Introducing Community Single Sign-On for EDIT

Lutz Suhrbier

Networked Information Systems (http://www.ag-nbi.de),
Department of Computer Science
Freie Universität Berlin
Königin-Luise-Straße 24-26
14195 Berlin
suhrbier@inf.fu-berlin.de

Abstract: The European Distributed Institute of Taxonomy (EDIT) platform, as well as biodiversity providers in general, provides a multitude of web-based taxonomic applications and services. Also, the diversity of service providers reflects the highly distributed, cross-national organisational infrastructure of taxonomic institutions and collections. This results in a problem of identity management. While the provider's system administrators have to register users and maintain individual access control lists for each offered service, users have to remember a variety of login/password combinations to use all these different services.

Therefore, EDIT promotes a Community Single Sign-On (CSSO) security infrastructure, which protects and provides access to all EDIT platform components based on a single identity per user. That way, users need to remember only one login/password combination to use EDIT's platform facilities. And, service providers can proceed to protect their resources and services by defining individual access control policies, but at considerably reduced administrative costs.

These fundamental enhancements can be achieved through the introduction of a Security Assertion Markup Language (SAML) based (Shibboleth) single sign-on framework, adapted to the requirements of the EDIT platform. Since, information infrastructures within EDIT are quite similar to those in the general biodiversity community, our approach shall motivate other providers to follow. Therefore, this document provides a first-hand report initiating single sign-on for EDIT.

1 Introduction

The European Distributed Institute of Taxonomy (EDIT) [Ed09] is a network of excellence bringing together 28 leading European, North American and Russian natural history collections-based institutions. EDIT's general objectives aim to reduce the fragmentation in European taxonomic research and expertise, to coordinate the European contribution to the global taxonomic effort and to improve society's capacity for biodiversity conservation.

In accordance with these general objectives, the creation of an Internet platform for Cybertaxonomy approaches to relieve typical taxonomic research activities and to provide additional services facilitating co-operation between taxonomists. Additionally, participating institutions' biodiversity informatics and IT resources have to be integrated into the platform.

