5 Final Results

A brief description about the time needed per archive and the minimum, maximum and average CPU and Memory usage can be shown below:

Configuration	1 MCP Client	2 MCP Clients	2 MCP Clients with allocated CPU
Fastest archive	30:41	07:17	06:02
Slowest archive	32:43	16:50	08:08
Average execution Time	31:12	10:55	07:10
Average throughput (MB/s)	0.43	1.23	1.88
Avg CPU usage	0.62	0.988	1.802
Avg Mem usage	3.54 GB	3.965 GB	3.889 GB

4.3.6 Created AIP Assessment

The created AIP can be downloaded from the platform's user interface or from Archivematica Storage Service. In the user interface, when an archivematica upload step is completed successfully, the AIP download button is exposed on top of the completed step at the archives detail page like in figure 4.7


Upload to AM

Step: SIP **UUID:** e32c86c2-721a-47fc-b3bb-323a67eb79a3 **Status:** COMPLETE

Directory: sip__cds__2823957_1666255852__Archive_16-e32c86c2-721a-47fc-b3bb-323a67eb79a3

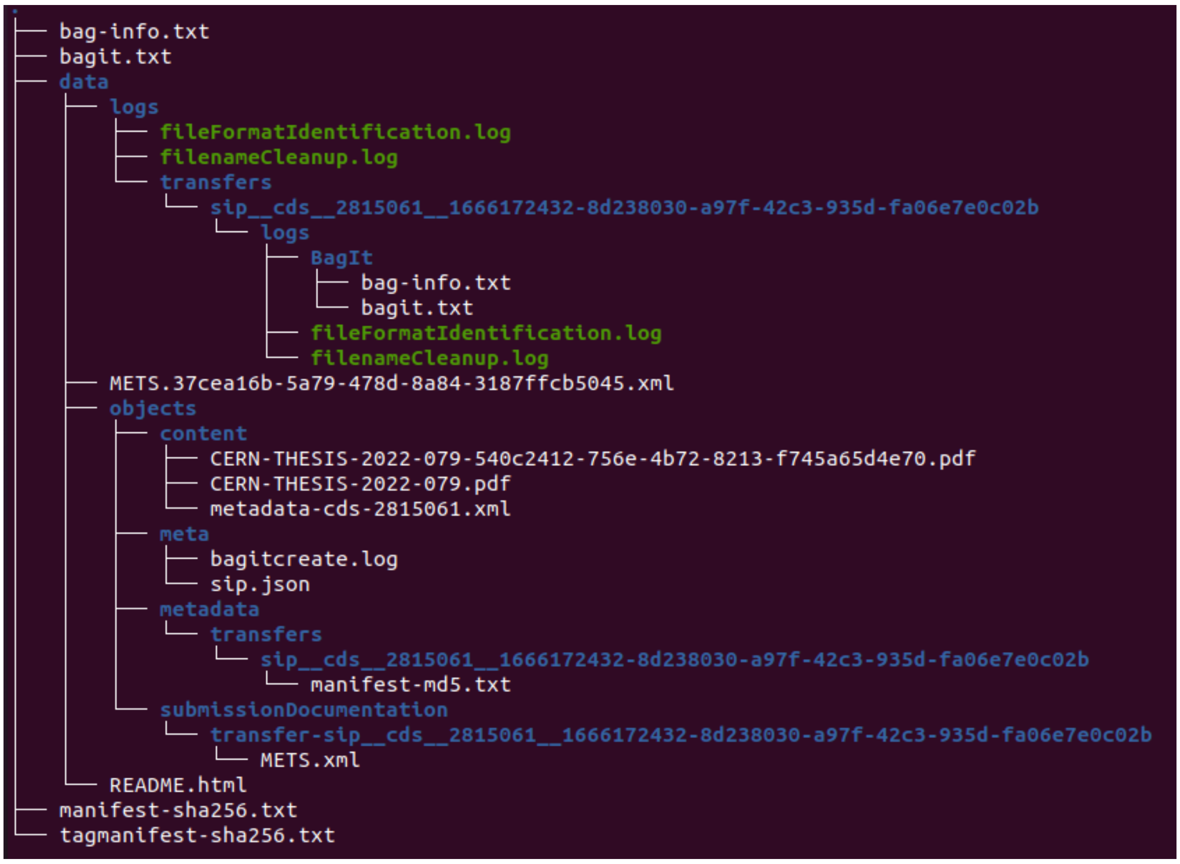
File Name: sip__cds__2823957_1666255852__Archive_16

Microservice: Remove the processing directory

Message: Fetched status for e32c86c2-721a-47fc-b3bb-323a67eb79a3 successfully.

