User Delegation in the CLARIN Infrastructure

Jonathan Blumtritt¹, Willem Elbers², Twan Goosen³, Mischa Sallé⁴, Menzo Windhouwer⁵

¹ Cologne Center for eHumanities – University of Cologne
² The Language Archive – Max Planck Institute for Psycholinguistics
³ CLARIN ERIC
⁴ Nikhef
⁵ The Language Archive – Data Archiving and Networked Services

E-mail: jonathan.blumtritt@uni-koeln.de, willem.elbers@mpi.nl, twan@clarin.eu, msalle@nikhef.nl, menzo.windhouwer@dans.knaw.nl

Keywords: web services, single sign-on, delegation

1. Introduction

The topic of this paper is the interaction between two of the pillars of the CLARIN research infrastructure: ¹ ease of access and integration of services. Ease of access has been implemented by enabling researchers to use their home institution credentials to access resources, tools and services offered by CLARIN on the web. This works well in many cases, but has turned out problematic for the cases where these services themselves need to access other services or resources on behalf of the researcher. To research possible solutions and implement them for a specific use case CLARIN-NL² has teamed up with the Dutch BiG Grid project.³ Last year another CLARIN-D⁴ use case has been solved using the same solution. This paper reports on the results of the research and implementation.

2. Shibboleth and User Delegation

Shibboleth is the underlying technology that enables users to use the credentials of their home institute in the CLARIN infrastructure. It is based on the Security Assertion Markup Language (SAML), as a Single Sign-On (SSO) system. Shibboleth⁵ is widely used in the research world, providing web single sign-on based on national federations, where the universities and research institutions function as Identity Providers (IdPs). The CLARIN centers that offer services, i.e., are Service Providers (SPs), have grouped together in a CLARIN federation, which makes it administratively easy for the IdPs to deal with the CLARIN SPs.

The wide support for Shibboleth has made it a good starting point for CLARIN, but it also has disadvantages. Shibboleth is typically aimed at users logging in and interacting with the SPs via their browser. Although the use cases described in this paper always start out in a browser session, the service invoked needs to invoke another service on behalf of the researcher. Shibboleth does not support this by default. In the next section possible solutions to enable such functionality are described.

2.1 Possible solutions

In the research phase of the CLARIN-NL/BiG Grid collaboration many solutions were considered and evaluated against the following requirements (grouped from 3 angles):

1) For the User:
   a) Single-Sign-On
   b) Access public and private services from within a portal (and other services)
   c) Transparent use, no required confirmation for every service or service access

2) For Services:
   a) Authentication by identity provider
   b) Authorization by service provider
   c) Nested service invocation possible (delegation)
   d) Easy to setup (for researcher)

3) For the System as a whole:
   a) Multi-federation authentication using SAML2
   b) REST and possibly SOAP
   c) Using proven technologies
   d) Operational effort minimal
   e) In-line with standards & best practices
   f) Can we start today?

In this section the considered solutions and their evaluations are briefly discussed, for a more extensive discussion see Van Engen and Sallé (2011). For convenience S1 indicates the service that calls another service, which is called S2, on behalf of the researcher.

Open

In this simple model all services trust each other. S1 includes the user identity with its request to S2, which accepts this without further checking. This is easy to setup, but does not scale up to the CLARIN infrastructure. OAuth ¹ (Hammer-Lahav, 2010)

This protocol is popular on the Internet and uses delegated security tokens for one site to access another site, e.g., allow LinkedIn to access your Google address book. When S1 wants to access S2 the researcher’s browser will be redirected to S2. There the researcher allows the access, and is redirected back to S1. The drawback is, the separate confirmation needed for each combination of services.

SAML ECP (SAML V2.0 Contributors, 2005)

Enhanced Client or Proxy (ECP) is developed to support SAML for programs other than the browser. It is actually supported by Shibboleth, but not enabled by default, and
SimpleSAMLphp\(^6\) does not support delegation via ECP. Since CLARIN cannot force the IdPs to enable ECP. And furthermore since ECP would require a configuration for each AP at each IdP, which does not scale, this is not a viable solution.

