CLARIN / BiG Grid Development Report

User Delegation in the CLARIN Metadata Infrastructure: connecting the component registry and ISO-DCR

Part I - Research

version 010; authors Willem van Engen & Mischa Sallé

20 October 2011

Table of Contents

1 Introduction

The European project CLARIN[1] is developing a research infrastructure for e-Humanities based on a service oriented architecture. An important aspect is security, and user delegation in particular. The vision of CLARIN is to have a collection of web services, accessible to the user via a (web) portal. A web service may need to call another service, which may require the credentials of the calling user. This has been the topic of a workshop on security for web services[2] organised by BiG Grid in April, 2010.

The next step has been found in a practical use-case where one web application (the component registry) needs to access a user's private data in another web application (ISO-DCR). This document is a study of the technical options that could solve this particular use-case, as a prototype for the delegation issue for CLARIN in general.

1.1 The specific use-case: component registry & ISO-DCR

Within the component metadata framework there are currently two software tools: the component registry[3]/editor[4] and the ISO data category registry[5] (ISO-DCR) which can be used in a collaborative scenario. In both applications separately, users can log in to edit their own data.

On the other hand, the component registry may contain references to the ISO-DCR. When a user adds such a reference, he may want to see his own private ISO-DCR information from within the component registry/editor. This requires the latter to have access to the user's *private* information in ISO-DCR.

Currently only public ISO-DCR information can be referenced. The end goal of this use-case is to also be able to refer to private information from ISO-DCR from within the component registry/editor.

1.2 Current situation of the tools

The CMDI component registry/editor is a web application that allows the user to create new metadata components and schemas for use in the CMDI infrastructure. Part of the creation process for metadata components require that every element in a metadata component is associated with a reference to a Data Category (DC) stored in the ISO-DCR. To make this easy for the user, a pick list of DCs is provided with information obtained from the ISO-DCR. Access to component registry is limited to authenticated and authorized users and the application is Shibbolized.

The ISO-DCR is a data category registry (compliant with ISO 12620:2009), that stores numerous DCs created by the linguistic community. DC are ordered in different profiles and may be either public or private. The ISO-DCR has a REST web service that allows an application to obtain a list of public DCs based on a search query. Non-authenticated users have only access to public DCs in ISO-DCR, but editing and access to private DCs requires authentication and authorization. The ISO-DCR currently uses its own authentication and user database, but a connection to Shibboleth is being worked on,

1 http://www.clarin.eu/
2 http://agenda.nikhef.nl/conferenceDisplay.py?confId=993
3 http://catalog.clarin.eu/ds/ComponentRegistry/
4 http://www.lat-mpi.eu/tools/arbil/
5 http://www.isocat.org/

and it will be connected to the same user database as the component editor/registry.

1.3 Delegation desired

We want to enable the component registry/editor to access the user's ‘private’ DCs from the ISO-DCR. With the larger CLARIN picture in mind, this can best be implemented by using the delegated credentials of the currently logged-in user to access the ISO-DCR from the component registry. If necessary, the ISO-DCR's REST implementation can be replaced by a SOAP one, to enable conveying security tokens as specified by OASIS WSS. However keeping the current REST implementation seems less complex.

1.4 Terms & symbols

User – End-user of the CLARIN infrastructure, a researcher in the social sciences or humanities.

Identity provider (IdP) - Party that verifies a user's identity, typically a username/password combination using the SAML Web SSO profile. The result is an assertion that proves a user's identity.

Service – Building block with which a user performs research. For the specific use-case both ISO-DCR and CMDI are services. Throughout this document S1 refers to a service the user accesses directly and S2 is accessed by S1 using delegation, unless indicated otherwise.

Portal – A user's entry-point to the CLARIN infrastructure, typically a web portal that can access services on behalf of the user. This is considered a service itself (S1), unless mentioned otherwise. A user can access both ISO-DCR and CMDI directly using a web browser, so both are portals.