Figure 4.7: AIP download button from the archive detail page.

In figure 4.8, we can see that in the `data/objects/content` directory, there are two PDF files, the one is the original PDF harvested from CDS while the second one is the normalized PDF type document.



```
.
├── bag-info.txt
├── bagit.txt
├── data
│   ├── logs
│   │   ├── fileFormatIdentification.log
│   │   ├── filenameCleanup.log
│   │   └── transfers
│   │       └── sip__cds__2815061__1666172432-8d238030-a97f-42c3-935d-fa06e7e0c02b
│   │           └── logs
│   │               ├── BagIt
│   │               │   ├── bag-info.txt
│   │               │   └── bagit.txt
│   │               ├── fileFormatIdentification.log
│   │               └── filenameCleanup.log
│   ├── METS.37cea16b-5a79-478d-8a84-3187ffcb5045.xml
│   ├── objects
│   │   ├── content
│   │   │   ├── CERN-THESIS-2022-079-540c2412-756e-4b72-8213-f745a65d4e70.pdf
│   │   │   ├── CERN-THESIS-2022-079.pdf
│   │   │   └── metadata-cds-2815061.xml
│   │   ├── meta
│   │   │   ├── bagitcreate.log
│   │   │   └── sip.json
│   │   ├── metadata
│   │   │   └── transfers
│   │   │       └── sip__cds__2815061__1666172432-8d238030-a97f-42c3-935d-fa06e7e0c02b
│   │   │           └── manifest-md5.txt
│   │   └── submissionDocumentation
│   │       └── transfer-sip__cds__2815061__1666172432-8d238030-a97f-42c3-935d-fa06e7e0c02b
│   │           └── METS.xml
│   └── README.html
├── manifest-sha256.txt
└── tagmanifest-sha256.txt
```

Figure 4.8: Tree graph of the resulted AIP.

The naming of the AIP directory, consists of the original name of the SIP submitted to archivematica, accompanied by a unique identifier (UUID) which is assigned during the package creation and can be used to identify the archive package.

The AIP directory structure is fully compliant with the BagIt Library of Congress specification[2] which has been analyzed also in chapter 3.2.1. To better understand the directory structure, we will use Figure 4.8 as reference, which represents a created AIP for a record containing a single PDF document. It is clear that the top directory files are `bag-info.txt`, `bagit.txt`, `manifest-sha256.txt`, `bagmanifest-sha256.txt` and the `data` directory as well. This is the same directory structure used for the SIP in 3.2.2.

The `data` directory consists of the `METS` file for the AIP, a `README` file and two folders, one for `logs` and one for the `objects`.[3]

- `METS.uuid.xml`[4] contains the full PREMIS implementation[5]. The role of the METS file is to link original objects to their preservation copies and to their descriptions and submission documentation, as well as to link PREMIS metadata to the objects in the AIP.

[2]https://blogs.loc.gov/thesignal/2019/04/bagit-at-the-library-of-congress/
[3]https://www.archivematica.org/en/docs/archivematica-1.8/user-manual/archival-storage/aip-structure/
[4]https://wiki.archivematica.org/METS
[5]https://wiki.archivematica.org/PREMIS_metadata:_original_files

- `logs/` folder includes the logs files generated in various stages of the package creation as well as the `transfer` directory which contains logs from processing that occured during the transfer workflow in the dashboard which is part of the SIP.
- `objects/` directory includes original objects, the normalized objects and their logs. More specifically, it includes the following directories:
  - `/content` where the original and the normalized files are contained.
  - `/meta` that includes the *bagitcreate.log* file and the *sip.json* which come from the SIP.
  - `/metadata` that includes further metadata that have been imported through the SIP.
  - `/submissionDocumentation` that includes submission documentation for each transfer which is part of the SIP and each transfer's METS.xml file and also donor agreements and transfer forms.
- `README.html` describes the basic structure of an Archival Information Package (AIP) generated by Archivematica.

