



omii europe
open middleware infrastructure institute



EU project: RI031844-OMII-Europe

Project no: **RI031844-OMII-Europe**

Project acronym: **OMII-Europe**

Project title: **Open Middleware Infrastructure Institute for Europe**

Instrument: **Integrated Infrastructure Initiative**

Thematic Priority: **Communication network development**

DJRA1.5

The first yearly report on the Virtual Organization Management task

Due date of deliverable: 30 April 2007
Actual submission date: 30 April 2007

Start date of project: **1 May 2006**

Duration: **2 years**

INFN

Revision [1.0]

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)		
Dissemination Level		
PU	Public	x
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Document Control Sheet

Document	Title: The first yearly report on the Virtual Organization Management task	
	ID: D:JRA1.5	
	Version: 0.6	Status: Draft
	Available at:	
	Software Tool:	
	File(s):	
Authorship	Written by:	Valerio Venturi (INFN) Sven van de Berghe (FLE) Morris Riedel (FZJ) Federico Stagni (INFN) Alberto Gianoli (INFN)
	Contributors:	
	Reviewed by:	
	Approved by:	

Document Status Sheet

Version	Date	Status	Comments
0.1	12 April 2006	Draft	All sections written
0.2	13 April 2006	Draft	Rearranged to meet Steve requirements
0.3	19 April 2006	Draft	Rearranged section 3.1
0.4	26 April 2006	Draft	Rearranged UNICORE section
0.5	26 April	Draft	Added introduction and conclusion
0.6	27 April	Draft	Added references
0.7	30 April	Draft	Reduced section 3.1.1 and extended conclusion.
0.8	30 April	Draft	Internal review
0.9	30 April	Final Draft	Last style changes
1.0	17 May	Final Draft	Corrected FLE effort

Executive Summary

This document describes the status of the Virtual Organization Management task of the JRA1 activity. The JRA1 activity aims to re-engineer existing grid and web services interface to improve quality, robustness and documentation and to integrate these services into other grid distributions.

The Virtual Organization Management task aims to ensure that there are common interfaces for Virtual Organization Management, or more generally for Attribute Authority, services across grid software distributions. VOMS, the Virtual Organization Membership Services, is being extended to support authorization specifications emerging from OGF standardization work. VOMS, already available under EGEE and GLOBUS, is being integrated into UNICORE. This document describes the status of the task at April 2007 (PM12) and the work done in the first year of the project. A similar document will be produced at PM23.

By April 2007 we have reached most of the expected outcomes. We have identified the interface for the Attribute Authority service. We have started discussing this interface in the relevant OGF working group. We have completed a full feature prototype of a VOMS service implementing the interface. We have identified how UNICORE may benefit of VOMS based authorization and the changes needed to the UNICORE protocol in order to transport the needed security tokens. A prototype for the integration of VOMS in UNICORE will be available by mid May 2007.

Table of Contents

1	Introduction	6
2	Partners and Efforts	6
3	Progress	7
3.1	Standardization and Community Efforts	7
3.1.1	Attribute Authority Interface	7
3.1.2	Attributes Format	8
3.2	VOMS Extension	8
3.2.1	VOMS Web Service Platform and SAML Implementation	8
3.2.2	Client Side Software and API	9
3.2.3	Testing	9
3.2.4	Building and Distribution	10
3.2.5	Relations with other activities	10
3.2.6	Exploitation path	10
3.3	Availability of VOMS under UNICORE	10
3.3.1	Prototype Implementation Progress	11
3.3.2	Web Services-based UNICORE Security Context Overview	11
3.3.3	UNICORE Identities for SAML-based VOMS Support	12
4	Conclusion	13
5	References	13

1 Introduction

The JRA1 activity aim is to re-engineer existing Grid and web services and their interfaces to improve their quality, robustness and documentation and to integrate these services into other Grid distributions.

OMII-Europe has chosen initially to ensure that there are common interfaces to the following types of services across a number of Grid software distributions: Database Access, Virtual Organisation Management, Accounting, Job Submission and Job Monitoring, Portal Interface.

The Virtual Organization Membership Service (VOMS) is used in EGEE to manage membership and position of users in Virtual Organizations (VOs). VOMS is available also under GLOBUS.

The JRA1 Virtual Organization Management task is extending VOMS to be compliant to emerging standards in the field of authorization, such as SAML and specifications coming from the OGF OGSA Authorization WG. This simplifies the other sub-task of the here described task to make VOMS available under UNICORE.

As virtual organization management is a basic, fundamental service for enabling VO-based authorization, this task has close relationships with all the other re-engineering tasks. All Grid services need to take authorization decisions based on the position of the subject accessing the resource in the VO. In particular, such need is already emerged clearly for job submission and database access services.