WS-Trust\(^7\)

WS-Trust defines the concept of a security token service for SOAP web services. It is a flexible but rather complex setup, and can also be problematic for REST services.

OAuth 2\(^8\) (Hardt, 2012)

This next evolution of OAuth supports more scenarios and is quickly gaining popularity and is replacing OAuth 1. As in the WS-Trust case a central service, an Authorization Service (AS), allows S1 to request a security token to pass on to S2, which can check the validity of the token and receive the user identity. Although this solution was fairly new at the time it was selected as the primary option to be further investigated.

GEMBus STS

The GEMBus framework\(^9\) is intended as a multi-domain communication environment and provides a number of services, including a security token service. At the time of evaluation GEMBus was alpha software.

X.509 certificates (Cooper, Santesson, Farrell, Boeyen, Housley, & Polk, 2008)

These certificates are the basis of the widely used SSL and TLS protocols. They are based on a public key infrastructure where trusted certificates are signed by trusted certificate authorities (CA). Delegation can be implemented using proxy certificates and is used as such in the ‘grid world’. At the cost of additional setup the, much feared, burden of managing the certificate/keypair can be hidden from the user. This solution was selected as the secondary option to be investigated in case the OAuth 2 solution would fail.

3. Configuring and Running an OAuth 2 Authentication Service

Figure 1 sketches the OAuth 2.0 delegation workflow: A user is logged in to Service 1 (S1), which is secured via a Shibboleth SP, using the IdP of his home institution. When the user triggers an action on S1 that requires access to a resource on Service 2 (S2), S1 redirects the user to the AS to collect an access token. Since the AS is also secured via an SP, it sends the user to the Discovery Service (DS) where he selects the IdP for authentication. The AS creates an authorisation code which is sent to S1 via the user. S1 uses it to request an OAuth2 access token from the same AS. S1 then passes this access token to S2,

\(^6\) https://simplesamlphp.org
\(^7\) http://docs.oasis-open.org/ws-sx/ws-trust/v1.4/ws-trust.html
\(^8\) http://oauth.net/
\(^9\) http://geant3.archive.geant.net/Research/Multidomain_User_Application_Research/Pages/GEMBus.aspx
which checks the validity of the token with the AS and
receives user attributes in return (such as the user ID
derived from the EPPN (EduPersonPrincipalName)). If
the token is valid and S2 authorizes the user for the re-
source (a decision based on the user ID), S2 sends back
the response to S1, which can then process it and com-
plete the action triggered by the user. For the lifetime of
the initial token, further communication between S1 and
S2 can occur without the need to request another token.
Van Engen and Sallé (2013) describe how first
S2 can occur without the need to request another token.

The solution based on X.509 certificates was not further
implemented, but Van Engen and Sallé (2013) state that a
pre-authentication filter and a dummy ‘UserDetailsSer-
vice’. 

On the ISOcat side OAuth for Spring Security could not
be used as it is not based on servlet technology. However,
this part of the AS interaction is relatively simple. The
security token is retrieved from the HTTP header and
passed on in a simple check token request to the AS. If the
token is valid the identity of the researcher is returned and
ISOcat can extend the search to include her workspace.
One implementation issue which still needs to be resolved
is the Component Registry’s use of frames for the AS
interaction. It was pointed out that this hides the URL of
the AS and IdP, which makes it hard for the researcher to
determine to whom she is providing her credentials.

4. CLARIN Use Cases

4.1 CMD Component Registry and ISOcat

The Component Registry is part of the Component
Metadata (CMD) Infrastructure (Broeder, et al., 2010)
implemented by CLARIN. It provides an online editor to
metadata modellers to create CMD profiles and compo-
nents. To enable semantic interoperability, these CMD
profiles or components contain references to, among
others, data categories stored in the ISOcat Data Category
Registry. The editor allows searching in ISOcat, where
the search is initiated by the Component Registry
backend, i.e., the backend plays the role of S1 and ISOcat
that of S2. Without user delegation only a search for
public data categories is possible. Hence the use case is to
extend the search for private data categories in the ISOcat
users workspace.

