



29-04-2011

Deliverable DJ3.2.1,2: Identity Federations Yearly Deliverable



Deliverable DJ3.2.1,2 v1.0

Contractual Date: 28-02-2011

Actual Date: 29-04-2011

Grant Agreement No.: 238875

Activity: JRA3

Task Item: T2

Nature of Deliverable: R

Dissemination Level: PU

Lead Partner: NORDUNET/UNINETT

Document Code: GN3-11-001

Authors: Andreas Solberg, Miroslav Milinovic, Leif Johansson, Kristof Bajnok, David Simonsen, Remco Poortinga, Lukas Hämmerle, Maja Gorecka-Wolniewicz, Torsten Kersting, Dubravko Voncina, Josh Howlett, Licia Florio, Shannon Milsom

Abstract

This deliverable covers GN3 JRA3 T2 Identity Federations continued work to research and develop ways for different identity federations to coexist and interact. Activities included developing tools and components for identity architectures and implementing protocols.

Table of Contents

Executive Summary	1
1 Introduction	3
2 Federation Lab	4
2.1 SAML Registry for SPs	5
2.2 Federation Lab OpenIdP	5
2.3 Automated Testing Tool	6
2.4 SAML 2.0 Debugger	8
2.5 Firefox SAML Debugger Plug-in	9
2.6 Metadata Editor in Javascript	9
3 Federation Harmonisation	12
3.1 De-provisioning	12
3.1.1 Privacy	13
3.1.2 Common Requirements	13
3.1.3 Patterns	13
3.1.4 Further work on De-provisioning	13
3.2 Single Log-Out	13
3.2.1 Current Implementations	14
3.2.2 Major Logout Challenges	14
3.2.3 Logout Deployment in Federations	16
4 Virtual Organisations	18
4.1 Attribute Queries in simpleSAMLphp and VO	18
4.1.1 Implementation	19
4.1.2 Privacy Considerations	19
4.2 VO Platform in Centralised Login Federation	19
4.3 Attribute Collector Based on Virtual IdP	21
5 Metadata Distribution and Cross-Federation Scalability	23
5.1 New Models for Distributed Metadata Aggregation	23
5.2 Metadata Aggregator Specification	24
5.3 Scalability of IdP Discovery: Location-based Discovery	25
5.3.1 Problem Statement	25

5.3.2	Functional Description	25
5.3.3	Future Work	27
5.4	Kantara ULX Scalability	27
6	Beyond WebSSO: Moonshot	29
7	Conclusions	30
	References	31
	Glossary	33

Table of Figures

Figure 2.1:	Example UI for registering an SP	7
Figure 2.2:	Example Test case report	8
Figure 2.3:	Debug log	8
Figure 2.4:	Example to decode SAML message	9
Figure 2.5:	Example of SAMLmetaJS editor	11
Figure 4.1:	VO implementation architecture	20
Figure 4.2:	Enabling cross-federation attribute exchange	21
Figure 5.1:	IdP selection window	26
Figure 5.2:	IdP selection window (zoom)	26
Figure 5.3:	Demonstration of PoC modifications to Kantara ULX	28

Executive Summary

Identity Federations are based upon the principle that a user's authentication is undertaken by the home organisation, referred to as Identity Provider (IdP), and that a resource, a Service Provider (SP), trusts what the home organisation states about the user. This model enables users of one domain to securely access data or systems of another domain seamlessly, without the need for redundant user administration.

The GN3 JRA3 Task 2, Identity Federations, aims to research different aspects to improve the inter-operability across federations in the Research and Education community and to support collaborative communities. It also aims to support new federation use-cases to expand the deployment of the federated framework. The Identity Federations Task looks ahead to solve some of the scalability and usability issues that arise when cross-federations succeed. GN3 Identity Federations works in close collaboration with GN3 eduGAIN.

This deliverable covers Identity Federation's continued work to achieve the goals mentioned above. The Task has identified several sub-tasks namely:

- Federation Lab
- Federation Harmonisation
- Virtual Organisations
- Metadata Distributions and Cross-Federation Scalability
- Beyond Web Single Sign-On with Moonshot
- User-centric identity

Following is a short summary of these activities in JRA3 T2.

During Y1 JRA3 T1 researched ways to support user-centric technologies (OpenId and CardSpace) within the existing SAML federations. An initial paper was produced with the intention to do more tests during Y2. However at the beginning of 2011, Microsoft announced they would not ship CardSpace any longer¹, which will in the longer-term mean the end of the technology. Due to this, and due to the fact that the research and education community has clearly chosen SAML federations, JRA3 T2 has decided not to pursue this work item any longer.

Main achievements regard the Federation Lab [[FEDLAB](#)], released for testing at the end of 2010. At this stage, feedback from users is important to further develop the work. The first release of the Federation Lab has been very focused on SPs. Future work will extend the audience to include IdPs and metadata validation services.

¹ <http://blogs.msdn.com/b/card/archive/2011/02/15/beyond-windows-cardspace.aspx>

The Federation Lab website (<https://fed-lab.org/>) contains the validation services, and it is being updated to act as a placeholder for some of the best practices documents produced within the Identity Federations Task. There are currently two documents available, the best practices on **de-provisioning** and a report containing **recommendations on single logout**.

The work performed on Virtual Organisations (VO) has been very useful and the results available so far are important for further work that will be carried out in Y3 and Y4. This work area is very complex and it will take years for the community to deploy a VO infrastructure that addresses all possible use-cases. Current plans are to focus on a limited information model, solving a selected number of real use-cases and making the technology available for developers.

The work on metadata distribution and cross-federation scalability has been fed into the eduGAIN project. JRA3 T2 will follow up aligning new requirements with revisions of the specifications. The next main challenge is to solve the scalability issue in the discovery service (see section 5.3). The current discovery service shows a poor user experience when the number of IdPs becomes large, which may become an obstacle to the deployment of eduGAIN (for example) or any other cross-federation activities. JRA3 T2 aims to solve this scalability issue and deliver a successful user experience. The results should be production-ready and available for SPs connecting to eduGAIN.

The GÉANT project is contributing effort towards the project *Moonshot* [[Moonshot](#)], an initiative to bring the benefits of Identity Federations to Internet protocols other than HTTP. The principal use-cases for this work are federated access to non-web applications, such as those used by High Performance Computing (HPC) facilities, Grid computing resources, email and instant messaging.

A working proof of concept (PoC) was demonstrated at the end of February 2011. It is anticipated that this code will be production quality by August 2011.

1 Introduction

This deliverable describes the work accomplished in Y2 by GN3 JRA3 Task 2, Identity Federations. It describes the main focus areas where significant progress has been made. These include:

- Federation Lab
- Federation Harmonisations
- Virtual Organisations
- Metadata Distributions and Cross-Federation Scalability
- Beyond Web Single Sign-On with Moonshot

The rest of this document outlines the result of work in each of these areas. The problem descriptions that were presented and discussed in detail in Y1 *Identity Federations* [\[DJ3.2.1.1\]](#) are not repeated in this document.

2 Federation Lab

The goal for Federation Lab [[FEDLAB](#)] is to provide an automated SAML (Security Assertion Markup Language) conformance test suite to be used as a Federation Lab for testing the interoperability of new components in SAML-based environments. Such a suite will enable federations to ensure interoperability with new components, without having disruptive effect on operational federations. Real-life testing between Service Providers (SP) and Identity Providers (IdP) raises several issues involving test-users and scalability. Therefore, alternative measures to ensure interoperability are crucial to offer a reliable user experience for end users.

Tools and components incorporated in the GN3 Federation Lab work include a SAML Registry for SPs, easy account creation, automated testing of SPs, a SAML debugging tool based on simpleSAMLphp and a JavaScript metadata editor for SAML XML Metadata.

The Federation Lab is being developed in close collaboration with the GÉANT eduGAIN [[eduGAIN](#)] project. The idea is that eduGAIN will directly benefit from some of the test and validation tools on Federation Lab for ensuring interoperability within eduGAIN.