A standard's based, common virtual organization management service is fundamental for interoperation of authorization across different Grid middleware platforms. Having that, a VO will be able to span over different Grid middleware systems and enable users to access resources under different distributions without changing authorization mechanisms.

2 Partners and Efforts

This task involves 4.6 staff years (3.2 funded, 1,4 unfunded). INFN has 3.8 staff years (2.4 funded, 1.4 unfunded), FZJ 0.2 staff years (0.2 funded), FLE 0.6 staff years (0.6 funded). The effort expended to date is summarized in the following table.

<i>Partner short name</i>	<i>Budgeted funded effort (years)</i>	<i>Actual funded effort (years)</i>	<i>Reasons for deviation from budgeted and actual funded effort</i>	<i>Actual unfunded effort (months)</i>
<i>FZJ</i>	0.1	0.1		
<i>FLE</i>	0.3	0.15	First year has been primarily design. The effort intensive implementation will be carried out in the second year	
<i>INFN</i>	1.2	1.2		0.7

INFN brings in EGEE expertise and is undertaking the extension of VOMS to support the OGF AuthZ standard, FZJ and FLE will bring UNICORE expertise and is undertaking the integration of VOMS in UNICORE. INFN leads the task.

3 Progress

The task comprises two sub-tasks: the re-engineering of VOMS to support OGF AuthZ standards and the integration of VOMS in UNICORE. Both sub-tasks depend on an active participation in the appropriate standardization bodies of the Grid community. The standardisation activity is described first, followed by the VOMS extension and finally the integration of VOMS in UNICORE.

3.1 Standardization and Community Efforts

Following the fashion of the JRA1 activity, the main duty of the task is to identify and discuss within the Grid community an agreed interface for an Attribute Authority service.

3.1.1 Attribute Authority Interface

The description of work refers to an OGF document describing the use of SAML for OGSi Authorization [GFD.66] as the specification to use during developments. While this recommendation indicates the use of SAML for the expression of assertion and for protocol messages, there were some issues that led us to reconsider whether this was the correct choice. These issues can be described as follows:

- the document refers to SAML V1.1. At the time the task started, SAML V2.0 had been an OASIS standard for over a year. As the first software outcome of this activity was planned for April 2007, we decided to directly focus on SAML V2.0.
- the idea behind the specification was related to an authorization service rather than that of an Attribute Authority (AA) like the VOMS system. An Authorization service answers authorization decision queries such as 'is user A authorized to get resources B' while an Attribute Authority answers attribute queries such as 'does user A have the attribute B'. While SAML V1.1 uses the same top level protocol elements for the two query types, SAML V2.0 has two different protocol elements for the two query types, namely <saml:AttributeQuery> and <saml:AuthorizationDecisionQuery>.
- the document contained an interface based on the Open Grid Service Infrastructure (OGSI) specification. At the time the task started, OGSI was already indicated as superseded by the Web Services Resource Framework (WSRF). OGSI and WS-RF both using state-full Web services and widely concerns emerged on the use of state for an Attribute Authority service.

The SAML V2.0 set of specifications comprises, along with an assertion and protocol document [SAMLCore], a document [SAMLBind] that specifies SAML protocol bindings for the use of SAML assertions and request-response messages in communications protocols and frameworks. This makes SAML already usable as a reference for an Attribute Authority service (e.g. VOMS), requiring only recommendation on the correct use in the context of Grid services. As a side remark, several documents within the OGF OGSA AuthZ WG's use the same approach.

The other main Attribute Authority component in the Grid community is GridShib/Shibboleth. Shibboleth provides a federated single sign-on and attribute exchange framework and is widely used in the education community. GridShib is a software product that allows for interoperability between the Globus Toolkit and Shibboleth, thus making the latter available for Grid authorization. The GridShib project's investigators are working on an OASIS standard for a deployment profile for X.509 subject to use with SAML V2.0 [SAMLX509]. This profile complements the SAML specifications with indications on how to use SAML with X.509 certificates.

We have agreed within the OGSA AuthZ WG that [SAMLX509] with SAML V2.0 set of specification was the specification to use for an Attribute Authority service. The interface is plain web services. It was decided that the use of state for such a service should be discussed within the WG and we offered to collect use case. However, the effort is not considered fundamental.

