Proper Language Resource Centers

Willem Elbers, Daan Broeder and Dieter van Uytvanck

Max Planck Institute for Psycholinguistics
Wundtlaan 1, 6525 XD, Nijmegen, The Netherlands
Email: willem.elbers@mpi.nl, daan.broeder@mpi.nl, dieter.vanuytvanck@mpi.nl

Abstract

Language resource centers allow researchers to reliably deposit their structured data together with associated meta data and run services operating on this deposited data. We are looking into possibilities to create long-term persistency of both the deposited data and the services operating on this data. Challenges, both technical and non-technical, that need to be solved are the need to replicate more than just the data, proper identification of the digital objects in a distributed environment by making use of persistent identifiers and the set-up of a proper authentication and authorization domain including the management of the authorization information on the digital objects. We acknowledge the investment that most language resource centers have made in their current infrastructure. Therefore one of the most important requirements is the loose coupling with existing infrastructures without the need to make many changes. This shift from a single language resource center into a federated environment of many language resource centers is discussed in the context of a real world center: The Language Archive supported by the Max Planck Institute for Psycholinguistics.

Keywords: federated, language-resource, centers

1. Introduction

A language resource center enables researchers to deposit their work, enriched with structural and descriptive meta data, into a reliable repository. The language resource center ensures the persistence, identifiability and publication of the deposited data. The language resource center where the data was stored first, the repository of record, controls the replication process and access rights, shown schematically in figure 1.

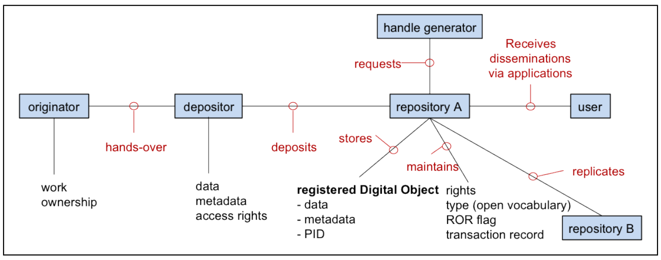

Figure 1: Language Resource Center

Single centers should have proper backup measures in place, however this does not provide a real long term persistency solution. In order to move towards a proper long term persistency solution the language resource center need to fullfill specific technical and non technical requirements and both the data and the services need to be replicated to multiple geographically different centers. Ideally all these replicas are also accessible by users trying to retrieve the data.

In the remainder of this article we will discuss the requirements of a proper language resource center in the context of the CLARIN (Vradi et al., 2008) project, expand these requirements to a federated LR center infrastructure, describe The Language Archive, maintained by the Max Planck Institute for Psycholinguistics, as a real world example of a proper language resource center and explain the steps we are taking and the challenges we encounter to move towards such a federated infrastructure.

2. CLARIN Centers

Within the CLARIN project, that aims to create a single domain of Language Resources (LR) and Language Technology (LT) for SSH researchers, language resource centers are considered as the infrastructure backbone. In this context proper language resource centers are of course subject to a number of requirements where the non-technical organizational issues are:

- The need to anchor the participation within the CLARIN infrastructure in its long term planning and not regard this as a short-term project.
- There is the need for the centers to act as a backup for other centers should one of these seize to function and so allow for a transfer of infrastructure services.
- The need to have a proper certificated repository or archiving system as RAC[1] or DSA[2], what in the last instance means that the center needs to be transparent about its archiving policies, so that data users and depositors know what they can expect. This transparency is a requirement for all the (CLARIN) services.

Technical requirements are manifold, ranging from the use of persistent identifiers (PIDs) for resources, the used data formats that should be open to ensure readability in the future to the use of component meta data (CMDI) (Broeder et al., 2010). How the centers should choose to fulfill these obligations is a matter of choice, they might want to outsource the storage of data to a friendly University

[1]http://cwe.ccsds.org/moims/default.aspx
[2]http://www.datasealofapproval.org/

data-center or buy storage from a cloud solution. The centers carry the responsibility to the users.

However some of the technology requirements do have bearings on the cooperation with other centers or the use of general infrastructure services from other providers. In particular there are the requirements for using a specific type of Authentication and Authorization Infrastructure (AAI).