4.3.7 AIP compliance with the OAIS standard

According to the OAIS Standard (see section 2.4), the AIP should contain all the Preservation Description Information and it must be itself a distinctive Information Object that is a container of other Information Objects.

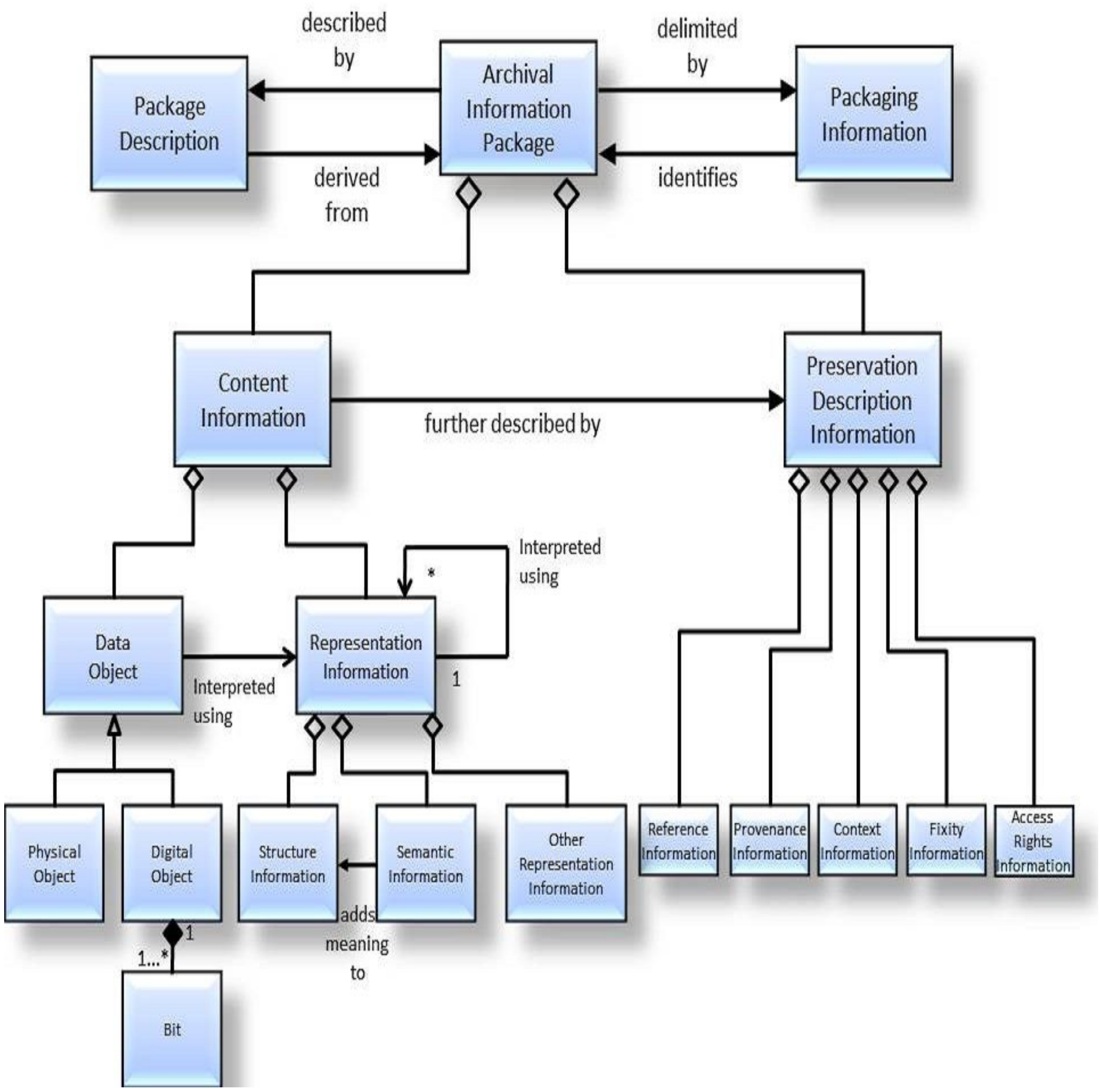


Figure 4.9: Archival Information Package (AIP) - Detailed View.