2 Requirements

General requirements for delegation in the CLARIN project have been discussed in the workshop[6] on this topic.

for the User

- Single sign-on
- Access public and private services from within portal (and other services)
- Transparent use, no required confirmation for every service or service access

for Services

- Authentication by identity provider
- Authorization by service owner
- Nested service invocation possible (delegation)
- Easy to setup (for researcher)

for the System as a whole

- Multi-federation authentication using SAML2 [7] [8]
- REST and possibly SOAP[9] [10]
- Using proven technologies
- Operational effort minimal
- In-line with standards & best practices [11]
- Can we start today?

In addition to this (though not a requirement) it would be good if only a part of a user's abilities can be delegated: constrained delegation.

2.1 Use-case: component registry & ISO-DCR

For the use-case connecting ISO-DCR and the component registry the requirements are more relaxed:

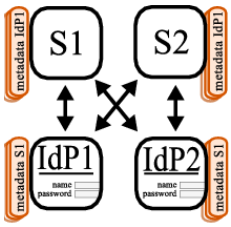
- Single delegation step (no nested invocation)
- Preferably at least REST

While ISO-DCR currently has a REST interface, a SOAP interface could be added if so desired. In general, authentication and authorization used with REST can easily be applied to SOAP as well. The reverse is not true: authentication and authorization used with SOAP is often based on WS-Security[12], which does not always translate transparently to REST.

6 See footnote 2 on page 1.
7 http://www.clarin.eu/files/trust_domain-CLARIN-ShortGuide.pdf
8 http://www.clarin.eu/files/wg2-6-requirements-doc-v2.pdf, §5.1.5
9 http://www.clarin.eu/system/files/SOA-CLARIN-ShortGuide.pdf
10 It appears that REST is popular. There are SOAP services as well, but they may be able to use the same security mechanisms as REST.
11 http://www.clarin.eu/files/standards-CLARIN-ShortGuide.pdf
12 see footnote 26 on page 8
Note however, that SOAP over HTTPS does not need WS-Security.

3 Authentication and metadata

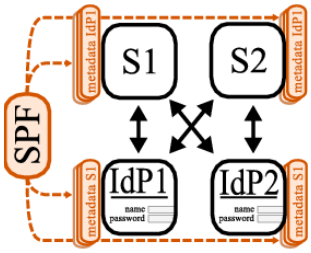
Within the world-wide education and research community there is a strong tendency to use SAML based single-sign-on authentication schemes[13]. Users can use their home institution's credentials to securely authenticate with any service that is connected, while service providers can be sure of a user's identity. The party that verifies the credentials of a user is called an Identity Provider (IdP), usually located at a user's home institution. Shibboleth[14] and SimpleSAMLphp[15] are two widely used software packages providing an IdP service.



When adding new services to such an infrastructure, it is necessary to provide it with information (metadata) about all the IdPs it trusts. In turn, each IdP needs to be configured to allow each service as well.

To make collaboration easier, such networks of IdPs and services are usually organized within national identity federations, making it easier to exchange metadata and to setup agreements on mutual trust. In order to collaborate across borders, some means of connecting multiple federations becomes necessary. The Geánt eduGAIN[16] initiative aims to form a pan-European confederation, providing the technical means to smoothly connect different federations.

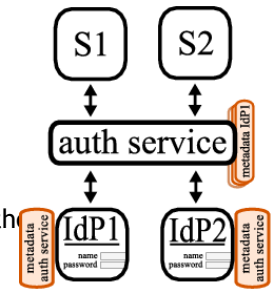
3.1 Service provider federation



Services in the CLARIN project need to be able to handle users of multiple federations and therefore would need to have metadata for IdPs across all these federations, making it hard to add new services and connecting new IdPs. While eduGAIN can ease the technical exchange of metadata, the need to cooperate with all IdPs to accept each service can still be a large barrier. This is where the CLARIN service provider federation[17] (SPF) comes in, which does this for all services at once: all services sign agreements with the SPF, and the SPF signs agreements with every national identity federation (and so effectively with each IdP). Services can use the SPF or eduGAIN to retrieve their metadata (the former is depicted).