3. Federated Language Resource Centers

Language resource centers are expected to offer long-term persistency of both the data and the services they provide. The CLARIN project currently does not impose any requirement, except to take over infrastructure services when a center seizes to function, with respect to long-term persistency on participating centers. Language resource centers in a federated infrastructure can synchronize data between each other to provide long-term persistency and redundancy while at the same time synchronize auxiliary data required by the services to provide long-term persistency of the services. Different centers use different repository and meta data infrastructures. We assume it is unrealistic to expect all participating centers to change their infrastructure. This emphasizes the need for a loose connection to the federation ensuring that centers can keep using their existing infrastructure.

Federated AAI is another challenge in a federated language resource centers infrastructure. SAML 2.0 federated identity (Ragouzis et al., 2008) provides a single sign-on solution where user authentication is performed by the user's home organization and user authorization is performed by the service the user tries to access. The authorization and authentication can therefore take place in different organizations and by signing in once, the user can access all connected services.

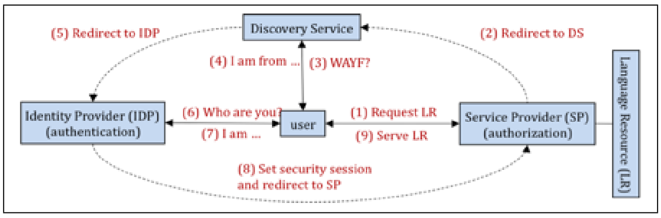

Figure 2: Shibboleth Single Sign On

The SAML 2.0 federated identity single sign on flow, see figure 2, typically goes as follows: (1) a user requests access to a resource protected by a service provider (SP), (2) the user is not yet authenticated, therefore the SP redirects the user to a discovery service [3] (DS), (3) The DS ask what the users home organization is, (4,5) based on the users reply the user is redirected to the identity provider of his/her choice, (6) the IDP asks the user to identify himself/herself, (7,8) if the user is authenticated the IDP creates a security session and redirects the user back to the SP, and finally (9) the SP has a valid security context and will grant access to the LR (if the user is authorized to access this resource). Naturally, if a security session is already available at step 1, step 9 is performed and all intermediate steps are skipped.

Each service is responsible to authorize the user; there is no need to share authorization information, however if the same service is replicated to multiple repositories, this changes. The information required to authorize a user needs to be replicated with the services in order to keep the authorization process synchronized over multiple repositories, making it transparent to the user which repository is used.

Another important aspect for language resource centers is the identification of digital objects. Persistent identifiers are used to identify the digital objects (DOs) and keep track of the specific instances of these objects, the data. If an instance of a DO is moved, the persistent identifier should be updated as well. Instances of digital objects can be identified by the use of URIs. URIs can offer some level of persistency already but they are always based on a hostname. It has turned out that hostnames might change, which causes difficulties in keeping the URIs persistent. Persistent identifiers solve this problem, since the reference to the instance of the DO can be updated. The persistent identifier system is required to supply a resolver that can resolve the identifier to an associated URI.

In a federated infrastructure the persistent identifier has to support multiple URIs for a single DO as shown in table 1. The DO instances can be present in multiple data centers and the persistent identifier has to facilitate support for multiple URIs. The same holds for the resolver that must supply support to resolve to any of the URIs. Besides the location a PID preferably is also able to contain other information such as checksum information. This can be used to assess the validity of all the copies across language resource centers.

How the specific URI is chosen is a matter of choice. E.g. the users location in relation to the center location could be used, directing the user to the closest language resource center. Another possibility is to take center load into account and direct the user to the language resource center with the lowest load at that moment.

With respect to administration of the PID record there is the question what level of administration is required. There are several approaches possible, each moving towards a more fine grained administration model. (1) All language resource centers in the federation have full administrative permissions on the PID record and can update the record as soon as a new instance of a DO is deposited in their center. (2) The slave centers do not have administrative permissions on the PID record and will notify the master center (the repository of record) with the location when a DO instance is deposited in their center. The master center is the owner of the PID record and can update the PID

[3] In a non-federated scenario, this SP can redirect directly to a single IDP. This will skip step 3,4 and 5

record with the new location information. (3) The master center (the repository of record) is the PID owner and all slave archive get administrative permissions to update only their location information in the PID record.

