eunis dublin 2011
EUROPEAN UNIVERSITY INFORMATION SYSTEMS
eunis
International Congress
Dublin, Ireland,
Wed 15th - Fri 17th June 2011
Maintaining a sustainable Future
for IT in Higher Education

Thursday 16th June 2011

Time: 11:30 - 12:00

An Identity Management Web Server for Privacy Preserving Course Authorization in Federated eLearning

1. ABSTRACT

Using learning management systems (LMS) across higher education institutions' (HEIs) borders has become an increasingly important topic over the past few years. On the one hand, it allows us, the lecturers, to share the results of our time-consuming labor of creating high-quality E-Learning material with a larger audience. On the other hand, students get the chance of participating in courses that are not offered by their home university, so they can study subjects of their interest and can also obtain corresponding certificates of achievement, which they can submit to their local examination office to advance their studies.

Some years ago, the lack of interoperability between LMS and national authentication and authorization infrastructures (AAIs), which are typically based on software like Shibboleth, was a major obstacle to get inter-HEI LMS scenarios working. The problem did not concern the purely technical aspects, because de-facto standards like Shibboleth were already supported by the major LMS vendors. Instead, it was rather regarding the data about students and courses that needed to be exchanged between each two involved HEIs, i.e., the student's home university and the university that offered the E-Learning course: Each university typically used its own denominators for study courses, had different LDAP data schemes, and different class enrolment software in place. To solve this problem in Germany, a major standardization effort, which was led by the national research and education network (NREN), DFN, resulted in a common syntax and semantics for students' personally identifiable information (PII) for the purpose of inter-organizational LMS usage. We reported about this milestone at EUNIS 2009 (Hommel, 2009).

Meanwhile, implementation projects at the Virtual University of Bavaria (VUB), which consists of more than 35 HEIs in Southern Germany, have identified privacy and stability issues that were not fully addressed by those standardization efforts. VUB operates a central E-Learning portal at which

students from any of the member HEIs can register. Using this portal, students can then choose from a large number of E-Learning courses, which fit into their curriculum and are offered by one of the member HEIs. In this context, VUB is a broker for de-centrally operated LMSs, but not a central LMS provider itself. However, VUB keeps track of which student has registered for which course, and a student must only be allowed to use the LMS that provides such a course after successfully enrolling through the VUB portal. Thus, when a student attempts to login, the LMS must check whether the user is enrolled in the course. Based on the AAI federation model, the LMS is a Shibboleth service provider (SP), which requests this information from the Shibboleth identity provider (IDP), i.e., the student's home university. In turn, the IDP retrieves the relevant information from two data sources: (1) the HEI's local identity management system, e.g., an LDAP server, in which basic information about all its users is stored (e.g., username and email address), and (2) VUB's course management database, from which the course enrolment information can be retrieved. While this workflow may be considered adequate for students using VUB courses, limitations of the Shibboleth software resulted in data queries made to the VUB course database whenever any user logged into the IDP, e.g., even if a HEI's employee wanted to use a non-VUB-service via Shibboleth. On the one hand, this has led to major privacy concerns (e.g., VUB may create profiles of users that are not even aware of VUB); on the other hand, it also could cause a decline in stability because any temporary unavailability of the VUB database would result in an error message for the user – again even if the service he wanted to use was not related to VUB.

In this presentation, we outline solution alternatives that were analyzed with organizational and technical aspects in mind. We then discuss the finally chosen solution in more detail: It uses a simple web services based protocol to include VUB course enrolment data in each HEI's local identity management system; it has also been implemented for several platforms. We conclude with a summary of the results so far and a discussion of our road ahead.

2. VIRTUAL UNIVERSITY OF BAVARIA: AN EXAMPLE OF FEDERATED E-LEARNING

VUB was founded in the year 2000 and had a steadily increasing number of students from the beginning. In 2010, over 26.700 Bavarian students used VUB to enroll in the more than 400 E-Learning courses that were offered, resulting in over 66.400 total course registrations. In average, considering all the courses, 60% of all students requested certificates of achievement that are honored by the local examination offices based on contractual agreements between VUB and the participating HEIs.