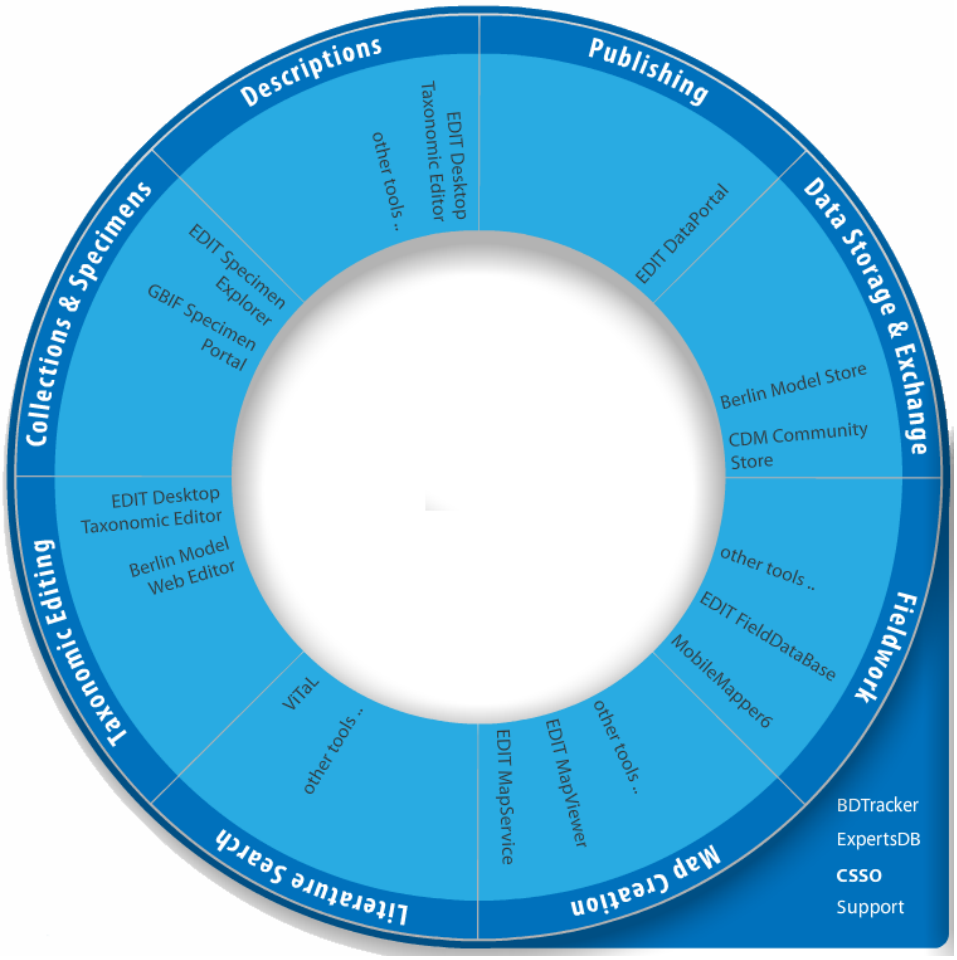

Figure 1: EDIT Internet platform for Taxonomy (http://www.cybertaxonomy.eu)

Further on, the highly distributed nature of EDIT partners reflects the general organisational (infra-)structures within the biodiversity research community. Among others, this community consists of a cross-national conglomerate of universities, institutes, botanical museums, (private) collections. In particular, any community member may contribute several service providers, databases, hosts, or other applications and a number of potential users to the community, but can employ a limited number of system administrators only. Finally, each member may have individual security or organisational requirements to be considered.

Considering the planning for a Cybertaxonomy platform, there is an inherent problem regarding identity management. While, users are obliged to authenticate multiple times in order to access all these different services offered by the platform, system administrators are burdened with the maintenance of multiple access control configurations responding to individual service requirements. Consequently, there is a need for a comfortable Single Sign-On (SSO) solution which will support particular security and organisational requirements of providers as well as privacy aspects of users. The ladder should be conform with the EU's data protection and privacy directive [Eu95].

Therefore, the Community Single Sign-On (CSSO) security infrastructure aims to integrate the security domains of various service providers into EDIT's Cybertaxonomy platform. Thereby, most biodiversity service providers demand to remain the sovereigns of their resources and services offered. Nevertheless, EDIT community members shall be enabled to access these services using a single identity within the community. Further on, community members should be registered only once, which could be done ideally by (re-)using an already existing user account at the user's home institution.

This document will describe EDIT's approach of installing the CSSO security infrastructure into its Cybertaxonomy platform. Therefore, the subsequent chapters will start with an overview of the application scenarios to be covered by CSSO in chapter 2. Next, a suitable single sign-on (SSO) framework has to be evaluated in chapter 3, meeting the requirements resulting from these application scenarios. Subsequently, chapter 4 presents an basic overview of the SAML protocol and briefly describes the components of the CSSO security infrastructure. Then, chapter 5 profiles how the application scenario requirements have been integrated with the CSSO Security infrastructure. Finally, chapter 6 combines a report on our general experience made within this ongoing process and has an eye towards the future.

2 CSSO application scenarios

This chapter outlines expected application scenarios for the Cybertaxonomy platform. Also, these application scenarios reflect some initial technical requirements to be covered by the CSSO security infrastructure.

While EDIT's Cybertaxonomy platform intends to provide various software components covering several aspects of taxonomic research, these components may be offered using different technologies. However, any of these components must be able to communicate *via* common web interfaces. So, any of these components must support the standard web protocol HTTP. Nevertheless, any of them can be assigned to one of the following application scenarios:

- web application
- web service
- desktop application

The following paragraphs will provide general definitions regarding these application scenarios together with some example components within EDIT context.

A web application is assumed to be a software program running on a web server and accessible over a network such as Internet oder Intranet. User interaction with this software can only take place using a web browser (client). EDIT components like EDIT Dataportal[1], Experts Database[2] and BDTracker[3] 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, desktop application or another web service as well. Web services can also be seen as network accessible APIs executing services on a remote system requested by a client. For example, EDIT platform components like CDM Community Store[4] and EDIT MapService[5] are implemented using REST [Fi00] based web service technologies.

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 network with other applications like web services or databases. If so, they are often called "rich client“ as opposed to e.g. web browser, which are called "thin clients". Currently, the platform component Desktop Taxonomic Editor[6] represents such a rich client application interacting with the web service component CDM Community Store.

The next step before we can build up the CSSO security infrastructure is to evaluate a suitable single sign-on framework, which meets the requirements coming along with the application scenarios mentioned in this chapter.

