heiARCHIVE, a long-term preservation service at Heidelberg University

Martin Baumann[1,2], Florian Heß[1,3], Leonhard Maylein[1,3], Tatjana Mechler[1,2], Benjamin Scherbaum[1,2] and Eric Volkmann[2]

[1]Competence Centre for Research Data, Heidelberg University
[2]University Computing Centre, Heidelberg University
[3]University Library, Heidelberg University

heiARCHIVE is a new institutional service for long-term data preservation at Heidelberg University. It offers researchers an easy-to-use end-user platform for archival of their research data as well as the possibility to do a OAIS compatible long-term preservation containing features like format recognition, validation and conversion of files of appropriate file formats. heiARCHIVE is developed and will be operated by the Competence Center Research Data - a joint service facility of the University Computing Center and Heidelberg University Library. This work outlines the concept of the service and its current status.

1 Introduction

heiARCHIVE[1] is an upcoming institutional service for long-term data preservation offering researchers an easy-to-use end-user platform for archival of their research data. It is a dark archive that is based on the OAIS reference model, cf. [1]. heiARCHIVE is based on an in-house software development that offers features like format recognition/validation and extraction of metadata from files. A storage abstraction is realized based on the open source data management software iRODS[2] to manage data copies and geo-replication and the BagIt file packaging format (RFC 8493[3]) is used for structuring and naming directories and files. A dedicated right and role concept including billing management is available. Through service-local identity management, also alumni can use the service and users will prospectively also be able to do authentification using their ORCID[4].

heiARCHIVE is a service developed and maintained by the Competence Centre for Research Data[5] (KFD), a joint facility of the university's Computing Centre and the Uni-

Das hier beschriebene Poster ist in der Open Access-Plattform der Universität Heidelberg heiDOK unter der DOI: `https://doi.org/10.11588/heidok.00029723` veröffentlicht.

[1] `https://heiarchive.uni-heidelberg.de/`

[2] `https://irods.org`

[3] `https://tools.ietf.org/html/rfc8493`

[4] `https://orcid.org`

[5] `https://data.uni-heidelberg.de`

Publiziert in: Vincent Heuveline, Nina Bisheh (Hg.): E-Science-Tage 2021.
Share Your Research Data. Heidelberg: heiBOOKS, 2022.
DOI: https://doi.org/10.11588/heibooks.979.c13740 (CC BY-SA 4.0)

versity Library. In accordance with Heidelberg University's Research Data Policy it is the mission of KFD to provide the best possible support for the comprehensive and coherent management of research data for the university and its researchers. Amongst others, KFD offers also an institutional repository for open research data heiDATA[6] that is based on the open source web application Dataverse[7]. This service is used for data publication which is preferred over pure archiving wherever allowed and useful.

The software behind the heiARCHIVE service was started to be developed in 2017 since no available software could be found that would fulfill all specific requirements. On the one hand, an archiving solution for researchers to preserve their research data was needed, i.e. the software must be usable for end-users and allow for potentially very large capacity. On the other hand, elaborate long-term preservation processes for data of cultural heritage must be feasible. The long-term preservation of data must be included in existing processes of digitalization, data presentation or publication which requires certain levels or automatization and also of continuous adjustment between the primary data location and the archive. Additionally, a billing system must be integrated. Based on these requirements and expecting furthermore to come in the future, we decided to start a software development project to be flexible and put the focus on the features that we have the strongest demands.

2 Modular design and implementation

The modular design of heiARCHIVE follows the OAIS concept in implementing the data flow in terms of SIP, AIP and DIP, see Fig. 1. The data flow is managed by process steps that run sequentially and correspond to one module each: the inbox (prepare the data), ingest (package the data), storage (securely store the data) and access (to access the data). These modules are conceptually separated, can run on different (virtual) servers and interact only through a dedicated API. The heiARCHIVE admin module controls the overall process and also the state of all archive packages. Scaling compute resources and network bandwidth is possible by adding additional (virtual) servers, and can be helpful e.g. for intensive checksum operations. Today, data transfers from/to heiARCHIVE are realized via SFTP, but further protocols are intended. There is a graphical web interface for user interaction (GUI), both for end users and for maintainers. The full software stack of heiARCHIVE is based on Python 3. For the GUI, the scheduling and API endpoints are realized using the high-level Python Web framework Django[8].

Within the heiARCHIVE admin module, an indexer based on Solr[9] is integrated. Its main task is to make certain element contents of the metadata of all archived data retrievable. Today, this feature can be used by heiARCHIVE admins only, but might be activated for end-users in the future.

