Extending the Collaborative Online Visualization and Steering Framework for Computational Grids with Attribute-based Authorization

Morris Riedel, Wolfgang Frings, Sonja Habbinga, Thomas Eickermann,
Daniel Mallmann, Achim Streit, Felix Wolf, Thomas Lippert
Jülich Supercomputing Centre, Forschungszentrum Jülich
D-52425, Jülich, Germany
m.riedel@fz-juelich.de

Andreas Ernst, Rainer Spurzem
Astronomisches Rechen-Institut, University of Heidelberg
D-69120 Heidelberg, Germany

Abstract

*Especially within Grid infrastructures driven by high-performance computing (*HPC*), collaborative online visualization and steering (*COVS*) has become an important technique to dynamically steer the parameters of a parallel simulation or to just share the outcome of simulations via visualizations with geographically dispersed collaborators. In earlier work, we have presented a* COVS *framework reference implementation based on the* UNICORE *Grid middleware used within* DEISA. *This paper lists current limitations of the* COVS *framework design and implementation related to missing fine-grained authorization capabilities that are required during collaborative* COVS *sessions. Such capabilities use end-user information about roles, project membership, or participation in a dedicated Virtual Organization (*VO*). We outline solutions and present a design and implementation of our architecture extension that uses attribute authorities such as the recently developed Virtual Organization Membership Service (*VOMS*) based on the Security Assertion Markup Language (*SAML*).*

1. Introduction

World-wide Grid infrastructures such as DEISA, EGEE, OSG, or TeraGrid provide a wide variety of Grid services to enable large-scale resource sharing for e-science. Virtual Organizations (VOs) allow to share such resources across organizational boundaries and to make efficient use of the provisioned computational Grid resources such as supercomputers or clusters. Many scientific applications within these VOs and underlying Grid infrastructures aim at simulations of physical, biological, chemical, or other types of domain-specific processes. While many Grid infrastructures exist today, infrastructures such as DEISA or TeraGrid, which are largely driven by high-performance computing (HPC) needs, run Grid applications with parallel computing techniques (i.e. MPI [24], OPENMP [10]) to simulate these processes. The outcome of these simulations is often analyzed in a separate post-processing step, for instance by viewing the results in a visualization application. Based on these intermediate results, a decision is made to change simulation parameters for another computational period.

In order to increase the efficiency of e-scientists, the collaborative online visualization and steering (COVS) technique emerged that performs simulation and visualization at the same time. Online visualization refers to e-Scientists that are able to immediately observe the processing steps during the simulation. This in turn allows for computational steering to influence the computation of the simulation during runtime on a supercomputer. In this context, we have shown in earlier work that the efficiency of e-scientists can be further improved by leveraging strong security environments and collaborative Web service-based features when using a COVS framework [22] in UNICORE Grids such as DEISA. In this paper, we discuss challenges that arise in geographically dispersed visualization sessions, creating a demand for fine-grained authorization. Attributes of end-users such as VO and group membership as well as different roles and capabilities are available in Grids today, but the COVS framework is limited to identity-based authorization (i.e. using X.509 certificates only). In this paper we present an extension that allows for attribute-based authorization.

This paper is structured as follows. After reviewing visualization and steering capabilities in computational Grids, Section 2 introduces the COVS framework implementation in UNICORE and lists limitations with regard to fine-grained authorization. Section 3 describes which standard-compliant attribute-based authorization technologies can be used within Grid environments today. Based on these technologies, we present extensions to our COVS architecture in Section 4, while Section 5 describes two scientific applications as use cases of this new feature. Finally, after surveying related work in Section 6, we offer our conclusion in Section 7.

2. Limitations of the Collaborative Online Visualization and Steering Framework

The collaborative online visualization and computational steering (COVS) framework enables Grid applications with interactivity (i.e. computational steering) and visualized feedback mechanisms. In earlier work [26], we have shown a prototype COVS technique implementation based on the visualization interface toolkit (VISIT) [13] and the Grid middleware of DEISA named as the Uniform Interface to Computing Resources (UNICORE) [28]. Since then the approach grew to a broader COVS framework [23] and we further published at the Grid 2007 conference in [22] that the approach taken is feasible and provides sophisticated performance. More recently, we investigated in [21] the impact of using the computational steering capabilities of the COVS framework implementation in UNICORE on large-scale HPC systems of DEISA (e.g. IBM BlueGene/P JUGENE with 65536 processors).