VUB always had the role of a broker: Instead of creating E-Learning content or operating LMSs itself, VUB offers a course management system. Lecturers from the participating HEIs can register their E-Learning courses, and students can enroll in these courses after signing up for the central VUB portal. VUB then ensures that accounts are created for the students in the LMSs that provide the courses they signed up for, and takes care of certificate achievements as well as statistics. To automate as much of these workflows as possible, VUB started to provide APIs and user import tools for popular LMS products quite early. However, because those interfaces were not based on any standards (which did not exist at this point in time either), integrating them into commercial LMS products turned out to be difficult, and several HEIs running open source LMS products did not want to spend much time on modifying their LMS installations either, especially with the risk of having to repeat these efforts with major LMS software revisions over and over again.

For this reason, VUB had a strong interest in the standardization effort that led to the so-called DFN-AAI E-Learning profile, i.e., a specification of the syntax and semantics, which HEIs and SPs shall adhere to whenever student PII is being exchanged between IDPs and SPs in the German research federation DFN-AAI. Most DFN-AAI participants use the de-facto standard software Shibboleth to run their IDPs and SPs, and so a couple of Shibboleth-based prototype implementations of VUB LMSs using the DFN-AAI E-Learning profiles were carried out and presented their results at the end of 2009.

The basic *logical* workflow that has to be realized is shown in Figure 1. We have to distinguish between the following four involved parties:

1. The user, i.e., a student interested in one or more of the courses that are offered via VUB. Using the VUB portal and the LMSs that offer the relevant courses should be as comfortable as possible, i.e., Shibboleth's single sign-on capabilities and the features to automatically setup user accounts in the target systems should be leveraged. The user must sign up at the VUB portal before he can enroll in courses (step 1).
2. The IDP, i.e., the user's home university, which is responsible for authenticating the user (e.g., by means of a web interface, which requests username and password) and providing the relevant PII based on the DFN-AAI E-Learning profile (step 2).
3. The VUB portal, which students can use to find suitable courses and to enroll therein. In the background, the VUB portal stores this course management information in a database. Afterwards, the student is guided to the appropriate LMS (steps 3-5).
4. The LMS, which again is operated by a HEI (in general not the user's home university, although this is a valid VUB use case). It shall receive all the required user information via Shibboleth, i.e., neither the user should have to sign up manually there, nor should it be required that LMS operators or lecturers import any user lists manually.

On the *technical* level, this workflow needs to be broken down into two separate groups of Shibboleth communication steps as shown in Figure 2. They reflect the following two use cases:

1. User logs into VUB portal: The VUB portal is a Shibboleth SP and the Shibboleth IDP sends the relevant user PII (e.g., name, study course, and email address) after successfully authenticating the user (using his campus credentials). Thus, VUB can update the user record it stores about the student at each login, and the student does not need to remember a separate username/password combination for the VUB portal as it was the case in the pre-Shibboleth era (steps 1-3). The user can then enroll in a suitable course and is forwarded to the appropriate LMS (steps 4-5).

