



Identity & Access Management using Social Networking Technologies

Final Report


Project Information			
Project Acronym	FOAF+SSL+SHIB		
Project Title	Identity & Access Management using Social Networking Technologies		
Start Date	01/01/2010	End Date	30/09/2010
Lead Institution	The University of Manchester		
Project Director			
Project Manager & contact details	Mike Jones Address: Research Computing Services, The University of Manchester, Manchester M13 9PL Email: mike.jones@manchester.ac.uk Tel: +44 161 275 7038 Fax: +44 161 275 6120		
Partner Institutions			
Project Web URL	 Web: http://www.rcs.manchester.ac.uk/research/FoafSslShib Wiki: http://wiki.rcs.manchester.ac.uk/community/FoafSslShib		
Programme Name (and number)	<i>Access and Identity Management: Innovation 08/09</i>		
Programme Manager	Chris Brown		

Table of Contents

Acknowledgements.....	2
Executive Summary.....	3
Background.....	4
Aims and Objectives.....	6
Methodology.....	6
Implementation.....	7
Outputs and Results.....	8
Outcomes.....	9
Conclusions.....	9
Implications.....	9
Recommendations.....	9
References.....	10
Appendixes.....	11
A) FOAF+SSL Globus Authentication and Authorisation.....	11

Acknowledgements

This project was funded by JISC as part of the Access and Identity Management: Innovation 08/09 Programme.

We would like to acknowledge Henry Story the lead on the FOAF+SSL activity who provided technical input as well as dissemination of this work.

Executive Summary

1. This project demonstrates that authentication based upon social networks as implemented by FOAF+SSL (Now renamed WebID) could technically be applied as an optional mechanism to access services on the NGS grid and within the UK Access Management Federation environment. The technology has the capacity to allow individuals to express their identity securely, without restricting them to a specific set of identity providers.
2. There are a number of caveats to applying this technology:
 - It is still relative immature, the libraries are still incomplete and the specification is still in development and therefore should be considered volatile/experimental.
 - The grid middleware to which we have applied this technology needs modification albeit only minor. This project has delivered a prototype which includes these modifications
 - The grid middleware also places certain constraints on how FOAF+SSL credentials are manufactured. Partial support can be achieved within Globus based middleware without further modification by using an approach which uses a stripped down Certificate Authority.
 - Levels of identity assurance apply differently to this system than to both grid and shibboleth federation in that there is no registration process *per se* in FOAF+SSL.
3. The advantages of using this system:
 - Lack of a formal registration process creates a more user friendly environment.
 - User control of identity means that identity is not linked or limited to any specific institute.
 - It is easy to revoke one's credentials, by modifying or removing references to key material in the one's FOAF file. In a similar manor relationships such as VO membership any corresponding roles can be revoked.
 - Transient relationships such as dynamic virtual organisations can be created.
4. While we are very excited with the possibilities this system can offer, we make the recommendation that this technology is not currently mature enough for production environments, such as the NGS.
5. We recommend that the community should continue to monitor the development of this technology. If this technology is to be further exploited in its current state, any such work should consider the following additional work packages:
 - The grid community should set up a WebID Service and seek IGTF accreditation.
 - The reference implementations developed for this project, if used, should be enhanced to better evaluate the social network graphs.
 - The functionality of any developing WebID server should be enhanced such that it will provide the ability for the individual to target specific services (a basic attribute release policy) and perhaps the ability to release or withhold information for those targeted services (fine-grained attribute release).
 - The applicability demonstrated in this project should be determined for other middleware which rely upon delegation such as gLite, UNICORE, etc.

Background

6. This project was undertaken to address a number of issues in current UK e-Infrastructure Identity Management environment. The focus was to create a flexible, scalable system of authentication and credential management which could be slotted into the existing Identity Management systems.
7. The two primary issues to be addressed were:
 - To remove the onerous registration processes involved in gaining access to grid resources (see e.g. [1] and [2] for discussion).
 - To tackle the barriers introduced by rigid membership structures in the UK federated access management system. Within this system there is currently a need to belong to a recognised Identity Provider usually part of a larger federation. The scope of any assertions are necessarily limited to the authority of those bodies. As such individuals can only be authenticated based upon their roles within these bodies: no affiliation means no authentication, and multi-site collaborations do not fit well into this model.
8. The project was formulated when the concepts of Virtual Organisations, Social Networks and Web Security were broached in a paper presented at the 6th European Semantic Web Conference [3]. At this point the work was called FOAF+SSL, (later FOAF+TLS, and now WebID) reflecting the basic structure of the technology: a merger between FOAF: a vocabulary used to describe individuals and their relationships, and SSL: the Secure Socket Layer used to underpin secure transactions on the Web as well as other on-line protocols.
9. The principles behind the system are fairly simple. In FOAF every entity has a unique identifier (a WebID) which is essentially a URI (for example <https://foaf.me/nimp0#me> is a URI which describes one of the authors). The FOAF vocabulary allows one to make various assertions about that WebID: who or what it represents, its properties and its relationships with other entities. In FOAF+SSL the specific assertion which provides the basis for Identity Management is that the URI is linked to a public key suitable for use in mutually authenticated SSL protocol handshakes.
10. To make use of this, FOAF+SSL relies on the fact that FOAF files are discoverable, in fact directly so, as the URI translates into a resource on the Web. By pointing a browser at the URI <https://foaf.me/nimp0#me> the browser would retrieve a rendering of the FOAF file located there which describes that entity.