PID	URI_1
	...
	URI_N
	$checksum$
	...

Table 1: PID record

4. A real world language resource center

The Max Planck Institute for Psycholinguistics manages a LR repository, The Language Archive, with approximately 80TB of data and 100.000 meta data records. The digital objects are e.g. structural and descriptive meta data files, multimedia files and annotation files. Currently, we are looking into solutions to provide long-term persistency of our data and our services, by implementing a new synchronization process.

The new synchronization process is implemented as a logical synchronization workflow which (1) is able to traverse the hierarchical structure defined in the meta data and (2) consists of multiple steps with at least the data synchronization step present. We aim to achieve a master-slave scenario with one master, the owner of the data (the repository of record), and many slaves as read-only archives. Currently we are cooperating with MPG data centers. Users trying to access a language resource should use the PID identifying the resource. The resolver will then redirect the user to any of the centers hosting the DO. If the user wants to make changes, the resolver redirects the user to the master center, since that is the only writeable center.

The data in The Language Archive is organized as IMDI[4] (Broeder and Wittenburg, 2006), which allows the definition of hierarchical structures on top of the data. This structure does not have to match with the organization on the filesystem. Therefore some extra logic is required for a synchronization based on the structure defined in the meta data. Typically we aim for a synchronization where a corpus manager, responsible for the synchronization, can select a DO in the master center and in the slave center. The synchronization workflows is required to select the files on the filesystem that are part of the subtree defined in the meta data starting with the selected source DO as root.

The COSIX tool has been implemented to support the synchronization between IMDI based archives. COSIX uses the hierarchical structure information in the IMDI files to traverse and index two IMDI based archives and perform the synchronization of the data and meta data. COSIX does not synchronize any of our services, access rights or any other auxiliary databases. Also, COSIX does not perform any administration of PID records. Currently we have synchronized archives containing over 60000 DOs consisting of data files and meta data files, summing up to a total volume of several hundreds of gigabytes.

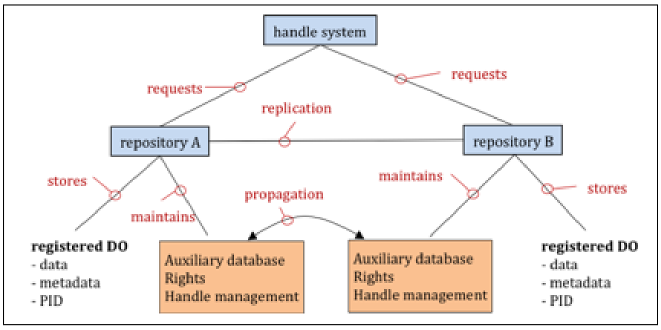

Figure 3: REPLIX

In the context of the EUDAT [5] project we have started the REPLIX [6] pilot project to investigate the use of iRODS (Moore, 2008) to implement a logical synchronization capable of achieving our goal to synchronize data and services, see figure 3. After the data synchronization step (replication in figure 3) a number of extra actions are performed (propagation in figure 3), such as synchronizing the auxiliary databases, synchronizing access rights and updating the persistent identifiers, shown in figure 3. Important goals of the REPLIX project are a loose coupling with existing infrastructure and a configurable synchronization workflow.

iRODS provides functionality to create multiple data grids and provides a rule engine to create configurable synchronization workflows. iRODS provides the possibility to mount a directory on a file system into the iRODS grid, without taking over ownership of the data. Direct access to the files on the file system, outside iRODS, is possible in this scenario. We propose to use such mounted collections providing the loose coupling we require. The downside of this approach is that we cannot associate iRODS user meta data to the objects in mounted collections and that no iRODS system meta data is kept for these files. This can result in problems when using certain icommands and micro services that require proper iRODS system meta data. However, we didn't run into serious issues caused by these limitations during the implementation of the initial synchronization workflow.