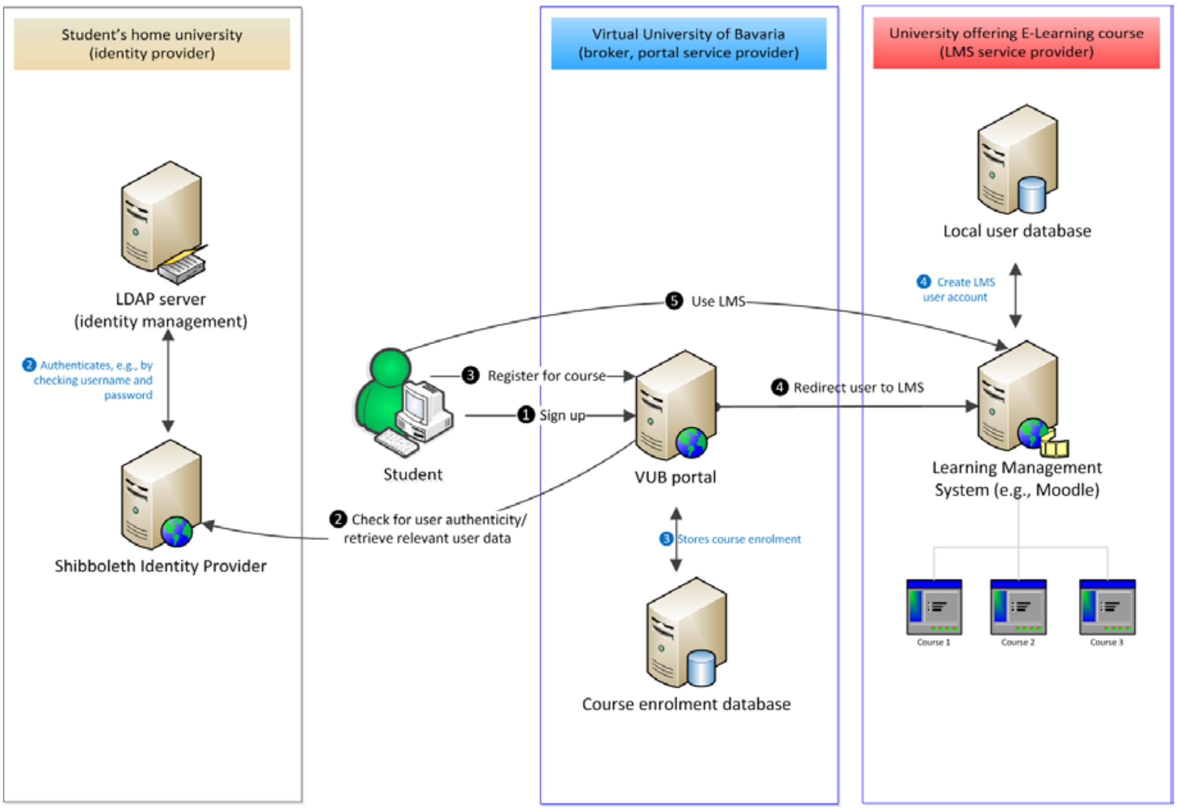
Student's home university (identity provider)
Virtual University of Bavaria (broker, portal service provider)
University offering E-Learning course (LMS service provider)
LDAP server (identity management)
2 Authenticates, e.g., by checking username and password
Shibboleth Identity Provider
Student
5 Use LMS
3 Register for course
1 Sign up
2 Check for user authenticity/ retrieve relevant user data
VUB portal
3 Stores course enrolment
Course enrolment database
4 Redirect user to LMS
Local user database
4 Create LMS user account
Learning Management System (e.g., Moodle)
Course 1
Course 2
Course 3

Figure 1: Logical workflow to enroll and participate in VUB e-learning courses

2. User logs into LMS: In this case, the LMS is a Shibboleth SP that requests the student's authentication and relevant PII from the IDP. The important aspect here is that the IDP must also send the information whether the student has enrolled for the course provided by the LMS. Because this information is stored in VUB's database, and thus not available locally at the IDP, the prototype implementations used the Shibboleth IDP configuration depicted in Figure 2, i.e., the IDP had two data sources: (a) the local HEI identity management system, typically an LDAP server, and (b) VUB's course enrolment database. The LMS then checks whether the user is authorized to access the offered course based on this data and grants access accordingly (steps 6-8).