The current architecture of the COVS framework is illustrated in Figure 1, which shows a collaborative scenario with two geographically dispersed participants (i.e. client tier A and B). Both run a scientific visualization, which is coupled with a COVS GridBean plug-in that extends the GPE UNICORE Grid Client [25]. The Grid client is used to access two COVS services that are implemented using the factory pattern of the Web Services Resource Framework (WS-RF) [1] implementation of UNICORE. Therefore, the client is used to call a COVS Factory Service, which creates COVS Session resources that are in turn accessible via the COVS Service. An instance of the session resource represents a collaborative visualization session managing different participants by controlling the VISIT Collaboration Server and the VISIT Multiplexer. While the VISIT collaboration server is used to exchange information between participants over dedicated connections secured with SSH, the VISIT Multiplexer is responsible to distribute the outcome of one parallel simulation to n participants using the same connections. These connections are created using the strong security features of the UNICORE Grid middleware and is described in more detail in [26]. To sum up, the scientific data of the simulation and collaboration data is transferred via secured dedicated connections with binary wire encoding to achieve satisfactory performance, while the simulation job submission and the management of collaborative sessions use Web service calls that in terms of the overall performance are non-critical.

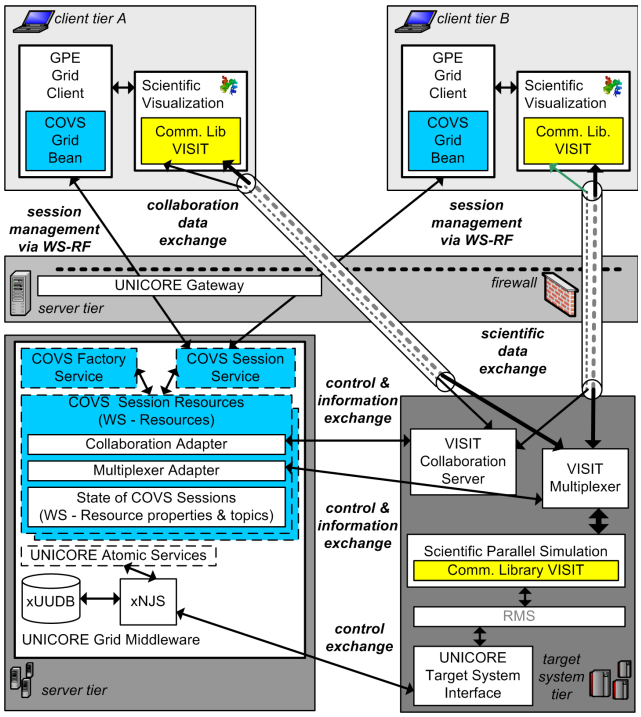

Figure 1. COVS Framework implementation in the UNICORE Grid middleware.

Although our framework implementation is used in production, we recently encounter several limitations of the framework with respect to fine-grained authorization capabilities, which motivated the approach in this paper respectively. In typically scenarios, the COVS service is used within a VO, but with different geographically dispersed VO members that act in different roles and possess multiple capabilities during one COVS session. In more detail, a person that use our framework is in the *participant role* if the person shares the view on one visualization of a parallel simulation with all other n-1 participants. While some people only act in the participation role, there are other people that may represent more than one role. This implies that the functionality of our framework for one role differs from the functionality offered to other roles. For instance, only people in the *master role* are able to use the framework for the submission and control of a parallel simulation that runs on a computational Grid resource. Hence, other participants do

not need (and should even not be allowed) to submit a simulation job, because the outcome of one submitted parallel simulation is shared with all others participants.

To circumvent that any end-user is able to join a session, we define the *approver role* that is responsible to make decisions which participants are allowed to join visualization sessions. They are making their decisions based on the different roles and pre-defined capabilities of the candidates that would like to join the session. Furthermore, technical capabilities to steer a parallel simulation during a collaborative session raises the demand for mutual exclusion of participants during steering. The steering process requires expertise in the field of the simulation and thus only a subset of participants are able to represent the role and only this sub-set should be allowed to change the behavior of the simulation. Therefore only one participant that represent the *steerer role* is allowed at the same time to steer a parallel simulation during a COVS session in order to ensure the consistency of the simulation and its computation. In addition, only participants in the *collaborator role* are allowed to change the view of the visualization. This role typically also needs expertise to choose, for instance, color codings of physical phenomena that can be understand by all participants and that make sense in the context of simulations.

