DA-NRW: a distributed architecture for long-term preservation

Manfred Thaller manfred.thaller@uni-koeln.de, Sebastian Cuy sebastian.cuy@uni-koeln.de, Jens Peters jens.peters@uni-koeln.de, Daniel de Oliveira d.de-oliveira@uni-koeln.de, and Martin Fischer martin.fischer@uni-koeln.de

Historisch-Kulturwissenschaftliche Informationsverarbeitung, Universität zu Köln, Albertus-Magnus-Platz, D-50923 Köln

Abstract. The government of the German state of North-Rhine Westphalia is considering the creation of a state-wide long-term repository for digital content from the cultural heritage domain, which at the same time will act as a pre-aggregator for the *Deutsche Digitale Bibliothek* and the Europeana. The following describes a software architecture that relies exclusively on existing open source software components to implement a distributed, self-validating repository, which also supports the notion of "executable contracts", allowing depositors a high degree of control over the methods applied to individual objects submitted for preservation and distribution.

1 Introduction: Background

North-Rhine Westphalia, as well as other political entities responsible for the cultural heritage in the public domain, faces the problem that - as of now - few, if any, workable solutions for the preservation of digital content exist. That is true for digital content created by projects within the field of retrospective digitization of cultural heritage, and it is even more true when we look at the safe-keeping of digital content created by public administration or arriving in the public domain through deposition in one of the deposit libraries of the state.

At the same time North-Rhine Westphalia is expected to support the creation of the Europeana, as one of many entities. As Germany has decided to channel its contribution to the Europeana through an intermediate layer, the *Deutsche Digitale Bibliothek*, the original metadata schemas of the content holding institutions have to be converted for both target systems. At the same time, few, if any, memory institutions would be willing to submit the very top quality of their digital holdings to a European (or any other) portal that allows the completely unrestricted use of that material. It is, therefore, necessary to convert the data submitted by the memory institutions to a form that can be distributed completely without restriction.

It was decided to attempt an integrated solution for both problems: a framework is to be developed under the name of *Digitales Archiv Nordrhein-Westfalen (Digital Archive North-Rhine Westphalia)* (DA-NRW) which would allow all memory institutions (archives, museums, libraries) of the state to submit their digital content to a state wide repository, which would follow the OAIS model [2] and specifically:

- Ingest the material into a long-term repository system, which allows for a technology watch, triggering migration if necessary, and other active methods.
- Perform automatic verification of the redundantly stored material between geographically distributed sub-repositories.
- Evaluate user-submitted contracts expressed in XML, describing in detail which of several options for storage as well as distribution to the public are to be provided for that object.
- Derive suitable representations of the administered material, keep them on a server which supports OAI-PMH (cf. [5]) and other protocols to make these representations available to various cultural heritage portals.

The system is to be based upon existing infrastructural institutions from different sectors: the *Hochschulbibliothekszentrum*, the Computing Center of the *Landschaftsverband Rheinland* and the Computing Center of the *Universität zu Köln*. The chair for Humanities Computer Science at the *Universität zu Köln* is responsible for design and implementation of a prototype.

In order to avoid performance and cost problems during the transfer from prototype to production system, and to create a scalable prototype in less than 18 months, the following decisions have been made:

- The system is built according to agile software development rules.
- Only Open Source components are being used.
- The prototype is expected to perform with 200 TB, being scalable without re-design by one order of magnitude, to 2 PB.

At the end of June 2011, after an initial preparatory phase and four months into the core development time of 14 months, a functionally complete pre-prototype is available.

2 Introduction: Overall Architecture

The three participating computing centers, referred to as the nodes of the DA-NRW, are to be understood as independent nodes of a network. The flow of data within each node is directed by an instance of a *content broker*, directing the flow of data from ingest into the archive, on the one hand, and that of derived copies of the data into a presentation area, on the other hand, where these data can be accessed by appropriate harvesters. For a diagram of the component structure of these *content brokers* see figure 1. The individual components will be described in the following sections of this paper.