The benefits of the Federation Lab suite are to guide providers into better understanding their systems, help them to test, debug, monitor and diagnose their systems, and lead providers to define more interoperable configurations for their systems.

The first public release of Identity Federations was made available for the public at the end of 2010. The Federation Lab is at <https://fed-lab.org>.

About the site setup

The Federation Lab website is being hosted on a virtual server by DFN. The site consists of a Wordpress content management system powered by *apache2* and *mysql*. The *wordpress* installation has been modified to allow for federated access in conjunction with simpleSAMLphp as a Service Provider (SP). The metadata of the site still needs to be fed into an official Metadata Service (MDS) provider for inter-federation access using eduGAIN when that service is officially in production. The site will then be configured to pull all metadata from that service on a regular basis (daily).

The initial content of the site provides various technical resources and tools for the Identity Federation task. The Federation Lab is a modular tool suite. As more work becomes ready for use by the public, it will be included in the Federation Lab. The following subsections describe the tools and the documents that are currently available on Federation Lab.

2.1 SAML Registry for SPs

One of the more important Federation Lab components will be the SAML Registry of Service Providers.

To enable some of the tools at Federation Lab for testers, such as the *Test Identity Providers*, it is necessary to register the Service Providers' metadata. One important aspect of Federation Lab is that a tester should only need to register an SP once, then a number of tools would be using the registered metadata.

The registry, therefore, has two interfaces:

- User interface, allowing testers to register one or more SP.
- Configuration interface towards the available tools at Federation Lab, providing each tool with a complete list of SPs that should be accepted.

Implementation of the user interface of the registry has led to two new independent libraries that may be re-used by others. A new javascript library named *SAMLmetaJS* is significantly improving the user interface by implementing a metadata parser and an editor. In addition, a new SimpleSAMLphp module named *metaedit2* allows users to login and register SPs and stores the metadata in an SQL backend. The *metaedit2* module makes use of the *SAMLmetaJS* for an improved user interface. *SAMLmetaJS* is described in more detail in section 2.6.

metaedit2 has been fed back to the simpleSAMLphp project and will be used as a basis for a new built-in metadata registry in SimpleSAMLphp. Federation Lab will contribute this module to the simpleSAMLphp project that will adopt and integrate it as part of the standard simpleSAMLphp distribution. Once integrated in the simpleSAMLphp, a wider community will be able to expand it. The simpleSAMLphp project expects this to be completed before the end of 2011.

2.2 Federation Lab OpenIdP

OpenIdP [[OpenIdP](#)] is available within the Federation Lab to allow for quick and easy account creation. These accounts can be used to test various SPs within eduGAIN, but out of one's own national federation. The OpenIdP has been set up using apache2 and simpleSAMLphp with the selfregister module and openidap. The OpenIdP is at <https://sisatestidp.dfn.de/simplesaml/module.php/selfregister/index.php>.

The Federation Lab OpenIdP is one of the tools that automatically becomes available for SPs registered in the Federation Lab Service Provider Registry.

The plan for Y3 of Identity Federations is to include a number of alternative OpenIdPs to be available for registered SPs, including:

- ProtectNetwork, <http://www.protectnetwork.org/>
- TestShib, <http://www.testshib.org/testshib-two/index.jsp>

Multiple OpenIdPs running various software and configurations would make it possible for an SP to ensure that the configured software is compatible with a wider range of IdP configurations.

2.3 Automated Testing Tool

As described in [\[DJ3.2.1.1\]](#), SAML 2.0 specifications are flexible and allows for various alternatives, which in turn may result in interoperability issues in large scale Identity Federations, such as GÉANT eduGAIN.

An approach to solving this issue is the saml2int profile [\[saml2int\]](#). The interest for saml2int is large and is gaining adoption. However, one of the main problems is that it is really complex to validate a provider against SAML 2.0 specifications. An operator may think that his/her software is configured and implemented to behave in a specific way, but testing all the border cases requires deep insight and a large amount of resource.

The Automated SAML 2.0 Testing Tool for Service Providers on the Federation Lab is a fully automated testing tool that can be configured to run through a series of tests and get a report of system misconfigurations.

An operational proof-of-concept (PoC) of this tool has already been implemented. The PoC has been very useful in the development of the simpleSAMLphp software, even before release of an official stable version. Version 1.7 of simpleSAMLphp, released in December 2010, fixed a large number of SAML compatibility issues based on output from the testing tool. A commercial company made contact to validate the behaviour of their SAML software. A number of issues were discovered and they are in the process of being fixed.

The procedure for an operator to follow to use the testing tool is simple. First the operator must configure the testing tool as a trusted Identity Provider (IdP) for the systems to be able to communicate. Next, the SP must be configured with a web page that shows the attributes for a user if it accepted the assertion sent from the IdP (or from the testing tool pretending to be an IdP). Figure 2.1 shows the user interface for registering an SP.

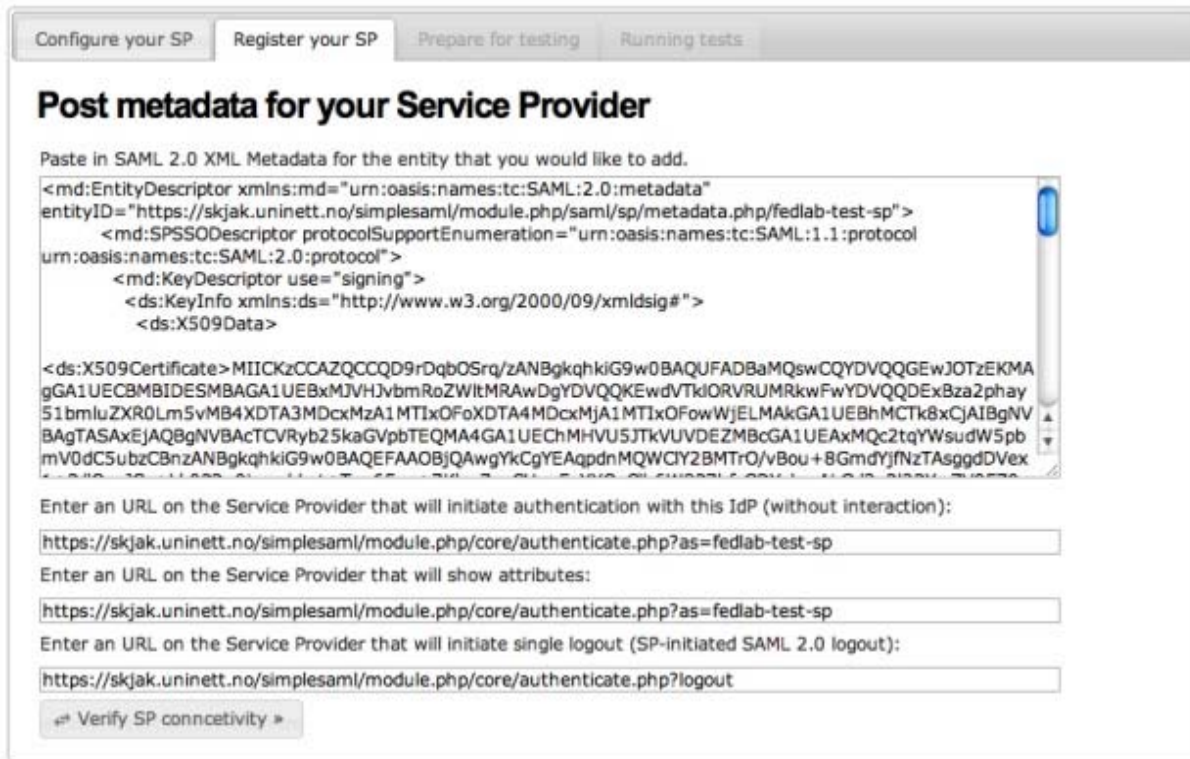


Figure 2.1: Example UI for registering an SP

When the SP is registered, the testing tool performs a simple basic test to verify if the login works under normal circumstances. If not, debug information displays to help to fix the configuration of the SP.