3.1.2 Attributes Format

As well as the interface of the service, the service attributes also need to be the focus of standardisation work. The OGSA AuthZ WG has already produced a document describing the attributes used in Grids and within the same WG the VOMS team is finalizing a document describing the format of the VOMS Attribute Certificates. GridShib/Shibboleth bases its set of attributes on the MACE-Dir [MACE-DIR] set of specifications. Tools that need to process VOMS attributes must be given a standard way to use <saml:Attribute> elements expressing VOMS specific attributes such as Fully Qualified Attribute Name (FQAN) and generic attributes. Our efforts so far have been in deciding how to express VOMS information using SAML elements and respecting the SAML protocol's constraint and processing rules. This work is nearly finished and the intention is to produce a document to be submitted to the OGF OGSA AuthZ WG.

3.2 VOMS Extension

One of the expected outcomes of the activity in the first year is the release at M12 of a fully featured prototype for the extension of VOMS using SAML and the agreed OGSA AuthZ interface. A more detailed description of the aspects discussed in the following sections is available with the design documents that were produced as MJRA1.v1. We are continuing to update the documents which will be available with the software.

3.2.1 VOMS Web Service Platform and SAML Implementation

OpenSAML is an open source toolkit for implementing solutions using SAML specifications. It is available for both C++ and Java. The current release supports SAML V1.1 and SAML V1.0 but a version implementing SAML V2.0 is expected for spring 2007. OpenSAML is developed by Internet2 and is the toolkit on which Shibboleth, the main software using SAML, is built. The teams developing OpenSAML and Shibboleth have large intersections with the team developing the SAML specification within OASIS. This, as well as the large adoption of Shibboleth, suggests this as a reliable product.

Whilst the SAML V2.0 version is still under development, a technology preview of the library has been released for testing purposes. The plans of the OpenSAML team were to release the toolkit during March 2007 but it seems now they are slightly delayed. However, as the production level for our components is planned for M24 (April 2008) this is not considered as a problem. In fact, while announced as not frozen, the technology preview of the toolkit implements all the functionalities that VOMS needs, and during testing proved usefully stable.

Given the reliability of the project, the liaison with the main SAML based middleware, Shibboleth, and the availability of a preproduction release; we decided to use OpenSAML as SAML implementation. As expected after the testing performed during analysis, the toolkit has all the features we need and has proven well designed, reliable and stable. However, we continue to follow the development process and announcements in order to be ready to react should problems arise.

Given the initial preference for using C++, which would have allowed sharing of code with the existing VOMS software, we initially looked at gSOAP and Axis C++, the main open source C++ SOAP toolkits. Unfortunately this analysis revealed that both had problems producing skeletons and stubs from the SAML XML schemas.

We then analyzed Axis Java and XFire, two Java based SOAP toolkits. With minor problems for Axis, both were capable of producing skeletons and stub from the schema.

The final choice was Axis Java, since it allows the use of Proxy Certificates when used with Globus Toolkit's libraries. Proxy Certificates are the most common way of authenticating users and are used in the platforms on which VOMS is most used; gLite, VDT and Globus. The VOMS support for these platforms is still considered essential.

Finally, the service container we used to test the implementation was Apache Tomcat.

3.2.2 Client Side Software and API

We decided not to provide an API for the service. Following the Web Services fashion, the WSDL is the API; clients should be free to use their favourite SOAP tool with the provided WSDL to write client software. Given the problems that SOAP interoperability may raise, the choice could have its risk in terms of usability of the services. We decided to follow this principle in the beginning, when it is easier to monitor the use of the service, allowing us the possibility of providing an API should the use of the services prove too difficult. This would be following a similar path to the OGF SAGA WG; the group writing the specifications for Grid APIs. (To which we may contribute, as there is currently no API available that covers the semantics of VOMS).

In order to minimise any problems that may arise from not providing an API, we have released examples of using the services with some of the most popular SOAP tools. There are examples using Axis and XFire (using JAXB and XmlBeans bindings). We are willing to do the same with other libraries such as gSOAP (a first attempt however failed), Axis2, SOAP:Lite and ZSI in the second year of the project.

3.2.3 Testing

The prototype contains unit testing covering 42% of the total amount of code, thus not excluding the code that is not strictly unit testable. The unit tests for the classes implementing the service interface contain most of the testing for the compliance to the agreed interface.

We also want to extrapolate the interface compliance testing to an external test suite. We had an informal agreement with the GridShib team (the other main implementer of the interface) to work on a common test suite. This work will be planned with the SA2 Quality Assurance activity, which is responsible for standard compliance testing within the project.

3.2.4 Building and Distribution

Following project indications, the software is built using ETICS and made available through the OMII-Europe repository. The source code is placed in an INFN maintained Version Control System (<http://forge.cnaf.infn.it>). While instructions are available to get and build the software from source, the recommended way of obtaining it is through the OMII-Europe repository. There we will make available the releases configured and built using ETICS.