Authentication

11. To achieve an authenticated access to some on-line service using FOAF+SSL the user connects to the service using a private key and X509 certificate (which may or may not be self-signed). So far there is nothing unusual about this; it is part of the standard SSL handshake. However, the user needs the service to understand who they are so they may do something useful while connected. The server needs to discover the user's FOAF file and verify the connection through this. To do this the user will have placed the public key of the certificate in their FOAF file and inserted their WebID into the certificate:
 - In the certificate this is achieved by using the Subject Alternative Name extension¹, which allows the certificate issuer to assert other identities for the certificate owner specifically in this case a URI which will be their WebID.

1 <http://www.ietf.org/rfc/rfc2459> section 4.2.1.7

- In the FOAF file the public key is stored broken down into its simplest representation, the modulus (usually in hexadecimal) and the public key (usually in decimal).
12. The service on receiving the certificate extracts the alternative name, resolves the WebID gains the FOAF file and parses it to find the public key which it then uses to conclude client verification within the SSL handshake.
 13. The result of a successful basic FOAF+SSL authentication is that the service knows that the client has the ability to place data in the location described by the WebID URL. Compare this to the grid where the service would know that the client had registered with a CA, or Shibboleth where the service would know that the client was known by an IdP within the federation:
 - WebID: `https://foaf.me/nimp0#me`
 - Distinguished Name: `/C=UK/O=eScience/OU=Manchester/L=MC/CN=michael jones`
 - SAML Assertion: `<saml:NameIdentifier>
_d6b5576a6a7fbcf49ee342cfa2eef9a1
</ saml:NameIdentifier>`
 14. Of course, the above examples describe just the primary authentication, i.e. the service can communicate securely with an entity and that entity has a unique handle. Real services require two more things: Trust and Authority.

Trust

15. Trust is arguably the hardest part of any digital security system. In the grid trust is established through Certificate Authorities (CAs); the equivalent for Shibboleth protected services are called Identity Providers. In order to reduce the burden of each service having to examine each of these *Trust Anchors* a Federation exists in both paradigms whose role it is to establish and assert the authenticity and compliance to a set of policies. The upshot being that each service essentially trusts either the UK federation or the International Grid Trust Federation to regulate/audit its members. In FOAF+SSL the IdP/CA entities are the users themselves, removing the central federation.
16. Each Trust Anchor will assert certain attributes about a registered entity. The important point is the scope of the assertions. A CA will generally assert that the entity possesses a private key and provides a Distinguished Name; the Grid CAs will scope any DN they assert to a certain name space in agreement with the IGTF. An IdP will assert whether an entity connected to the service is known, it may provide any other assertion it feels able to but will generally only provide a targeted persistent identifier and perhaps a generic Role at that organisation. FOAF+SSL provides only one assertion: the WebID, the rest of the FOAF file provides self asserted information but this is verifiable through the Social Web.
17. At this point the reader may feel that WebID possesses a flaw: How does the service trust the location of the WebID that it is modifiable only by the user. It is true, that the user may choose to locate their WebID at a URI for which they do not have full control. The issue of how a user controls their identity credentials is unfortunately often the weakest link in any security system. Users may choose weak passwords, share passwords, lose private key material *share* private key material, etc. The service provider must at some point assume some level of user key management. The user will have probably signed a terms and conditions during the registration process with the IdP/CA; for FOAF+SSL this agreement is directly between the service and the user and needs to be implicitly so.

18. The FOAF+SSL assertion however only states that the entity has write access to the WebID URI. It does not involve an explicit third party.

Authority

19. Authorisation to use a service relies on the mapping between the identity and various attributes, and the mapping between possession of attributes and allowed actions.

- For Shibboleth protected services the Service Provider may be presented with all the required attributes or may try to obtain them directly from the Identity provider.
- For the grid these attributes often come in the form of signed assertions from a Virtual Organisation. In grids, unlike Shibboleth, authorisation is devolved from identity management.
- In FOAF+SSL these assertions may be from individuals, organisations, virtual organisations, other services, anything that may be expressed using RDF and placed in a Web accessible resource. These attributes will generally not be signed, instead they are statements of relationships. To resolve them the user's FOAF file will describe the relationships using the WebID of the other entities. From this a graph can be constructed. The verification of any assertion is achieved by finding a reciprocal statement in the other entities' RDF.

20. It is the job of the entity that provides the assertions to agree with the service provider some level of service for its community. This is true for Institutions (i.e. IdPs), and for VOs whether expressed via usual grid methods or via FOAF.

21. Advantages of using FOAF+SSL over Shibboleth and grid mechanisms:

- Registration – no onerous matriculation process need be completed.
- Affiliation – not institutional but individual.
- User Mobility – no need to renegotiate each user credential each change of affiliation.
- Attribute Release – user control of what is released how much is released and potentially to whom.

22. FOAF+SSL² is an emerging technology that aims to address the issues related to security in social networking environments, so as to be able to leverage the flexibility of the models highlighted above in a secure way. In particular, its initial use cases are the solution of problems relating to the granularity of access, disclosure of data and trustable self-assertion of identity. It also makes use of links between the users and their social network to build a web-of-trust (which is then used by services to make authentication and authorisation decisions). Although FOAF+SSL is quite a new approach, it has already gained some visibility in the semantic web community and been implemented in a number of existing services (such as wikis³ and micro-blogging⁴).

