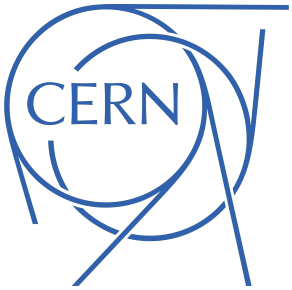
Digital Memory
Enhancement of the Preserve Platform
Summer Student Program Report

Author: Maria-Teona Olteanu
Supervisors: Jean-Yves Le Meur, Panna Georgina Lipták

October 10, 2024



Abstract

In 2016, CERN launched the Digital Memory Project to digitally preserve its heritage data and ensure that all acquired knowledge will be accessible to future generations. One of the critical initiatives - the Preserve Platform, is focused on providing existing information systems with OAIS-compliant archival workflows. Throughout this report, we present the enhancements brought to this service and future improvements in view of contributing to CERN's long-term digital preservation efforts.

1 Introduction

Since the digital revolution, society has transformed into a data-driven world, generating vast quantities of information at a rapid rate. Thus, the risk of losing data by not preserving it digitally in an adequate manner would inevitably lead to a hole in humankind's knowledge. The scientific community does not escape this problem. It is even more present, keeping in mind the responsibility to ensure that past discoveries and historical information are available to future generations. Consequently, one of CERN's missions is to make sure all its assets, such as raw data, multimedia content, documents, and code, will be accessible years from now on.

It is worth noting that digital preservation is not an easy feat - in reality, we only realize we need certain information once we no longer have access to it. So, the problem of what we should archive arises. Moreover, especially when preserving digital artifacts, there are significant risks such as bit rot, hardware or file format obsolescence, dissipation, ambiguous intellectual property etc[1]. Evidence of bad practices can also be observed in CERN's past[2], for example, quadriplex tape data being lost due to hardware deprecation or essential documents stored on a personal laptop becoming inaccessible. The OAIS (Open Archival Information System)[3, 4], the ISO standard for digital preservation, outlines a clear framework on how to overcome these challenges. Consequently, the implementation of this model enables the scientific community to ensure long-term digital preservation.

Within the Digital Memory Project, the Preserve Platform represents an OAIS-compliant system built on top of CERN's cloud to streamline the archival process for information systems at CERN, such as Indico or repositories like Zenodo or the new CERN Document Server (CDS), which are based on Invenio-RDM, by enabling the harvesting, ingestion and long-term archival of existing records. During the Summer Student Program, I focused on improving several components of the platform, particularly in areas such as archive filtering, pushing information packages to tape, and the archival processing pipeline.

2 Preserve Platform

As stated, the platform is designed based on the OAIS Reference Model framework. Moving forward, for a better understanding, we will explain the main proposed functional entities of the model displayed in Figure 3, which are included in the Preserve Platform implementation.

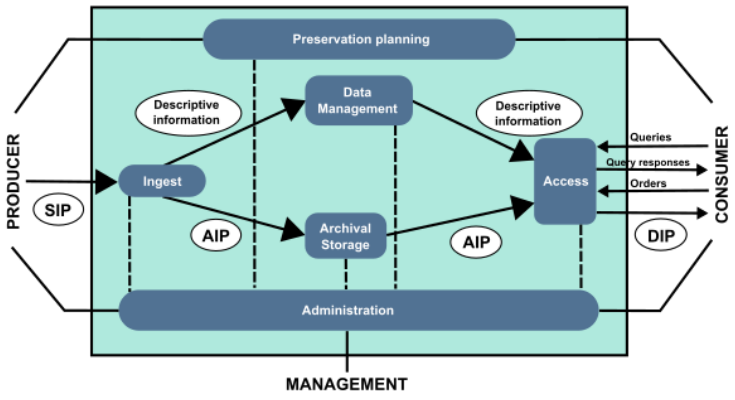


Figure 1: OAIS Reference Model

Ingest This component implies validation and preparation for long-term storage of Submission Information Packages (SIPs) by creating Archival Information Packages (AIPS) from one or multiple

SIPs. Before Ingestion, the Preserve Platform also enables the creation of SIPs from records sourced from CERN's information systems.
Archival Storage This function conducts the long-term storage of the AIPs; in CERN's case the information packages is written to tape, as well as providing a way for the archived information to be accessed by the Consumer.
Data Management The Data Management component handles the maintenance and management of the packages metadata and execution queries sourced by the Access module.
Access This functional entity refers to providing Consumers a way of requesting and searching for archived information and also delivering preservation information through DIPs (Dissemination Information Packages).