3.2.5 Relations with other activities

During the first year of the project, we have disseminated the work we were doing to other re-engineering activities in the project. Being a basic component for Grid authorization, use of the service should be of interest for all other reengineered services. Data services developed by the JRA1 Database activity (OGSA-DAI) and Basic Execution Services developed by the JRA1 Job Submission activity (UNICORE BES, CREAM BES, GridSAM) have started supporting VOMS for authorization, some currently in its legacy version based on Attribute Certificate, some using the prototype component we are developing.

We also have collaborated with the JRA3 activity, which Task 1 focus on Common Security Infrastructure. Furthermore, VOMS is an represents an important cornerstone of the multi-platform Grid infrastructure of JRA3 – Task 2 that deals with the interoperability between the OMII – Europe components. First experiences with interoperability between different components indicated that VOMS can be perfectly used as a new Grid platform-independent Attribute Authority service.

3.2.6 Exploitation path

It is fundamental for a service like VOMS, which can be fully exploited only after integration with other services, to find users. Thus, besides other OMII-Europe activities, some of which have manifested interests as said in the previous section, we need to raise interest and find potential users in the Grid community.

The most natural customers could be Grid middleware already using VOMS, such as gLite, VDT, and also GLOBUS. The plan for the first months of the second year is to disseminate the work we have done, and to deploy services for testing purpose.

Also PERMIS, an authorization framework using SAML and XACML, has manifested interests in testing the interoperability with the component.

Finally, in order to make the adoption of the component easier, we will make sure it is interoperable with Shibboleth.

3.3 Availability of VOMS under UNICORE

The other expected outcome of the activity in the first year is the release of a prototype for the integration of VOMS in UNICORE at M12. A more detailed description of the aspects mentioned in the following sections is available with the design document that was produced as MJRA1.v1 although, as described below, this has been superseded by other developments. We will provide an update of this document with the prototype software.

3.3.1 Prototype Implementation Progress

Following discussions within the UNICORE community the decision was taken to concentrate the implementation on the UNICORE 6 platform. This is an entirely Web services-based platform that removes any use of pre Web services UNICORE components while retaining the overall UNICORE architecture and approach with respect to security. This change of focus has delayed the implementation of the prototype while the decision was taken and the UNICORE 6 alpha software was evaluated. It also means that the design of MJRA1.v1 can be simplified, as there is no need to support UNICORE 5 interactions, mainly because UNICORE 6 will go into production in 2007.

During the next period we will produce a prototype implementation that will allow a client select the attributes (VO, groups, etc.) to request from the VOMS SAML service and pass these on to a UNICORE server using the protocol outlined below. We will also update the initial design document to reflect more recent implementation decisions. Further work will be carried out to enhance the prototype, bring it up to production quality and integrate it with the OMII-Europe repository.

3.3.2 Web Services-based UNICORE Security Context Overview

One of the major tasks that need to be performed before implementing UNICORE access to VOMS is to develop a way of carrying the security tokens released from a VOMS service between UNICORE clients and servers. This needs to be capable of expressing the UNICORE security model of the pre Web Services versions [ETD] as well as carrying tokens produced by VOMS or any other Attribute Authority service. This work was carried out in collaboration with the wider UNICORE community, most notably Krzysztof Benedyczak of the Chemomentum project. This section briefly describes the UNICORE security in this context. It will be implemented as part of the prototype that will be delivered for OMII Europe.

The major challenge is the way of how security related information is transported that is needed by the UNICORE server-side software to incarnate and perform a task. This information includes:

- the task itself (not directly security related),
- the identity of an entity on whose behalf the task is to be executed, this is called the User identity,
- and additional attributes supporting the authorization process, which will be expressed as a tuple (task, User).

The server must be able to verify that the information is genuine and it may also use alternative sources of security related information for other (e.g. logging) purposes.

3.3.3 UNICORE Identities for SAML-based VOMS Support

The task together with any addressing information, credentials and other metadata is referred to as a request. The most typical example of a request is a SOAP Envelope together with its TLS context.

In this context, UNICORE understands two types of identities:

- Consignor: the identity of a client that send requests. This identity is mandatory, i.e. it must be always available

- User: the identity of an entity on whose account the task is to be executed. It is optional. If it is not directly available, then the Consignor identity is used.

Identities in UNICORE are usually transmitted as X509 certificates. X509 certificates refer to the same entity when certificates are equal and not when just their Distinguished Names (DN) are the same (this statement may be modified in the light of further experience).