According to Figure 4.9, one main component of the Archival Information Package (AIP) is the Content Information which is further described by the Preservation Description Information (PDI).

The PDI in the created AIP is recorded in a `METS.xml`[6] file. The METS file usually consists of the following METS sections:

- **<mets:metsHdr>** *(METS header)* Includes basic information about the METS file.
- **<mets:dmdSec>** *(descriptive metadata section)* Includes descriptive metadata about the digital objects.
- **<mets:amdSec>** *(administrative metadata section)* Includes technical and prove-

[6]METS is maintained by the Library of Congress, which defines it as "a standard for encoding descriptive, administrative, and structural metadata regarding objects within a digital library, expressed using the XML schema language of the World Wide Web Consortium." [23]

nance information about the digital objects.

- **<mets:fileSec>** *(file section)* Includes a list of the digital objects together with an indication of their role in the AIP.
- **<mets:structMap>** *(structural map)* Includes an ordering of the digital objects.

Moreover, technical and provenance information in the METS *amdSec* is recorded in the PREMIS[7] metadata. These entities are wrapped in the METS file as shown below:

- **<mets:amdSec>**
  - **<mets:techMD>** *(technical metadata)*
    - **<premis:object>** e.g. UUID, size, checksum, format, original name, extracted technical metadata
  - **<mets:digiprovMD>** *(digital provenance metadata)*
    - **<premis:event>** e.g. ingestion, message digest calculation, virus scan, format identification, validation, normalization, fixity check
    - **<premis:agent>** for each PREMIS Event there are three associated Agents: the organization, the digital preservation system (e.g. Archivematica 1.x) and the logged-in user
  - **<mets:rightsMD>** *(rights metadata)*
    - **<premis:rights>** Rights pertaining to the preservation, reproduction and use of the preserved digital objects (only included if the user added rights metadata prior to or during ingest)

We can see though that all parts of the PDI are included in the PDI as follows:

- **Reference:** Unique identifier (UUID) assigned by Archivematica
- **Provenance:** Included in the <*mets:digiprovMD*> field in the *mets.xml* file
- **Context:** Included in the original *sip.json* (e.g. the url to the original source and the invenio instances)
- **Fixity:** Included in the *manifest-sha256.txt*, *tagmanifest-sha256.txt* and *manifest-md5.txt*.
- **Access Rights:** Included in the <*mets:rightsMD*> field in the *mets.xml* file

In the created AIP, the Content Information contains the original objects submitted and all normalised for preservation versions of these objects created. These copies have the same name with the UUIDs appended to the name of the file and some times with different file extension, depending on the preservation policy. This is part of the data object and it is stored in the `objects` folder. The Representation Information is stored in the metadata folder that includes information about the object including structure information and semantic information.

[7]PREMIS is also a Library of Congress standard, and is described as "the international standard for metadata to support the preservation of digital objects and ensure their long-term usability." [24]

Additionally, the AIP, is identified by the Packaging Information which according to the OAIS Standard is the information that binds the components of the package into an identifiable entity on specific media. In this case, this information is included in the *bag-info.txt* and includes the UUID and the software used to create the package. Package description is includes additional metadata which describe the contents of the AIP. This descriptive metadata are provided by the *METS.xml* and the *sip.json* files. Moreover in the *data/* directory, there is the *README* file which provides a self describing information about the package so future users can understand the structure of the package.

According to the package contents shown in 4.8 and the assumption we made above, we can conclude that the created AIP contains all the parts of the AIP specification shown in 4.9, making it OAIS-compiant.