Because switching from a previously self-developed LMS interface solution to a Shibboleth-based one with more than 35 organizations and over 100 LMS instances involved is a complex project, there was a migration project planning phase in mid-2010. In this phase, the prototype installations, which were working fine and just as expected with a couple of test users, were investigated in more depth to provide a technical specifications document and an estimation of the effort required at each IDP and LMS. Only then one major technical drawback of the solution architecture discussed above was discovered, which has serious privacy and organizational consequences:

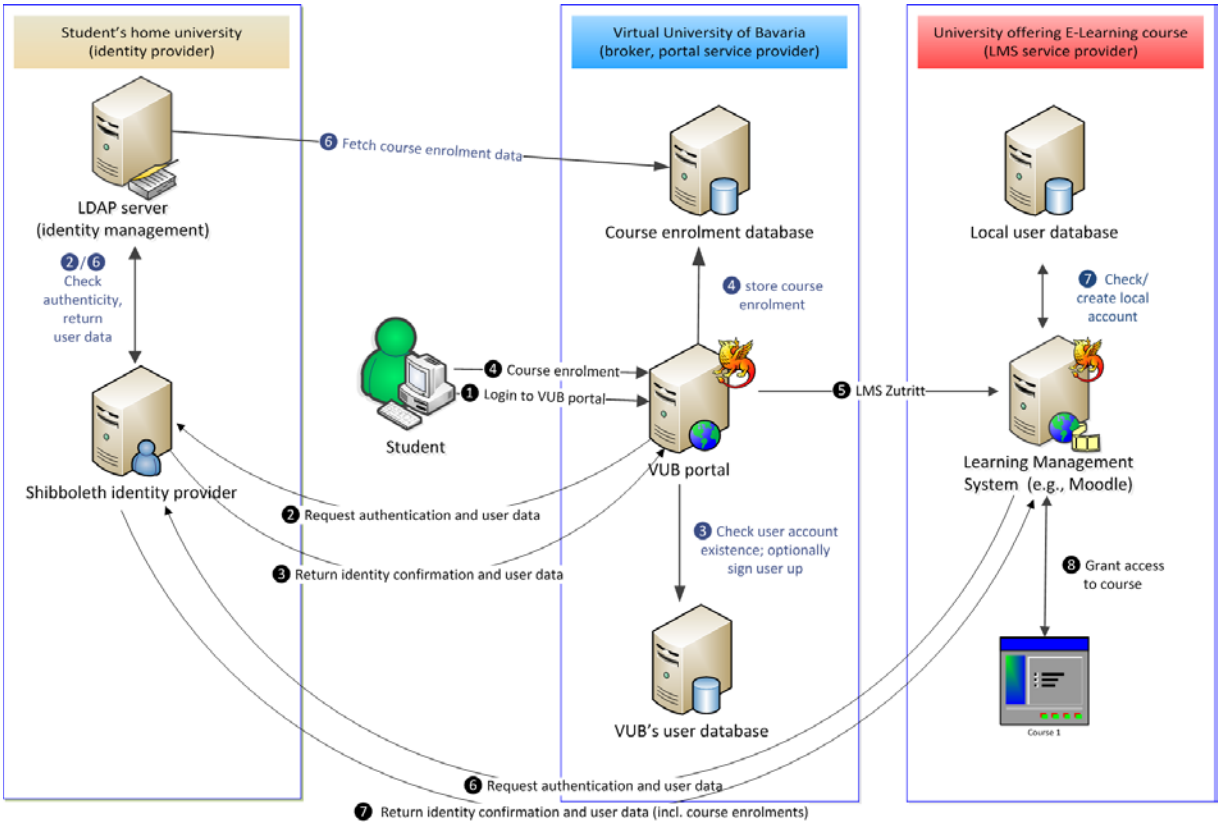
Student's home university
(identity provider)
Virtual University of Bavaria
(broker, portal service provider)
University offering E-Learning course
(LMS service provider)
LDAP server
(identity management)
6 Fetch course enrolment data
Course enrolment database
Local user database
2/6 Check authenticity, return user data
4 store course enrolment
7 Check/ create local account
4 Course enrolment
1 Login to VUB portal
5 LMS Zutritt
Student
VUB portal
Learning Management System (e.g., Moodle)
Shibboleth identity provider
2 Request authentication and user data
3 Return identity confirmation and user data
3 Check user account existence; optionally sign user up
8 Grant access to course
VUB's user database
Course 1
6 Request authentication and user data
7 Return identity confirmation and user data (incl. course enrolments)