3 Evaluating a single sign-on framework for EDIT

The first decision which has to be made before integrating single sign-on (SSO) into the EDIT platform is about a suitable SSO framework. The framework to be selected must meet the specific requirement deduced from the application scenario described in chapter 2. Particularly, the highly distributed organisational infrastructure and the integration of existing identity management systems or user databases respectively has been given top priority.

[1] http://dev.e-taxonomy.eu/trac/wiki/CdmDataportal
[2] http://dev.e-taxonomy.eu/trac/wiki/ExpertsDatabase
[3] http://dev.e-taxonomy.eu/trac/wiki/BDTracker
[4] http://dev.e-taxonomy.eu/trac/wiki/CommunityServer
[5] http://dev.e-taxonomy.eu/trac/wiki/GeographicComponents
[6] http://dev.e-taxonomy.eu/trac/wiki/TaxonomicEditor

Since, it matches almost perfectly our organisational demands, the evaluation process result was to start using Shibboleth [In09], which founds on the Security Assertion Markup Language (SAML) [Oa09] standard family. Shibboleth is an Internet2 Middleware Initiative project that has created an architecture and open-source implementation for federated, identity-based authentication and authorization infrastructure based on SAML. SAML is a product of the OASIS Security Services Technical Committee and represents an XML-based standard defining secure exchange statements for authentication and authorisation information between security domains. Other qualities are attribute based authorisation enabling privacy-preserving access to individually protected online resources, the federation concept and the availability of stable open source implementations like Shibboleth or OpenSSO [Co09]. Therewith, SAML excels versus other single sign-on (SSO) approaches like Passport [Mi09], Kerberos [MIT09], OpenID [Op09] or CAS [Ja09].

The Passport approach violates the requirements regarding privacy aspects, since the sovereign of any personal data stored there is the Passport operating company (Microsoft). Additionally, every user must create a Passport account first. Thus, it is not possible to reuse existing accounts at the users home institution. Kerberos lacks on its centralist design, which does not reflect the heterogeneous nature of the biodiversity community. Beyond that, the availability requirement for the central server makes it a single point of failure. OpenID follows an highly decentralised approach, but does not support something similar to the federation concept. So, it is on the given service provider to maintain lists of trusted OpenID identity providers. Furthermore, OpenID is suspected to be vulnerable for several phishing and man-in-the-middle attacks. CAS pursues a similar approach to Shibboleth, but missed an implemented federation concept at time of decision. Finally, the status of SAML as an XML based OASIS standard, its principal applicability in different environments and the availability of Shibboleth, as a stable and secure implementation of SAML's web profile specification were decisive factors for using SAML.

A federation is a framework enabling multiple scalable trust and policy sets. It is build up by a group of organisations abided by a common set of policies and practices, like e.g. mutual acceptance of user identities. The ladder prepares for interaction between federated parties without negotiating antecedent, bilateral agreements. That way, the federation concept allows for integrating various security requirements required by the CSSO security architecture.

Using SAML's attribute exchange framework, various existing identity management solutions may be integrated into the federation seamlessly. Hereby, attribute assertions can hold direct identity information (e.g. user id) for user authentication. Also, access may be granted or refused indirectly referring to other attributes like group membership or origin site as well (attribute based access control). Within federations, a standard (yet extensible) attribute-value vocabulary should be agreed, to ensure that attribute assertion will be interpreted identically by each member. For instance, the eduPerson schema includes widely-used identity attributes in higher education.

Through the abstraction level introduced by attribute based access control, SAML based solutions enable for active privacy management also. That means, users are the sovereigns of their data and control what kind of information will be forwarded to service providers. For individual, user based attribute management, suitable web interfaces are available and may be installed on identity information providing sites. So, SAML complies with our initial requirement for privacy protection also.

Finally, a SAML based single sign-on system like Shibboleth conforms to all basic EDIT requirements. It maps to the highly distributed organisational infrastructures through its federation concept. Existing identity management solutions can be integrated into the platform by connecting them *via* attribute exchange protocol. Furthermore, service providers remain the final instance to protect their resources. Simultaneously, users keep control of their privacy by enforcing service provider specific rules for attribute release.

Based on this decision for SAML, the next chapter will show the construction of the CSSO security infrastructure.

4 CSSO security infrastructure

Building up a SAML based security infrastructure requires the implementation of some basic components reflecting the given information flow of the SAML attribute exchange protocol. At the beginning of the EDIT project, apart from the programming library OpenSAML [In09a], only Shibboleth could have been investigated as providing a solid and quite comfortable implementation of the SAML web browser profile. Since, Shibboleth comes with a web server module, its use is somewhat limited to environments with administrative access rights to the web server instance. Meanwhile, other open source alternatives like simpleSAMLphp [Fe08] or OpenSSO are extending the operative range of SAML.