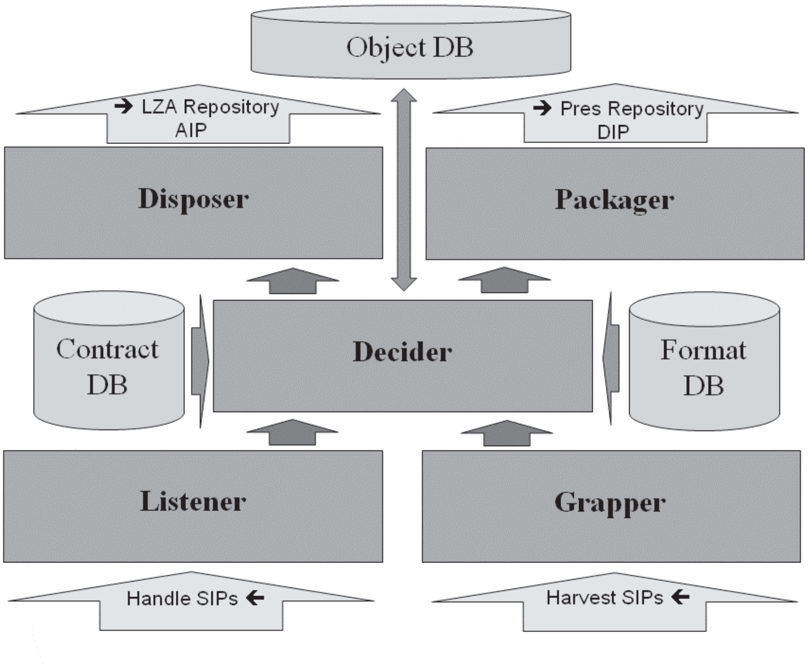

Fig. 1. Component structure of the *content broker*

The individual nodes are bound together by a synchronizer and deliver their data into a presentation component, which is separated from the actual long term preservation components by appropriate firewall techniques.

3 Ingestion methods

One key feature of systems providing long-term preservation is the delivery of digital objects from an institution to a preservation node. In our system this can be accomplished in two different ways.

The first one allows contractors to build *Submission Information Packages* (SIPs) [2] on their own. In this case, however, the structure of the SIPs has to be valid prior to ingestion into the archive. That means the SIPs have to contain structural metadata in a format supported by DA-NRW (e.g. METS[1]). If contractors decide to build their own SIPs, they are also responsible for creating checksums for the package contents in order for the *content broker* to be able to check for consistency.

The second possibility of building valid SIPs is to use the *DA-NRW SIP Builder*. This tool enables users to create SIPs in a very simple manner. In order

[1] cf. `http://www.loc.gov/standards/mets/`

to make the tool available for a wide audience, the *SIP Builder* is written in Java and therefore constitutes a platform-independent application. It provides a graphical user interface for comfortable usage. After choosing a destination path where the SIPs will be created, the user chooses which strategy to use for compiling the SIPs. On the one hand, one can choose a metadata XML file which describes the package structure. The tool then collects the files referenced in the XML. On the other hand, the tool is able to compile valid SIPs from directories taking into account folder structure and contained metadata files.

Another important aspect of the *SIP Builder* is the possibility of declaring contracts in a user-friendly way. Statements generated by the *SIP Builder* are serialized as a machine-readable contract in a PREMIS-based XML (see [1]) format that can subsequently be evaluated by the *content broker*.

4 Content broker

The central part of the architecture is called the *content broker*, a tool written in Java. This component is responsible for manipulating complete information packages in various ways. It does so by executing predefined chains which correspond to use cases such as ingest, retrieval or the migration of information packages. Each chain consists of atomic actions which in turn operate on the aforementioned information packages. Examples for actions are: the self-explanatory 'FormatConversionAction' that converts bitstreams and/or metadata into target formats, or the 'RegisterObjectAction' that registers an information package at the global object database. Administrators can define different chains for different tasks. Chains can be configured in an easily readable XML syntax.

Format conversion and identification are also implemented in a highly flexible manner in the overall design. As far as format identification is concerned, 3rd party software (such as DROID[2], JHOVE[3] or the Linux FILE-Command) can easily be plugged into the the workflow. Format conversion policies can also be configured from a set of XML files.

Migration happens along the same lines. Policies and corresponding conversion routines have to be defined in order to automatically retrieve and convert packages which are marked as containing deprecated formats. At this stage two aspects have to be stressed: first of all, there is the problem of 'marking' formats as deprecated. At present this is done manually, but for the future we plan to use an automatic approach by connecting the system to an automated obsolescence notification system, as currently discussed within some preservation infrastructure projects.

The second aspect refers to the selection of appropriate conversion routines. Here an administrator of a node, or an administrator of the whole DA-NRW system, is requested to choose which conversion routine delivers the best results in terms of quality for long-term preservation. That means it first has to be chosen which target format serves as a long-term preservation format. Once the

[2] cf. `http://droid.sourceforge.net`
[3] cf. `http://hul.harvard.edu/jhove`

format is chosen, the next decision will be which program to use with which parameters to achieve good results.

5 Presentation repository

The architecture of DA-NRW also includes a *presentation repository* that acts as a central service provider for different partnering institutions and interdisciplinary portals – such as Europeana, the *Deutsche Digitale Bibliothek* and the North Rhine-Westphalian portal developed at the HBZ during the course of this project. Also the *presentation repository* can serve as a data source for subject-specific repositories aggregating specialized collections. Finally, small institutional repositories can harvest the central repository in order to implement own applications for the presentation of their data. While doing this they can profit from the format conversions and normalizations that the packages undergo on their way through the services of the digital archive as a whole.