Even if the user is logged in using single sign-on, he still needs to approve each service's access the very first time (for the release of, possibly privacy-sensitive, attributes).

3.2 Central authentication service



An alternative approach would be to provide a central authentication service. Instead of interacting with the IdPs directly, services interact with the authentication service. The latter would be known and trusted by each IdP and vice versa, but now the metadata exchange

13 Security Assertion Markup Language, an open standard for authentication and auth XML; http://saml.xml.org/
14 http://shibboleth.internet2.edu/
15 http://simplesamlphp.org/
16 http://www.edugain.org/
17 For a general overview, see http://www.clarin.eu/system/files/SPF-CLARIN-ShortGuide.pdf. Current participants can be found at http://www.clarin.eu/node/2965 , while adding a new service is described at http://www.clarin.eu/node/3227 .

has to happen only once for the whole infrastructure. This basically would create a security domain separate from the IdPs, with the authentication service as a bridge.

The burden of maintaining the IdP's metadata (and vice versa) now lies with one authentication service instead of many services.

3.3 Authentication versus authorization

Authentication is the verification of a user's identity, which is done by an IdP. Authorization is the permission to access a service, for which the decision is taken by the owner of the service (generally using the result of authentication).

Some of the technology options discussed in section 4 address both of these (SAML ECP). Other mechamisms, such as OAuth 1, address only delegation. They need a separate mechanism for authentication at the point where a user enters the system, which in this use-case is expected to always be a portal.

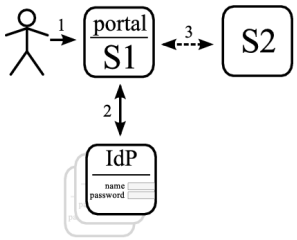
4 Options for delegation

There are several technological options that enable delegation. This chapter discusses a number of them. While this list is not exhaustive, the most important options should be present. Each is explained briefly, including a rough sketch of how it would work in practice. Though the most important benefits and drawbacks are listed in each case, their applicability as a solution will be discussed in chapter 5.

Legend for the diagrams:

stick figure: user
S1: service 1 (which can be a portal)
S2: service 2 (delegated access)
solid line: browser interaction (including redirection)
dotted line: service-to-service communication

4.1 Open



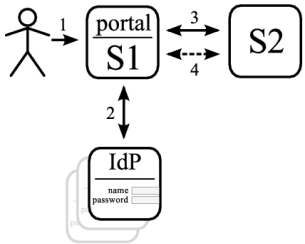
The most simple model is one in which all services trust each other. When the user *(1)* access the portal, he needs to *(2)* authenticate first with an IdP. When S1 *(3)* sends a request to S2 it includes the user's identity, which is accepted by S2 without further checking the validity of the statement.

This approach could be manageable when administered by a single person. Adding a service requires trust by all other services, which raises the barrier to creating a new service.

+ Easy to setup & maintain

– Doesn't scale
– Barrier to adding services & portals

4.2 OAuth 1



OAuth 1[18] is used on the world wide web (e.g by Google[19] and Twitter[20]) to allow users to share their private resources stored on one site with another site using delegated tokens instead of username and password. Authentication of the user is completely separate for each service, as this is not addressed in OAuth 1 (see section 3.3).

The user *(1)* accesses S1 after *(2)* authentication. When S1 wants to access S2, it *(3)* redirects the user's web browser to S2. There the user confirms that S1 is allowed to access it, and is redirected back to S1. S1 receives a token with which it can *(4)* access S2. This token can be persistent.

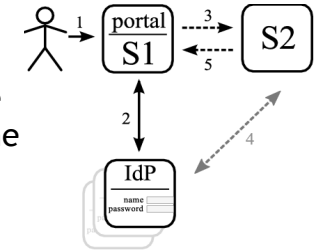
This one-step delegation is managed by the service S2 itself. Each combination of S1 and

18 "The OAuth 1.0 Protocol", RFC 5849
19 http://code.google.com/apis/accounts/docs/OAuth.html
20 http://dev.twitter.com/pages/sign_in_with_twitter

S2 needs user confirmation at least once.