Figure 2: Technical realization using Shibboleth for the Identity Provider and the Service Providers

Whenever a user authenticates at its home university's IDP, VUB's database is searched for any VUB courses this user has registered for. While this is the obviously desired behavior for students using a VUB-related LMS or the VUB portal itself, those database queries are also being performed if any non-student user wants to login to a completely unrelated SP, e.g., if a staff member of the university wants to use any of the library services via Shibboleth.

Given the current software version of the Shibboleth IDP, there is no configuration solution to query the VUB database only if the user currently attempts to login to any of the VUB-related services. This leads to the following two issues:

1. Privacy concerns: From an organizational and juristic perspective, VUB is a legally independent institution, and thus not a part of the user's home university. Thus, whenever the user properly authenticates, this third party is being informed about the successful login of a user who might not even know that VUB exists. Because at most universities' IDPs, the user's email address is used as login name, VUB gets to know each user's email address (this login name is also used to search for courses the user has signed up for in VUB's database) and a timestamp of the last successful login. Consequently, PII that could be abused for profiling users would be leaked to VUB without the user's consent, and thus this configuration setup must be considered very critical w.r.t. privacy.
2. Stability concerns: As the VUB database, which is one of the two data sources for each IDP, is operated centrally at VUB, there have been discussions about its availability and performance. For example, network failures between the HEIs and the data center where the VUB database is running, power outages, and scheduled maintenance would make the VUB database temporarily unavailable for the IDP, which in turn leads to timeouts and error messages displayed to the user (although they could be suppressed, they are useful in the case of, e.g., a local system failure).

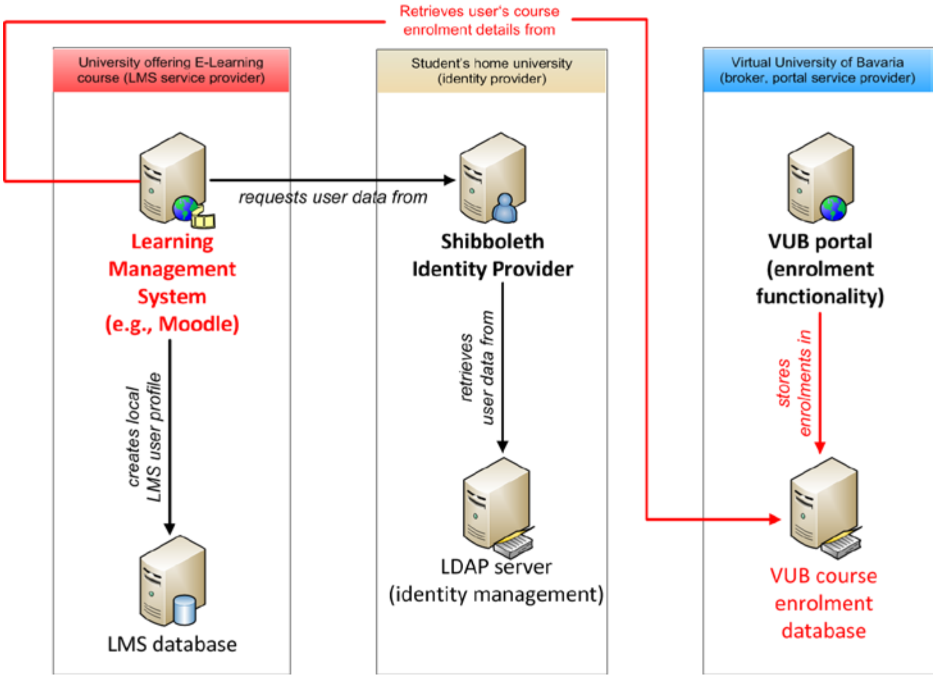
Retrieves user's course enrolment details from
University offering E-Learning course (LMS service provider)
Student's home university (identity provider)
Virtual University of Bavaria (broker, portal service provider)
requests user data from
Learning Management System (e.g., Moodle)
Shibboleth Identity Provider
VUB portal (enrolment functionality)
creates local LMS user profile
retrieves user data from
stores enrolments in
LMS database
LDAP server (identity management)
VUB course enrolment database