The overall management of a COVS session therefore requires *authorized session management control actions* taking the roles and capabilities of participants into account. In the current implementation however, the enhanced UNICORE User DataBase (XUUDB) that deals with authorization in UNICORE only allows definitions of one defined role that is strictly bound to one X.509 certificate identity of the end-user (i.e. identity-based authorization) for any service within UNICORE. Hence, there is no functionality how all different roles and capabilities of one end-user can be mapped to the X.509 certificate identity used within the Grid middleware for authorization decisions so far.

Another limitation of the current design and implementation is that anyone who is allowed to use the UNICORE Grid middleware and its deployed COVS services is also automatically allowed to join any COVS session available at this site. In our scenarios we would like to restrict the access to COVS sessions only for certain members of a VO that are actually part of the respective groups that created the COVS session. But so far, the authentication and authorization of end-users using the COVS framework was purely based on full X.509 certificates. This only allowed a raw-grained authorization approach based on the identity provided via the certificate used to check whether end-users have access to COVS services (and all sessions) or not. All in all, the security in the COVS framework can be significantly improved for collaborative scenarios, while there is already strong security on the data connection level.

3. SAML-based Attribute Authorities in Grids

Many authentication and authorization infrastructures (AAI) for Grids are using basic authorization mechanisms based on the distinguished name (DN) of the end-users proxy X.509 certificate or the full X.509 certificate (e.g. within UNICORE). Thus these certificates are not only used to authenticate end-users, but also to base authorization decisions on them as long as no further information describes the end-users, his/her roles or capabilities. Experts refer to this approach as identity-based authorization. But many frameworks and services in Grids need more information to achieve fine-granular decisions [30], and, also the previous section clearly raised a demand for fine-grained authorization based on different roles and capabilities (collectively named as attributes) of end-users.

This demand is not new and thus there are solutions in Grid environments that deal with these kind of requirements named as attribute-based authorization mechanisms. This approach needs two additional components compared to the pure identity-based approach. First, an attribute authority (AA), which issues attributes in a trusted way is required. Second, a so-called policy decision point (PDP) [7] using these attributes for authorization is the complementary component often offered via the Grid middleware itself. The attributes are encoded as fully qualified attribute names (FQANs) [29] containing VO membership, groups, roles and capabilities within that VO.

At the time of writing, two major attribute authorities are available in Grids that are the Virtual Organization Membership Service (VOMS) [7] and Shibboleth [4]. Both Shibboleth and the recently developed new VOMS service [30] are based on the Security Assertion Markup Language (SAML) standard [9]. We enable our framework with attribute-based authorization using VOMS since it is closer to our use case than Shibboleth and its federation approach [17]. In addition, VOMS is following the recommendations of the OGF OGSA-Authorization working group and is thus compliant with the OGF security standards.

The basic idea of VOMS is illustrated in Figure 2 that shows that an administrator is able to use the VOMS Admin Client to configure VO information (i.e. attributes) of end-users. This information is stored in a VO database, which is used by the VOMS service in order to fulfill requests by providing information. Hence, VOMS represents an AA while the message exchanges using the SAML protocol over Web services. The released FQANs (e.g. roles and capabilities) are encoded in an XML-based SAML assertion (i.e. `<saml:AttributeStatement>` element), which is signed by VOMS. Every technology that would like to base its authorization decisions in the PDP on this SAML assertion have to check this signing and thus trust the VOMS service.

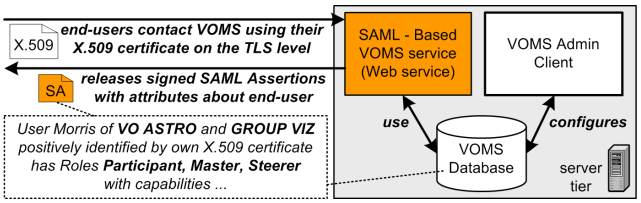

Figure 2. Attribute authority (i.e. SAML-based VOMS service), which releases signed SAML assertions with attributes stating the position and roles of an end-user in the VO.

