Federated Identity Management for Grids

Henri Mikkonen and Mika Silander
Helsinki Institute of Physics
PL 9250
02015 TKK, Finland
Email: firstname.surname@hip.fi

Abstract—Identity federation is a novel technology allowing end users’ identity information and preferences to be communicated between service providers. While in complete control over what personal information is interchanged, the users benefit from identity federation in several ways, e.g. the services can be tailored according to the users’ preferences and all the services can be accessed by logging in once to the community of collaborating service providers.

Computing Grids have recently gained popularity in the research community because of the superior capacity of computation they offer. This characteristic is ideal for solving complex problems that consume exceptional amounts of computing resources. Using current Grid middleware requires, however, fairly proficient computing skills thus excluding out less experienced but nonetheless potential users.

We explore the possibilities to combine identity federation and Grid services in order to simplify the usage of Grids and thus lower the barrier of entrance for Grid usage. Our prototype implementation leverages the identity federation standards and provides a simple and secure end user interface to Grid services.

I. INTRODUCTION

Grid technologies have become common in the world of research for enabling distributed services over the Internet. Their idea is to harness idle computing, storage and network resources. One of the hurdles preventing the Grid technologies from gaining broader acceptance lies in their usability. Nowadays, a typical Grid user is a researcher working for an academic research institute who accesses Grid services with command line tools. They are efficient in the hands of an experienced user, but very difficult for average users accustomed to Graphical User Interfaces (GUI).

As the Internet is nowadays accessible for almost everybody everywhere, a Grid portal seems to be a good solution to simplify the use of Grid services. It offers a GUI that can be used with a Web browser and browsers are bundled to virtually every common operating system distribution. Depending on the user authentication methods, Web portals can also be accessed from a wide varieties of devices, including mobile phones and Personal Digital Assistants (PDA).

A Grid portal makes the use of Grid services easier, but there is still one important thing it needs: a permission to execute jobs and access Grid services on the user’s behalf. In practice, a user grants this permission by issuing a user proxy, the user’s Grid identity. In Grid Security Infrastructure (GSI), a commonly used Grid security module, the users’ and resources’ identities are defined by X.509 certificates [1]. A long-term certificate is issued to a user by a trusted Certificate Authority (CA) after authentication to a Registration Authority (RA). The user certificate together with its private key are then used to initialize the user proxy. This in turn consists of a proxy certificate [2], its corresponding private key and the signer’s certificate chain. As the private key is included unencrypted in the user proxy, the authentication to the services can be done without user intervention (Single Sign-On, SSO). Because of the security risk this implies, the user proxy is valid only for a short time, far shorter than the user’s long-term certificate. The proxy is generated with a proxy initialization program that must be installed to a computer where the signer’s certificate and private key are accessible.

As a result, a user who wants to access Grid services via a Grid portal still needs to obtain at least a personal certificate from a CA and install the proxy initialization software on the workstation. All this may be quite straightforward for an experienced user but inusperable for an average Internet user.

The paper is organized as follows. First we provide an overview of the main components of our solution in Section III. The interaction of the components is clarified in the prototype’s use case in Section IV. The proxy extension implemented for managing the end users’ Grid credentials in an identity federation framework is explained in Section V. We conclude by listing the results and by identifying the issues requiring further attention.

II. RELATED WORK

Online Credential Repositories (OCR) resemble conceptually the prototype presented in this paper. They can host the users’ short-term proxy certificates and if desired, also their long-term certificates [3]. In the latter case, they completely release an end user from the burden of certificate maintenance and thus enable renewal and delegation of the users’ access rights to Grid agents acting on the users’ behalf. Combined with a browser accessible GUI, an OCR can be used for building a Grid portal.

MyProxy [4], released in 2000, is an example of such an OCR. By integrating an OCR with a CA, an accounting system is able to request a certificate for the user as part of the enrollment process. The Portal-Based User Registration Service (PURSe) [5] implements such a system with the help of MyProxy. In PURSe, the long-term user certificate is stored directly to the user’s account at the MyProxy server. The user proxy can then be initialized at the MyProxy server and retrieved by the user without having to manage the long-term...
credentials. Also, the proxy can be delegated directly to a Grid portal.

III. OVERVIEW

Physically, a Grid consists of resources offered by the participating organizations. Users within these organizations have different roles and responsibilities and can be given access accordingly to different subsets of resources. It is customary to group users into what is called Virtual Organizations (VO) and assign the access rights to Grid resources on a per VO basis. The members of a VO can pertain to any of the collaborating organizations and it is often the case that the VO spans several member organizations.