Contractors of the DA-NRW can define if and under which conditions an object will be available through the *presentation repository*. These conditions include restrictions on the quality of the presented material, such as resolution and bit rate, restrictions on the content, e.g. by allowing only specific parts of the data and metadata to be retrieved, as well as time-based statements in order to be able to represent "moving-walls" or the expiration of copyright.

Currently the *presentation repository* is based upon the Fedora Commons Repository Software and supports the dissemination of Dublin Core (DC) and Europeana Semantic Elements (ESE) [4] metadata for every object in the repository. These standards represent a common basis for the heterogeneous objects we have to deal with. However, we are planning to support richer metadata formats in the presentation of objects and are examining ways to make the data available as part of the ongoing efforts to support Open Linked Data.

6 Storage layer

Our basic approach in long-term preservation regarding storage is to synchronize the stored information across at least three different storage locations, technically and geographically independent, across the state of North Rhine-Westphalia. To accomplish this major goal, we decided to use the iRODS (Integrated Rule-Oriented Data System) Data Grid Software [7].

In order to test our system under realistic conditions and with real data at a relatively early stage of development, we chose an iterative approach for the design and realization our project. In terms of iRODS, we implemented the basic features related to the data storage part corresponding to the final stage of the archival workflow after the *content broker* actions have already taken place. So we first focused primarily on the storage capabilities of iRODS. In the upcoming iteration we plan to use iRODS, in particular its "Rule Engine" and its "Microservices", more intensively in the entire workflow of the archival storage process as well as the ongoing data curation process in the years to come.

The consistency of each digital object will be ensured by physical checksum comparisons and by keeping the minimum number of replicas of each object on the desired nodes after AIPs (Archival Information Packages) being "put" to the node. These use cases will be implemented using "Rules", statements executed on the data by the Rule Engine.

7 Future research

As mentioned in the introduction, the architecture described here is a pre-prototype version which was developed within four months. A "pre-prototype" means that all major components exist and can be used. However, quite a few details are missing (e.g. the notification of an end user about the results of the ingest process). Furthermore, it means that major processes, which shall run automatically at the end of development, have to be started explicitly at this point.

In the near future we plan to replace a large part of the orchestration of individual services, which has now been rapidly prototyped in Java, by a stronger reliance on iRODS Micro-services. In other words: we plan to shift from just storing data in the iRODS-Grid at the final stage of our existing archival workflow-chain to a more iRODS-centric architecture by making the features of "Rules" and "Micro-services" do the major work. This will also ensure computational scalability. The leading design principle in our already developed components was to develop fine-grained actions which are only loosely coupled. These actions can now easily be replaced by or be incorporated into iRODS Micro-services. A lot of research has to be done on the second question, i.e. how Rules can help us build up "policies" for archived content itself. A major part of our work in the next months will be the usage of iRODS Rules to execute policies on our stored objects.

We are currently also evaluating the use of PREMIS OWL [3] and triple stores for the representation of contracts in RDF (Resource Description Framework). This allows for easier extension of the contract format, reduces the mapping overhead between the XML format and the relational database, and simplifies the organization of machine-processable contracts. We are also investigating different RDF-based variants for wrapping package metadata. One approach might for example be the application of OAI-ORE as an alternative for METS as proposed in [6]. This would allow us to incorporate contract, format, structural and descriptive metadata into one unifying RDF model.

References

1. Brandt, O.: PREMIS. Nestor-Handbuch: eine kleine Enzyklopädie der digitalen Langzeitarchivierung pp. 60–62 (2007)
2. CCSDS - Consultative Committee for Space Data Systems: Reference Model for an Open Archival Information System. Blue book. Issue 1 (January 2002)

3. Coppens, Mannens, Evens, Hauttekeete, Van de Walle: Digital long-term preservation using a layered semantic metadata schema of premis 2.0. In: Cultural Hertiage on line. International Conference Florence, 15th-16th December 2009 (2009)
4. Europeana Foundation: Europeana Semantic Elements Specification, version 3.4 edn. (March 2011)
5. Lagoze, C., Van de Sompel, H., Nelson, M., Warner, S.: Open archives initiative - protocol for metadata harvesting - v.2.0 (Jun 2002)
6. McDonough, J.: Aligning mets with the oai-ore data model. In: Heath, F., Rice-Lively, M.L., Furuta, R. (eds.) JCDL. pp. 323–330. ACM (2009)
7. Rajasekar, Wan, Moore, Schroeder: iRODS Primer. Morgan Claypool Publishers (2010)