4. Extending the COVS framework with Attribute-based Authorization Capabilities

Extending our rather complex framework design with fine-grained authorization capabilities is not a straightforward task and implies changes to the client and middleware. So far, we identified that attribute-based authorization could improve our framework design to overcome its limitations described in Section 2. This lead to the design and implementation of an architecture extension of our framework to work with attribute authorities that release attributes of end-users. We also noticed that the VOMS approach fits nicely into our Grid environment and thus we present our attribute-based authorization extensions based on VOMS. Nevertheless, with little modifications, the whole system should also be able to work with Shibboleth, because both agree to the same SAML standard. Figure 2 illustrates that necessary information (e.g. roles) can be encoded within a standardized SAML assertion when contacting a VOMS service. These pieces of information are used to enable fine-grained authorization for COVS session and thus providing answers to questions like 'which user is authorized in which sessions and has which roles and capabilities in the context of the visualization and steering process'.

The VOMS integration work basically starts with precise definitions of roles as well as VO and group information. We mapped the VO concept and attribute capabilities of VOMS to our specific needs. This is possible by configuring the VOMS server and we provide examples of attributes and their FQANs as released by VOMS in Table 1. In fact, these FQANs are encoded inside the `<saml:AttributeStatement>` element as an `<saml:AttributeValue>` element within a SAML assertion. While each role is represented by one FQAN, it's particularly important for our approach that the VOMS approach allows for multiple `<saml:AttributeValue>` elements that enable multiple roles for end-user encoded in one SAML assertion. To provide an example, we can define that one end-user represents the *master*, *participant*, and *steerer* role at the same time and all this information is encoded in one SAML assertion released from VOMS (cp. Figure 2).

A further advantage for our extension approach with VOMS is the possibility to define and work with so called capabilities. These capabilities are basically 'key-value pairs' following the format `name=value`. We did not list all precise examples in Table 1 since such capabilities are mostly used to express scientific domain-specific expertise. For instance, we can define an end-user with certain roles of the VO astro and group viz that has the domain-specific expertise (i.e. capability) of being an expert in n-body problems within astro-physical phenomena and simulations. This would be encoded as `/expert=n-body` and thus underlines that the approach with such rather generic attributes is very much extensible – a further benefit of using the attribute-based approach within our framework.

Attributes	FQAN	Desription
VO	/astro	End-user belongs to astro VO
Group	/astro/viz	End-user is member of group viz
Role	/astro/viz/Role=master	End-user can submit Grid applications (creates COVS session)
Role	/astro/viz/Role=participant	End-user is able to join an existing COVS session
Role	/astro/viz/Role=approver	End-user is able to approve group members for COVS session
Role	/astro/viz/Role=steerer	End-user can steer Grid applications
Role	/astro/viz/Role=collaborator	End-user is able to change the view on scientific visualizations
Cap.	/astro/viz/Cap=value	Additional capabilities of end-user
expert	/astro/viz/expert=n-body	Capability of an end-user stating being an expert of n-body

Table 1. Attributes used in the COVS framework and possible capabilities extensions.

So far, the end-users use purely their X.509 certificate stored and configured within the GPE Grid client. Thus, in order to use SAML assertions as addition to end-user certificates, we have extended our COVS GridBean plug-in for the GPE Grid client as shown in Figure 3. This particular extension basically represents a VOMS Web service client, which invokes the `samlp:AttributeQuery` operation [9] of VOMS. The connection between the client and VOMS is based on the X.509 certificate of the user and thus the VOMS is able to get the identification of the end-user from the TLS connection [12]. The response of this Web service call carries a SAML assertion with attribute information that is temporarily stored at the COVS GridBean plug-in. Afterwards it is subsequently used for each COVS service invocation. In more detail, for each COVS service invocation, the SAML assertion is transported within the SOAP header of the Web service message exchange using the Web Service Security Extensions [6] standard.

Obviously, just using these SAML assertions with attribute information for service invocations is not enough to realize fine-grained authorization. Therefore, we also have implemented several extensions within the hosting environment of the Grid middleware UNICORE. Most notably, we implemented, as being part of the OMII-Europe project [2], a security handler that is called before any COVS service invocation is taking place in the middleware. First, the handler extracts the SAML assertion from the SOAP header for further processing. Then the handler checks whether the SAML assertion is still valid, because lifetime information is also encoded in the SAML assertion within a `saml:condition` element [9]. Finally, the handler checks whether the SAML assertions is signed from a valid attribute authority that is being trusted. Only if both steps are successful, the handler puts the SAML assertion in the security context of UNICORE that is used for security enforcements later.