4.3.8 System compliance with the OAIS Standard.

Apart from the Information Package specification described before, the system as a whole must have some additional features to comply with the OAIS standard. According to the standard (see section 2.4), it is required that Information Packages have an associated Descriptive Information which is used to locate them. *In our platform this information is included in the sip.json and additionally in the METS.xml file in the AIPs.*

Moreover according to section 3.1 of the OAIS standard [7], there are some mandatory responsibilities that an OAIS archive should have *(in italics there is the implementation on our system)*:

- Negotiate for and accept appropriate information from information Producers: *The platform accepts information from the producers according to the CERN SIP standard 3.2 and uses bagit-create to retrieve relevant information.*
- Obtain sufficient control of the information provided to the level needed to ensure Long Term Preservation: *The platform has full control of the provided information and the users can monitor and control the archival process through the user interface.*
- Determine, either by itself or in conjunction with other parties, which communities should become the Designated Community and, therefore, should be able to understand the information provided, thereby defining its Knowledge Base: *The designated community is for now the CERN personnel and researchers who can use the platform to archive information that they consider of significant importance to the organisation. Access to the platform from the designated community can be done through the CERN SSO.*
- Ensure that the information to be preserved is Independently Understandable to the Designated Community. In particular, the Designated Community should be able to understand the information without needing special resources such as the assistance of the experts who produced the information: *Package metadata and the README file in the information to be preserved such as the AIP, guarantees that information will be easily accessible in the future.*
- Follow documented policies and procedures which ensure that the informa-

tion is preserved against all reasonable contingencies, and which enable the information to be disseminated as authenticated copies of the original, or as traceable to the original: *From the creation of the SIP all the steps are recorded in the database and are tied to that specific archive so the preserved content can be easily traced back to the original. Moreover the original sip.json is part of the created AIP and can be used to authenticate the originality of the content.*

- Make the preserved information available to the Designated Community: *After the creation of the archival content, the user can log in again with the CERN account and have access to their archived information.*

Chapter 5

Discussion and Conclusion

5.1 Conclusion

The main goal of this diploma thesis was to design and implement an application for digital preservation complied with the OAIS standard. Initially, we showed why digital preservation is more relevant than ever and we presented the main components of the platform.

Using our platform, the users are able to harvest records from the supported repositories or to upload their local files. The platform then creates a Submission Information Package (SIP) which is fully compliant with the OAIS standard 2.4 and the CERN SIP Specification 3.2. These packages are then transferred to Archivematica, an open-source application that provides out-of-the box functionalities such as file normalisation, antivirus scanning, metadata extraction and results in the creation of the Archival Information Package (AIP) which is the preservation package and it is fully compliant with the OAIS standard.

An important feature of our platform is that archive creation can be easily controlled using the User Interface where created archives can be monitored, managed and organized. Alternatively, it is possible to use the platform through the rest API 3.9 and perform the same actions by using the CLI or through another application.

We also showed how the application can easily be deployed on Openshift or on Kubernetes using Helm Charts and how this procedure can be automated by using the Gitlab CI/CD pipelines.

Moreover, we examined the performance of the system especially that of the deployed Archivematica instance and we saw that we can achieve a data rate of 1.88 MB/s by using two MCP Client instances with more CPU power allocated. During our benchmarks, we saw that the workload is concentrated on the MCP Clients. The throughput specified above can be increased in the future by adding more quota for the Openshift deployment and by using more MCP Client instances.

Finally, we evaluated the created AIP produced by Archivematica, and we saw that it is fully compliant with the OAIS standard. Also we made sure that the system as a whole complies with the requirements posed by the standard. This means that the created packages can be stored and have increased chances to be available in the long term.

5.2 Further Improvements

Even though the basic features for the creation of a digital archiving platform for CERN were implemented, there are some features that will be useful to be included in the future either to increase the performance of the platform or to extend the current functionality. Some of the basic features are given below:

- **DIP creation request from the UI:** We show that a fundamental part of an OAIS archive is the creation of the dissemination information package (DIP) on request of the user. Currently Archivematica creates DIPs by default after creating the AIP as we saw in 3.5. The created DIP is available for download through Archivematica's Storage Service. This feature could be moved to the user interface where the user should be able to ask from the platform to create an up-to-date DIP and when available, to be able to download it.