The federated identity management targets a different problem: it tries to simplify the access to services and allow the tailoring of them according to the user’s preferences. The user has complete control over what personal attributes and preference information are distributed between the community of cooperating service providers and hence, to what extent the services offered can be tailored. Liberty Alliance [6] is an industry driven standardization consortium defining and promoting standards for federated identity management. From its point of view, Grid collaborations and their member organizations form Circles of Trust (CoT). The concept of CoTs is similar in spirit to Grid VOAs as it assumes all its member organizations and companies adhere to commonly agreed identity information exchange conventions and policies. In a CoT, there are Service Providers (SP) and Identity Providers (IDP). The SPs are entities offering services to the end users. IDPs offer the SS0 authentication service to end users and vouch for the authentication with respect to the SPs, i.e. IDPs communicate the end users’ authentication events to the SPs.

In order to offer Grid services from within the identity federation framework and get the additional benefits of identity management, provisions were made for transporting the end users’ Grid credentials between the identity management framework and the Grid infrastructure. This concerns in particular the users’ user proxies. Figure 1 illustrates the proxy extension scenario of our prototype implementation fulfilling this requirement. In the solution, the services of a VO are accessed from a Grid portal but the portal also acts as a SP in the Liberty CoT. This dual role portal capable of handling the end users’ proxies was called the GridSP. The IDP augmented with the proxy extension functionality and conforming to the Liberty Identity Federation Framework (ID-FF) standard [7], was named the Identity Broker (IDB).

The IDB manages its users’ long-term certificates and the corresponding private keys. The problem is that the GridSP and subsequent Grid servers must have the user proxy that is generated with the long-term certificate in order to be able to execute jobs on the Grid. The process of delegating a user’s rights to a Grid agent acting on their behalf, is referred to as proxy delegation [2]. In the sequence below, the delegatee is the entity requesting a proxy certificate and the delegator is the entity granting this proxy certificate by signing the delegatee’s request:

1) The delegatee generates a new public/private keypair.
2) The delegatee generates a certificate request and attaches the public key to it.
3) The delegatee sends the certificate request to the delegator, the owner of the delegating certificate.
4) The delegator signs the certificate request and thus generates a proxy certificate.
5) The delegator sends the proxy certificate to the delegatee.

After a successful completion of the process, the delegatee has the proxy certificate together with the corresponding private key, i.e. the user proxy. In this case, the IDB service acts as the delegator and the GridSP service as the delegatee. The GridSP may now initiate jobs on the Grid and further delegate the proxy to other Grid servers when necessary.

IV. THE PROTOTYPE

Liberty Alliance allows extending the ID-FF protocols for application specific purposes. To this end, there are Extension elements in their messages that developers may employ to add further XML encoded information to the message exchanges. Applications unaware of these extensions, must, according to Liberty Alliance, simply ignore them to ensure interoperability. This possibility of extension was leveraged to implement the obligatory initial step of proxy delegation. This approach ensures conformance and seamless integration with other Liberty Alliance enabled software.

For the subsequent transfer of the proxy certificate between the GridSP and the IDB, we implemented a new protocol, the proxy delegation protocol. This protocol in turn was implemented by reusing and extending the ID-FF specifications.

Our proxy extension consists thus of the augment to an existing Liberty ID-FF protocol and the proxy delegation protocol. More specifically, it enables performing the entire proxy delegation process between the GridSP and the IDB. The following use case demonstrates the proxy extension and compares it to the MyProxy server functionality.

Fig. 1. The architecture of the prototype implementation
A. Use case

We assume the user has existing accounts on both the IDB and the GridSP as stipulated in the Liberty Alliance account federation. The user’s long-term certificate and the corresponding private key are generated as part of the enrollment process and stored in the IDB’s database.

To begin with, the user must federate his accounts between the IDB and the GridSP. The account federation can be established after the user has successfully authenticated to both services. This is a standard Liberty Alliance procedure. The existing federations are registered to the IDB, where the user can manage delegation authorizations for each GridSP. By contrast, the usage of the MyProxy server does not necessarily require an existing user account at the Grid portal as the essential user information is obtained from the delegated proxy certificate.

Once the federation is established, standard Liberty ID-FF functions enable SSO: instead of authenticating to the GridSP, the user is redirected to the IDB with an authentication request. At this point, the GridSP may specify further requirements for the authentication performed by the IDB, e.g., preferred authentication algorithms and key sizes for adequate cryptographic strength. In addition to the standard operations of the Liberty ID-FF SSO procedure, the GridSP queries for the proxy certificate subject Distinguished Name (DN), see Section V-A. If the user has not already been authenticated to the IDB, the authentication is performed now. After a successful authentication, the IDB redirects the user back to the GridSP together with an authentication response. The response contains an assertion expressed in SAML v1.1 [8] about the user authentication. If the user had earlier authorized the GridSP to delegate the user’s proxy certificates, the subject DN is included in the response message. With this authorization, the GridSP has now all the required attributes to request a user proxy certificate from the IDB and subsequently initiate the user’s Grid jobs.