When all handlers configured for a particular service such as COVS have been processed, the UNICORE PDP (cp. Section 3) makes a callout to a policy that is compliant with the OASIS extensible Access Control Markup Language (XACML) [20] standard for authorization decisions. That means the attribute information of the SAML assertion within the security context is used in conjunction with the policy to check whether the end-user got the right attributes to get access. Using this policy we have defined rules that define which end-users of which VOs and groups are actually allowed to work with COVS services as a first step towards a more fine-grained authorization. Whenever someone is trying to invoke COVS services and would like to join a session the handler is called and subsequently the XACML policy is checked whether the correct attributes have been presented at the Grid middleware (via the SAML assertion).

A benefit of the UNICORE design was that the security context and thus the stored SAML assertion is available in the COVS session service. That means we extended our service to take the different roles of the end-users into account to check whether certain actions are allowed or not. To provide an example, only persons that presented a SAML assertion with attributes expressing the *steerer role* are actually allowed to steer the application via the VISIT toolkit. That means the COVS session service implementation offers or restrict certain actions to end-user based on the different roles that an end-user possess. While the same approach can be basically implemented with the capabilities (i.e. *expert=n-body*), we initially just use them to give this information as trusted additional information about participants in a session. Hence, by using the above mentioned extensions we have been able to overcome the limitations stated in Section 2 and are thus able to present in this work a solution for the problems that arise in collaborative scenarios.

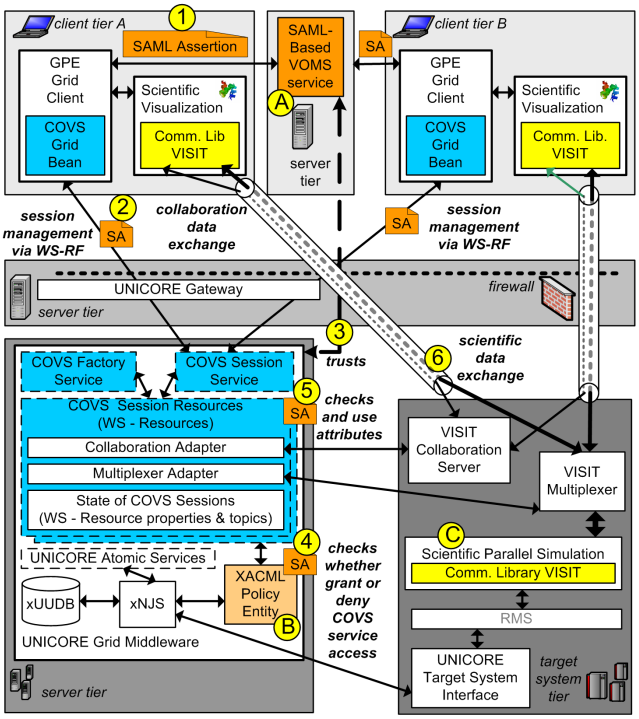

Figure 3. COVS architecture extensions to leverage SAML-based VOMS and use SAML assertions for attribute-based authorization. The XACML policy is firstly used to enforce policies based on information in SAML assertions, later the attributes are used in the service itself for further authorization of dedicated actions.

Before end-users actually can use a deployed COVS framework in their daily work we have to assume some preconditions that are marked with single alphabetical characters in Figure 3. First, in (A) we have to configure the VOMS with general information such as VO and group status. In addition we have to define the roles and capabilities about end-users (cp. Table 1). Afterwards we have to setup XACML policy rules that match the attribute statements of acceptable users (B). To provide an example, we define that only end-users in the VO astro and group viz are allowed to use the COVS factory and COVS session service. Finally, we assume that one end-user has already submitted a computational job via the Grid middleware (C) by being in the *master role*.

We summarize the usage of our framework extensions in a step-wise fashion (cp. marked numbers in Figure 3) in the following paragraph. The first step in our approach is to contact the VOMS with the COVS GridBean plug-in in order to retrieve a SAML assertion with attribute information (1). This SAML assertion is then transmitted during the COVS service invocation within the SOAP header (2). This COVS service invocation represents a COVS session join request. In step (3) the implemented VOMS handler is activated and checks whether the SAML assertion is still valid (i.e. lifetime checks) and signed by an attribute authority that is being trusted. Only if this step is successful, step (4) checks the provided SAML assertion in conjunction with the pre-configured XACML rules to enforce first parts of the fine-grained authorization approach. This checks whether an end-user is allowed to use the COVS services or not.