Next, the user may initiate running the real test series. The user interface shows a live progress bar and each test result pops up as soon as it is ready. Each test may result in **OK**, **Notice**, **Warning** or **Error** displayed with varying background colours.

At present, the testing tool has implemented over 80 different test cases. In comparison to manual testing, each of these test cases requires careful configuration; many of the test cases would not be possible to configure without writing dedicated testing tools. This makes the automated testing tool extremely cost effective for testing configurations of SPs. Running through all the test cases takes approximately 30 seconds. The final report quickly displays a test that failed.



Figure 2.4: Example to decode SAML message

2.5 Firefox SAML Debugger Plug-in

JRA3 T2 will also offer an even simpler workflow for debugging SAML messages than the one described in section 2.4 using the SAML debugger.

By creating a Firefox plug-in, we can create a tool where both the message capturing and the SAML decoding is handled and presented in the same tool.

For end users, this plug-in gives them an extra browser window with a live view of well-formatted SAML messages sent on the wire using the other browser window.

It is assumed that this tool will be very popular and JRA3 T2 has a working PoC version already. It is planned to release a final stable public release of the tool before the end of 2011.

2.6 Metadata Editor in Javascript

One of the most important building blocks in setting up identity federations is SAML Metadata XML documents and the exchange of these documents. These metadata documents establish the trust relationship between the Service and Identity Providers (including certificates for signing and encryption) and endpoints that providers should use to exchange SAML messages.

The metadata format is extensible. Recently there have been a number of proposals to provide more information into the metadata. Following are some of the relevant extensions that might be used in our community:

- *SAML 2.0 Metadata Entity Attributes* [[MDEntAttribs](#)]: The Entity Attributes extension, which allows for generic key value pairs associated with entities.
- *Identity Provider Discovery Service Protocol and Profile* [[IdPDiscServ](#)]: Defines fields for use of the Discovery Service.
- *SAML Metadata Document and Registration Information Extension (MDAttribs)* [[MDAttribs](#)]: Defines placeholders for registration information of metadata.
- *Discovery and Login UI Metadata Extension Profile* [[DiscLoginUIMDExt](#)]: Defines several fields, such as human readable name, description, geolocation and logo.

The metadata XML document format also allows for some variations. Several profiles now specify context-specific rules for what metadata should look like in a federation or in a cross-federation such as GÉANT eduGAIN or Kalmar Union [[Kalmar](#)].

The SAML 2.0 Metadata Interoperability profile also puts restrictions on how to use metadata, mainly focusing on the use of certificates:

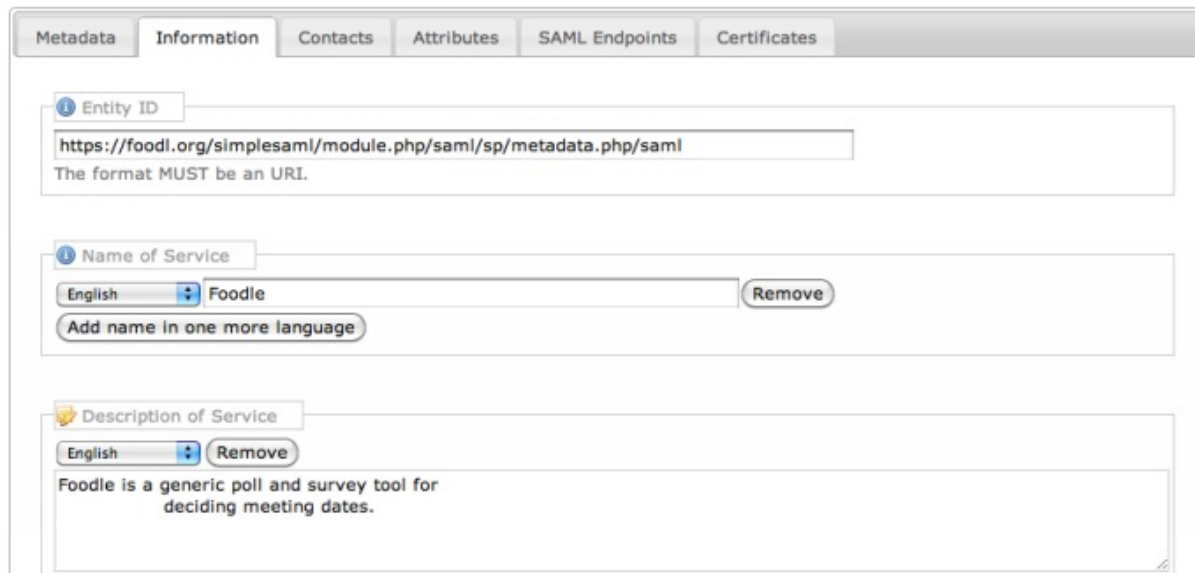
- *SAML 2.0 Metadata Interoperability Profile* [[MDIOP](#)].

Metadata documents can be generated in a number of ways. Often SAML software generates a very basic metadata document with the most crucial elements. It is common for provider operators to add necessary information afterward; including some of the extensions mentioned above. Then these metadata documents often are published through a federation's web-based registry system, which also may add adjustments to the document.

This complex XML format with all of its available extensions and requirements is already or will be an extra burden on administrators who would like to connect to a federation. Also, with the flexibility in the format, we are almost guaranteed to see many variations of how to express more or less the same data. This is something that might, in turn, result in interoperability problems in providers spanning multiple federations. This issue is highly relevant in GÉANT eduGAIN and was experienced in the eduGAIN pilot. JRA3 T2 will work closely with eduGAIN on this topic. It is expected that the SAMLmetaJS will become a tool to help reduce this complexity for real users.

JRA3 T2 is creating a tool that will guide operators to be able to craft sensible, well-structured SAML metadata documents: the SAMLmetaJS (SAML Metadata JavaScript) editor. The tools will allow any web-based text field which requests the user to provide metadata to be a rich metadata editor with a simple-to-use interface. This tool is written entirely in JavaScript and is independent of any programming language used on the server side. By including a few HTML tags, any of the various metadata registry systems can easily enable the metadata editor for all its users.

Demonstration of SAML Meta JS



The screenshot shows the SAML Meta JS editor interface with the following sections:

- Entity ID:** A text input field containing the URL `https://foodl.org/simplesaml/module.php/saml/sp/metadata.php/saml`. Below the field is a note: "The format MUST be an URI."
- Name of Service:** A section with a language dropdown set to "English" and a text input field containing "Foodle". A "Remove" button is to the right. Below the field is a button labeled "Add name in one more language".
- Description of Service:** A section with a language dropdown set to "English" and a "Remove" button. Below the field is a text area containing the description: "Foodle is a generic poll and survey tool for deciding meeting dates."

Figure 2.5: Example of SAMLmetaJS editor

In November 2010, the SAMLMetaJS was made available as a proof of concept (PoC) version. Various mailing lists were notified to gauge interest for including such a tool in federations' own systems. The response was very positive.

Plans for Y3 are to continue working on the SAMLmetaJS editor to make the software more stable and to add more features.

SAMLmetaJS will be made available as an independent software library. It is open source and a stable version will be available for download by October 2011.

3 Federation Harmonisation

The aim of the harmonisation of Identity Federations is to increase interoperability when different federations are inter-connected.

This encompasses the collection of tasks related to the management of a federated identity infrastructure, such as de-provisioning, single logout (SLO) and new SAML 2.0 profiles. Best practices documents can help guide federations to make the same choices and increase the chances for cross-federation interoperability.

The work in this area includes the creation of guidelines and best practices documents to help identity federation operators make implementation choices that will not impact on interoperability.

The JRA3 T2 task during Y2 focused on two aspects related to federation harmonisation, de-provisioning and single log-out.

3.1 De-provisioning

De-provisioning involves a set of standard changes performed on a service delivery platform to prevent a user whose identity has been de-provisioned from an IdP to obtain a service; essentially the reverse of provisioning. Provisioning within an enterprise IT ecosystem typically involves traditional IT integration tools, such as an enterprise service bus, web services, messaging systems, etc. In the controlled environment of enterprise IT systems, provisioning and de-provisioning are complex, but not conceptually difficult.