+ Widely used technology	– Single delegation only – Confirmation for each service pair

4.3 SAML ECP



While current IdPs provide web-based single sign-on, SAML ECP[21] (Enhanced Client or Proxy) can do authentication and authorization for programs other than a web-browser, and can be used to provide delegated authentication. This could make a service accesible in the same way using delegation as for an end user directly.

The process flow is as follows. The user *(1)* accesses the portal, *(2)* authenticates at the IdP requesting delegation. When this succeeds, the portal *(3)* initiates a request to S2 on behalf of the user, which it *(4)* checks for validity. S2 subsequently *(5)* returns the response.

Both the SP and IdP need to support SAML ECP. This is the case for Shibboleth IdP 2.1.3 or higher[22] and Shibboleth SP 2.2 or higher. Quite a number of IdPs appear to use[23] Shibboleth 2, but a significant portion still use Shibboleth 1.x, SimpleSAMLphp[24] or something else and hence do not support it.

This approach requires that the metadata for each service is configured at each IdP and vice versa (see section 3). Each IdP additionally needs to configure which services may delegate to which others. For a large number of services and IdPs this becomes unmaintainable.

+ Based on already deployed technology	– Requires SAML ECP support at all IdPs – Metadata exchange necessary for each service & IdP, this does not scale

4.4 WS-Trust

SOAP web services are defined by a collection of specifications[25]. This includes

21 As specified in the SAML profiles standard http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf (SAML ECP as described here is sometimes referred to as SAML ECP 1.0); version 2.0 is currently in draft http://wiki.oasis-open.org/security/SAML2EnhancedClientProfile

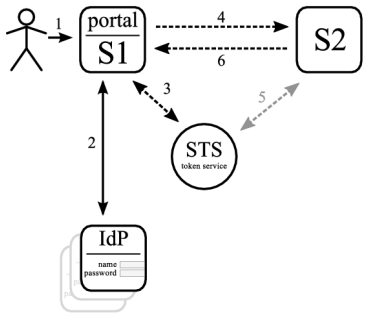
22 For the IdP, a plugin is needed. IdP 2.3 onwards has built-in support for the basic ECP, but the delegated ECP which is needed for our use case will only be available in the next major version, https://spaces.internet2.edu/display/ShibuPortal/Configuring+Shibboleth+Delegation+for+a+Portal; see also https://wiki.shibboleth.net/confluence/display/SHIB2/ECP and https://wiki.shibboleth.net/confluence/display/SHIB2/IdP+ECP+Extension. For the Shibboleth service provider component, version 2.2 or higher is required.

23 https://refeds.terena.org/index.php/FederationProtocol

24 Which does not support SAML ECP; http://markmail.org/message/54pbhpzqwr6dlzd5

25 http://en.wikipedia.org/wiki/WS-* for a list; Figure 1-1 in "The WSIT Tutorial" by Sun Microsystems (Version 1.0 FCS) provides a more friendly explanation of how these fit together. See also footnote 12.

WS-Security[26] for signatures and encryption for (parts of) SOAP messages, and WS-Trust[27] for the concept of a security token service (STS). These technologies could be used to setup the desired security infrastructure using standard components. A central STS would provide the necessary information to support delegation, based on an initial assertion from the IdP after user login.



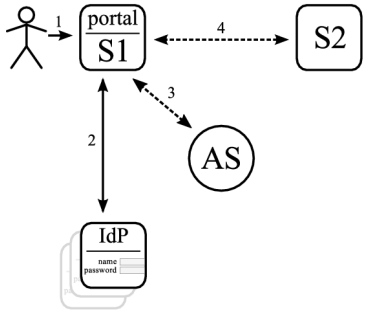
A similar situation is worked out in detail in SURFnet reports on webservices[28]. Our situation is different in the sense that the solution should work across federations.