The second part of this fine-grained authorization approach is undertaken in the COVS session service implementation afterwards. In more detail, within a particular instance of a COVS session resource accessible with the COVS session service (cp. Figure 3). The state of the COVS session consists of the joined participants or those that still require approval from someone that possess the *approver role*. To influence the state of the COVS session via Web service operations, we use the role information of the SAML assertions as a base of authorizing certain actions (5) within this particular COVS session. This means according to the presented roles stated in the SAML assertion, the end-user is able, or not able to influence the behavior of the visualization session or change the scientific data stream that is transferred to all participants (6). To provide an example, only a participant in the *steerer role* is able to influence the application during its runtime. This is internally realized by forwarding suitable actions or commands via the multiplexer adapter, which in turn controls and manage the VISIT multiplexer [22]. The same approach is implemented in the COVS session service in terms of the *collaborator role* that uses the collaboration adapter to control and manage the VISIT collaboration server [22].

5. Scientific Use Case Applications

We have evaluated our design approach of the extensions with two scientific scenarios, however, its difficult to show results since the attribute-based authorization with SAML assertions is basically only present behind the scene and not visible to end-users. This can be considered as a feature since e-Scientists that use the Grid for research typically do not want to know much about the details of fine-grained authorization. They just would like to use the framework as provided.

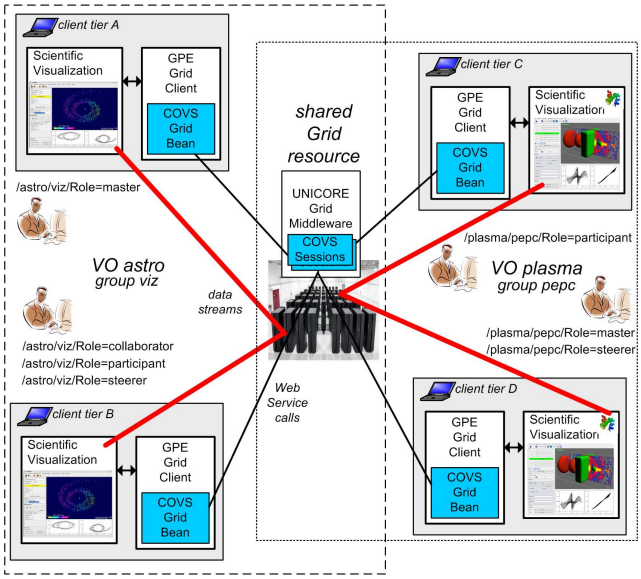

Figure 4. Two independent scientific visualization sessions share one computational resource accessed via one Grid middleware. The resource is shared between VO astro and VO pepc that both use attribute-based authorization taking roles into account.

The two use case applications we describe are both n-body problems and both have been instrumented with VISIT in earlier work [13]. Such n-body problems appear in many scientific areas such as astrophysics, plasma-physics, molecular dynamics, and fluid dynamics. Therefore we developed in earlier work the scientific visualization Xn-body, which visualizes n-body problems [5] and which interfaces the GPE Grid client to get access to the Grid via Grid middleware. N-body problems are commonly solved using divide-and-conquer mechanisms or parallel computing techniques. In this context, the first use case application of our attribute-based authorization is the Nbody6++ program [27] used in the field of astrophysics. Nbody6++ is a parallel variant of the Aarseth-type N-body code nbody6

suitable for N-body simulations on supercomputers within HPC-driven Grid infrastructures such as DEISA. This Grid application is typically used to simulate dynamics of star clusters in galaxies and their centres, respectively, formation of planetary systems and dynamical evolution of galactic nuclei. In Figure 4, we illustrated one session of the VO astro that is running this application on a supercomputer and share the view of it using the recently developed attribute-based authorization. The attributes of the respective end-users are also shown as FQANs.

Another use case application of our architecture extension is also shown in Figure 4 in the context of the plasma VO. This VO with experts from plasma-physics run the Pretty Efficient Parallel Coulomb Solver (PEPC) code [16], which is a massively parallel code to perform potential and force summation of N charged particles in a time O(N log N) using a hierarchical tree algorithm. While this simulation is running on the supercomputer, e-Scientists are able to obtain a step-wise visualization of the computational process and are able to influence the behavior based on their attributes such as the roles encoded in the FQANs.