SimpleSAMLphp is a pure PHP-based implementation, which makes it particularly applicable within most hosted web space environments. The addition of simpleSAMLphp to the supported SAML implementations of the CSSO security infrastructure result from the direct requirement of the EDIT All Taxa Biodiversity Inventories (ATBI)[7], which are running in such a hosted web space. OpenSSO develops towards a complete, administrative interface for SAML federation management. It also provides basic identity management facilities and remote administration of service providers. Therefore, we started to use OpenSSO for EDIT federation management.

The following subsections will give an overview an brief introduction on all components forming the CSSO security infrastructure.

[7] http://www.atbi.eu

4.1 CSSO components

Independently of the solution installed, the CSSO security infrastructure includes the following components

- Identity Provider(s) (IdPs)
- Service Provider(s) (SPs)
- Discovery or Where are you from service (WAYF)
- Public Key Infrastructure

The IdP's role includes the following tasks

- identity management of the EDIT federation
- authentication of federation members (users)
- attribute management and provision to SPs
- privacy protection

Regarding identity management, IdPs have to manage information of all identities being member of their domains (e.g. institutions). Identity information may include attributes like user ids, credentials (e.g. password), real names, group/role memberships etc.. This information has to be stored on the IdP's platform in a secure manner. Furthermore, the IdP is responsible to authenticate accurately those users belonging to its security domain. The authentication methods have to be commonly agreed within the federation policies. Next, the IdP should provide tools to ease the management of the attribute assignment for administrators in relation with the identity management system of its security domain. Also, the IdP is responsible to securely transmit those attribute information to federated SPs requesting them e.g. during SSO authentication. Finally, IdPs should provide tools facilitating users to manage their privacy concerns. That means, users should be enabled to determine the pieces of information released to SPs.

Service providers grant access to their web resources based on attributes requested from a federated identity provider. Initially, users requesting access to resources of a service provider will be redirected to the IdP of their home institution. The next step depends on the number of IdPs within a federation. If there is more than one IdP, users will be redirected to the WAYF service. The only task of the WAYF service is to redirect users to the IdP of their home institution for authentication. The home IdP can be selected from a list of federated IdPs presented to the user. If there is only one IdP within a federation, the WAYF service is not needed.

After successful login at their home IdP, users get a secure token and will be redirected to the initially requested resource at the SP. Once, a user received that token, it will be cached in his browser and can be presented to any SP to access its web resources. The secure token permits SPs to retrieve any attributes released for the user presenting it. Finally, the SP verifies the validity of the secure token and grants access to the requested resource or not after having validated the user attributes against the local access control policies (see Figure 2).

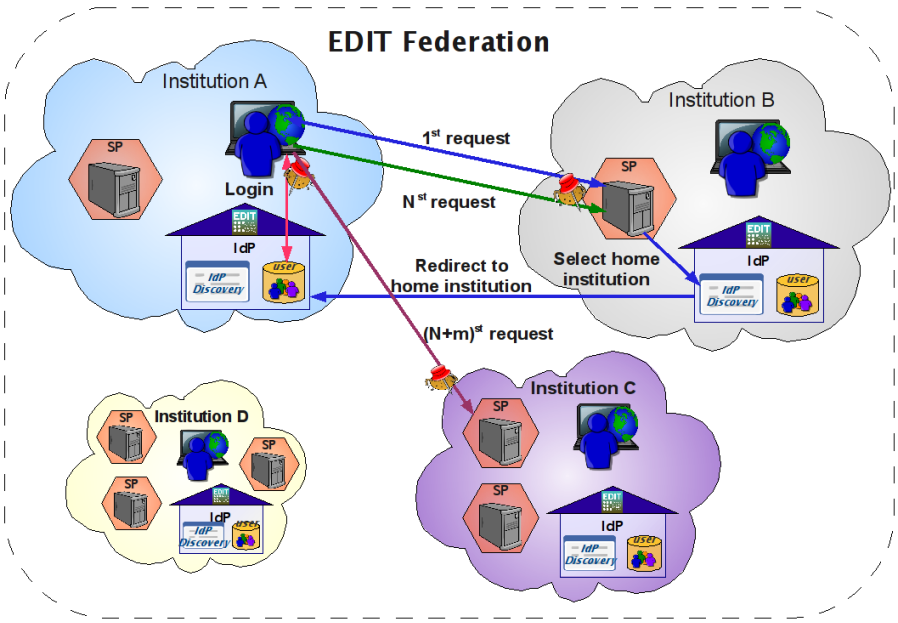

Figure 2: EDIT federation: CSSO information flow

In order to enable secure communication between the components of the CSSO security infrastructure, all components need to be equipped with X.509 certificates. Therefore, it may be necessary to set up a Public Key Infrastructure (PKI) for the EDIT federation in order to issue server certificates whenever needed.