To enable this, the Component Registry has been extend-
eded with OAuth for Spring Security, providing the follow-
ing functionality:

1) A method to check if a security token is available in
   the current session;
2) A method to initiate the request for a security token,
   i.e., to interact with the ndg_oauth AS including
   logging in and giving permission for delegation;
3) A method to query ISOcat while passing on the
   security token.

Enabling OAuth for Spring Security required the already
present Shibboleth authentication layer to be ‘bridged’
with Spring Security. This was solved by a simple, though
not entirely obvious mapping, involving a custom
‘pre-authentication filter’ and a dummy ‘UserDetailsSer-
vice’.

The CLASS web application implements tools for
searching and analysis based on the Poio API, and also
provides easy-to-use web interfaces to facilitate field
linguists’ research. Apart from hosting scripts the main
function of the CLASS application is to serve as a gate-
way to the archives that maintain annotated corpora. The

4.2 CLASS: Cologne Language Archive Services

The CLASS web application implements tools for
searching and analysis based on the Poio API, and also
provides easy-to-use web interfaces to facilitate field
linguists’ research. Apart from hosting scripts the main
aim is to offer a convenient web-based workflow, that enables the user of the application to access resource files for analysis directly from the repository. The Cologne use case targets the DoBeS corpus, a core resource hosted by The Language Archive (TLA) at the Max Planck Institute for Psycholinguistics (MPI), a CLARIN center. Most of the collections within the corpus are protected on a personalized level for privacy and ethical reasons. They may only be accessed by the corresponding owner or research group, hence the retrieval of data by external services was unviable in the past. It was soon noticed that this was another case that called for a solution of the delegation issue with the CLASS web application playing the role of S1 and a TLA service that of S2. With the availability of the AS the realization of this layout was possible (see Figure 2).

TLA has implemented a servlet, also known as the TLA facade service, which allows delegated access to the resources in the archive. Contrary to ISOcat this servlet can and does use the OAuth for Spring Security. The services provided by the TLA facade are:

1) accessRights: receive the access rights (none, read or read/write) the logged-in researcher has for one or more resources;
2) accessFile: fetch a specific resource for the logged-in researcher (if she has the right to do so).

The CLASS application uses the rauth18 library written in Python as an OAuth 2 client to talk with the AS and call the TLA facade services. OAuth 2 is specifically designed to reduce complexity on the client side. Tie-ins with common web frameworks are smooth and well documented. Now researchers can run the tools provided by CLASS on resources residing in The Language Archive.

5. Future Work and Conclusion

Apart from these two first use cases other uses are possible. For example, in addition to accessing archived resources, CLASS tools could also issue delegated calls to protected remote tools, i.e., web services residing on different sites. The same can potentially be done by WebLicht19 web services. Another potential extension is multi-step delegation: the current solution supports single step delegation, i.e., from S1 to S2, but S2 cannot request a security token from the AS to call a next service, Sn. Support for such multi-step delegation is currently under investigation.

Not all IdPs release sufficient information for the AS to allow identification of the logged-in researcher. Rather than a universally identical user identifier, such as EPPN (EduPersonPrincipalName), the IdP might release a EPTID (EduPersonTemporaryId). Although the IdP gives the same EPTID each time the researcher accesses a certain SP (so it can use it to identify the return of the researcher), it gives a different EPTID for the same researcher to each different SP. When the AS and S2 thus are hosted at different SPs the EPTID cannot always be used to identify the researcher. Thus researchers with such an IdP are likely to have problems using delegation.

The ndg_oauth AS is currently an experimental service at TLA. In the future this or another AS could be a CLARIN service, but to realize this service, the stability and high availability options have to be investigated first.

6 Acknowledgements

The authors like to acknowledge the valuable support of all the experts in the Big Grid/CLARIN project and the current CSNE informal expert group, especially Willem van Engen who implemented an easy to use and understandable ndg_oauth and OAuth for Spring Security demo setup.20 We also like to thank Shakila Shayan for the implementation of the TLA facade servlet.

7 References


---

17 http://tla.mpi.nl/
18 http://rauth.readthedocs.org
19 http://weblicht.sfs.uni-tuebingen.de
20 https://github.com/wvengen/oauth2-demo