Figure 3: Alternative workflow based on modified LMS instances

As shown in Figure 3, these issues could be avoided by delegating the VUB database access from the IDP to the LMS: In this case, there would be no privacy violations because the VUB database is only accessed when the user actually logs into a VUB-related LMS that needs to check for the course registration anyway. Also, if the VUB database was temporarily unavailable, the LMS, which typically has a local user database of its own, could re-use the locally stored, last known course registration status as a workaround. In this case, only the first-time LMS login would fail if the VUB database was unavailable at the same moment.

However, while such an adaption of the LMS would technically be possible for several (open source) products, such as Moodle, the combination of (a) using Shibboleth to authenticate the user and retrieve the user's PII, and (b) using an additional external database to check for course registrations, might not be feasible for all LMS suites and is generally considered to require too much effort to be implemented for each LMS instance. Thus, an alternative solution was implemented as described in the next section.

3. WEB-SERVICE-BASED SYNCHRONIZATION OF COURSE ENROLMENT DATA WITH UNIVERSITY IDENTITY MANAGEMENT SYSTEMS

The solution presented here has been designed in order to realize the rationale behind the DFN-AAI E-Learning profile, i.e., to have the home university IDP deliver all required user data to the external LMS without any further necessary interactions between the LMS and any third party data sources, such as the VUB course enrolment database. It also solves the privacy and stability issues discussed in Section 2 by fetching the relevant course enrolment database entries from VUB and storing them in the local identity management system: The privacy concerns are mitigated because no third party or external database is involved anymore during the actual authentication of a user. Also, the stability concerns are no longer valid because the availability of the course enrolment data is the same as the availability of the other PII and authentication data, e.g., based on the availability of the local LDAP server.

Figure 4 shows the basic workflow: Whenever a student enrolls in a course via the VUB portal, the student's home university is being notified about this change by means of a web service call. Course cancelations are handled similarly, e.g., if the student pulls out on his own or is excluded from the course for other reasons. The design of this web service focused on the following core aspects:

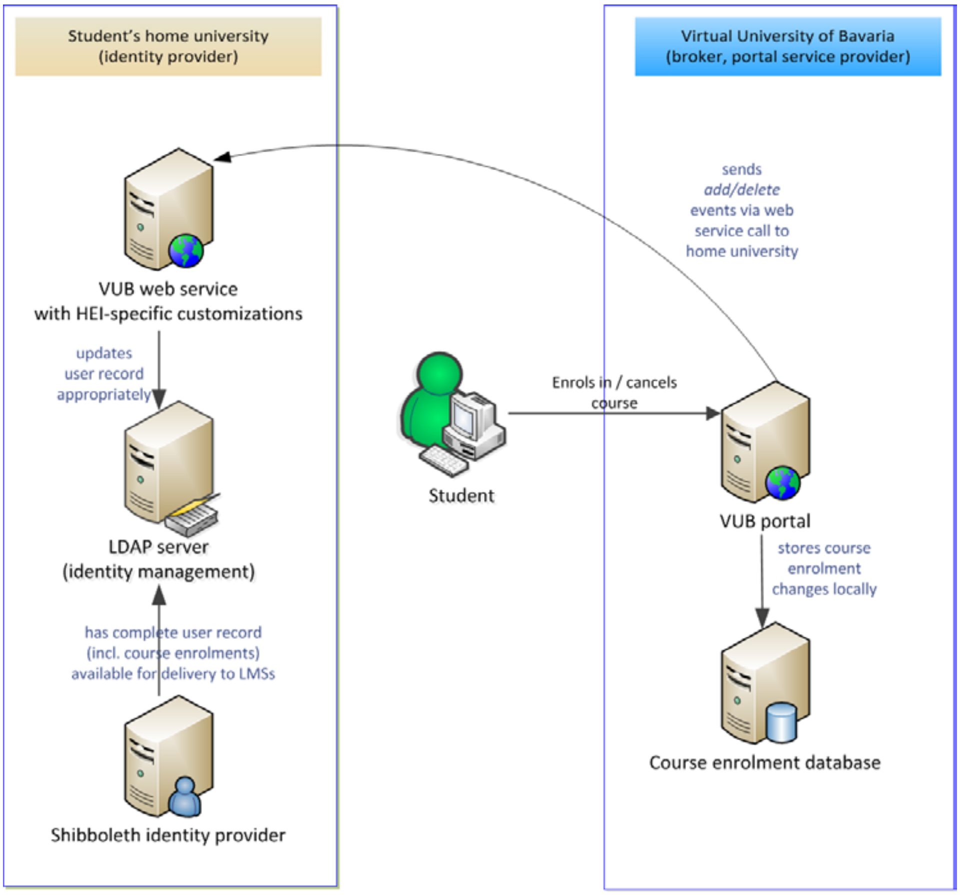

Figure 4: Web service based synchronization of course enrolments

- **Functionality**: *add*- and *delete*-events may be sent for each single course enrolment per user to foster near-real-time event propagation, but each web service call can consist of an arbitrary number of such events. This enables a full re-synchronization of the home university's identity management system, which is necessary, e.g., for initial deployment and disaster recovery (please note that a local backup of the identity management system is not sufficient in the latter case because course enrolments at VUB might change while the home university's identity management system is being restored from a backup). The web service, which is operated by the home university, must acknowledge the successful update of the local identity management system, so VUB is also informed about potential delays, e.g., due to an LDAP server's temporary unavailability. This would mean that the student cannot start to use the course he just signed up for immediately, and so the user can be informed accordingly.
- **Security**: To ensure the confidentiality, integrity, and authenticity of the transferred course enrolment event messages, standard web services security mechanisms are applied, such as the use of HTTPS with certificates that are created within the German NREN's DFN-PKI. Additional site-specific restrictions including packet filtering firewalls and request counter thresholds can be applied to further minimize the risks of denial-of-service and similar attacks by ensuring that only VUB can connect to each HEI's web service.
- **Platform independency**: Each HEI needs its own instance of this web service. Given the variety of operating system preferences and identity management system products at the more than 35 HEIs, the ability to run the web service reliably on multiple platforms was considered mission-critical (the alternative approach to provide a pre-configured virtual machine image was dismissed due to the expected fussiness of updating local customizations when new web service software versions are released). As a consequence, the current requirements are that the web service runs on both the Linux and the Microsoft Windows Server operating systems (including the predominant web server products, Apache and Microsoft IIS), and supports LDAP servers as well as relational database management systems (via SQL) as back-ends.
- **Managing the metadata**: Manually managing IP address / DNS name / port number changes for the web service endpoints of over 35 HEIs would be tedious over the years. Especially VUB itself would have to manually respond to any change in the setup of the web service at any of the HEIs. Thus, a feedback channel is necessary, which HEIs can use to inform VUB about changes in their web service endpoint configuration.