[6] https://heidata.uni-heidelberg.de/

[7] https://dataverse.org/

[8] https://www.djangoproject.com/

[9] https://solr.apache.org/

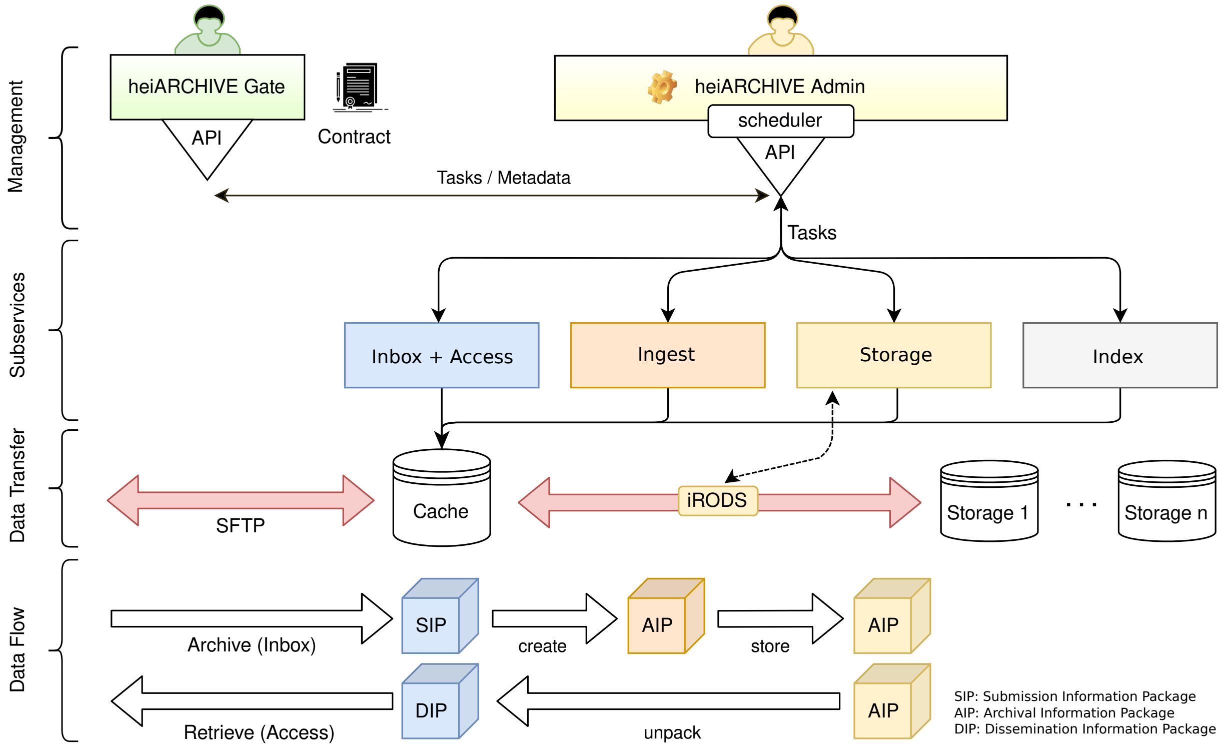


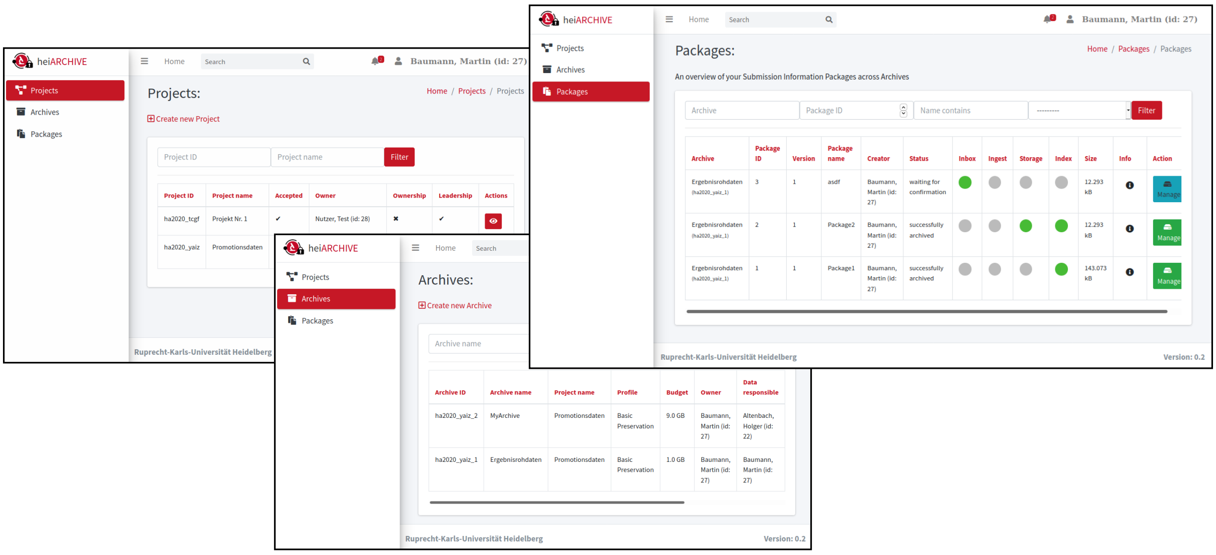
Figure 1: Illustration of the data flow, data management and modules that correspond to subservices of heiARCHIVE.