In the federated model, provisioning is often accomplished by on-demand provisioning where the identity claim used to authenticate a user carries enough information to allow the provisioning of a service to take place.

A real life use-case for de-provisioning is represented by the Confusa [\[CONFUSA\]](#) certificate portal service; this is a web application with federated authentication that issues eScience certificates to users. In this case, if a user no longer has an account on his/her IdP (for example, if the user changes job or if the user graduates), the user's identity is de-provisioned, resulting in the user's certificates being revoked.

Another example is represented by eduroam. When a user's identity is de-provisioned, the user will no longer be able to connect to the eduroam network.

3.1.1 Privacy

During the de-provisioning phase, the amount of Personal Identity Information (PII) that needs to be transmitted is minimal: it is limited to the membership status. Because of this, no user-consent is needed, in contrast to what happens at login time when a user may be asked to approve the information transmitted from the IdP to the service that the user wants to access.

3.1.2 Common Requirements

A set of common requirements has been identified as a basis for evaluating the various proposed solutions to the federated de-provisioning problem:

- It must not be possible for a user to avoid de-provisioning.
- It must be possible to apply policy to de-provisioning events separately from other types of events that happen to use the same protocol mechanisms.
- Repeating the same de-provisioning event must not raise an error condition.

3.1.3 Patterns

It is possible to identify a few common patterns that most proposed solutions to the federated de-provisioning problem fall into:

- Revalidation: Periodically ask users in an SP to re-authenticate to prove that they have an active relationship with their IdP.
- Polling: Out-of-band refresh by the SP; requesting user-state information from the IdP.
- Notification: Out-of-band push of updates from the IdP to the SP regarding user-state.

3.1.4 Further work on De-provisioning

During Y3, work will be carried out to develop a de-provision engine that the federations can set up and make available to institutions. The aim is to facilitate institutions to de-provision services and enhance the quality. Institutions can benefit from this engine when needed. The expected outcome will be an implementation that can serve as a reference implementation for others.

3.2 Single Log-Out

While Single Sign-On (SSO) means creating a new session based on a session at the IdP, single logout profile specifies a method to remove all sessions (both at the IdP and all of the SPs) at once. The trivial approach would be to reverse the process of SSO; however, this would not work in practice since the user might get stuck between redirects if one of the parties does not respond. The biggest threat to Single Logout (SLO) is that it can lead to a false sense of security by giving the user the impression that he/she can leave the browser

unattended. Every logout implementation must, therefore, ensure that each active session is terminated before it reports logout success to the user.

On a technical level, IdPs and SPs communicate with each other by using SAML 2.0 LogoutRequest/LogoutResponse messages. These messages can be transferred using either front-channel binding (HTTP-Redirect in practice) or back-channel (SOAP). The SAML specification recommends that if the Single Logout process is initiated by an SP, then the initiating request should be made via front-channel. In other cases, SAML 2.0 metadata may be used to determine the appropriate binding and endpoint.

3.2.1 Current Implementations

At the time of writing of this deliverable, the major SSO software supports SLO as follows:

- Shibboleth IdP: does not support SLO (a [fork](#) which provides SLO support is maintained by NIIF).
- Shibboleth SP: supports SLO on front-channel and back-channel bindings.
- simpleSAMLphp IdP: supports Single Logout on front-channel bindings.
- simpleSAMLphp SP: supports SLO on front-channel. In addition, as the latest stable branch (1.7) introduced SQL session store, it is possible to use SOAP logout as well.

3.2.2 Major Logout Challenges

3.2.2.1 Third party cookies

Both the simpleSAMLphp and the Shibboleth IdP implementations solve the problem of the daisy-chain of redirects with an iFrame-based user interface for front-channel logout. Via this interface, the browser is redirected to the associated SP in an iFrame (the redirect contains the necessary elements for a SAML 2.0 LogoutRequest). However, if certain third-party cookie restrictions apply, the SP does not get its cookies. Third party cookie handling is very browser-dependent. Moreover, it seems impossible to come up with a solution that can reliably transmit the cookies within a frame (or iFrame, IMG, etc.), unless the SPs share a common domain or third party cookies are enabled (the latter is the default for the most popular browsers).

- **Alternative 1:** Do not rely on cookies when using front-channel. It requires the SP to use a session store that works without cookies. It is possible with simpleSAML SP 1.7, and is possible in theory with Shibboleth SP as well, although currently it does not operate this way. See section 3.2.2.2 for more information.
- **Alternative 2:** Use back-channel logout, which also does not require cookies, therefore, it also needs heavy application support. However, it has at least three other problems:
 - Unlike other back-channel messages, IdP-initiated back-channel messages require the response to be signed, even if it is using an encrypted TLS channel. The reason is that the SPs often use a well-known certificate for the HTTPS endpoint, which might not be the same as the one that can be authenticated with the metadata. Therefore, to authenticate the remote peer, explicit signing must be turned on.
 - If the SP is clustered without shared SP sessions (thus solely relying on node affinity), back-channel requests would probably not reach the SP node with which the user is associated.

- Many IdPs have a firewall, which might not allow outgoing TCP connections to remote hosts.

3.2.2.2 *Application integration*

Traditionally application integration efforts ended with creating an application session based on another session (e.g. Shibboleth SP). It does not involve run-time SP session checking and handling messages or notifications. However, offering single logout without proper application support could result in users leaving their application sessions without requiring them to re-authenticate. This undermines the system's security.

If an application has its own sessions, there are two approaches to prepare for logout:

1. When loading the application session, the application should check that the SP session is still valid. This requires direct modification of the application's session handling.
2. The application should provide an interface to remove a user's session (only Shibboleth SP provides this mechanism). In the case of back-channel binding, the application should be able to do the session removal based on the SP's Session-ID. Therefore, the application sessions should be indexed by this, which again needs direct modification of the application session initiation code. In theory, for front-channel binding, cookies could be used instead. As explained in section 3.2.2.1, frame-based logout UIs cannot deliver the cookies reliably. As a result, this mechanism will not work with such front-channel single logout implementations.

Probably to facilitate cookie-based application session removal, Shibboleth SP refuses to invalidate SP sessions if it receives the LogoutRequest on some front-channel binding without a cookie. This restriction is a policy decision. However, unless a cookie-safe logout controlling user interface is invented, this behaviour blocks front-channel single-logout deployments, even if the application can handle logout events in some other way.

3.2.2.3 *Avoiding stale sessions*

In a Single Sign-On scenario, there are at least four sessions for a user. Each is created on top of the previous session:

3. IdP authentication session.
4. IdP session (active authentications, associated SPs, etc.).
5. SP session.
6. Application session.

These sessions are usually implemented with different session stores. They expire or time out independently. However, if single logout is introduced, it becomes a security problem if a session remains usable after the parent session has expired.

To avoid stale sessions, the IdP session should have a limited inactivity timeout ("soft timeout", the maximum time between two user requests), but an unlimited session lifetime ("hard timeout", independent from user activity). The SP sessions must have a session lifetime, which is not longer than the inactivity timeout of the IdP session. The IdP must set the SessionNotOnOrAfter attribute in the authenticating Assertion according to its

session inactivity timeout value. The SP must honour this attribute and set its session lifetime based on this information.

If the SP session lifetime or inactivity timeout is shorter than the IdP session inactivity timeout, the user may get an error message during single logout because of the expired SP session. IdP implementations can, in theory, exclude the SPs from the SLO whose sessions are initiated earlier than the inactivity timeout. It is not possible to avoid logout errors caused by SP inactivity timeout. It is not an option to treat "Session not found" errors as a successful logout response because of the issue with back-channel logout and clustering, described in section 3.2.2.1.