23. This project demonstrates the benefits that can be gained by drawing on the strengths of social trust models. The FOAF+SSL methodology will be applied to authentication and authorisation middleware used by two large JISC initiatives which rely heavily on the Access and Identity Management programme: the UK NGS and the UK Access Management Federation. By doing so, we believe that we have demonstrated practical solutions to issues of user centricity, granular access policy, and delegation of authority.

2 <http://esw.w3.org/topic/foaf+ssl>

3 <http://trunk.ontowiki.net/>

4 <http://foaf.me/shout/>

Aims and Objectives

24. This project's aims were to demonstrate the benefits of emerging social networking technologies for access and identity management in the context of UK academia, more specifically for the UK Access Federation and for the National Grid Service.

Methodology

25. The project undertook to develop two demonstrators. FOAF+SSL is a developing technology. In order to evaluate its potential suitability for the UK community it was necessary to show that the system would work when applied to the two environments where the UK Access Management Federation currently struggles to provide functionality, and where grid security has been described as onerous. It was also necessary to provide an implementation that would slot into these environments without modifying the middleware stacks.

Implementation

26. The implementation was decomposed into two common components and two parallel streams. The common components were the certificate verification library and the WebID certificate hosting and creation service. These were used to support the usage and implementation following two streams: the Shibboleth IdP and the Globus bridge.
27. The development process of all of these components was iterative. By essence, this technology relies on a number of services interacting with each other. Where possible, some early implementations of those services (e.g. the WebID repository) were available in the FOAF+SSL community, so we tried to make use of them as far as possible until new features were required, also taking into account the fact that those services changed outside our control.

Integration with Shibboleth

28. The initial implementation steps consisted of producing a Shibboleth IdP capable of authenticating a user via a WebID certificate. This IdP was built in a Servlet environment and relies on the OpenSAML library (from Internet2, as is the reference Shibboleth implementation) to process SAML messages, at the heart of Shibboleth. The first implementation did not provide an attribute request service. However, it was able to pass simple assertions to the service provider via the browser. We were able to make it interact with the Shibboleth module for Apache Httpd and with the simpleSAML library (a PHP implementation which can be used as a Shibboleth service provider library independently of the Internet2 implementations).
29. The verification of the WebID certificates was done using the FOAF+SSL Java library developed in parallel, mainly by Henry Story (formerly from Sun Microsystems, lead architect of FOAF+SSL) and Bruno Harbulot. Because the specifications of FOAF+SSL were not stable at the time (it has only been going through a draft standardization process since July 2010), some of the API and code was evolving during this A&IM project, which had some impact, albeit limited, on the implementation of the IdP.
30. The main impact of the changes in the FOAF+SSL specification seem to have come from the implementation of the WebID repositories, provided by a number of contributors in the FOAF+SSL community. Because these services were mainly experimental and not under our control, it was difficult to predict which and when changes would be made; in addition, since these services were provided by community members on a friendly best-effort basis, they were not always available when needed. As a result, we had to implement a WebID repository service slightly earlier than we anticipated, but this was beneficial to the project as we had a more stable environment to evaluate the technology

and we were able to produce a system that enables richer WebID profiles, more adapted to the goals of the project, with respect to the Grid and UK Access Federation use-cases.

Certificate Management

31. Since FOAF+SSL relies on certificates, we also made an effort to make dealing with certificates as easy as possible for the end-user. Thus, we looked into in-browser generation of keys and certificates, for both a Microsoft Internet Explorer environment and for other browsers, more particularly Firefox and Chrome. While this technology is already in use in Certification Authorities such as the UK e-Science CA, our system allowed for a simpler flow for the user, since, by nature of FOAF+SSL certificates, the validation of the identity is less stringent at the point of issuance.
32. The initial tools for FOAF+SSL used self-signed certificates (since the signature in the conventional PKI terms is not relevant). However, browser key-generation mechanisms do not allow for generation of self-signed certificates. Instead, we used a service that issues certificates instantly based on key-pairs generated within the browser. This effectively works as a CA that does not verify anything at the issuance point. Browsers such as Firefox and Opera rely on the HTML <keygen> element to do this (this is a non-standard element introduced by Netscape); this element has been integrated in the HTML 5 specification (although none of our tools use HTML 5 yet). Microsoft browsers do not support this element (and have publicly said they had no intent to support it with HTML 5⁵); on this platform, the key-generation mechanism uses ActiveX controls via JScript (in turn, the controls depend on the version of Windows where Internet Explorer is running: XP or Vista/7).
33. In addition, to illustrate the ability for WebID repositories to publish information asserted by a third party, the WebID repository was also extended to connect to an institutional LDAP server. While this is not the general use-case for the technology, it can be envisaged that institutions could provide their own WebID repositories, so as to be able to interact with other institutions or individuals. In this example case, we linked the WebID repository to the institutional identity management system so as to enable the repository to assert various attributes regarding individuals (if they were able to tie their WebID to their institutional identity).
34. As well as ease of certificate management, the provision of a WebID repository allows the system to provide an additional assertion about Level of Authentication this may be useful when addressing issues discussed in paragraph 17 on page 5.