4.2 Federation setup

At first, setting up a federated security infrastructure for the EDIT Cybertaxonomy platform requires the creation of a local federation for EDIT. Due to the highly distributed nature of institutions in biodiversity, the EDIT federation must remain independent from individual or national authentication and authorisation infrastructures (AAIs) like e.g. DFN-AAI. For the time being, the EDIT federation is prospected to consist of EDIT partner institutions only. A future opening towards other biodiversity institutions is envisioned. Of course, this needs political consent of EDIT federation members. For a start, the EDIT federation consists of one IdP holding the initial user registry of the EDIT federation and two institutions providing SPs.

Another crucial point setting up the EDIT federation concerns the set of commonly agreed attributes. Usually, this has to be commonly agreed with all federated partners. Further on, it must be ensured that attribute values are interpreted identically. Also, potential differences in user registration procedures has to be agreed. Failures within one of these fields may lead to severe security leaks, if e.g. user to group assignments or registration policies were interpreted differently or simply not respected. Actually, only approved EDIT member staff can be added to the EDIT user registry.

At the moment, the common set of attributes for the EDIT federation is oriented towards integration of Cybertaxonomy platform components. So, only attributes which are required within web applications for user authentication (e.g. login id, groups/roles), user communication (e.g. email address) or editing user profile information (e.g. name) are included. In addition, each user gets an unique identity identifier which is not changing during system lifetime. That way, identities can be recognised even they are changing e.g. their institution and/or login id. Current attribute names in EDIT have been oriented towards the eduPerson schema, which is often recommended as starting point for higher education environments. Table 1 presents the current set of attributes defined for EDIT.

Attribute	***Authentication***	***Description***
eduPersonPrincipalName	*yes*	login id
eduPersonAffiliation	*yes*	groups/roles
eduPersonTargetedID	*yes*	unique id
mail	*no*	email address
givenName	*no*	given name
postalAddress	*no*	address
telephoneNumber	*no*	phone number
sn	*no*	surname
cn	*no*	common name

Table 1: Common attributes of the EDIT federation

4.3 Identity Provider Integration

This section describes different implementation options evaluated when preparing the IdP integration within the EDIT federation. While user authentication and attribute release to SPs represent IdP core functionalities, all of the SAML implementations presented in chapter 4 support these basic requirements. Additionally, they provide configurable connectors to common types of identity stores like directory services (e.g. LDAP) or databases (SQL). Particularly, OpenSSO offers a web based management console assisting users editing their personal profiles or supporting administrators in managing federations, identity management, attribute release or remote administration of SPs protected by OpenSSO agents.

For those institutions already running identity management systems, Shibboleth or simpleSAMLphp would be recommended solutions. Since, EDIT will not run any identity management system, OpenSSO was selected to be implemented as IdP instance because of its simple but convenient identity management component. By experience of the current OpenSSO version, please note that it is a comfortable and handy tool for federation management, but its configuration is very complicated and a clearly structured manual is missing.

Regarding attribute management, some other tools like ShARPE [Li09a], Group Management Tool (GMT) [Sw09a] or Grouper [In09b] have been evaluated during the course of the project. Finally, they have been rejected because they are outdated (ShARPE), too focussed on a specific federation infrastructure (GMT) or missed identity management facilities (Grouper).

Considering the privacy protection aspect, the OpenSSO just provides the edition of user profiles. Here, uApprove [In09c] provides an interesting approach by getting the users consent before releasing attributes to the requesting SP.

4.4 Service Provider Integration

This section describes the currently evaluated options to integrate SPs within the EDIT federation. The Shibboleth-SP is available in form of a module for different web servers (e.g. Apache). Therefore, administrative rights will be required on the server machine hosting the web server instance to enable installing and configuring the module. As a reference implementation, Shibboleth is a quite flexible and stable service provider. Therefore, it is the recommended choice for most environments in EDIT.