Regarding technical components, the Preserve Platform comprises of a REST API interface, OAIS platform[5], which can independently support archival flow requests, and a user interface, OAIS web[6]. The backend exposes multiple HTTP endpoints using Python and the Django Framework, Celery and Redis for asynchronous execution, and PostgreSQL to store information about triggered archival processes. The UI is developed in React with semantic-ui. During the Summer Student Program, my tasks involved working on both components.

3 Push To Tape

The Preserve Platform utilizes key CERN services to store AIPs long-term - from now on we will refer to this process as pushing to tape. After the ingestion process, a newly created AIP is stored in the service's EOS space, and consequently will be pushed to the service's CTA space on request. To transfer data to or from the CERN's tape the FTS service is used by way of a Python3 module. While the main part of the Push To Tape operation was already implemented, I had to investigate if the workflow is still functional, given the updates of the services mentioned above, and to identify possible solutions. After reviewing the behaviour I concluded that there was a problem with authentication of the service when conducting file transfers despite the fact that the correct certificate was provided.

3.1 Certificate Authentication

The FTS service uses X509 Certificate authentication to check if a user or a service has permissions to push or retrieve data from CERN's storage spaces. Thus, in order to execute file transfers the FTS client should be initialized with the service's Robot Grid Certificate and additionally it is necessary to inquire the mapping of the certificate to the OAIS service so that it can access the EOS and CTA designated paths.

To set up and obtain the user certificate and encrypted private key the following steps are executed:

1. Inquiry EOS and CTA services to map the **DN** to the service account
2. Access **CERN Certification Authority** and request a **Robot Grid Certificate**
3. Download the related **.p12** file and extract the certificate and key:

```
# Get the public part
openssl pkcs12 -in myCertificate.p12 -clcerts -nokeys -out usercert.pem
# Get the private one. A passphrase is required.
openssl pkcs12 -in myCertificate.p12 -nocerts -out ./userkey.pem
# Remove the passphrase from the private part
openssl rsa -in userkey.pem -out private.nopwd.key
```

In addition to providing the certificate, in order for the service to authenticate to the FTS instance and transfer files, a trusted entity issued by a CA (Certificate Authority) is needed to sign the OAIS service's certificate. Therefore, for certificate validation, I added a script in the platform's Dockerfile that installs a CERN Root and Grid Certificate downloaded from the EGI repository[7].

4 Archival Pipeline

The Preserve Platform supports the execution of different operations such as creation of an AIP (Preserve), extraction of metadata (Extract Title), pushing the archive's information to the Digital Memory registry (Push to Registry) or preserving AIPs on tape (Push to Tape); after harvesting records these processes are triggered individually on request for a single archive or for batches. Accordingly, carrying out operations on archives requires constant manual input from users. Thus, to enable automation I focused on developing the creation of pipelines for archival processing. This functionality required full-stack development, by implementing two new endpoints and creating a pipeline creation display, so that users could easily create and trigger chains of steps.

4.1 Pipeline Logic

While designing the pipeline solution we considered multiple use cases:

- Creation and execution of custom pipelines
- Addition of new steps to a running pipeline
- Retrial of failed steps in the pipeline
- Continuation of a pipeline after a failed step
- Creation and execution of periodical pipelines

The implemented solution supports the first four use cases but can be easily extended to contain the last functionality by creating a new periodic task that calls already implemented functions. Two endpoints were established; one for creating and executing a custom pipeline 1 and another for starting or retrying the execution of an existent step in the chain 2.

```
POST /archives/{archive_id}/pipeline/
Parameters:
- archive (dict): Target archive
- pipeline_steps (list): List of steps to add to the archive's pipeline
```

Listing 1: Pipeline creation endpoint

```
POST /archives/{archive_id}/run-step/
Parameters:
- archive (dict): Target archive
- step_id (number): ID of the step to be executed
```

Listing 2: Run individual step endpoint

4.2 Pipeline Implementation

To address the requirements mentioned above, especially the possibility of adding more steps to a pipeline while it is executed, we developed the solution based on the producer-consumer paradigm. As such, I added a new field to the archives table in the DB, called `pipeline_steps`, which contains the IDs of steps to be run and it is handled as a queue; each archive stores the current state of its pipeline. In this scenario the consumption of the FIFO buffer happens when the pipeline is being executed, while the production of new steps occurs only on user request.