Integration with NGS

35. FOAF+SSL certificates are standard X.509 certificates. The Grid Security Infrastructure, used by NGS Globus based middleware, relies upon the user possessing an X.509 certificate using it to authenticate to grid services. Globus client side tools do not need to derive the trust of a user's certificate so it was only necessary to provide a bridge between grid services and the FOAF+SSL authentication libraries.
36. The main grid services provided by the NGS are based upon a hybrid of Globus and gLite middleware. In order to evaluate Virtual Organisation membership of a user NGS makes use of gLite's LCAS/LCMAPS⁶ libraries. These gLite libraries provide a number of modules with which to make authorisation decisions. The project initially envisaged writing a new plugin module to this environment. However, much of the emerging FOAF+SSL tools are written in Java whereas LCAS/LCMAPS is written in C, so a callout to a separate Web Service Policy Decision Point (PDP) was developed. The Site Central

5 See e.g. <http://lists.w3.org/Archives/Public/public-html/2009Aug/0389.html>

6 https://www.nikhef.nl/pub/projects/grid/gridwiki/index.php/Site_Access_Control

Authorisation Service⁷ (SCAS) LCMAPS plugin provides a means to talk with such a service using XACML2, SAML2 and SOAP. Further development of this was needed as FOAF+SSL requires the full user certificate and neither LCMAPS nor SCAS propagate this information.

37. The Web Service *FoafAuthz* developed to handle the request, verify the FOAF+SSL session and establish the Social Network graph was implemented within a Jetty⁸ web service container. It is comprised of four components, one to handle incoming requests, one to verify the session, one to process the RDF and one to cache FOAF files.
38. Globus unfortunately only allows authorisation to be handled outside of the gatekeeper, performing the authentication internally. This has two side effects:
 - Self signed certificates do not work. Globus requires that a Trust Anchor be present on the system: All CA hierarchies must be present otherwise preliminary authentication checks fail.
 - FOAF+SSL does not use the Subject Field as a rigorous name the way GSI does. Leading to the possibility of clashes in GridMapfile based authorisation.
39. These two issues can be resolved by the required use of a WebID repository as was developed in the other parallel stream. Firstly, the CA for this service may be distributed in a similar method to those in the International Grid Trust Federation. Distinguished Name issuance can be controlled so as to avoid clashes. Furthermore the use of such a repository can be controlled in such a way as to provide assurances about the write access to a URI (see issue paragraph 17, page 5). Other methods of incorporating FOAF+SSL into Globus would be to provide a patch to the Globus distribution, although previous attempts to do so by the community have resulted in a fork of software (e.g. VDT distribution of Globus).
40. Further detailed information on the NGS evaluation can be found in Appendix A.

Outputs and Results

41. The project has produced a WebID repository, a WebID/FOAF+SSL to Shibboleth bridge and a WebID/FOAF+SSL to Globus bridge. In addition, we also enhanced the FOAF+SSL verification library as well as provided input into the establishment of a WebID/FOAF+SSL specification (this process is still on-going). Software outputs can be found as follows:
 - Web Framework <https://github.com/harbulot/corypha>
 - WebId Repository <https://github.com/harbulot/webidrepository>
 - Shibboleth Bridge <https://github.com/harbulot/foafsslshib>
 - Patches to the LCAS/LCMAPS/SCAS middleware were delivered to the gLite developers
 - Globus Bridge Code can be found via the project page
 - Installation Instructions can be found on the project wiki <http://wiki.rcs.manchester.ac.uk/community/FoafSslShib>

Outcomes

42. While we have demonstrated that the objectives set out by the project are technically possible, by providing implementations to support it, some non-technical issues remain. The expressiveness of the semantic web is powerful and open and, in the same way as SAML assertions could potentially be about anything, there would need to be an

7 <https://www.nikhef.nl/pub/projects/grid/gridwiki/index.php/SCAS>

8 <http://jetty.codehaus.org/jetty/>

agreement in the same way as there is an agreement within a Shibboleth federation regarding the semantics of the attributes.

Conclusions

43. It is possible to build both a Shibboleth IdP and a Globus Authorisation Module using FOAF+SSL as the primary method of authentication. There are a number of technical requirements for the credentials and these services to overcome in order to be able to use FOAF+SSL with grid middleware based upon Globus at this point in time. We have not evaluated other middleware outside of the current NGS software stack.

Implications

44. It has been noted that it is in essence possible to use this technology within the UK's eAuthentication frameworks. The technology opens up a much larger range of terms for describing relationships, entitlements and preferences. Possible future directions to exploit may be.

- **WebID Service:** To encourage compatibility of FOAF+SSL certificates with grids the eScience community could host a WebID service. This could also provide certain assurances to the consuming services which would help them to trust the model better. For example, a hosted service can provide better control of who edits whose FOAF files; it might also impose an entry EULA which places requirements on users to comply with grid Acceptable Usage Policies; it could also manage the distribution of the CA credentials.
- **W3C WebID Incubator:** WebID is not a completed standard, we intend to partake in the W3C WebID incubator as it progresses.
- **Graph evaluation:** In this project the evaluation of FOAF files created simple graphs based only upon membership of a group WebID. This may be sufficient for the standard VO authorisation environment. However, expanding the graph evaluation will allow for a much richer authorisation space, for example: Dynamic VOs, delegation of authority, etc.
- **Preferences rules via FOAF File:** In our test environments we have provided no facility for an individual who satisfies a number of authorisation conditions. Allowing a user to specify a set of rules in their FOAF file for services or groups thereof would provide a mechanism to do so.
- **Data release policies:** Users may prefer not to release information about themselves except to specific entities. This would need to be expressed and included in any WebID service functionality; it also has the possibility to aid preference rules (above).
- **Other middleware:** How will FOAF+SSL work in a gLite environment? How will this work with Workload Management Systems e.g. gLite WMS and different submission endpoints e.g. CreamCE. This project was limited to NGS baseline services.
- **IGTF Accreditation:** If a WebID service were to be provided what would be required for it to get IGTF accreditation.