One approach would be the following. When the user (1) accesses the portal, he is (2) redirected to the IdP, where he logs in. With this token the portal can (3) obtain a security token and (4) access S2. S2 (5) checks if the token is valid, and (6) returns the result upon success. Verification of the token can be done offline (using signatures) or contacting the STS to validate it.

It still needs some work to figure out how this should be implemented exactly. The tools are available (see also footnote 28, 1st report, §2.2), but flexibility comes at a cost of complexity.

- \+ Widely used technology
- \+ Flexible
- – Complex
- – SOAP-based, REST can be a problem
- – Central security token service

4.5 OAuth 2



OAuth 2[29] is the next evolution of OAuth (still in draft though already in use[30]), which supports many more scenario's than OAuth 1. Both use tokens, but OAuth 2 splits the roles between those providing the resource, the access token and the authorization. Obtaining a token from an IdP's SAML assertion is among the possibilities[31]. RedIRIS' OAuth2lib[32] is an implementation of an OAuth 2 client and server that does exactly this.

In a typical situation, the user *(1)* goes to the portal. In the portal, he *(2)* authenticates with an IdP. The portal *(3)* sends the assertion obtained to the authorization service, which returns an access token with a validity for this user, portal, scope (i.e. which services) and lifetime. The portal uses this token to *(4)* send a request to the resource

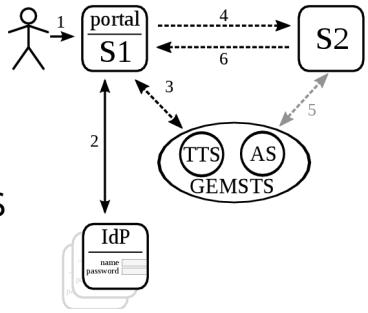
26 http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss#technical
27 http://docs.oasis-open.org/ws-sx/ws-trust/v1.4/ws-trust.html
28 "Webservices en SURFfederatie" (Jun, 2009) indi-2009-06-012 and "PoC Webservices en SURFfederatie" (Sep, 2009) indi-2009-10-014; the first gives a good general technology overview.
29 http://oauth.net/2/,see also the helpful diagram at http://www.independentid.com/2011/03/oauth-flows-extended.html and other posts on the same website. A good introduction to OAuth 2 can be found at http://hueniverse.com/2010/05/introducing-oauth-2-0/
30 by Facebook and github, for example.
31 "SAML 2.0 Bearer Assertion Grant Type Profile for OAuth 2.0", IETF draft draft-ietf-oauth-saml2-bearer and http://iiw.idcommons.net/SAML_Profiles_for_OAuth
32 http://www.rediris.es/oauth2/, first released in Q1 2011. Although written by the same authors as GEMBus (section 4.6), it is a different product.

S2.

To allow delegation a central authorization service[33] needs to be maintained. When the resource S2 receives a request it needs to check the token provided. The resource can make the final authorization decision at that moment.

+ Implementation that supports much of general use-case available
+ Constrained delegation possible
- Fairly new technology
- Central authorization service

4.6 GEMBus STS



The GEMBus framework[34] is intended as a multi-domain communication environment and provides a number of services, including a security token service (GEMSTS)[35]. It consists of two separate entities, a ticket translation service (TTS) handling authentication and token issuance, and an authorization service (AS). Both can obtain their input from external sources. The TTS is the part the user-side will talk to, while the AS is queried by services to validate the security tokens. The GEMBus framework uses a preferred common security token, but the TTS part of the GEMBus STS can provide the necessary translations from one format to another. The use of tokens will reduce repeated authorization overhead and make it possible to relay trust from one service to the next.

The flow is identical to the WS-Trust scenario, only the implementation is different. It is unclear whether step (5), validating the token, is optional.