Particularly, in environments where insufficient access rights prevent the installation of the Shibboleth module on a web server (e.g. hosted web environment), simpleSAMLphp will be a suitable alternative. As simpleSAMLphp provides a pure PHP-based SAML protocol implementation, it actually represents the only known option to configure SPs (e.g. EDIT ATBI services) in hosted web spaces, which are generally supporting PHP.

The third option is to install one of the OpenSSO agent modules. This might be the recommended SP implementation, when centralised, remote administration of SP instances is wanted. Unfortunately, OpenSSO also needs administrative rights, since it must also be installed as web server module. In addition, the OpenSSO console application needs to be installed, which should usually be the same instance housing the OpenSSO IdP instance as well.

Any of the SP implementations stated above share the same tasks. First, they are needed to run the SAML protocol and protect the ultimate web applications. Secondly, they extract the requested attributes from the authentication statements received from the authenticating IdP. Finally, the extracted attributes will be handed over to the web application by adding them to the environment of the incoming client connection. For more details considering the integration of web applications, web services or desktop applications, please advance to chapter 5

4.5 IdP Discovery Service (WAYF)

The recent version 2 of the SAML protocol introduced the IdP discovery service. It replaces the former WAYF service and asks the user which IdP to use for authentication. The user selects his home institution and will instantly be redirected to the corresponding IdP's login form. This feature is supported by all SAML implementations supported by the CSSO security infrastructure.

4.6 Public Key Infrastructure (PKI)

The CSSO relies on secure communication channels. Therefore, all services need to be equipped with digital certificates. Therefore, we set up a small OpenSSL [Op09a] based PKI to enable us issuing certificates for federated EDIT servers whenever needed. Other organisational or legal constraints may require referring to approved commercial suppliers. Some larger institutions may also dispose of a qualified certification authority. Alternatively, certificates may be requested at the community-driven certificate authority Cacert.org[8].

Within this chapter, a brief overview of the basic components of the SAML framework and its integration into the CSSO security infrastructure has been given. Furthermore, the cooperation of these components have been shortly described. The next chapter will describe, how these components have been adopted to the application scenario requirements shown in chapter 2.

5 Profiles

Considering the application scenarios described in chapter 2, this section presents a general description on how these scenarios have been integrated into the CSSO security infrastructure. Therefore, the next sections will elaborate profiles according to the given application scenario name. Particularly, the web application profile comprehends a general overview of the SAML integration based on Shibboleth. Both subsequent profiles will just discuss the differences, options or necessary additions to the web application profile.

5.1 Web application

Web applications for the EDIT Cybertaxonomy platform were mainly developed based on the content management system Drupal [Bu09]. In the early stages, we adopted an existing Drupal authentication module to our needs. Meanwhile, the Drupal community has released a dedicated Shibboleth authentication module[9].

[8] http://www.cacert.org/
[9] http://drupal.org/project/shib_auth

The module is not restricted to be used with Shibboleth and fits any basic requirements regarding integration (mapping) of SAML attributes with the internal structures of Drupal's authentication and access control system. For instance, the EDIT web applications BDTracker, ExpertsDB and ATBI services are developed with Drupal.

Recently, we evaluated the upcoming Spring Security SAML module, which will enable us to integrate Spring based applications like the EDIT Data Model Portal as well. The module integrates well with the Spring Security Framework. So, this may exemplify the envisioned step-by-step integration process of web applications into EDIT's CSSO security infrastructure.

Most other web applications may simply be integrated by overlaying the web server default environment variable for login ids REMOTE_USER with the string value of the eduPersonPrincipalName attribute. For instance, we integrated some of the web applications denoted under the term EDIT developer tools, namely Trac and Subversion. Since, for both applications no modules or add-ons supporting the integration of SAML attributes are known, federated login based e.g. on group information are obsolete. So, only user based login can be offered.

5.2 Web service

Like web applications, web services are usually running on web servers. So, on server side, the same options as to protect web applications apply for web services. Since, web services may access the web servers' environment variables transmitted by their web service container, they can evaluate these variables for authentication and authorisation.