Recommendations

45. Code base is too volatile at this point and the current implementations of FOAF+SSL are too immature to base any production service on top of.

46. The concept however contains implicit features for much deeper relationship management and expression than the monolithic-in-comparison VO architectures of grid infrastructures and the institutional structures of current federated ID management environments. Therefore the progress of this activity should remain monitored and future

R&D activities with a requirements for a pervasive easy-to-use authentication and authorisation should consider adopting this technology.

References

[1] Beckles, B., Welch, V. & Basney, J. 2005 Mechanisms for increasing the usability of grid security. *Int. J. Human-Computer Studies*, 63(1-2), 74–101. (doi:10.1016/j.ijhcs.2005.04.017)

[2] Martin, A. & Spence, D. 2008 Trust and security in virtual communities, report on first workshop:the application-led security agenda for escience. Workshop report, University of Oxford.

[3] Henry Story et.al. *FOAF+SSL: RESTful Authentication for the Social Web, Proceedings of the ESWC2009 Workshop on Trust and Privacy on the Social and Semantic Web (SPOT2009)*, Heraklion, Greece, June 1, 2009.

Appendixes

A) FOAF+SSL Globus Authentication and Authorisation

Introduction

47. The Globus based services: gatekeeper (GRAM), GSISSH and GridFTP use certificates for authentication and authorisation. Can FOAF+SSL certificates be used for that purpose? The answer is yes, but with restrictions:

- Globus makes use of GSI-proxy certificates (an extension to X.509 certificates⁹) to effect single sign-on, the user usually creates one – deriving it from their standard X.509 grid credential – before connecting to the service. The standard Globus tools for dealing with proxy certificates work with FOAF+SSL certificates, they can be used in the same way as normal certificates.
- The Globus server libraries, which are responsible for the secure communication between client and server, verify the client's proxy certificate. They require that the Certificate Authority (CA) certificate which has issued the client's certificate is installed on the server. This means that self-signed FOAF+SSL certificates will not work. Only certificates that were issued by a CA that is supported by the server can be used.
- The verification of the client's proxy certificate fails if there are any critical extensions in the certificate that are not supported by the Globus libraries. This includes the standard subject alternative name (SubjectAltName) extension which is used to store the WebID. Therefore only FOAF+SSL certificates which have this extension marked as non-critical will work – this is usually the case.

48. If the above requirements are satisfied then FOAF+SSL certificates can be used for authentication / authorisation with Globus.

Globus Authentication and Authorisation

49. The standard Globus authentication and authorisation is very basic. A client is authenticated by verifying its proxy certificate when the connection to the server is opened. The authorisation is done by searching for the Distinguished Name (DN) of the certificate in a map file which maps it to a local user account. Globus allows these mechanisms to be either replaced (authorisation, map files) or extended (authentication, client certificate verification) by configuring additional plugins as the standard mechanisms are not always sufficient. One of the plugins, the GT4-LCAS/LCMAPS¹⁰ Interface, integrates the gLite Local Centre Authorization Service (LCAS) and Local Credential Mapping Service (LCMAPS) into Globus. LCAS and LCMAPS themselves are frameworks which allow a dynamic configuration of authorisation and authentication modules and therefore the functionality of their standard modules can easily be extended.

50. FOAF+SSL certificates may be issued by a Certificate Authority (CA) or self-issued, those that are self-issued are not wholly compatible with the standard functionality of Globus and LCAS/LCMAPS: Globus libraries require all the signing authorities' certificates to be present on the server. However, this still leaves credentials that are issued by some CA.

9 [IETF RFC 3820](#)

10 GT4-LCAS/LCMAPS was contributed by The University of Manchester on behalf of the NGS to the gLite project.

51. Care is necessary when installing CA certificates as *trust anchors* on grid services because grid environments place stringent requirements on identity management, registration and naming. Most CAs (Commercial and ad-hoc CAs) are not generally considered trustworthy for grid purposes as they issue certificates to anyone who asks for them: resulting in either many entities to one DN mappings or a weak link between the credential and the individual. Therefore if such CAs are used the client cannot be considered authenticated (the DN is no longer sufficient to identify an individual) even if with a valid certificate unless the FOAF+SSL mechanism is being used. Globus has a mechanism which stops authentication taking place if a CA issues a credential outside of a configured namespace thus providing functionality to run FOAF+SSL alongside the standard authentication mechanisms, if the FOAF+SSL CA in question issues certificates in such a namespace. The WebID service produced as part of this project issued credentials of the form “/CN=<some name>”, this was sufficient to write the necessary rules for testing as there are no other Grid CAs issuing certificates whose DN begins with “/CN=”, although FOAF ME now issues credentials of the form “/CN=FOAF ME Cert <some WebID>” which is perhaps better as a regular expression rule can be constructed tying that CA to that namespace.