3.2.2.4 Administrative logout

If a user's session is compromised or, for some other reason, an individual user needs to be disconnected, the administrative logout can be used to terminate IdP and SP sessions without restarting or otherwise clearing all of the IdP sessions. However, none of the implementations offer administrative logout. Administrative logout can only work on back-channel bindings.

NIIF has developed a [proof-of-concept implementation](#), partially solving this with Shibboleth IdP. It is capable of invalidating an existing IdP session. Another option is to send a LogoutRequest to every SP associated to that session. However, it is not a complete solution because the current IdP session API does not provide a way to list all sessions of the user. A user might have multiple sessions at a time if, for example, he/she is logged in from multiple browsers. The implementation can only manage the last created session. The API is subject to change with Shibboleth IdP v3, which will make such features possible. At the time of writing, there is no estimated release date.

3.2.2.5 Interoperability problems

The interoperability problems are mostly not product-specific or implementation-specific, but rather are due to a difference between configurations. The default configuration of Shibboleth IdP SLO requires the logout messages to be signed, but neither Shibboleth SP nor simpleSAMLphp signs front-channel messages by default. Either the requirement should be dropped (which is a violation of the standard and is a possible security risk when using back-channel) or all of the SPs need to be configured to sign messages.

3.2.3 Logout Deployment in Federations

Typically, to ensure that all the users' sessions are closed once the users disconnect, users are asked to kill their browsers. Although this practical approach solves the problem, it is more desirable to offer a single logout button to users.

The work on this led to a recommendation that only those SPs that have proper underlying application support should promote SLO endpoints.

It is also not recommended that Federations deploy SLO without carefully considering all the challenges and how the Federation may be affected.

The document on SLO can be considered complete, but in Y3 JRA3 T2 will perform a detailed analysis of how third party cookies are handled through iFrames in various browsers. Not only does this affect the iFrame-based Logout approach, but it may also impact some of the technology used in VO work. A report on the result of this analysis will be published through the Federation Lab website by March 2012.

For more information, see the detailed report at: <https://fed-lab.org/best-practises/single-logout/>.

4 Virtual Organisations

This task aims to develop the identity architecture to better support users' collaboration beyond organisational boundaries. A typical example of such collaboration is the GÉANT project where users' identities are provisioned by the users' home organisations, but the ability of a user to access a number of resources is determined by the role these users cover in the project.

To aid this, JRA3 T2 has introduced the concept of a Virtual Organisation Platform or simply Virtual Organisations (VO); this is a technical infrastructure that supports collaborative projects, such as GÉANT, which spans across multiple federations and institutions. The VO provides authorisation and additional user data and synchronises this information across multiple services. However, the data is maintained and stored in a single place.

The work in this area resulted in:

- An attribute collector based on previous work carried out by WAYF and SURFNet; this provides important functionalities to implement VOs.
- A suggested technical infrastructure to support VOs based on **SAML AttributeQueries**. NIIF has developed a simpleSAMLphp module, which provides SOAP AttributeQuery support for simpleSAMLphp. By using this extension module, it is possible to use simpleSAMLphp SPs in a VO.
- An implementation of the VO functionalities in a federation with centralised login service (such as the Croatian federation, the Norwegian federations and a few others), developed by SRCE.

Plans for Y3 are to focus on a selected number of real use-cases, to define a VO protocol and to offer an implementation for the protocol. The final result will be released (at the end of Y3) so developers can integrate the VO software in their applications.

4.1 Attribute Queries in simpleSAMLphp and VO

Back-channel attribute queries offer an easy way to integrate SAML SPs with VO platforms (VOP). The VOP provides a SAML Attribute Authority which can be used to retrieve additional information (attributes) from the current federated user. The VO platform and the SP must share a common identifier of the user. In many cases, this is a federation-wide identifier eduPersonPrincipalName.

4.1.1 Implementation

NIIF has developed a simpleSAMLphp module which provides SOAP AttributeQuery support for simpleSAMLphp [[simpleSAMLphpAQ](#)]. By using this extension module, it is possible to use simpleSAMLphp SPs in a VO. (Shibboleth IdP can be used to implement a SOAP Attribute Authority.) The module, called saml2Aggregation, is a post-authentication processing filter. It takes the configured session attribute (e.g. eduPersonPrincipalName) and queries a given Attribute Authority for additional attributes. It then adds the received attributes to the user's session.

This module is not only useful in the SP+VO scenario, but in any deployment where direct access to the attribute store is unfeasible (SQL, LDAP) and the flexibility of SAML is required.

The module can be adapted to both simpleSAMLphp-1.6 and the newly released simpleSAMLphp-1.7, but not without patches to the SSP core itself (see SSP issue #333 [[SSP333](#)]). It is hoped that the upcoming release of simpleSAMLphp-1.7.1 will solve the integration issues between the core and the extension module.

4.1.2 Privacy Considerations

When using eduPersonPrincipalName as an identifier for a VO, care should be taken to restrict the set of SPs for which the AttributeQueries are answered because the identifier is not transparent and could be used to collect VO data from individuals.

The current module implementation can only use simple string identifiers (such as eppn, mail, etc). It can be easily extended to support compound identifiers (such as the new-style eduPersonTargetedId or SAML2 Persistent NameID). To use targeted identifiers, AffiliationDescriptor should be supported by the IdP as well.

4.2 VO Platform in Centralised Login Federation

CARNET/SRCE has developed an implementation of the VO functionalities in a federation with a centralised login, such as the Croatian federation. This implementation is based on the simpleSAMLphp tool and introduces a centralised VO platform. The implementation has been developed and tested as a part of the Croatian federation (AAI@EduHr).

Figure 4.1 shows the architecture of the VO implementation.

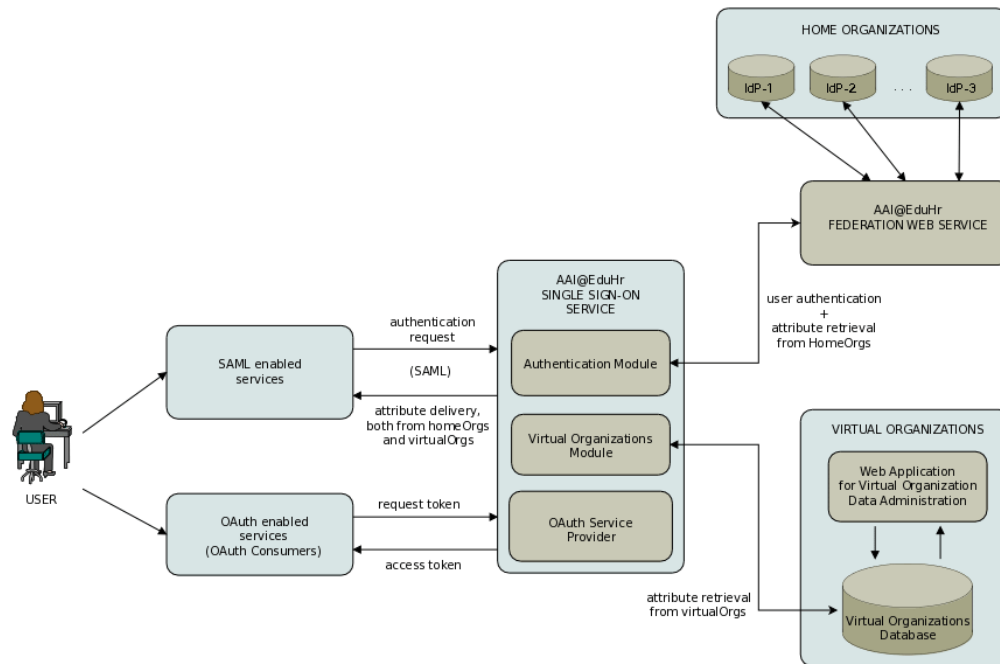


Figure 4.1: VO implementation architecture

In March 2011, after a pilot phase, this implementation was moved into production and is now part of the Croatian federation. For more information, go to: <http://www.aaiedu.hr/vo/> (note that access is limited only to users from the AAI@EduHr federation).