Specifying the Consignor: It is the role of the UNICORE server-side to establish the Consignor identity. This can be from a client certificate retrieved from transport layer (TLS/SSL v3) or from the signature of the whole request in case of message level security. The Consignor identity must always be an X509 certificate.

Specifying the User: Whenever a Consignor wants to execute a task contained in a request on behalf of an entity other than itself, the User identity must be provided. To achieve this, the Consignor adds a SAML assertion to the request header, which contains the identity of the user and related attribute information (it is this information that the prototype implementation will obtain from a VOMS server).

Delegation of Trust¹: This means of how to specify an assertion that delegates (part of) the privileges of one entity (custodian) to another (receiver). This delegation is defined as signed SAML assertion. Whenever trust is delegated, the delegating entity must be identified by an X.509 certificate.

The trust delegation assertion is encapsulated by the TrustDelegationOfUser attribute statement. This attribute must hold the initial custodian of trust. It is the same as the issuer of assertions in simple situations. However, whenever delegation is further delegated (i.e. U delegated privileges to A and A delegates them further to A2) it plays an important role as it shows whose privileges are actually transferred. Any relaying software must always use this attribute to verify the validity of delegation from Consignor to User. The conditions of Trust delegation are expressed as a saml:Conditions assertion element, such as *NotBefore*, or *NotOnOrAfter*. Those are standard SAML attributes that should be used. The trust delegation is valid only in time frame of those attributes and *ProxyRestriction*. That is another standard SAML element and whenever presented its value limits the maximum length of delegation chains. Trust delegation chains come into play whenever privileges from an initial custodian to a final recipient are delegated through one or more proxies. It is composed of multiple trust delegation assertions that:

- have common value of TrustDelegationOfUser attribute (initial custodian),
- there is initial assertion which issuer is the same as initial custodian,
- for every assertion, except initial, with issuer A, there is other assertion with Subject A,
- every assertion condition is satisfied.

4 Conclusion

The task has reached most of the expected outcomes. Of great importance, an agreement has been established within the relevant standardization body on the interface for the Virtual Organization Management services. Minor details remain to be sorted out, but the core work has been done.

A prototype for VOMS using the agreed interface has been completed. Other re-engineering activities have started using the service for authorization.

¹ Trust delegation is beyond the scope of the initial prototype. It is included here as a potential for future work.

The integration of VOMS in UNICORE has been slightly delayed. The main reason was the decision of concentrating on UNICORE 6 rather than on previous versions. Following that, effort has been spent on designing a protocol for UNICORE 6 able to express the UNICORE security model as well as carrying VOMS security tokens. An initial alpha prototype support of UNICORE 6 was demonstrated at the all hands meeting at FZJ.

For the second year of the project, the main aim as stated in the description of work is making the prototypes ready for production environments and thus more stable. This means at minimum to satisfy the quality request of the project and put releases into the project repository. The first months of the second year will be dedicated to be ready to undergo the demanding quality assurance process of the project. Another aim is to disseminate the work being done for having other re-engineering activities, as well as other subjects developing Grid components, start integrating VOMS in their authorization frameworks as described in section 3.2.6. This could require for example support for other SOAP tools to be tested and eventually writing an API. Finally, the standardization effort started in the first year has to be finalized which includes the following tasks:

- preparing a document describing the SAML profile for VOMS attributes; this will make it a lot easier for other authorization tools to integrate VOMS
- collect the work being done in a profile for the interface of Attribute Authority services to be used as a reference within the OGF OGSA AuthZ WG.

5 References

[ETD] Explicit Trust Delegation: Security for Dynamic Grids, Dave Snelling, Sven van den Berge, Vivian Li. Paper published in FUJITSU SCIENTIFIC & TECHNICAL JOURNAL (FSTJ) - Special Issue on Grid Computing, December 2004 Issue (Vol.40, No.2).

[GFD.66] Use of SAML for OGSF Authorization, available at <http://www.ogf.org/documents/GFD.66.pdf>

[SAMLCore] Assertions and protocols for the OASIS Security Assertion Markup Language (SAML) V2.0, available at <http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf>

[SAMLBind] Bindings for the OASIS Security Assertion Markup Language (SAML) V2.0 , OASIS Standard, available at <http://docs.oasis-open.org/security/saml/v2.0/saml-bindings-2.0-os.pdf>

[SAMLX509] SAML V2.0 Deployment Profiles for X.509 Subjects, OASIS Draft, available at <http://www.oasis-open.org/committees/download.php/21568/sstc-saml2-profiles-deploy-x509-draft-01.pdf>

[MACE-DIR] Middleware Architecture Committee for Education (MACE), available at: <http://middleware.internet2.edu/dir/>