FOAF+SSL Authentication and Authorisation modules

52. An authentication module which verifies the FOAF+SSL certificate and an authorisation module which makes its authorisation decision based on the WebID and the FOAF file hosted at that location have been developed to use FOAF+SSL certificates with Globus. LCAS/LCMAPS provides a much more powerful framework for authorisation and authentication modules than Globus, therefore the choice was to use an LCMAPS module. LCMAPS modules are responsible for the mapping of a client credential to a local user account. The nature of the mapping process takes care of the authorisation as well and therefore no LCAS module is required.

53. Integrating the whole authorisation and authentication for FOAF+SSL certificates into an LCMAPS module would mean that the developed code would be tied to the LCMAPS framework. LCMAPS modules have to be implemented in C/C++, but most of the existing code that deals with FOAF+SSL and RDF is written in Java. It was decided that, instead of integrating all code into a LCMAPS module, it was preferable to implement a FOAF+SSL authentication and authorisation web service (FoafAuthz) that acts as a policy decision point (PDP). An LCMAPS module then delegates the authorisation decision to this service and afterwards acts as a policy enforcement point (PEP) by enforcing the returned result. The communication between the LCMAPS module and the FoafAuthz service uses SAML2 and XACML2 in a SOAP message that is sent via HTTPS. A major factor in that decision was that there is an LCMAPS module that is based on the same technologies to communicate with a Site Central Authorization Service (SCAS). The XACML profile¹¹, which is used by SCAS, was defined by participating members of a collaboration between the VO Services project (for OSG), EGEE, EGEE-INFN, Globus, and Condor. The profile allows all information that are required by the FoafAuthz service, but the LCMAPS SCAS module does not provide the optional certificate attribute. Therefore modifications to LCMAPS allowing the certificate chain to be included were necessary.

Design of the FoafAuthz web service

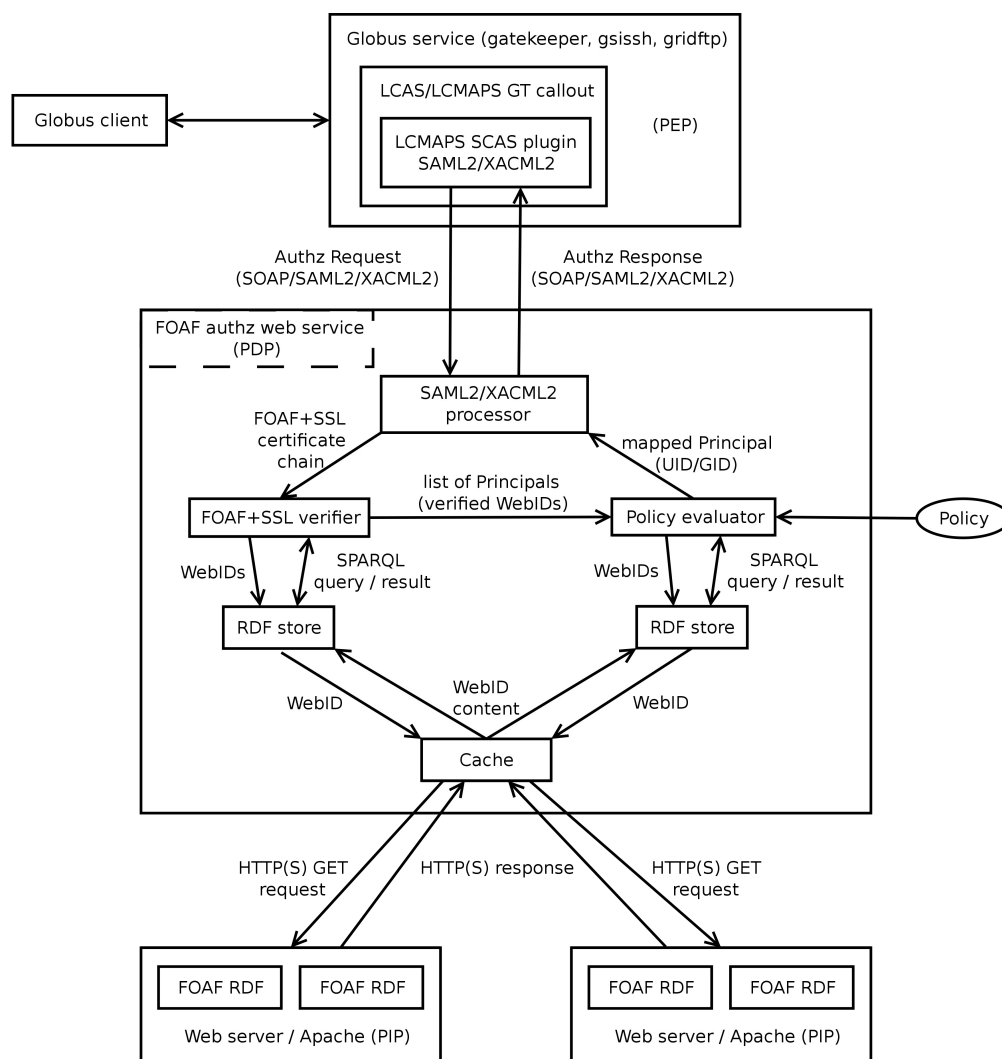
54. This section describes the overall design of the FoafAuthz web service. Some of the more advanced features described here could not be implemented within the scope and time-scale of this project (paragraphs 62 detail what was implemented).