- **Integration with long term digital storage services:** This integration will be useful in order to actually have many storage mediums available to put the created archival packages. One possible integration already specified is the CERN Tape Archive (CTA), but similar storage facilities can be added to the platform as well.

- **Further integration with InvenioRDM:** As we saw in 3.6.2, an Invenio RDM instance is planned to be used along with the platform to support indexing and versioning of the archives. Right now the invenio instance is completely isolated by the platform and hosted seperately. In the future, the invenio instance can be customised and integrated further with the platform by using the same hostname and the same user interface.

- **Updated Archivematica preservation policy:** Preservation policy and rules should be evaluated and updated in regular intervals. Each time, the best normalisation and transcription policies should be chosen in order to guarantee that the normalised files will be in the most secure format available at that moment. Given the fact that in the scope of this thesis we used mainly Archivematica's default preservation policy, this can be reviewed and potentially modified and tailored to CERN's specific needs.

- **Improvement of the deployed Archivematica instance:** As we saw in chapter 4, the current Archivematica configuration works efficiently with the current data volume. We also saw that we can increase the performance of the system by increasing the instances of the MCP Client pods or by allocating more resources. The true bottleneck of the system is that we may still not be able to ingest many archives at the same time. This can be fixed by creating a queue which will constantly check how many transfers are active at any given moment and create new ones only when a transfer slot is available. This will increase the availability and reliability of our system as package creation will be guaranteed.

Appendix A

Appendix

A.1 Rest API

A.1.1 Archives API

- `GET /api/archives`

Returns a paginated list of archives

- `GET /api/archives/{id}/`

Returns all details for the archive specified by the id.

- `GET /api/archives/{id}/delete/`

Deletes a specific archive (works only for staged archives)

- `GET /api/archives/{id}/next-step`

Creates the next Step of the passed Archive

- `GET /api/archives/{id}/save-manifest/`

Gets a new manifest object and creates a new manifest step which keeps track of the manifest changes and adds the new manifest to the manifest field of the archive object

- `GET /api/archives/{id}/details/`

For the given archive, returns other archives in the registry with same source and record id

- `GET /api/archives/{id}/search/`

For the given archive, returns other archives in the registry with same source and record id

- `GET /api/archives/{id}/steps/`

For the given archive, returns all the associated steps.

- `GET /api/archives/{id}/tags/`

For the given archive, returns all the tags that have been applied.

- `POST /api/archives/{id}/unstage/`

Unstages the given archive (sets the staged boolean to false)

- `POST /api/archives/details/`

Given a list of archives, returns the same list with additional information about similar archives (same source and record id)

- `POST /api/archives/unstage/`

Given a list of archives, unstages all of them (sets the staged boolean to false)

- `GET /api/get-archive-information-label/`

Gets the number of archives that are in the staged area for the current user (staged field equals to true)

A.1.2 Steps API

- `GET /api/steps/{id}/`

Given the id of the step returns the details of the step

- `POST /api/steps/{id}/approve/`

Approves the identified step (for administrators only)

- `POST /api/steps/{id}/reject/`

Rejects the identified step (for administrators only)

A.1.3 Tags API

- `GET /api/tags/`

API endpoint that returns a paginated list of tags that have been created or are accessible by the user.

- `GET /api/tags/{id}/`

Given the id of a tag, returns details for that specific tag.

- `POST /api/tags/{id}/add/`

Given a list of archives and the id of a tag, adds the identified tag to the passed archives.

- `GET /api/tags/{id}/archives/`

Given the id of a tag, returns a list of the archives that are associated with this tag.

- `POST /api/tags/{id}/delete/`

Given the id of a tag, deletes that tag.

- `POST /api/tags/{id}/edit/`

Given the id of a tag, the name and the description, updates the specified tag with the new name and description values.

- `POST /api/tags/{id}/remove/`

Given a list of archives and the id of a tag, removes the identified tag from the passed archives.