GEMBus STS is still in alpha shape and is not available until Q3 2011. It aims to be compatible with eduGAIN and has plans for interoperability improvements (X.509, bidirectional tokens, OAuth 2 support for TTS input and AS output, to name a few). The GEMBus session tokens are foreseen to be JWT (JSON Web Token), which are compatible with OAuth 2.

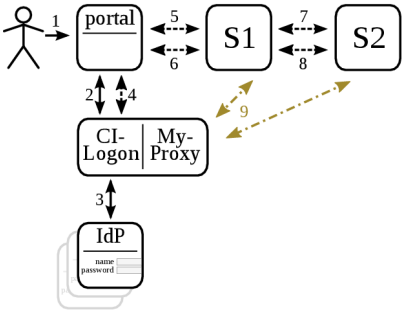
+ Constrained delegation possible
+ REST based (using JWT[36])
- Alpha software, not yet available
- Central GEMSTS service

4.7 X.509 certificates

X.509 certificates[37] form the basis for among others SSL and its successor TLS[38]. Both are

33 What OAuth 2 calls an authorization service (AS) does no more than provide the information to the service which it can use to make the an authorization decision.
34 https://tnc2011.terena.org/core/presentation/14
35 "The GEMBus STS", talk at EUGridPMA, Prague, May 2011; http://agenda.nikhef.nl/materialDisplay.py?contribId=9&materialId=slides&confId=1481 . Its function is similar from to a WS-Trust STS but the latter is a standard based on SOAP, while the GEMBus STS is a particular (REST-based) implementation.
36 JSON Web Token, which can be seen as a lightweight SAML equivalent, for example to use in HTTP headers (both REST and SOAP); http://tools.ietf.org/html/draft-jones-json-web-token-04
37 http://tools.ietf.org/html/rfc5280
38 HTTPS (secure internet access) is based on this.

widely used protocols for which exist efficient and fast implementations[39]. They are based on a public key infrastructure (PKI) where trusted certificates are signed by trusted certificate authorities (CAs) (and user or host private keys).



Delegation can be implemented using proxy certificates. This is what is used in the 'grid world'. In the standard scenario the end-user maintains its end-entity certificate and private key.

Services can delegate the credentials further using e.g. the GridSite[40] delegation service, which is implemented in SOAP over SSL. This is a push model, where the first service (or user tool) requests a proxy delegation at the next service. For the SSL handshake, the client-side uses its (proxy)certificate as credential, while the (next) service uses a host-certificate. This (second) service creates a new key-pair and returns a certificate signing request (CSR) to the client. The first service signs it with its own (proxy)certificate and returns the signed proxy certificate. The next service now has a delegated proxy certificate.

By using the CILogon[41] setup, the first certificate/keypair can be hidden inside the portal, thus removing the burden of maintaining the private key from the user. It combines a MyProxy[42] online CA with either OpenID[43] or Shibboleth authentication. Once authenticated, OAuth 1[44] is used behind the scenes for getting a certificate/keypair on the portal. The certificate/keypair itself remains hidden from the user and can be used for delegation.[45]

A typical scenario would be that the user (1) accesses the portal. For authentication he is (2) redirected to CILogon, which (3) obtains authentication from the IdP. The user is redirected back to the portal, which can then (4) retrieve a certificate. When the portal (5) accesses S1 it uses this certificate for encryption and authentication. To enable S1 to access S2, the portal (6) performs a certificate delegation exchange with S1. This results in a proxy certificate at S1, which it (7) uses to access S2. It can again (8) perform a delegation when needed. If a proxy certificate would expire during the process, there is an option to (9) renew it.

Constrained delegation is supported by standard tools, but only based on validity time and maximum delegation depth. Using custom proxy policies would an option, though more involved.