11 <https://edms.cern.ch/document/929867>

55. The FoafAuthz web service is responsible for authenticating the FOAF+SSL certificate and making an authorisation decision based on a policy. The three main parts of the web service are the SAML2/XACML2 processor, the FOAF+SSL verifier and the policy evaluator.
56. **The SAML2/XACML2 processor** parses the SOAP message which it receives from the LCMAPS SCAS module and extracts the SAML2 and then the XACML2 part. It iterates over the subject attributes inside the XACML2 message and passes each certificate of every found certificate chain to the FOAF+SSL verifier. The processor uses the OpenSAML 2 library¹² for processing the message.
57. **The FOAF+SSL verifier** searches for all WebIDs in the certificate. It downloads the content of the FOAF RDF file from the location specified by the WebID and adds it to an RDF store. The data is retrieved via a shared data cache to reduce network traffic and to allow a time limited use of the cached content in case of networking problems. A new in-memory RDF store is used for each WebID to prevent that content of other FOAF files interfere with the authentication process. The RDF store is queried for all public keys and each of them is compared to the public key of the certificate. If both public keys match then the authentication is successful and a new principal is created and added to a list of principals. This list is passed to the policy evaluator.
58. **The policy evaluator** itself is just an interface to allow an easy implementation of different policy evaluators. The actual evaluator that is being used is specified in the web services configuration together with the location of its configuration file. It takes the list of principals and, after processing its policy, it returns a user mapping (user ID, group ID) which will be enforced by the PEP later. A policy evaluator implementation has access to the shared cache, but it is responsible for creating and managing its own RDF store.
59. The example implementation of the policy evaluator uses a simple configuration file to map FOAF groups, which are specified by their own WebIDs (emulating VOs), to user IDs and group IDs. All group FOAF files are downloaded and put into an RDF store. The store is searched for each WebID from the list of principals. If one, and only one match is found then the user ID and group ID that are assigned to that FOAF group in the policy file are returned.
60. **The RDF store** provides an abstraction layer for RDF libraries. This abstraction layer allows the use of different RDF libraries. The main advantage in supporting multiple libraries is the better support for different RDF file formats. FOAF RDF files are created by the users themselves and it is up to them which file format they use. There are already a wide range of file formats in use to store RDF data and none of the libraries, which exists at the time of this project, support all of the formats. The web service can be configured to use multiple libraries, one of these is defined as the default library and provides the back-end store that is used by the web service. If a FOAF file uses a file format that is not supported by the default library, one of the additional libraries can convert the file into a format that the default store understands. The main disadvantage in using an abstraction layer is the limited functionality it can provide, as it is restricted to functionality that is common to all libraries.

12 <http://www.opensaml.org>

Illustration 1: FOAF+SSL Globus Authentication / Authorisation



61. **The cache** is responsible for retrieving the data located at a WebID and the caching of it. The cache handles WebIDs based on their Universal Resource Identifier (URI) scheme. A URI handler for each supported scheme is responsible for retrieving the data and providing it to the caller. Each handler should employ methods to retrieve the original data only if it has been modified since it was last accessed otherwise the cached version should be returned. For example the http and https handler, which are provided by the initial implementation, always request the data from the web server, but send Etag and modified headers with the request. If the web server supports either method, it only returns the full FOAF file if it has changed.

Current State of the FoafAuthz web service implementation

- 62. The FoafAuthz web service is implemented as an HTTP servlet which runs in an Jetty web service container¹³.
- 63. It uses the simple FOAF group policy mentioned in the description of the policy evaluator. The configuration file for this evaluator is the policy file itself. The policy is loaded when

13 <http://jetty.codehaus.org/jetty/>

the first instance of the servlet is created. At the moment the web service container has to be restarted to apply changes to the policy file. A future feature could provide means of monitoring the policy file and reload it automatically if it has been modified. The policy file uses the following format:

- “<WebID>”: <uid>, <gid>

64. Each line in the policy file contains one of the above mappings. Each WebID may only occur once. An example file could be:

- "https://www.kato.mvc.mcc.ac.uk/gridsite/foafssl/ngs": 10030,10030
- "https://www.kato.mvc.mcc.ac.uk/gridsite/foafssl/uom": 10031,10031

65. Invalid lines are ignored and cause a warning to be printed into the log file. All members of the group that are identified by a WebID are mapped to the same user account. Pool accounts are not supported in the current implementation. Only one valid mapping is allowed. If a user is a member of more than one group, which is being used in the policy file, then the mapping fails, as the policy evaluator cannot decide which mapping to use. The implementation of the simple FOAF group evaluator makes one assumption which has serious security implications: The FOAF data that is located at the group WebIDs, which are specified in the policy file, is trusted entirely. This is similar to the trust level that is given to a normal CA and means that the evaluator will accept membership assertions that a group makes for a different group, e.g. to use the example above, if *https://www.kato.mvc.mcc.ac.uk/gridsite/foafssl/ngs* says a user is a member of *https://www.kato.mvc.mcc.ac.uk/gridsite/foafssl/uom* then the mapping succeeds and results in the user being mapped to uid/gid 10031/10031. A future version of the evaluator should verify that a group only makes assertions about its own members.