- `POST /api/tags/create/`

API endpoint to create a new tag given the tag name, tag description and an optional list of archives.

A.1.4 Upload API

API endpoint that allows to create UploadJobs, add files, and submit

- `POST /api/upload/jobs/{id}/add/file/`

Adds the given file to the specified UploadJob.

- `POST /api/upload/jobs/{id}/sip/`

Creates an SIP calling bagit create on the specified UploadJob.

- `POST /api/upload/jobs/create/`

Initializes an UploadJob, returns its id and its corresponding temporary directory.

- `POST /api/upload/sip/`

Uploads a compressed SIP record to the platform.

A.1.5 Users API

- `GET /api/users/`

Returns a list of all the users of the system.

- `GET /api/users/{id}/`

Returns details for a user identified by the passed id.

- `GET /api/users/{id}/archives/`

Returns all the archives created by a specific user.

- `GET /api/users/me/`

Returns information and settings about the User.

- `POST /api/users/me/`

Updates information about the User using the passed values to overwrite.

- `GET /api/users/me/sources/`

Returns the status of all the upstream sources available to the specific user.

- `GET /api/users/me/staging-area/`

Returns all the staged archives of the specific user.

- `GET /api/users/me/stats/`

Returns all Steps and their status for the specific User

- `GET /api/users/me/tags/`

Returns all Tags created by the User

A.1.6 Search API

- `GET /api/search/{source}/`

Given the search query and the source to search returns a paginated list of results that correspond to that search term.

- `GET /api/search/{source}/{recid}/`

Given the source and the ID of the record in the upstream repository, returns this record.

- `GET /api/search/parse-url/`

Given a URL from the supported repositories, returns the pointed record.

A.1.7 Other API calls

- `POST /api/login/`

Given the username and the password, if they are correct, logs in the user.

- `POST /api/logout/`

API endpoint to log out the current user.

- `POST /api/get-archive-information-labels/`

Gets the number of all the staged and unstaged archives for a specific user.

- `POST /api/harvest/{id}/`

Creates an Archive given the Source and Recid and assigns a havert Step to it

- `GET /api/settings/`

Returns a collection of (read-only) the main configuration values and information about django configuration.

- `POST /api/records/check/`

Gets a list of records and searches the database for similar archives (same recid + source) and returns the list of records with an archive list field which contains the similar archives

Appendix B

Acronyms

SIP Submission Information Package

AIP Archival Information Package

DIP Dissemination Information Package

MCP Manage, Control and Plan

PDI Preservation Description Information

METS Metadata Encoding and Transmission Standard

PREMIS Preservation Metadata Maintenance Activity

UUID Unique Universal Identifier

OAIS Open Archival Information System

CERN European Organization for Nuclear Research

LHC Large Hadron Collider

CDS CERN Document Server

COD CERN Open Data

CTA CERN Tape Archive

RDM Research Data Management

UI User Interface

API Application Programming Interface

ID Identification

RECID Record ID

SSO Single Sign-On

CLI Command-Line Interface

CPU Central Processing Unit

Bibliography

[1] A. G. Holzner, R. Gokieli, P. Igo-Kemenes, M. Maggi, L. Malgeri, S. Mele, L. Pape, D. Plane, M. Schröder, U. Schwickerath, R. Tenchini, and J. Timmermans, “Data Preservation at LEP,” Tech. Rep., 2009, comments: 7 pages, contribution to proceedings of the First Workshop on Data Preservation and Long Term Analysis in HEP. [Online]. Available: https://cds.cern.ch/record/1228028

[2] J. Shiers, “Distributed File and Tape Management System,” 1991, cERN Program Library Short Writeups. [Online]. Available: https://cds.cern.ch/record/2050894

[3] R. Pryterch, *Harrod’s librarians’ glossary and reference book (10th edition)*. Ashgate, 2005.