The indexer reads and validates the metadata files generated during the ingest process. In order to feed Solr with items in the expected format, XSLT transformations are used. Additionally, a selection of admin database items is indexed for faster retrieval compared to SQL-based queries.

3 Archive- and role management

heiARCHIVE has a hierarchical management structure and a related right and role concept for the users. The main navigation of the web interface reflects this management structure, see Fig. 2. The highest-level structure is denoted by “project” that can only be created by entitled persons (e.g. professors). A project establishes the link to a cost center (“Kostenstelle”) and can also define a financial quota.

A project can contain one or multiple “archives” each of which is related to a data responsible person (e.g. a PhD student). The archive involves a set of archival parameters, e.g. the archive mode that is set to be either “long-term archiving” to include format validation or to be “bitstream-preservation”. Descriptive metadata such as title, short description, project context, etc. is set at this level as well.



Projects:

Home / Projects / Projects

⊞ Create new Project

Project ID	Project name	Accepted	Owner	Ownership	Leadership	Actions
ha2020_tcgf	Projekt Nr. 1	✓	Nutzer, Test (id: 28)	✖	✓	👁
ha2020_yaiz	Promotionsdaten					

Archives:

⊞ Create new Archive

Archive ID	Archive name	Project name	Profile	Budget	Owner	Data responsible
ha2020_yaiz_2	MyArchive	Promotionsdaten	Basic Preservation	9.0 GB	Baumann, Martin (id: 27)	Altenbach, Holger (id: 22)
ha2020_yaiz_1	Ergebnisrohdaten	Promotionsdaten	Basic Preservation	1.0 GB	Baumann, Martin (id: 27)	Baumann, Martin (id: 27)

Packages:

Home / Packages / Packages

An overview of your Submission Information Packages across Archives

Archive	Package ID	Version	Package name	Creator	Status	Inbox	Ingest	Storage	Index	Size	Info	Action
Ergebnisrohdaten (ha2020_yaiz_1)	3	1	asdf	Baumann, Martin (id: 27)	waiting for confirmation	●	○	○	○	12.293 kB	ℹ	Manage
Ergebnisrohdaten (ha2020_yaiz_1)	2	1	Package2	Baumann, Martin (id: 27)	successfully archived	○	○	●	●	12.293 kB	ℹ	Manage
Ergebnisrohdaten (ha2020_yaiz_1)	1	1	Package1	Baumann, Martin (id: 27)	successfully archived	○	○	○	●	143.073 kB	ℹ	Manage

Figure 2: Exemplary views of heiARCHIVE's GUI: Users have an overview of their projects, archives and archive packages including the states of the archive packages within the archiving pipeline.

The archive is the container for one or multiple "archive packages" – the structure which finally contains the data to be archived. The data responsible person has the permission to upload data into an archive package, add metadata and start the packaging and storage process.

4 Metadata

For subsequent use of the preserved data and also for a description of the preservation process, a minimal set of mandatory metadata is stored in the heiARCHIVE database and the index, but also within the AIPs. Some metadata is demanded from the user, e.g. the creator of the data and additional descriptive information. Other metadata can be determined from the user's data to be archived (denoted by *payload*) itself. More extensive descriptive metadata may be stored in a pre-defined location within the data package that might be considered in the future by the indexer.

The metadata procedure during the ingest process is sketched in Fig. 3. Metadata is collected from the user via the GUI or via the API and is stored in the file"heiarchive-metadata.xml" complying a custom schema. Using a custom schema for this intermediate format has practical reasons, since it is easy to use and we may change at our sole discretion to meet new institutional conditions. During the ingest workflow, the data of this file is read into an SQLite database which is also the reporting target of several tools that analyze the structure and content of the payload. After modifications to certain items in the database, e.g. placeholder replacements and reference resolutions, all items are processed and put into place to forge piece by piece the final XML file which can be