66. The SAML2/XACML2 processor only uses the certificate chain attribute from the XACML2 message, all other attributes are ignored. Further development should include parsing of additional attributes and passing them to the policy evaluator. They can provide additional information to an evaluator and help it to make more complex policy decisions.

67. The RDF store abstraction layer is very minimalistic and should be extended to support additional features of RDF libraries. The most important one that should be implemented is the addition of scoped RDF data into the store.

68. All data in an RDF document can be given a context (scope, namespace, as a URI). Normally, this should be the WebID where the document is located, but documents are allowed to define data with different contexts. As an example, the group with WebID *ngs* can specify that the person with WebID *user1* is a member of the group with WebID *uom*, even though it has no authority to make this assertion. This can have severe implications on security if this information is used for authentication and authorisation purposes. Some RDF implementations have the feature to specify explicit contexts (scopes) for the data when adding it to the store, and therefore overwriting the contextual information in the data itself. In our example, the data of the group *ngs* can only be added to the store using its own context (WebID *ngs*) any assertion from this context outside of this scope are therefore invalid. i.e. *ngs* cannot assert that the user *user1* is a member of group *uom*. This feature is a prerequisite for the robust verification of the group – membership assertions of the simple FOAF group policy evaluator.

69. The current RDF store implementation supports the Jena¹⁴ and the Sesame 2¹⁵ frameworks.

14 <http://jena.sourceforge.net/>

15 <http://www.openrdf.org/>

70. The web service only works with RDF files in XML format. As it is one of the most common formats in use and widely supported by RDF libraries, it should remain the default format for the main RDF store of the web service, but support for other file format has still to be added, as well as the conversion feature. The feature to use the cached data in case of networking problems is not implemented yet. The current implementation only uses the cache to reduce network traffic. If a web server hosting the data is not reachable then the web service denies access even if the content is already cached. If this feature is going to be integrated, then the web service has to allow the service administrator to define the expiration time¹⁶ of the cached data. This expiration time period should be as short as possible to avoid security problems. Another feature that should be added is a regular clean up of the cache. This helps to reduce the memory requirements of the web service. The clean up period can and should be much longer than the expiration time period. Even if the cached data may not be used for authentication purposes, it can still reduce network traffic if the data on the web server has not been changed since the last successful retrieval.

Required Modifications to LCMAPS and LCMAPS SCAS Module

71. LCMAPS provides different calling interfaces which have significant differences in their behaviour. This mainly effects the certificate information which is retrieved from Globus and passed to the LCMAPS modules. The interface which is being used by the NGS version of LCMAPS does not make the full certificate chain available to the modules. The chain is required by the verify-proxy module which has to run before the SCAS module. The LCMAPS interface in question has been patched to provide the certificate chain as well.
72. The FoafAuthz service (PDP) requires the certificate chain of the user to make an authorisation decision. The XACML2 profile that is being used allows the chain to be sent in the request to the PDP, but it is optional and the current implementation of the SCAS module does not include it in the request. The functionality is already implemented in the SCAS module, it is just not being used. Therefore the SCAS module was modified to include the certificate chain in the request.

Issues and Possible Solutions

73. Two problems were encountered during development, the limited usability of FOAF+SSL certificates with Globus, and multiple mappings for a single user.
74. The limited usability of FOAF+SSL certificates with Globus is a major problem caused by the Globus SSL libraries which verify the client certificate when a connection to a Globus service is opened. This verification requires that the client certificate are issued by a CA and that the CA certificate is installed on the machine that is running the service. This restricts the usability to FOAF+SSL certificates that were issued by a FOAF+SSL certificate issuing service that acts as a CA by signing the issued certificates. Self signed certificates cannot be used with Globus. As the verification is done within a Globus library, there is the theoretical possibility to patch or replace this part of Globus to accept FOAF+SSL certificates in addition to its usual verification processes. FOAF+SSL certificates can be identified easily by checking if there is a subject alternative name containing a WebID in any of the certificates in the certificate chain. Implementing this solution would mean that sites that want to use it would have to replace part of the Globus stack. Most site administrators will be unwilling to do this. As self-signed FOAF+SSL certificates could not be used during our evaluation due to this problem, it is

¹⁶ The expiration time defines the time period for which cached data can be used to make an authorisation decision. This does not mean the data is invalid, just that it cannot be fully trusted any more.

unknown at this point whether there are additional related issues with LCAS/LCMAPS and the use of self-signed certificates. However, the use of a WebID service similar to that produced in the other portion of the project provides a straightforward means for credential manufacture and may even contribute to a higher Level of Assurance as the service provider is able to place restrictions on who can alter the contents of WebIDs' FOAF Files.

75. If a user is a member of multiple groups which are configured in the policy then the simple policy evaluator cannot make a decision which one it should use. Therefore it succeeds if there is exactly one mapping, otherwise it denies access. As it is very likely that a user is involved in multiple projects, multiple mappings cannot be prevented. The policy evaluator could pick one (e.g., the first in the policy file), but this has serious implications for the user, as it depends to the local configuration to which account the user is mapped to. A proper solution to this problem should allow the user to specify which account should be used. This can either be done by adding this information to the certificate, or by providing this information in the FOAF file which is specified by the WebID in the certificate.

Outcome

76. The software developed for this project demonstrates that it is possible to use FOAF+SSL for Globus authentication, but also that there are significant complications which can only be solved by modifying integral parts of Globus to allow the use of entirely self-signed certificates.