[4] J.-Y. Le Meur and N. Tarocco, “The obsolescence of information and information systems: CERN Digital Memory project,” *EPJ Web Conf.*, vol. 214, p. 09003, 2019. [Online]. Available: https://cds.cern.ch/record/2649765

[5] G. Pallab, “Google’s vint cerf warns of ’digital dark age’,” Feb 2015. [Online]. Available: https://www.bbc.com/news/science-environment-31450389

[6] B. Lavoie, “The Open Archival Information System (OAIS) Reference Model: Introductory Guide (2nd Edition),” Digital Preservation Coalition, Tech. Rep., Oct 2014. [Online]. Available: https://www.dpconline.org/docs/technology-watch-reports/1359-dpctw14-02/file

[7] C. C. for Space Data Systems (CCSDS), “Reference model for an open archival information system (oais),” *Recommended Practice, issue 2, CCSDS 650.0-M-2.*, 2012. [Online]. Available: https://public.ccsds.org/Pubs/650x0m2.pdf

[8] D. S. Rosenthal, “Lockss: Lots of copies keep stuff safe,” 2010.

[9] P. Caplan, “Daitss and the florida digital archive.” [Online]. Available: https://slideplayer.com/slide/13238718/

[10] ——, “Daitss, an oais-based preservation repository,” in *Proceedings of the 2010 Roadmap for Digital Preservation Interoperability Framework Workshop*, ser. US-DPIF ’10. New York, NY, USA: Association for Computing Machinery, 2010. [Online]. Available: https://doi.org/10.1145/2039274.2039291

[11] “Daitss core service repository.” [Online]. Available: https://github.com/daitss/core

[12] F. Louise and P. Sébastien, “A data-first preservation strategy: Data

management in spar," Tech. Rep. [Online]. Available: https://www.ifs.tuwien.ac.at/dp/ipres2010/papers/fauduet-13.pdf

[13] C. Billenness, "E-ark project, year 3 summary," Tech. Rep., 2017. [Online]. Available: https://www.eark-project.com/resources/annual-summaries/100-annual-project-summary-year-3/E-ARK%20Summary%20Year%203.pdf_%3b%20filename_%3dUTF-8%27%27E-ARK%2520Summary%2520Year%25203.pdf

[14] J. Allinson, "Oais as a reference model for repositories," Tech. Rep., 2006.

[15] R. Tansley, M. Bass, and M. Smith, "Dspace as an open archival information system: Current status and future directions," Hewlett-Packard Laboratories and Massachussets Institute of Technology Libraries, Tech. Rep. [Online]. Available: https://core.ac.uk/download/pdf/4395968.pdf

[16] Dataverse, "The dataverse project." [Online]. Available: https://dataverse.org/about

[17] Archivematica, "Archivematica's technical architecture." [Online]. Available: https://www.archivematica.org/en/docs/archivematica-1.13/getting-started/overview/technical/#technical-arch

[18] P. Van Garderen, "Archivematica: Using micro-services and open-source software to deliver a comprehensive digital curation solution," Tech. Rep. [Online]. Available: https://ipres-conference.org/ipres10/papers/vanGarderen28.pdf

[19] J. van Kemenade, "The CERN Digital Memory Platform: Building a CERN scale OAIS compliant Archival Service," 2020, presented 28 Jun 2020. [Online]. Available: https://cds.cern.ch/record/2728246

[20] "Report on long-term electronic archiving (ltea)." [Online]. Available: https://cds.cern.ch/record/1028139/files/cer-002686760.pdf

[21] J.-Y. Le Meur, "The Digital Memory of CERN: status and input for a Preservation Strategic Plan," CERN, Geneva, Tech. Rep., Jul 2016. [Online]. Available: https://cds.cern.ch/record/2200146

[22] E. Cano, V. Bahyl, C. Caffy, G. Cancio, M. Davis, O. Keeble, V. Kotlyar, J. Leduc, and S. Murray, "CERN Tape Archive: a distributed, reliable and scalable scheduling system," *EPJ Web Conf.*, vol. 251, p. 02037, 2021. [Online]. Available: https://cds.cern.ch/record/2799972

[23] T. library of Congress, "Metadata encoding & transmission standard." [Online]. Available: https://www.loc.gov/standards/mets/

[24] ——, "Preservation metadata maintenance activity." [Online]. Available: https://www.loc.gov/standards/premis/