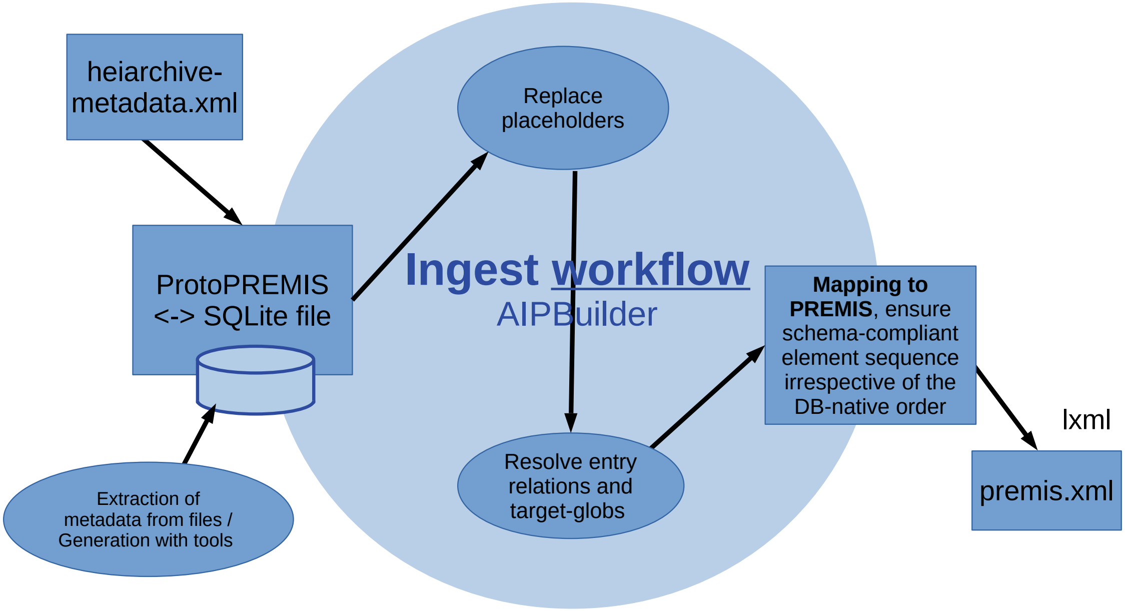

heiarchive-metadata.xml
Replace placeholders
ProtoPREMIS <-> SQLite file
Ingest workflow
AIPBuilder
Mapping to PREMIS, ensure schema-compliant element sequence irrespective of the DB-native order
lxml
premis.xml
Resolve entry relations and target-globs
Extraction of metadata from files / Generation with tools

Figure 3: Process of metadata collection, its metadata processing towards a standardized schema and finally the creation of a standard conforming metadata file.

written in one single pass using the Python library lxml[10]. The SQLite database is used as a cache and collection facility for easy enrichment of the data without need to read in and write out large portions of XML by separate tools in rather resource-inefficient ways. The database file is stored within the AIP besides the“heiarchive-metadata.xml” to be on the safe side when detecting and resolving issues.

The XML file resulting at the end of the process complies different standards. The METS standard[11] defines a container for descriptive, administrative, and structural metadata. The PREMIS standard[12] defines the metadata for the preservation of the data objects and their long-term usability. And, most likely, DataCite[13] will be used to represent the descriptive metadata in the future (currently in investigation). DataCite is considered to be a suitable schema for descriptive metadata and - due to its wide distribution - facilitates inter-operation with other archive or repository services, also at University Library.

5 Status and next steps

The main features of the software behind heiARCHIVE are implemented, the submission process is running and the dissemination is technically prepared. Extensive testing has been done to ensure the GUI and backend functions are working reliably. Currently, the

[10] https://lxml.de/
[11] https://www.loc.gov/standards/mets/
[12] https://www.loc.gov/standards/premis/
[13] https://schema.datacite.org/

metadata model and the author's contract are not finally defined and the access of the available tape library via iRODS is under investigation. Geo-replication has not been realized yet but is in preparation. Although heiARCHIVE is a dark archive, a publicly accessible registry is planned that contains an excerpt of the metadata of the archived data together with a persistent identifier.

In the next few months, these tasks will be tackled and the service will afterwards be started step by step towards productive operation: First, selected researchers will be invited to do an archiving for research data that has no challenging requirements (e.g. not very large capacity). Then, the service will be opened for general use for Heidelberg University researchers. At the same time, a connection between services at University Library to heiARCHIVE will be realized via the API for an automated long-term preservation of data of these services.

Acknowledgements

The development of heiARCHIVE has been funded in parts by the ministry of science, research and arts of the state of Baden-Württemberg, Germany.

Bibliography

[1] CCSDS - Consultative Committee for Space Data Systems. "Reference Model for an Open Archival Information System (OAIS), Recommended Practice." Recommendation for Space Data System Practices, CCSDS 650.0-M-2, Magenta Book, June 2012.