For the execution of each step a Celery task is spawned and executed asynchronously. To obtain the chain logic, the queue of steps is consumed after the task was successfully completed. Thus, after a step is completed the first available one in the archive's pipeline is processed. Ultimately, when the user requests the creation and execution of a pipeline, the given steps are created, and their IDs are queued in the mentioned buffer. Subsequently, if there is already a running pipeline, the given steps are only queued, while if there is no ongoing execution, the target steps will also start to be processed one by one.

It is to be mentioned that because of the usage of Celery tasks, two processes can try to write or consume the pipeline's content. Therefore, to assure DB consistency and synchronize access, atomic operations supplied by Django were used, as well as the `select_for_update()` method, which locks a row in the DB, consequently the FIFO buffer of an archive.

Besides using the API call, users can trigger the execution of an archive's pipeline through the Preserve Platform UI. To facilitate this, I focused on adding the option of creating pipelines and also enhancing the display of the chain of steps in the archive's details page. To add the option of creating a pipeline to the action menu, I refactored the component used for the pipeline display and reused it to allow selection of steps in a pop-up modal.

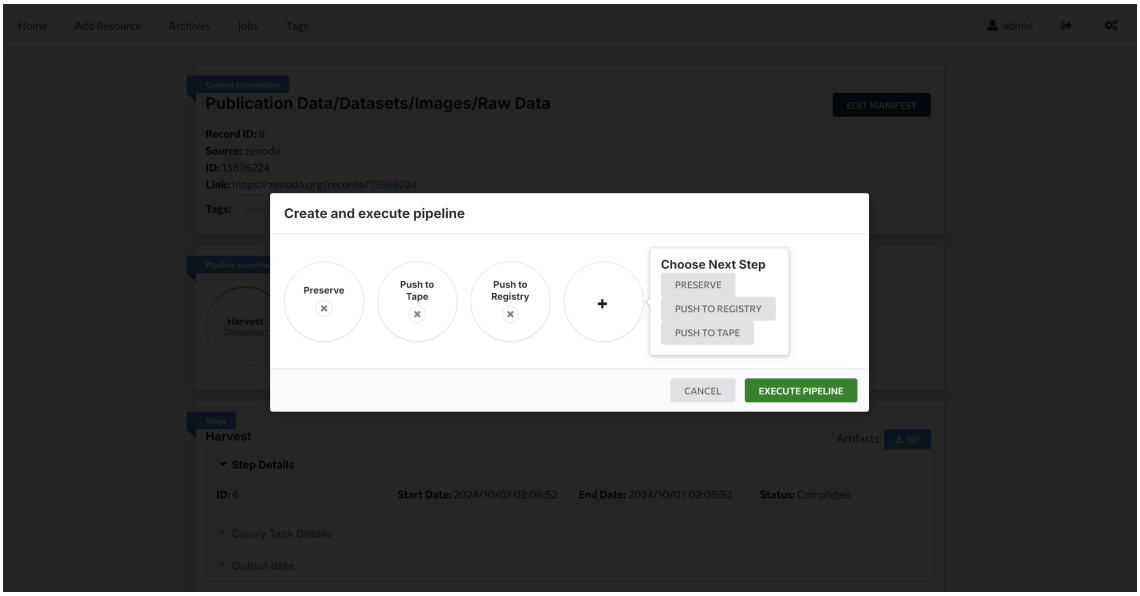


Figure 2: Pipeline creation modal

In case of failure, a step can be rerun on the UI, and if it is successful, then the pipeline continues. In addition, the pipeline can also be continued if a step failed, by executing the next possible step (not previously run) in the pipeline.