As the platform is based on AAI@EduHR architecture, SRCE will further enhance the software independently by the JRA3 T2 Task. Two main enhancements are planned by the end of 2011:

- Enabling different protocols for attribute retrieval. Currently, only OAuth is implemented, but other protocols may be added.
- Enabling cross-federation attribute exchange scenario, as Figure 4.2 illustrates.

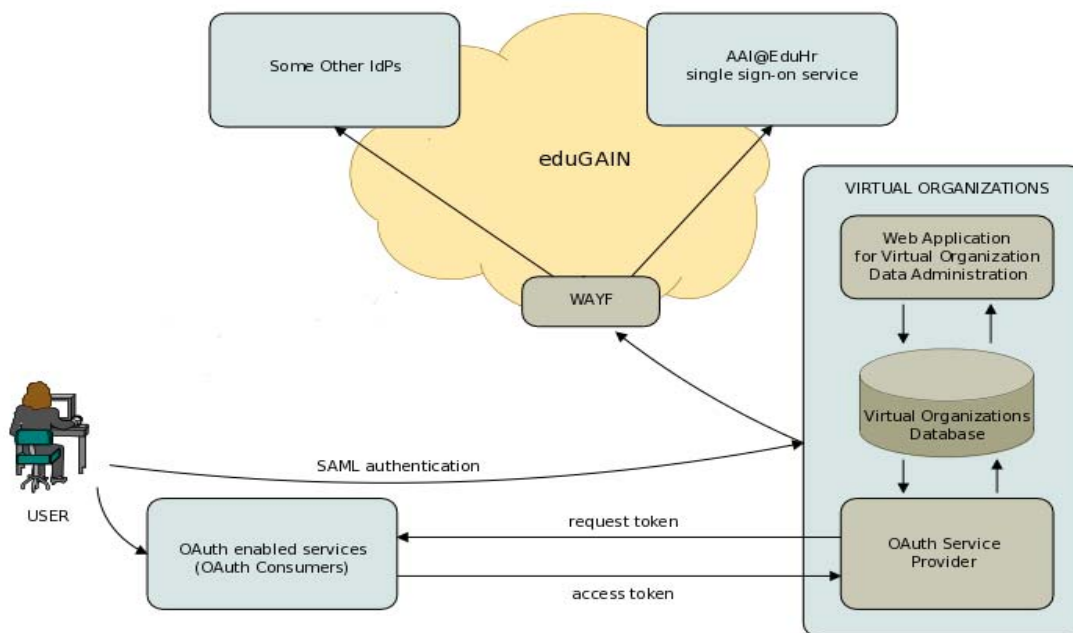


Figure 4.2: Enabling cross-federation attribute exchange

4.3 Attribute Collector Based on Virtual IdP

A use-case investigated by JRA T2 is related to the problem of collecting information about a user from multiple sources into an information superset, then making a subset of that collected information available to a given SP. This type of functionality is typically needed for creating VO support across SPs (or across federations).

WAYF [WAYF] is implementing such an attribute collector based on the Corto [CORTO] system. Corto is based on the virtual IdP concept, where an IdP is substituted by a corresponding virtual IdP. This setup allows the virtual IdP to collect additional information about a user from various sources that already have well-established trust relationships with the (virtual) IdP in place. The virtual IdP can then present (a subset of) the information to an SP as if it were coming directly from the (virtual) IdP itself. In other words, from an SP's point of view the virtual IdP acts exactly as (and is indistinguishable from) a regular IdP. This relieves the SP from the task of gathering additional attributes about the user itself, which it otherwise would need to do to realise a VO.

The introduction of proxy endpoints for all connected entities in hub-and-spoke federations, as well as the use of virtual IdPs and meaningless URLs (solely used as identifiers), enable Corto to act as a transparent attribute collector in both peer-to-peer federations and hub-and-spoke federations. The functional description of Corto is at <https://sites.google.com/site/cortopages/>.

Corto was initially called SpringFika [[SpringFika](#)], (described in the deliverable, *Identity Federations. [DJ3.2.1.1]*). The code has been further developed since then, both in JRA T2 and by WAYF (Denmark) and SURFnet (the Netherlands).

SURFnet implemented Corto as an attribute collector in the COIN [[COIN](#)] infrastructure, which dynamically builds a central database containing user information from several administrative domains. This setup, as an example, allows group management across SPs, i.e. the same group relations can be used in different SPs without defining them separately for each and every SP where they are needed. COIN is part of the bigger SURFconext [[SURFconext](#)] vision of a next generation collaboration infrastructure that creates new opportunities to collaborate online based on a (user composed) combination of applications from different providers.

WAYF is also using Corto to implement proxied federation endpoints for all services and IdPs. The goal is to publicly present entities in a peer-to-peer federation fashion, while at the same time keeping functional advantages of the hub-and-spoke architecture, such as real-time protocol translation, centralised attribute release policies, data release consent functionality and administration, and simple metadata administration.

5 Metadata Distribution and Cross-Federation Scalability

SPs want to offer their services across federations to as many users as possible. Federations also want to offer as many services as possible to their users (those users who are associated with the IdPs of these federations). All these actions require establishing trust between IdP and SP entities.

Inter-entity trust establishment must be scalable, particularly when the entities belong to different federations. An SP cannot afford to join all federations due to the overhead associated with trust establishment and maintenance. In a federative setting, SPs ideally should be able to join one federation and expect the federation to take care of establishing and maintaining their trust relationships with IdPs of (other) federations.

Inter-entity trust establishment in a federation is mainly realised through “trustful metadata exchange”. In higher education federations, the trustful metadata exchange is realised by exchanging signed XML metadata files. The scalability of inter-entity trust establishment, therefore, directly relates to the scalability of the exchange process of signed XML metadata files.

The scalability of signed XML metadata exchange across federations consists of making:

- Trust bootstrapping mechanism for cross-federation metadata exchange more scalable.
- Metadata exchange process more scalable.

The trust bootstrapping mechanism provides a means to distribute the signing keys of XML metadata files “trustfully” among the parties involved in cross-federation metadata exchange. The metadata exchange process, which utilises the bootstrapping trust relations, transfers the metadata among system building blocks.

5.1 New Models for Distributed Metadata Aggregation

Current solutions based on the Simple Metadata Aggregation Profile [[simpleMAP](#)]² architecture may suffer from scalability problems because it assumes a central aggregator, for both metadata updates by publishing entities and for metadata usage by consuming entities. Such a central component is also very vulnerable to attacks since it is a single point of failure.

² This profile was described in the DJ3.2.1 chapter 3.1.1

In this deliverable an alternative approach is described. This approach is based on the distinction between the **trust fabric** and the **core trust fabric**. The former is the trust that must ultimately exist between entities from different federations (IdPs and SPs). The latter, the core trust fabric, is the trust which can be established between local Metadata Aggregators (MA), of which there is one in every federation. The trust core fabric is much easier to achieve and presents fewer scalability issues, since the number of local MAs is limited (approximately 20 in the case of eduGAIN). The required *intra*-federation trust (the trust between entities within a federation) is already present because of the very nature of a federation. *Inter*-federation trust can be achieved in a scalable manner, since local MAs link the *inter*-federation trust with the *intra*-federation trust.

Although the core of the work was done in a Dutch national program, the draft results were discussed with JRA3 T2. The results were used to improve and complete the document, which is available at <http://www.surfnet.nl/nl/Innovatieprogramma%27s/gigaport3/Pages/Resultaten2010.aspx> [Result2010] under a Creative Commons licence.

5.2 Metadata Aggregator Specification

The goal of this activity is to create a requirement specification for a metadata aggregator. An MA is a component that collects SAML 2.0 metadata from different entities, processes the resulting aggregation of metadata, signs it and publishes it again.

The eduGAIN Metadata Service (MDS) operated by GN3 SA3 requires an implementation of an MA. Consequently, the idea is that the MDS in the long term will use an implementation that meets the requirements that results from this activity.