Besides the loose coupled connection to the archive data we implemented a loose coupled connection to a number of our services as well. This loose connection has been realized by developing a set of micro-services communication with the services via XML-RPC and REST-full interfaces. An example of such a connection to a service is the synchronization of authorization information. As one

[4] The transition to CMDI is currently in progress.

[5] http://www.eudat.eu

[6] http://www.mpi.nl/replix

of the last steps in our synchronization workflow an export of our authorization database is created by a XML-RPC call. This export is transferred to the destination archive where it is import again via a XML-RPC call. Finally all databases are refreshed with this new authorization information with a REST call to our access management system, AMS.

Since our archive contains a large amount of meta data objects a typical synchronization contains many small files with a low total volume, while the bulk of the volume comes from the actual resources. Our synchronization therefore has to handle a large amount of files and at the same time a relatively large data volume [7]. iRODS is very capable in transferring large volumes. We have reached speeds of over 500 megabits in transfers to the US over a shared gigabit connection. There are several options to improve the performance of transfers with many small files. For example bundling many small files into a bigger file for efficient transfer. iRODS comes with functionality to create and extract tar bundles.

Shibboleth (Scavo and Cantor, 2005) is the federated identity solution used in The Language Archive. A disadvantage of the use of Shibboleth is its human user, browser, centric approach. The human user initiates the AAI flow and the authentication session is stored in the users browser. This imposes problems in a machine-to-machine communication scenario. There is support for this scenario, but it is not yet widely used and it still has to prove itself. Other approaches to solve this problem are investigated. E.g. the use of security tokens for machine-to-machine based communication; Again this approach has not been widely used and is not proven, although it looks like a promising alternative to the Shibboleth built in machine-to-machine communication.

In The Language Archive we propose the use of handles (Kahn and Wilensky, 2006). The handle system fulfills both the requirements we set for a persistent identifier framework. The creator of the digital object is the handle administrator. In our use-case this typically is the repository of record. This brings some complexity in the handle administration after the synchronization of a DO instance; the owner of the handle has to be identified and informed about the location of this new instance. The owner is responsible for updating the handle record.

This is approach 2 of the PID administration as described in section 3. PID administration option 3 as described in section 3 might provide better administration options. However, this level of administration of handle records is currently not available in the handle system. An alternative is the use an extra layer on top of the handle API such as EPIC [8]. Such a layer could implement these extra functionalities on top of the handle API independently of the developments within the handle system.

[7] Ranging from several gigabytes to multiple terabytes.

[8] http://www.pidconsortium.eu

5. Conclusion

We have described what we consider language resource centers in the context of CLARIN and how we envision a federation of language resource centers. At the same time we discussed the requirements and challenges, such as long term persistency of both the data and the services, interoperability, federated authentication and authorization infrastructure and persistent identifiers.

The Language Archive is presented as an example of a language resource center, which is exploring ways to move towards a federated infrastructure, following the requirements we have set for proper federated language resource centers, with read-only accessible backup sites to provide long term persistency of both our data and our services. The REPLIX project is started as a pilot project to investigate the possibilities of iRODS. We conclude that iRODS provides the functionality we need. However there are still some challenges to be solved, e.g. scalability of the federation administration.

While the REPLIX project tries to solve practical problems for The Language Archive with a hands-on approach, the EUDAT project investigates similar challenges on a larger and more general scale.

6. References

D Broeder and P Wittenburg. 2006. The imdi metadata framework, its current application and future direction. *International Journal of Metadata, Semantics and Ontologies*.

D. Broeder, M. Kemps-Snijders, et al. 2010. A data category registry- and component-based metadata framework. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, pages 43–47.

Robert Kahn and Robert Wilensky. 2006. A framework for distributed digital object services. *International Journal on Digital Libraries*, 6(2), April.

Reagan W. Moore. 2008. Towards a theory of digital preservation. *International Journal of Digital Curation (IIDC)*, 3(1), June.

Nick Ragouzis, John Hughes, et al. 2008. Security assertion markup language (saml) v2.0 technical overview. Technical report, OASIS.

Tom Scavo and Scott Cantor. 2005. Shibboleth architecture. Technical report, Internet2.

Tams Vradi, Peter Wittenburg, et al. 2008. Clarin: Common language resources and technology infrastructure. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, may.