On the other hand, web service protocols are usually unaware of SAML. Therefore, intermediate components must be introduced to enable web service clients for single sign-on. At the beginning of the EDIT project, the idea of a Shibboleth Proxy component developed, hiding the Shibboleth authentication procedures from client software. This idea will be discussed in section 5.3. Meanwhile, SAML also profiles REST based web services. OpenSSO supports now both kind of web services and provides respective client APIs for web service developers. Shibboleth scheduled web service support to be released for the next version. Currently, we have no SSO enabled EDIT web services implemented.

5.3 Desktop application

Generally, common application software is not designed to operate with SAML SSO frameworks like Shibboleth. As networking in the CSSO infrastructure has been limited to the HTTP-protocol, it is technically possible to enable desktop applications for SSO. Therefore, the following components have been developed in order to integrate desktop applications

- CSSO Shibboleth Proxy
- CSSO Application Programming Interface (API)

The aim of the Shibboleth Proxy is to provide an intermediate filter to the SAML authentication exchange protocol. This filter enables SAML unaware applications to connect automatically and make use of http-based service providers protected by the CSSO security infrastructure (e.g. Shibboleth SP). Shibboleth Proxy provides seamless access to CSSO protected SPs by filtering out any http-protocol messages required for authentication (see Figure 3).

CSSO-API results from the Shibboleth Proxy implementation and disburdens CSSO integration for client application developers. According to the EDIT guidelines, the Shibboleth Proxy has been developed in Java for compatibility reasons. So, CSSO-API may applied directly to EDIT platform software component. Currently, only basic functions like the initialisation of the CSSO connection (e.g. authentication) and the transmission of HTTP-requests to SPs is supported. Additional functionality will be added on request of Cybertaxonomy platform developers.

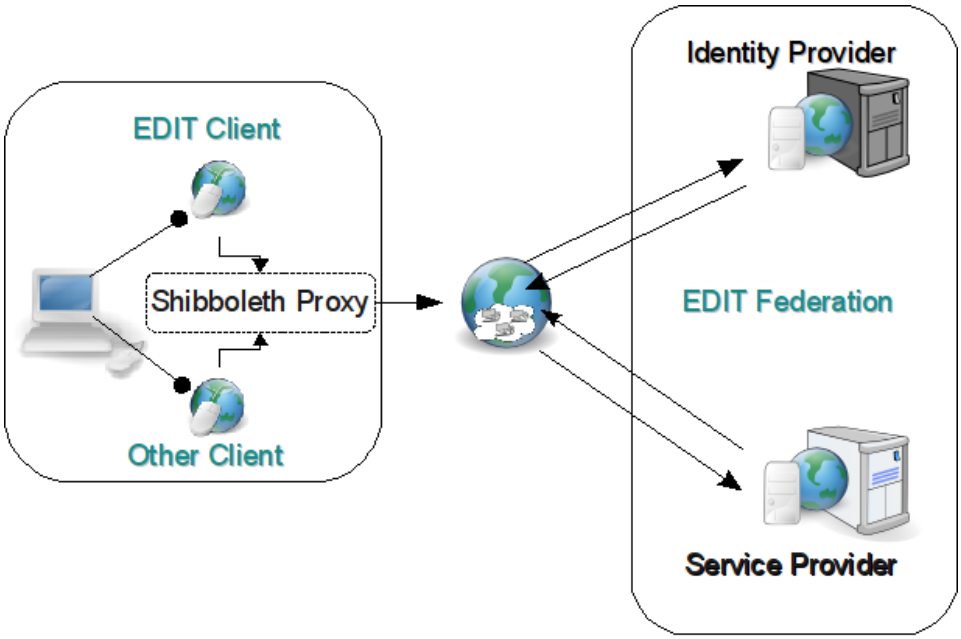

Figure 3: Shibboleth Proxy data flow

Instead of delivering these messages to the respective desktop application, Shibboleth Proxy proceeds the authentication procedure in favour of the user running the desktop application. Therefore, Shibboleth Proxy must be configured with the credentials (e.g. user id and password) of the user before running it. Finally, any http compatible desktop applications will run without any software modifications against CSSO protected SPs. The only requirement for desktop applications is to support https proxies.

Finally, both components can not only be used for client applications, but also to provide client functionality to web services (see chapter 5.2).

6 Conclusion and outlook

In general, the main benefit from single sign-on systems is on reducing administrative maintenance costs for service providers and minimising the number of passwords to be remembered by users. Simultaneously, these main advantages represent the main threats as well by creating single point of failures to both ends. Besides administration, the central positioning of IdPs and their user registries turns out to present valuable targets for potential attacks. On users' side, a stolen or weak password permits attackers to access a wide range of services. However, these threats can be compensated by a higher accuracy level towards a reduced number of services to be maintained by administrators. Also, an educational advertising provided, the minimized number of credentials to be remembered may help to increase the quality and protection level of user credentials significantly.