6. Related Work

There is plenty of related work in the field of visualization and steering within Grids. Brodlie et al describes in [8] a high-level framework for distributed and collaborative visualization and how it can be potentially implemented by visualization systems, but not considering attribute-based authorization as it is available in Grids today.

One of the specific research areas of the Japanese National Research Grid Initiative (NAREGI), among Grid middleware and Grid networking, include visualizations and limited steering scenarios. Kleijer et al describes in [18] the an API for Grid-based visualization systems of the NAREGI Grid infrastructure. The API consists of a visualization library and a Grid visualization service API. While the library is used to connect simulations by the provisioning of visualization functionalities, the visualization service API wraps the library to provide Web service-based functionalities. Although this approach is very similar to ours in terms of the Web service layer, this framework only supports identity-based authorization mechanisms, while we extend our scope to attribute-based authorization.

Another interesting work is developed in the Austrian Grid and Koeckerbauer et al describes in [19] the Grid Enabled Visualization Pipeline (GVID), which provides high quality Grid-based visualization of scientific datasets on thin clients such as SONY Playstations. In this approach, the data of the scientific visualization are efficiently encoded with the H262 code into a video stream and transferred to the thin client afterwards. The client in turn decodes the video stream and visualized the scientific data. While this technology is rather decoupled from Grid middleware, we implemented our services as higher-level services within the Grid middleware UNICORE to leverage the strong security infrastructure, which makes it also easier to achieve the attribute-based authorization of our approach.

A complete different approach was realized by the UK RealityGrid project [3] that focused on how scientists can make more effective use of a Grid and its visualization resources. In fact, this approach is similar to our approach, since more recent prototypes of the RealityGrid steering library have been renewed to be conform with the Open Grid Services Architecture (OGSA) [15]. It thus was realized within the Imperial College e-Science Networked Infrastructure (ICENI) [11] that partly based on Globus Toolkit technologies and the Grid Security Infrastructure (GSI) [14]. To the best of our knowledge their is no work describing how the RealityGrid steering library approach is used in conjunction with the GSI and attribute-based authorization. However, we know that GSI has been enabled with attribute-based authorization using attribute-certificates that are embedded in X.509 proxies. Since we rely on the HPC-driven Grid middleware UNICORE, which only supports full X.509 certificates, our approach is also different from the proxy-based GSI approach that is the security foundation for the ICENI middleware.

7. Conclusions

The evaluation with two use case applications proved that our approach is feasible and thus overcomes the limitations identified in the COVS framework with respect to fine-grained authorization. We have shown how the evolution from identity-based authorization (i.e. using pure X.509 certificates) towards attribute-based authorization using roles and capabilities of end-users can be applied to Grid visualization and steering in general, and the COVS framework in particular. By adding fine-grained authorization to our framework, we implemented a unique approach of having visualization and steering of HPC applications within Grids massively supported by Grid middleware and SAML-based attribute authorities. To realize that, we have been working in the OMII-Europe project as an early use case driver of SAML-based VOMS adoptions with the VOMS developers and thus contributed to the UNICORE and SAML-based VOMS development. Some future work in the field of attribute-based authorization would be an integration with SAML-based Shibboleth federations or an approach that allows for more dynamically definitions of attributes, for instance during a run-time of a COVS session. Other interesting work continue the investigation of Grid steering towards computational Grid resources towards peta-scale performance, e.g. soon we expect systems with 1/2 petaflop/s at our institute. Then computational steering become diffi-

cult to use and new approaches have to be identified using potentially more hierarchical or tree-based steering mechanisms.

Acknowledgments

The work presented in this paper has been supported by the OMII - Europe project under EC grant RIO31844-OMII-EUROPE, duration May 2006 – April 2008. We also would like to thank members of the OGF OGSA - Authorization group for their valuable advise and the developers of the SAML-based VOMS server, in particular Valerio Venturi (INFN, Italy). Finally, also the work of the EGEE middleware security group (MWSG) was helpful with respect to attribute-based authorization.

References

[1] OASIS - WSRF Technical Committee. http://www.oasis-open.org/committees/wsrf.

[2] OMII - Europe. http://omii-europe.org/.

[3] RealityGrid. http://www.realitygrid.org.

[4] The Shiboleth Proejct, Internet2/MACE.