Since the last deliverable, four video meetings were held. At first, work continued on one single document which dealt with various aspects of metadata aggregation. This included metadata format requirements as well as operational requirements. It then was decided to split this into multiple separate documents, because some of the requirements belong to the SA3 eduGAIN task.

The main document on which the work focused is the *Basic Metadata Aggregation Profile* [BMAP]. It contains a very generic metadata aggregation specification, which is not specifically targeted at the eduGAIN MDS.

The document is available on the Federation Lab website. However it may be desirable to publish it via one of the standardisation bodies. Because of the very general nature of the profile, OASIS [OASIS] seems to be most suitable body to submit a draft.

Further comments to refine the document are expected once eduGAIN runs in production mode (April 2011). The Identity Federations Task will follow up on requirements and experience drawn from eduGAIN.

5.3 Scalability of IdP Discovery: Location-based Discovery

5.3.1 Problem Statement

Federated services all rely on remote authentication systems, typically at the user's home organisation or IdP. The user must select his/her own IdP among the various IdPs available on a service page to login. This is known as IdP Discovery. As federations grow in number and size, the IdP discovery becomes cumbersome and difficult to handle. The problem is how to ease this task using a new combination of lists, maps and dynamic interrelations between them.

The new HTML 5 specification is expected to become the de facto standard in the coming years. It hosts features and APIs that can be used to ease the user's burden of IdP discovery. Among those is the Geolocation API, which reveals a user's physical, geographical location to the web browser.

Assisted matching of a user's physical location with other geographical knowledge about IdPs, etc., might augment the task of selecting IdPs. Section 5.3.2 describes a prototype that has been implemented.

5.3.2 Functional Description

The implemented solution uses simpleSAMLphp's standard discovery service with a few modifications and standard simpleSAMLphp flatfile metadata with location information added. The solution is implemented as a small JavaScript library that utilises the Google Maps API and the browser's native Geolocation API, as defined by the World Wide Web Consortium (W3C) [[W3C](#)].

When arriving at the IdP discovery service, the browser is queried for the user's physical location via the browser's Geolocation API. If the user accepts to reveal his location, a map displays (centred on the user's actual location). If the user does not allow his location to be revealed or the browser does not implement the Geolocation API, the map is centred at a predefined position.

All IdP metadata is processed and the geographical location is marked on the map. In addition to the map, a list of IdPs is visible on the map at the current zoom level. All IdPs have a one-to-one representation with a marker.

When the map is panned or zoomed, the list updates accordingly to only display the IdPs visible on the map. This means that if the IdP you are looking for is not on the location on which the map is initialised, you can change the map view to locate the IdP for which you are looking.

To select an IdP, click the marker on the map or on the IdP name in the list. Hover the mouse over a marker on the map to display the name of the IdP that refers to that marker.



Figure 5.1: IdP selection window

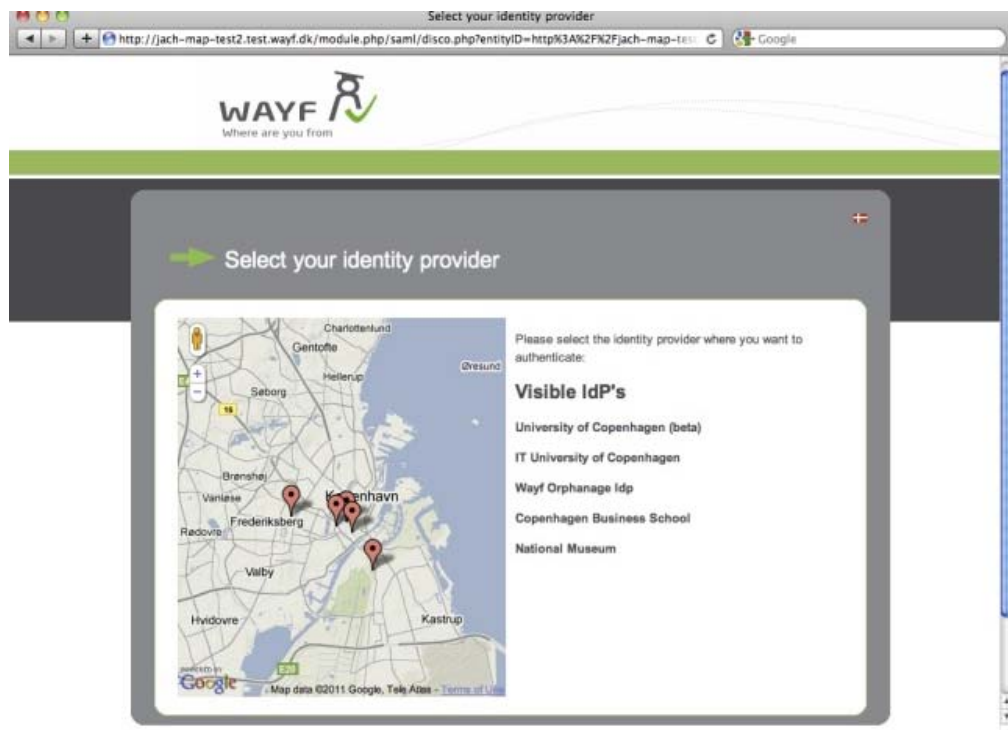


Figure 5.2: IdP selection window (zoom)

5.3.3 Future Work

The implemented solution has a number of limitations and issues on which work will focus in Y3:

- An IdP may have multiple locations associated with it. This is not supported in the current implementation.
- More detailed information about the IdP can be revealed by hovering above the IdP's marker on the map.
- How to represent IdPs with no location information.
- Graceful deterioration if the map service is not available.
- Parsing of SAML 2.0 metadata containing location information (IdP Discovery and Login UI Metadata Extension Profile).

5.4 Kantara ULX Scalability

As GÉANT eduGAIN becomes successful and the set of interconnected SAML providers increases, the user experience of selecting which IdP to use for logging in becomes a major issue. First generation user interfaces that present dropdown menus for the available providers simply do not work with thousands of available providers.

The Kantara Initiative has a working group, Universal Login Experience (ULX) [[KantaraULX](#)], which is working with an improved user interface for selecting providers. Attention has been given to how this work can be applied and used in a context such as GÉANT eduGAIN. It has included several modifications from the Kantara mock-up. This targeted mainly the ability to scale to a very large set of providers. The Kantara ULX mock-up is also very focused on displaying logos instead of text, which might be less optimal for educational providers than for big commercial companies.

Figure 5.3 shows a demonstration of PoC modifications to Kantara ULX.



Figure 5.3: Demonstration of PoC modifications to Kantara ULX

The current demo also contains a number of features that did not exist in the original Kantara ULX mock-up, which include being able to:

- Display a combination of logos and text with an additional dynamic text description.
- Support keywords for searching.
- Categorise by type.
- Categorise by country.
- Automatically discover the current country of the user by using IP lookup.
- Support for HTML5 Geolocation API to discover the user's location.
- Support for geotagging of providers.

Usability of provider discovery user interfaces is a hot topic and is currently being discussed in several forums. Further work in this area will be coordinated with GÉANT eduGAIN (in particular), Terena REFEDs [[REFED](#)] and the Kantara ULX Working Group [[ULX](#)].

6 Beyond WebSSO: Moonshot

The GÉANT Project is contributing effort towards Project Moonshot, an initiative to bring the benefits of Identity Federations to internet protocols other than HTTP. The principal use-cases for this work are:

- Federated access to High Performance Computing (HPC) facilities. The primary goal is to make HPC services more robust by enabling users and jobs to be transferred transparently between HPC facilities. The project has worked with the UK High Performance Computing Special Interest Group to define use-cases and requirements.
- Federated access to Grid [\[Grid\]](#) computing resources. The primary goal is to improve the user experience by enabling the use of credentials other than X.509 certificates, which are often perceived by users as complex and difficult to use. The project has worked with the UK National Grid Service and CESNET to define use-cases and requirements.
- Federated access to email and instant messaging. The primary goal is to make it easier to outsource the provision of these services to third parties. The project has worked with several institutions and Google to define use-cases and requirements.