The delegation protocol used in the MyProxy server does not utilize the subject DN field in the certificate request [9]. A similar subject DN exchange process is thus not required.

In Liberty Alliance, protocol definitions are two-fold: the exact structure of a protocol’s messages are given in protocol schema definitions and the protocol’s message exchanges in protocol profiles definitions. As mentioned earlier, the proxy delegation process is started by the delegatee, the GridSP in this case. The underlying protocol for the proxy delegation is defined as the protocol schema definition in Section V-B and the protocol profile definition in Section V-C. After a successful protocol message exchange, the GridSP is in possession of a user proxy that allows it authenticate to Grid resources as the user. The whole proxy delegation process can basically be hidden away from the users: if the user requested a Grid service that requires the user proxy and no valid proxy is stored at the GridSP’s database, or, the validity time is about to expire, the portal would automatically request a new one from the IDB without user intervention. This resembles the proxy renewal service of the MyProxy server, but in that case the workload management system of a Grid resource needs to implement the renewal service instead of the portal [10].

V. LIBERTY ID-FF PROXY EXTENSION

First we describe the extension to the existing ID-FF protocol that performs the obligatory initial step prior to the execution of our own proxy delegation protocol. Then we proceed as in Liberty Alliance specifications by first describing the structure of the messages of our proxy delegation protocol and then the message exchanges.

A. Proxy Certificate Subject DN Exchange

In order to act on the user’s behalf, the GridSP needs to construct a complete user proxy out of the proxy certificate signed by the IDB. To this end, it attaches the private key it originally generated for the request to the returned proxy certificate. The original request also contained the subject DN of the certificate. The subject DN could have been stored to the portal’s user database manually by a user, but this was automated to maximize usability as follows.

The Liberty ID-FF specifications allow the subject attribute to be retrieved from the IDB by using <Extension> elements in both the request and the response of the SSO and federation protocol. In our solution, the structure of the Extension-element in the request is as follows:

< Extension>
   <proxyCertificateSubjectDN request="true" xmlns:pd="[address]"/>
</Extension>

As the Liberty ID-FF specifications require a namespace to be used in Extension-elements, the [address] must be replaced by the address for the namespace definition XML schema. However, as it can be seen from the two listings, the schema is relatively simple in this case and the definition is thus omitted.

The response, on the other hand, is illustrated in the listing below:

< Extension>
   <proxyCertificateSubjectDN xmlns:pd="[address]"/>
   <subject/>
</Extension>

The [subject] contains the subject DN of the certificate the IDB can delegate to the GridSP. It can be e.g. /C=FI/O=HIP/OU=Tech/OU=hip.fi/CN=Henri Mikkanen. The user must be able to control which SPs can obtain the subject DN, as the response message is a kind of advertisement for the storage of the user’s long-term certificate and the private key.

B. Protocol Schema Definition

The proxy delegation protocol is a request/response protocol that can be used to delegate a proxy certificate from an IDB to a GridSP. The generic delegation protocol was described in Section III: the request of the protocol is used in the third phase of the proxy delegation process and the response in the fifth phase.
To request a proxy certificate, the GridSP sends a <ProxyDelegationRequest> message to the IDB. The next listing illustrates the schema fragment of such a request:

```xml
<xs:complexType name="ProxyDelegationRequestType">
  <xs:sequence>
    <xs:element ref="samlp:IDPProvidedNameIdentifier" minOccurs="0" maxOccurs="1"/>
    <xs:element ref="ProviderID" minOccurs="0" maxOccurs="1"/>
    <xs:element ref="ProxyCertificateRequest" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

The message must be signed by the GridSP. The elements of the message are as follows:

- **<Extension>** (Optional) - Like in any Liberty ID-FF protocol, this is an optional container for protocol extensions established by predefined agreements between providers.
- **<ProviderID>** (Required) - The identifier of the delegatee provider (GridSP’s identifier).
- **<IDPProvidedNameIdentifier>** (Required) - This element holds the name identifier the IDB uses for exchanging information about the users and their identity attributes.
- **<ProxyCertificateRequest>** (Required) - This element contains the proxy certificate request conforming to the PKCS #10 v1.7 [11] standard. The request itself is encoded in the Privacy-Enhanced Mail (PEM) [12] format.
- **<RelayState>** (Optional) - As in other Liberty ID-FF protocols, this element contains state information that will be relayed back in the response.

The recipient IDB of a request message must first validate the signature. The provided signature must be made by the provider contained in the **<ProviderID>** element. If the signature is valid, the IDB checks next whether the user contained in **<IDPProvidedNameIdentifier>** element has authorized the IDB to delegate his proxy to the requesting GridSP. A user is in control over the identity attributes that are used for the authorization decision. If the IDB is authorized, it continues to the fourth phase of the delegation process and signs the proxy certificate request.

The IDB responds to a request with a <ProxyDelegationResponse> message. The type of the response message is an extension of the samlp:ResponseAbstractType and its schema fragment is illustrated in the following listing:

```xml
<xs:complexType name="ProxyDelegationResponseType">
  <xs:sequence>
    <xs:element ref="samlp:Status" minOccurs="0" maxOccurs="1"/>
    <xs:element ref="ProviderID" minOccurs="0" maxOccurs="1"/>
    <xs:element ref="ProxyCertificate" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="RelayState" minOccurs="0" maxOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

Likewise, the response message must be signed by the IDB. The elements of the message are as follows:

- **<Extension>** (Optional) - Like in any Liberty ID-FF protocol, this is an optional container for protocol extensions established by predefined agreements between providers.
- **<ProviderID>** (Required) - The identifier of the delegator provider (IDB’s identifier).
- **<ProxyCertificate>** (Optional) - This element contains the proxy certificate chain encoded in Privacy-Enhanced Mail (PEM) [12] format.
- **<Status>** (Required) - The status of the request processing.
- **<RelayState>** (Optional) - This element contains the state information if it has been supplied in the request.

The recipient GridSP of the response must first validate the signature of the message. The signature must be made by the provider contained in the **<ProviderID>** element. If the signature is valid, the GridSP checks next the contents of the **<Status>** element. The alternatives for the values for the `<samlp:StatusCode>` in the element are as follows:

- **samlp:Success** - The request was successful and the delegated proxy certificate can be obtained from the element hosting proxy certificates.
- **samlp:InvalidSignature** - The signature of the request message was invalid.
- **samlp:NotAuthorized** - The recipient of the original proxy delegation request (IDB) was not authorized to delegate a proxy to the requestor (GridSP). This alternative is also used in situations where a proxy certificate cannot be delegated to the requestor, including inability to retrieve the user certificate and/or private key from the IDB’s database.

If the protocol ended affirmatively, the delegatee, i.e. the GridSP, now has both the proxy certificate and its corresponding private key, i.e. the user proxy.

C. Protocol Profiles Definition

protocol profiles. They apply to the ones given in this paper too in addition to the other common requirements defined in the same section. Our proxy delegation protocol profiles are described on purpose in the same fashion as the standard Liberty ID-FF protocol profiles in [13].

A GridSP can initiate the proxy delegation protocol by using a HTTP-Redirect-based or a SOAP/HTTP-based profile. Two profiles are provided like with the existing Liberty ID-FF protocols.

The first profile relies on an HTTP 302 redirect for the communication between the GridSP and the IDB. This means that all the protocol messages pass via the end user’s client device, the user agent, and the user is redirected to the IDB with the delegation request. In principle, this enables user interaction during the proxy delegation, such as typing in the passphrase of the private key coming from the IDB and used for signing the proxy certificate. If implemented, this would further improve security as the IDBs wouldn’t need to store end users’ private keys in clear text. On the other hand, the end users would have even stronger control on the delegation of their credentials.

The second profile relies on a SOAP (Simple Object Access Protocol) call from the GridSP to the IDB. SOAP is also used in the response. The SOAP profile does not involve the user agent for the message delivery, but rather the providers call each other directly. Any direct user interaction with the IDB is not thus possible as the user agent remains communicating with the GridSP only.

Both profiles make use of the elements defined in the Liberty Metadata Description document [14]. Among other things, the document describes the SOAP endpoint location of the desired provider to which Liberty SOAP messages are sent [13]. In addition to that, the protocol requires three further elements:

- **ProxyDelegationProtocolProfile**: The GridSP’s preferred proxy delegation protocol profile, which should be used by the IDB for this protocol. It is referred to with a URI-based identifier.
- **ProxyDelegationServiceURL**: The URL used for the HTTP-Redirect-Based proxy delegation protocol profile.
- **ProxyDelegationServiceReturnURL**: The GridSP’s redirecting URL for use after the HTTP-Redirect-based proxy delegation has taken place.