Currently there are two prototype implementations available. First, a Perl-based implementation that has to be run as a CGI script by a web server, such as Apache or IIS, which is very light-weight, includes a simple Perl-based LDAP back-end and can easily be customized by each HEI. Second, a stand-alone Java implementation that is based on the Axis2 framework; it works without additionally requiring a separate web server and makes use of Java's SQL and LDAP APIs, but customizations made to the code require a local development environment to build the executable Java classes.

The Java prototype relies on a SOAP-based, WSDL-specified interface. The messages follow the request-response-paradigm and contain explicit function calls defined in the WSDL document. In this setting, the entire web service layer, including the WSDL document itself, can be generated automatically by the Axis2 framework out of the plain Java methods (plain old java objects, POJOs). Furthermore, Axis2 allows for the automatic generation of client (caller) stubs.

Further benefits of this solution are:

- The implementation operated is always consistent with the documentation (WSDL).
- The semantics on the operational level is clearer: Which values must be *added* or *deleted* from which entry. Non-mentioned values or entries are not affected unless a *removeAll* operation request is sent.

The Perl CGI script expects simpler, yet proprietary XML documents via HTTPS POST, containing enrolments and de-registrations as delivered by the course administration software on VUB's side (which is based on FlexNow, a software suite that is also used by many other German universities). Parsing this XML document and calling the appropriate actions must be coded explicitly within the web service script. The lack of automatic generation of WSDL and client stubs is, however, negligible, since VUB will be the only client (caller) of the web service, whereas the very easy deployment (simply one single CGI script to be run on any existing web server) is an important benefit for each HEI that is offering VUB courses.

Both implementations are platform-independent, but require a local installation of Perl or the Java Runtime Environment, respectively. To offer further customization possibilities, upcoming versions of these implementations should support the execution of external scripts for each type of event (such as adding a course enrolment to a student’s LDAP object). The rationale behind this extension is that many HEIs already have custom scripts (written for Microsoft PowerShell, Perl, Python, and other scripting languages) in place that can be leveraged, and that the dependencies of site-specific script customizations (e.g., logging changes to an audit database, sending local notification emails, etc.) on the web service interface and its reference implementation can be minimized.

Course enrolments are stored in the *eduPersonEntitlement* attribute of user records, e.g., in local LDAP servers, and delivered by the IDP to the VUB-related LMS. The syntax chosen for VUB course enrolments features a unique prefix that allows the IDP to filter other *eduPersonEntitlement* values, which are not relevant for a LMS, such as *common-lib-terms*, to further improve the privacy protection by avoiding the leakage of too much information about the users.

4. LESSONS LEARNED AND THE ROAD AHEAD

Although our initial Shibboleth prototype was quite easy to set up and provided the expected functionality, a closer investigation of the communication steps below the surface revealed that the overall architecture and configuration had led to an unsatisfying solution that raised privacy and stability concerns. This made it obvious to us that the design of inter-organizational services needs to be carefully planned and checked. Given the limited configuration options of the state-of-the-art software involved, an additional course enrolment synchronization web service became necessary. Although kept as simple as possible, the requirement to run it independent of the used operating system, identity management back-end, and local scripting language makes this work-around a non-trivial challenge.

We are currently in the process of customizing and deploying the implemented web service for selected VUB HEIs in order to establish pilot operations by Q3 of 2011. Based on experiences made with the site-specific efforts that are necessary for customizations, and depending on pilot lecturers’ and students' feedback, a rollout plan for the more than 35 involved HEIs will be created. While our prototypes are currently tailored for VUB, the solution architecture and code framework can be re-used for other federated e-learning scenarios, in which course enrolment is handled by an external third party. After successfully finishing of the pilot phase, the resulting version 1.0 of the source code, the related API documentation, and a detailed documentation of the architecture and workflows will be published.

5. REFERENCES

Hommel, W. (2009). E-Learning in Shibboleth-based federations: The design rationale behind the German DFN-AAI E-Learning Profile. *15th congress on European University Information Systems (EUNIS 2009)*