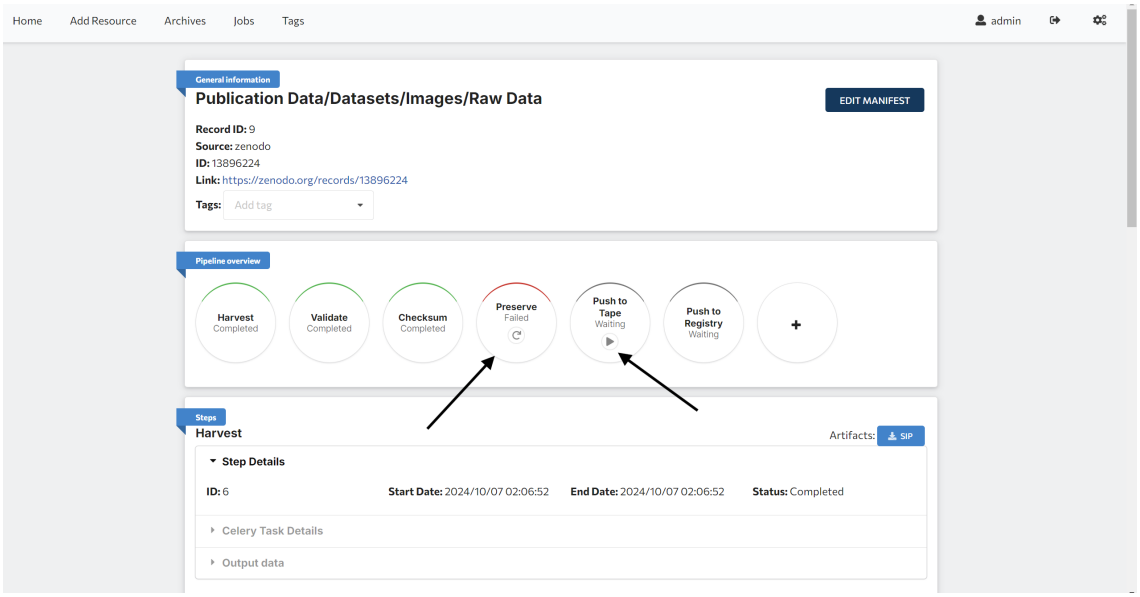


Figure 3: Pipeline display

5 Future Work

One upcoming step in developing the Preserve Platform is integrating a new CDS archival flow. The objective is to automatically harvest records, a functionality that can be enabled by scheduling periodical pipelines (mentioned in the Archival Pipeline chapter). In addition, a planned feature is to notify the source information system of the archival state of the record, more precisely, if the preservation of the record was successful. This capability entails adding a new pipeline step, Notify Source, which enables the asynchronous push of artifacts related to the record's archival state. Thus, this functionality will be available for CDS and all repositories built on Invenio.

Another future direction is to provide DRO (Departmental Record Officer) approvals for executing specific steps on the archive. For example, if a regular user creates a pipeline and the step needs approval, it will not continue executing until the action is permitted by the superuser.

Acknowledgments

I would like to thank Panna and Jean-Yves for the opportunity to work at CERN and be part of the Digital Memory project through the Summer Student program. This experience allowed me to gain valuable knowledge and learn new technologies, which enhanced my skill set and helped me understand the relevance of digital preservation. I am truly grateful for your assistance and encouragement, and I am thankful for the time and effort you invested in helping me grow.

References

[1] J.-Y. Meur and N. Tarocco, "The obsolescence of information and information systems cern digital memory project," *EPJ Web of Conferences*, vol. 214, p. 09003, 01 2019.

[2] J.-Y. L. Meur, "The digital memory of cern: Status and input for a preservation strategic plan," 2018.

[3] Consultative Committee for Space Data Systems, "Reference model for an open archival information system (OAIS)," April 2019. Recommended Practice.

[4] C. Lee, "Open archival information system (OAIS) reference model," December 2011.

[5] oais-platform repository. `https://gitlab.cern.ch/digitalmemory/oais-platform`. Accessed: 2024-09-27.

[6] oais-web repository. `https://gitlab.cern.ch/digitalmemory/oais-web`. Accessed: 2024-09-27.

[7] EGI repository. `https://repository.egi.eu/sw/production/cas/1/current/RPMS`. Accessed: 2024-08-27.