1) **HTTP-Redirect-Based Profile**: The profile relies on using HTTP 302 redirects to transport the protocol messages between the GridSP and the IDB. The transaction is illustrated in Figure 2:

1) The GridSP redirects the user agent to the proxy delegation service at the IDB. The Location HTTP header must include: (1) the IDB’s proxy delegation service URL with https as the URL scheme and (2) a [query] component containing the protocol request message as defined in Section V-B and with the formatting specified in Section 3.1.2 of the Liberty ID-FF Bindings and Profiles Specification [13].

2) The user agent accesses the IDB’s proxy delegation service URL in order to deliver the proxy delegation request to the IDB.

3) The IDB processes the request message with the rules that were defined in Section V-B.

4) The IDB redirects the user agent to the proxy delegation service return URL at the GridSP. If the request message sent in step 2 contained a <RelayState> element, the IDB has to include the same element in the response message. Now, the Location HTTP header must include: (1) the GridSP’s proxy delegation service return URL with https as the URL scheme and (2) a [query] component containing the protocol response message as defined in Section V-B and with the formatting specified in Section 3.1.2 of the Liberty ID-FF Bindings and Profiles Specification [13].

5) The user agent accesses the GridSP’s proxy delegation service return URL.

6) The GridSP processes the response message with the rules that were defined in Section V-B.

2) **SOAP/HTTP-Based Profile**: The profile relies on using SOAP over HTTP to transport the protocol messages between the GridSP and the IDB. The transaction is illustrated in Figure 3. The steps utilize the SOAP binding for Liberty as defined in Section 2.1 of the Liberty ID-FF Bindings and Profiles Specification [13]:

1) The GridSP sends a proxy delegation protocol’s request
message to the IDB’s SOAP endpoint.
2) The IDB processes the request message with the rules that were defined in Section V-B.
3) The IDB sends a proxy delegation protocol’s response message to the GridSP’s SOAP endpoint. The <RelayState> element is included only if it was included in the request at step 1.
4) The GridSP processes the response message with the rules that were defined in Section V-B.

VI. Conclusions
In this paper, we have proposed a design that simplifies the usage of Grids. It is an extension to the existing identity federation framework and enables the management and issuance of Grid proxies to be outsourced to a trusted third party, an IDB. Also, the user proxy can be delegated from the IDB to Grid portals (GridSPs) where it can be used for accessing Grid resources on behalf of the user. The deployment of our extension requires no modification of the underlying Grid security infrastructure.

Similar functionality is offered by the MyProxy server, but every proxy delegation procedure to a portal forces the user to re-authenticate to the MyProxy server. This has been identified as a major problem in use cases which involve a client using a Web portal to access the Grid services [15].

In our case, the user authentication is required only once at the IDB to enable SSO to the GridSP as well as the other SPs inside the CoT. Also, the proxy delegation procedure from the IDB to the GridSPs can be performed as many times as desired after the single authentication at the IDB. This allows the end users to access resources in independent Grids with a single login (i.e. SSO to several Grids) as long as the independent Grids (GridSPs) are members of the same CoT. This assumes of course that the user possesses proper credentials and is entitled to access each of those Grids individually. Different Grids may require proxies based on different certificates, but they can all be stored at the IDB with the user still retaining complete control of their delegation authorizations.

VII. Future Work
The current proxy delegation protocol is a proposed extension to the Liberty ID-FF. However, as the standard Liberty ID-FF functionality is already included in the SAML v2.0 specifications [16], they are not being further developed. The mechanisms for enabling the proposed protocol’s functionality with SAML v2.0 specifications should be investigated. For example, the proxy certificate subject DN distribution in Section V-A could be implemented using SAML attributes instead of the extension fields of the existing protocol. The IDB service should then act as a SAML Attribute Authority.

On the other hand, the GridShib project [17] is working on the integration of Shibboleth [18] and the Globus Toolkit. Like Liberty Alliance, Shibboleth is based on SAML. The assertions or the address of the attribute authority are attached to the proxy certificates. They are used at the Grid resources for authorization purposes. The IDB could attach this information to the proxy certificates too. Also, other authorization systems like the Virtual Organization Membership Service (VOMS) [19] could be supported by the IDB in the future.

ACKNOWLEDGMENTS
This work has been done as a part of the NetGate project [20] mainly funded by Tekes, the Finnish Funding Agency for Technology and Innovation. The work of Mika Silander is supported by the Academy of Finland grant 205961.

REFERENCES

0-7695-2622-5/06/$20.00 (c) 2006 IEEE