Project no. 018340

Project acronym: EDIT

Project title: Toward the European Distributed Institute of Taxonomy

Instrument: Network of Excellence

Thematic Priority: Sub-Priority 1.1.6.3: “Global Change and Ecosystems”

C5.82 Specification of CSSO profiles for the platform covering the use cases web application, web service and desktop application

Due date of component: Month 27
Actual submission date: Month 27

Start date of project: 01/03/2006 Duration: 5 years

Organisation name of lead contractor for this component: 9 FUB-INF

Revision draft

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)		
Dissemination Level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

C5.82 Specification of CSSO profiles for the platform covering the use cases web application, web service and desktop application

Objectives

The general objective of the activity is the design of a security infrastructure for the platform and introduction of a secure Single Sign-On service (SSO) for the platform. The task to be achieved within the third JPA concerns the integration of a Community Single Sign-On (CSSO) security infrastructure within the platform.

The CSSO enables the various EDIT service providers to protect their services and resources defining individual access control policies, while users can access different services using only one identity. The security infrastructure bases on the SAML protocol family (e.g. Shibboleth) and provides its federation concept to realise the community aspect.

This component reports on the specification of CSSO profiles for the platform covering the use cases web application, web service and desktop application.

Use cases

EDIT intends to provide various software components covering several aspects of taxonomic research and using different technologies. Nevertheless, any of them is supposed to falling into one of the use cases classes web application, web service or desktop application. Furthermore, all use cases must rely on the standard web protocol HTTP for communication.

A typical **web application** is software program running on a web server, which is accessible over a network such as Internet or intranet. User interaction with this software can only take place using a web browser (client). For instance, the EDIT components *Community Websites*, *CDM Dataportal*, *Experts Database* or *BDTracker* are examples of those web applications.

Unlike web applications, a **web service** is a software program running on a web server delivering information in a structured data format (XML). This information is intended to be further processed by any kind of client application, which may be a web application, a specific desktop application or other web services. Web services can also be seen as network accessible APIs executing requested services on a remote system which may be used by EDIT components like *CDM Community Store*, *GeographicComponents* or *LiteratureComponents*.

Desktop applications (or application software) are computer programs being installed on the user's desktop computer. In contrast to web applications, desktop applications are running on a local computer and usually have an individual user interface. Nevertheless, these applications may communicate over a network with other applications like web services or databases also. If so, they are often called "rich client" as opposed to e.g. web browser, which are called "thin clients". The EDIT component *Taxonomic Editor* will be realised as rich client or application software.

Security Infrastructure

Within the SAML protocol family, we have evaluated **Shibboleth** as suitable framework covering the EDIT requirements regarding CSSO. The Shibboleth middleware software itself is Open Source and runs on Open Source servlet containers like Apache Tomcat in combination with the widely used Apache web server. This includes several connectors to external identity management interfaces enabling the integration of existing infrastructures. Shibboleth provides a suitable implementation of the SAML web browser profile supporting distributed SSO authentication and authorisation for web application and web services. Herewith, Shibboleth already conforms to these EDIT use cases. As desktop applications are designated to communicate with components of both other use cases, this profile has to provide means to integrate them into Shibboleth. Initially, single instances of the Shibboleth components **Identity Provider (IdP)** and **Service Provider (SP)** will be integrated into EDIT forming a local EDIT federation. An IdP is responsible for performing user authentication and delivering related user attributes requested by SPs reliably. SPs are providing services to the EDIT federation and are relying on these user attributes for access control. Later on, the installation can be scaled up whenever other institutions may want to affiliate themselves to the community. This includes the integration of individual IdPs as well as the provision of further services running on individual SPs. The former probably involves the installation of a Shibboleth **"Where are you from" (WAYF)** component, which enables users to select their home institution requested for identification. Other applications supporting core administration issues of federations (e.g. user attributes, group/role memberships or users' privacy concerns are subject to evaluation and will be integrated whenever considered appropriate.

Finally, setting up a small **Public Key Infrastructure (PKI)** is necessary, because Shibboleth components must be equipped with X.509 certificates in order to establish secure communication channels between all federated instances.

Profiles¶

Considering the use cases described above, this section presents a brief overview on how these scenarios shall be provided using Shibboleth.

Since, **web applications** represent the basic design objective of the Shibboleth initiative, installing an appropriate EDIT federation based on the security infrastructure sketched above will meet the requirements for this use case. First, the EDIT component *Experts Database* is considered for integration. The Experts Database bases on the *Drupal* Content Management Systems which was evaluated as software platform for EDIT community components in general. Therefore, it has to be connected with CSSO first. Furthermore, the Experts Database is a candidate for a primary attribute source to foster attribute based identification of EDIT taxonomists.

As web applications, **web services** are also covered by the Shibboleth framework. Usually, web service client software (e.g. web services or desktop applications) is unaware of the Shibboleth authentication framework, additional software components will have to be developed to provide CSSO facilities to web services clients. Due to the fact that both scenarios apply to the desktop application profile as well, please refer to the next section.

Usually, **desktop application** software is not designed to operate with the Shibboleth authentication framework. Fortunately, the EDIT desktop application use case sketched above limits the external networking application range of those software components to the HTTP-protocol. So, those rich client software applications can be integrated into Shibboleth's SSO authentication framework. To support the integration of those applications, the additional components *CSSO-ShibProxy* and *CSSO-API* will be provided.

The **CSSO-ShibProxy** provides an intermediate filter to the Shibboleth authentication system. This filter enables Shibboleth unaware applications to connect automatically and make use of HTTP-based service providers protected by the Shibboleth authentication framework.

As the **CSSO-API** represents the client part of the Shibboleth Proxy implementation, this part will be provided to EDIT software developers as Java-API (according to the EDIT guidelines) in order to enable them "shibbolising" their client application easily.

Details

For more detailed information regarding CSSO or the current development state of this activity, please refer to the CSSO section within the WP5 developer's wiki[1].

[1] Detailed CSSO description in the WP5 developers wiki (http://dev.e-taxonomy.eu/trac/wiki/CSSO)