+ Standard tools
+ Fast

39 In addition there is special hardware to enable SSL acceleration.
40 Certificate-based authentication and authorization with HTTPS; http://www.gridsite.org/wiki/Delegation_protocol
41 http://www.cilogon.org/
42 Standard grid software for accessing and delegating grid certificates. For the approach mentioned it would need to run as a Certificate Authority; http://grid.ncsa.illinois.edu/myproxy/
43 A standard for decentralised authentication; http://openid.net/
44 Detailed at https://docs.google.com/Doc?docid=0Afb6l_ntpfB8ZGNiMzdtcTZfOWRiem5tdDZu
45 Outsourcing the key management to a remote party is allowed by the recently adopted Guidelines on Private Key Protection, http://www.eugridpma.org/guidelines/pkp/ which are currently incorporated only in version 4.3 of the Guidelines for the Classic X.509 CA, http://www.eugridpma.org/guidelines/classic. This was driven by the desire to bridge the gap between Web SSO and grid credentials.

+ Compatible with Grid
− CILogon/MyProxy service must be maintained
− Delegation constraints not so flexible
o Services perform delegation themselves

5 Viable options

5.1 Selection

Now that the advantages and disadvantages of the available technologies are known, a selection can be made. Both the general CLARIN and the specific use-case that connects the component registry and ISO-DCR are covered.

	Specific use-case	***CLARIN general***	*code maturity*	*possible showstoppers*
OAuth 1	+	–	production	user confirmation required for each service pair
SAML ECP	–	–	production	not feasible to require ECP support at each IdP
WS-Trust	o	o	production	SOAP-based, complex
OAuth 2	+	+	early production	beta software
GEMBus STS	o	o	alpha, unreleased	not mature
X.509 certificates	+	+	production	handling of CRLs (depending on implementation)

This leaves OAuth 2 and X.509 certificates as general options. Although the GEMBus STS looks very promising, it is too early to consider it a viable option at this stage. WS-Trust does does not support REST so easily but would be an option otherwise. For the specific use-case OAuth 1 is also an option and may be a last resort when the other options appear to be too time-consuming.

5.2 OAuth 2

This is possibly the solution that is easiest to get started with. There may be a migration path from OAuth 2 to the GEMBus STS scenario, once that software is ready. Although OAuth 2 is not an official standard yet, it has quite some momentum.

5.2.1 Needed software

- The OAuth2lib authorization server needs to be deployed either as a central service, or decentralised.
- Each of the services would need to run an OAuth2lib resource server, as well as an OAuth 2 client to invoke other services. OAuth2lib provides a client, but it should be possible to use other implementations as well.
- OAuth2lib is written in PHP, so each the authorization service and each service would need to run a webserver with PHP.

5.2.2 Implementation and maintenance effort

In a simple setup with one or a number of fixed authorization servers, OAuth2lib should be ready to use. A more decentralised setup would require some thought and implementation work.

Since OAuth2lib is beta-software, some work may be required to get it ready for large-scale production.

5.2.3 Links

- OAuth2lib
  - http://www.rediris.es/oauth2/, with downloads at http://forja.rediris.es/frs/?group_id=801
  - documentation: http://www.rediris.es/oauth2/doc/
- Optionally a separate OAuth 2 client. There are many implementations for different programming languages, incl. Perl, Java, Ruby, Python, etc.; see http://oauth.net/code/
- In certain situations an OAuth2 proxy would be useful
  - mod_oauth2 for Apache httpd[46]
  - standalone proxy[47]
  - this is on the roadmap of OAuth2lib as well[48]

46 A very early implementation seems to be https://github.com/apache/tuscany-sca-cpp/tree/trunk/modules/oauth

47 For example, https://github.com/mojodna/oauth-reverse-proxy

48 “Generic resource server” in their terminology

5.3 X.509 certificates

This solution is based on production software, therefore less prone to bugs, has the most documentation and can be relatively easily tested. Another advantage is that it can optionally be made compatible with Grid.

5.3.1 Needed software

- The CILogon software stack, which includes among others MyProxy and GridShib-CA, would be needed. As central service the online CA would be needed to run and maintained. The portal(s) would also need software from the CILogon project.
- Each of the services would need on one hand to run the GridSite delegation service, and on the other hand client functions to request a delegation at another service. GridSite comes with an example client, but this would need adapting.