While, at the beginning of the EDIT project, Shibboleth was the only reasonable open source based SAML implementation, nowadays the situation has improved significantly. More and more implementations, tools or application modules are coming up helping to reduce the complexity and work load setting up the EDIT federation. This will not mean that the SSO topic becomes less complex through these tools. But, complexity appears to become manageable and interested institutions or service providers can revert to some experience and support regarding the integration into such a SSO infrastructure. This is an important point, particularly in a research field like biodiversity, where expertise in computer science usually is sparsely widespread.

Despite the progressing development of tools and a certain amount of publicity evoked by the installation of some national authentication infrastructures (e.g. UK), it is not an easy job to convince and motivate people joining into the EDIT federation. Particularly, in consideration of the species richness in underdeveloped regions of the world, it is important to gain hands-on experience from the integration of a broad base of leading edge biodiversity research institutions. Not least, this task must be accompanied by a certain level of inter-institutional political activities and goodwill with regard to the constitution of a common future federation for biodiversity.

So, our planning for the near future is two-folded. While focussing on the enlargement of the EDIT federation by further identity sources or service providers, this aim comes along with a targeted diversification of integrable software platforms into the CSSO security infrastructure. By these activities, institutions and service providers shall be motivated to integrate their own applications or user registries and thus, build up the EDIT federation in a step-by-step process from scratch.

7 References

[Bu09] Buytaert, Dries: Drupal. Online available at http://drupal.org/. Last visited 24 April 2009.

[Co09] CollabNet: OpenSSO. Online available at https://opensso.dev.java.net/. Last visited 24 April 2009.

[Ed09] European Distributed Institute of Taxonomy: EDIT - European Distributed Institute of Taxonomy. Online available at http://www.e-taxonomy.eu/. Last visited 24 April 2009.

[Eu95] European Parliament and Council: Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data. InEuropean Parliament and of the Council (Ed.): *Official Journal L*, 281, pp. 31-50, 1995.

[Fe08] UNINETT: simpleSAMLphp. Online available at http://rnd.feide.no/simplesamlphp. Last visited 24 April 2009.

[Fi00] Fielding, Thomas Roy: Architectural Styles and the Design of Network-based Software Architectures. University of California, Irvine.2000.

[In09] Internet2: Shibboleth®. Online available at http://shibboleth.internet2.edu/. Last visited 24 April 2009.

[In09a] Internet2: Home - OpenSAML - Internet2 Wiki. Online available at http://www.opensaml.org/. Last visited 24 April 2009.

[In09b] Internet2: Grouper. Online available at http://middleware.internet2.edu/dir/groups/grouper/. Last visited 24 April 2009.

[In09c] Internet2: uApprove. Online available at http://www.switch.ch/aai/support/tools/uApprove.html. Last visited 24 April 2009.

[Ja09] Jasig: CAS - Jasig Community. Online available at http://www.jasig.org/cas. Last visited 24 April 2009.

[Li09a] Liong, Bruce: Shibboleth Attribute Release Policy Editor (ShARPE). Online available at http://www.federation.org.au/twiki/bin/view/Federation/ShARPE. Last visited 24 April 2009.

[Mi09] Microsoft: Windows Live ID - Simplify your sign in. Online available at https://accountservices.passport.net/ppnetworkhome.srf?lc=1033. Last visited 24 April 2009.

[MIT09] MIT Kerberos: Kerberos: The Network Authentication Protocol. Online available at http://web.mit.edu/kerberos/www/. Last visited 24 April 2009.

[Oa09] OASIS: OASIS Security Services (SAML) TC. Online available at http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security. Last visited 24 April 2009.

[Op09] OpenID Foundation: OpenID. Online available at http://openid.net/. Last visited 24 April 2009.

[Op09a] The OpenSSL Project: Welcome to the OpenSSL Project. Online available at http://www.openssl.org/. Last visited 24 April 2009.

[Sw09] SWITCH: Easy Demo. Online available at http://switch.ch/aai/demo/easy.html. Last visited 24 April 2009.

[Sw09a] SWITCH: Group Management Tool. Online available at http://www.switch.ch/aai/support/tools/gmt.html. Last visited 24 April 2009.