This sub-task started in April 2010. The sub-task's significant achievements since this date include:

- The inputs that led to identifying the work that required standardisation efforts. This resulted in the formation of a new Internet Engineering Task Force (IETF) Working Group, ABFAB [\[ABFAB\]](#), to standardise the Moonshot architecture. The Working Group is chaired by individuals from Cisco and NORDUnet. The first Working Group meeting was held in Beijing during IETF 79. The next meeting will be at IETF 80 in Prague.
- The inputs to specify the Moonshot architecture within IETF [\[IETF\]](#) draft documents, the involvement of other industry partners and the work that was in scope with JRA3 T2's objectives.
- The implementation of aspects of the Moonshot architecture, including a new Apache authentication module and Firefox extension; a new RadSec [\[RadSec\]](#) library, *libradsec*, has also been implemented, based on radsecproxy [\[radsecproxy\]](#) that was developed during GN2.

The final version of the code is expected in August 2011.

7 Conclusions

As described in this document, significant work has been carried out by the JRA3 T2 group.

Based on the results achieved in Y2, JRA3 T2 will focus during Y3 on the following areas:

- Federation Lab. This suite will be extended to include IdPs and metadata validation services. JRA3 T2 plans to do more promotion for this suite towards the end of Y3.
- Further work will be done on the discovery service. The first release of the software is expected on 17 May 2011. This release of the software will be used in production and, based on the feedback received, a new version will be produced at the end of Y3. As the work builds on the initial proof-of-concept developed within Kantara, it is expected that the results will be exposed to a wide community, not only limited to the project partners. A presentation of this work will be made during the TERENA Conference in May 2011.
- Based on the results of the tests made in Y2, it has been agreed that the VO work will focus on a limited information model, solving a selected number of real use-cases and delivering the technology so developers can integrate it in their platforms. The final result will be released (at the end of Y3) so developers can integrate the VO software in their applications.
- Alongside best practices documents, the Federation Harmonisation sub-task will create a de-provisioning engine that the federations can set up and make available to institutions. The aim is to facilitate institutions to de-provision services and enhance the quality. The outcome (which will become available at the end of Y3) will be one real-life implementation that can serve as a reference implementation for others. More information on the progress of this work will be shared at:

<https://fed-lab.org/best-practises/provisioning/>.

- Further results are expected from the project Moonshot, where JRA3 T2 is contributing manpower. The main result will be to support some of the (so-named) beyond-web SSO use-cases. The first main result is expected in August 2011.

References

[ABFAB]	https://datatracker.ietf.org/wg/abfab/
[BMAP]	<i>Basic Metadata Aggregation Profile</i> http://dl.dropbox.com/u/2381403/SAML/BasicMetadataAggregationProfile-0.4.pdf
[COIN]	https://projectcoin.surfnet.nl/
[CONFUSA]	http://www.assembla.com/wiki/show/confusa
[CORTO]	https://sites.google.com/site/cortopages/ http://www.phpbuilder.com/lists/php-db/2002092/0135.php
[DiscLoginUIMDExt]	https://spaces.internet2.edu/download/attachments/9731/saml_ds_login_ui_02.odt
[DJ3.2.1,1]	<i>Identity Federations</i> , GN3-10-039 http://www.geant.net/Media_Centre/Media_Library/Media_Library/GN3-10-039-DJ3-2-1-1_Identity_Federations-FINAL.pdf
[eduGAIN]	http://www.edugain.org/
[FEDLAB]	https://www.fed-lab.org
[Grid]	http://www.egi.eu/
[IdPDiscServ]	http://docs.oasis-open.org/security/saml/Post2.0/ssstc-saml-idp-discovery.pdf
[IETF]	http://www.ietf.org/
[JANUS]	http://www.wayf.dk/wayfweb/janus_coderepository.html
[Kalmar]	http://www.kalmar2.org/
[KantaraULX]	http://kantarainitiative.org/confluence/display/ulx/Home
[MDAttribs]	https://spaces.internet2.edu/download/attachments/9731/saml_md_dri_01.odt
[MDEntAttribs]	http://docs.oasis-open.org/security/saml/Post2.0/ssstc-metadata-attr.html
[MDIOP]	http://www.oasis-open.org/committees/download.php/36645/draft-ssstc-metadata-iop-2.0-01.pdf
[Moonshot]	http://www.project-moonshot.org/
NIIFShibIpdSLO]	https://wiki.aai.niif.hu/index.php/ShibIpdSLO
[OASIS]	www.oasis-open.org/
[OpenIdP]	https://sisatestidp.dfn.de/simplesaml/module.php/selfregister/index.php
[ProtectNetwork]	http://www.protectnetwork.org/
[RadSec]	http://wiki.freeradius.org/RadSec
[radsecproxy]	http://software.uninett.no/radsecproxy
[REFED]	http://www.terena.org/activities/refeds/
[Result2010]	http://www.surfnet.nl/nl/Innovatieprogramma%27s/gigaport3/Pages/Resultaten2010.aspx
[saml2int]	http://saml2int.org/
[simpleMAP]	https://rnd.feide.no/2009/07/13/simple_metadata_aggregation/
[simpleSAMLphp]	http://rnd.feide.no/simplesamlphp
[simpleSAMLphpAQ]	https://repo.niif.hu/gitweb/gitweb.cgi?p=simplesamlphp.git;a=commitdiff;h=d1c529c8643863f4ce68f9ba40d98c2a06a6117;hp=3fc89ecf47f0b7469ced47e2ac454dcf113391d0

References



[SpringFika]	http://code.google.com/p/springfika/
[SSP333].	http://code.google.com/p/simplesamlphp/issues/detail?id=333&can=1&sort=-id
[Terena]	http://www.TERENA.org
[TestShib]	http://www.testshib.org/testshib-two/index.jsp
[ULX]	Kantara ULX Working Group
[W3C]	http://www.w3.org/
[WAYF]	http://code.google.com/p/wayf/

Glossary

ABFAB	Application Bridging for Federated Access Beyond web
API	Application Programming Interface
back-channel flow	Protocol messages are exchanged directly between IdP and SP.
Corto	PHP open source software package for WebSSO; based on the SAML 2.0 specification.
eduGAIN	AAI confederation to interconnect a set of national/community-wide AAI federations
front-channel flow	Involves the user and the user-agent in the protocol flow.
GN2	GÉANT second generation network
HPC	High Performance Computing
HTTP	Hypertext Transfer Protocol
IdP	Identity Provider
IETF	Internet Engineering Task Force
IP	Internet Protocol
JANUS module	simpleSAMLphp extension built within WAYF federation
LDAP	Lightweight Directory Access Protocol
MA	metadata aggregators
MDS	Metadata Service
Moonshot	Initiative to bring the benefits of Identity Federations to internet protocols other than HTTP
NREN	National Research and Engineering Network
OASIS	Organization for the Advancement of Structured Information Standards
OCSP	Online Certificate Status Protocol
PII	Personal Identity Information (PII)
PoC	Proof-of-Concept
RADIUS	RADIUS Remote Authentication Dial In User Service - Transport protocol for AAA purposes.
RadSec	A modified RADIUS protocol. RadSec - a secure, reliable RADIUS Protocol
radsecproxy	Implementation of RADIUS/TLS
REFEDs	Research and Education Federations
SAML	Security Assertion Markup Language
SLO	Single Log-Out
SOAP	Simple Object Access Protocol
SP	Service Provider
SSP	Storage Service Provider
SQL	Structured Query Language
SSO	Single Sign-On
TCP	Transmission Control Protocol
TLS	Transport Layer Security

UI	User Interface
ULX	Universal Login Experience
VO	Virtual Organisation
VOP	Virtual Organisation Platform
W3C	World Wide Web Consortium (
WAYF	Where Are You From; Shibboleth-specific service. Allows user to choose appropriate IdP when attempting to access a resource protected by SP
WebSSO	Web Single Sign-On
XML	eXtensible Mark-Up Language