5.3.2 Implementation and maintenance effort

The implementation and maintenance effort depends on whether the online CA should be IGTF accredited (necessary for Grid interoperability) or not and whether the certificates can be short-lived (up to 11 days) to prevent the need for certificate revocation lists. Most of the implementation effort will go into adapting the configuration of the different components to the specific use case. Building a test setup with a test CA should not be difficult, given the relative code maturity.

5.3.3 Links

- CILogon
  - software stack: CVS http://cilogon.cvs.sourceforge.net/ or as source tarballs http://sourceforge.net/projects/cilogon/files/
  - Documentation at https://demo.cilogon.org/ and http://www.cilogon.org/ and about the CA specifically at http://ca.cilogon.org/
- The GridSite delegation service
  - software can be found at http://www.gridsite.org/
  - documentation at http://www.gridsite.org/wiki/Delegation_protocol

6 Conclusion: next steps

This study has examined 7 different solutions for the delegation use-case posed by the European CLARIN project, in which an end-user authenticates at one service which can request information from a second service using delegated credentials.

All of the viable solutions show the need for a central security token service. For the decentralized CLARIN project maintaining such a central service is difficult. A central security token service does ease the interoperability of multiple federations as in the CLARIN project.

Two viable options have come out of the research phase: OAuth 2 (sections 4.5 and 5.2) and X.509 certificates (sections 4.7 and 5.3). While the GEMBus Security Token Service (section 4.6) looks promising, it is not yet publicly available and hence it is not yet clear whether this can be used. When it will be available it can be interesting to see whether it can be used as a replacement for the OAuth 2 scenario. There may also be a possibility to use WS-Trust (section 4.4), but that would need some more research[49].

Given the difficulty in this stage to decide which of the two remaining options would be best suited to the specific and general use cases, we propose to build two test setups, one for the solution based on OAuth 2 and one for X.509 (CILogon). The outcome of that will then form the basis for the actual solution.

For the CILogon based X.509 setup as proposed, it has to be discussed and decided whether the CA needs to be IGTF accredited and whether the certificates can be short-lived (no need for CRLs) or long lived. It also needs to be discussed if constrained delegation can be implemented sufficiently.

Next steps

Independent of the specific solution, it would be useful to do the following in the implementation phase:

1. Initial development:
   a) Select and obtain server and client software (where applicable),
   b) Setup a test portal and service that does delegation,
   c) Write and/or package a software solution for use by services,
   d) Document the package.
2. Apply it to ISO-DCR and the component registry:
   a) Deploy delegation service, if needed;
   b) Modify ISO-DCR to accept the new technology as (delegated) authentication,
   c) Modify the component registry to delegate credentials when accessing ISO-DCR.
   d) Document the changes needed in the ISO-DCR and component registry.
3. A standard way for services to handle authorization (in addition to authentication) may be something to think about and could be provided as part of the package

49 Since WS-Trust is SOAP-based, it is not directly suitable for REST. There are possibilities to combine the two, but this was outside the scope of this report.

(1c).

Steps 1 and 2 could optionally be combined depending on available time and effort, but splitting them in the above way is probably preferred, since it may allow to make a choice between OAuth 2 and X.509 already in the first stage. The second stage can then be implemented for only one of them.

7 Acknowledgements

This research, conducted at the request of the CLARIN[50] project, is funded by the Dutch National Grid Initiative BiG Grid[51]. We like to thank Daan Broeder en Menzo Windhouwer of CLARIN and the Max Planck Institute for Psycholinguistics[52], Joost van Dijk and Remco Poortinga-van Wijnen of SURFnet[53] and Diego R. López of RedIRIS[54] for useful discussions and feedback on this report.

50 http://www.clarin.eu/
51 http://www.biggrid.nl/
52 http://www.mpi.nl/
53 http://www.surfnet.nl/
54 http://www.rediris.es/