[5] XNBODY. http://www.fz-juelich.de/zam/xnbody.

[6] OASIS - Web Service Security: SAML Token Profile 1.1, 2006. http://docs.oasis-open.org/wss/oasis-wss-SAMLTokenProfile-1.1.

[7] R. Alfieri et al. From gridmap-file to voms: managing authorization in a grid environment. In *Future Generation Comp. Syst., 21(4):549-558*, 2005.

[8] K. Brodlie, D. Duce, J. Gallop, J. Walton, and J. Wood. Distributed and Collaborative Visualization. In F. Berman, G. C. Fox, and A. J. G. Hey, editors, *Computer Graphics Forum, Volume 23*, 2004.

[9] S. Cantor, J. Kemp, R. Philpott, and E. Maler. *Assertions and Protocols for the OASIS Security Assertion Markup Language*. OASIS Standard, 2005. http://docs.oasis-open.org/security/saml/v2.0/.

[10] R. Chandra et al. *Parallel Programming in OpenMP*. Morgan Kaufmann, 2001. ISBN 1-55860-671-8.

[11] J. Cohen, A. McGough, J. Darlington, N. Furmento, G. Kong, and A. Mayer. RealityGrid: an integrated approach to middleware through ICENI. In *Philosophical Transactions of The Royal Society A, 363*, pages 1817–1827, 2005.

[12] T. Dierks and C. Allen. *The TLS protocol version 1.0, Internet Engineering TaskForce, RFC 2246*. 1999. http:///www.ietf.org/rfc/rfc2246.txt.

[13] T. Eickermann, W. Frings, P. Gibbon, L. Kirtchakova, D. Mallmann, and A. Visser. Steering UNICORE Applications with VISIT. In *Philosophical Transactions of The Royal Society Journal, London*, 2005. (doi:10.1098/rsta.2005.1615).

[14] I. Foster, C. Kesselman, G. Tsudik, and S. Tuecke. A security architecture for computational grids. In *5th ACM Conference on Computer and Communications Security*, pages 83–91. Assoc. Comput. Mach Press, New York, 1998.

[15] I. Foster, H. Kishimoto, A. Savva, D. Berry, A. Djaoui, A. Grimshaw, B. Horn, F. Maciel, F. Siebenlist, R. Subramaniam, J. Treadwell, and J. Reich. *Open Grid Services Architecture, Version 1.5*. Open Grid Forum Draft 80, 2006.

[16] P. Gibbon. *Short Pulse Laser Interactions with Matter: An Introduction*. Imperial College Press/World Scientific, London/Singapore, 2005. ISBN 1-86094-135-4.

[17] C. Grimm et al. Trust issues in shibboleth-enabled federated grid authentication and authorization infrastructures supporting multiple grid middleware. In *Proc. of the 1st International Interoperability and Interoperation Workshop (IGIIW) at e-Science 2007, Bangalore*, 2007.

[18] P. Kleijer, E. Nakano, T. Takei, H. Takahara, and A. Yoshida. API for Grid Based Visualization Systems. In *GGF 12 Workshop on Grid Application Programming Interfaces*, 2004.

[19] T. Köckerbauer, M. Polak, T. Stütz, and A. Uhl. GVid - Video Coding and Encryption for Advanced Grid Visualization. In *Proceedings of the first Austrian Grid Symposium, Linz*, 2005.

[20] T. Moses et al. *eXtensible Access Control Markup Language*. OASIS Standard, 2005.

[21] M.Riedel et al. Computational steering and online visualization of scientific applications on large-scale hpc systems. In *Proc. of the e-Science 2007, Bangalore, India*. 2007.

[22] M.Riedel et al. Design and Evaluation of a Collaborative Online Visualization and Steering Framework Implementation for Computational Grids. In *Proc. of the 8th IEEE/ACM Int. Conf. on Grid Comp, Austin, USA*. 2007.

[23] M.Riedel et al. Requirements and Design of a Collaborative Online Visualization and Steering Framework for Grid and e-Science Infrastructures. In *Online Proc. of German e-Science Conference, Baden-Baden*. http://edoc.mpg.de/display.epl?mode=doc &id=316630&col=100&grp=1414.

[24] P. Pacheco. *Parallel Programming with MPI*. Morgan Kaufmann, 1996. ISBN 1558603395.

[25] R. Ratering et al. GridBeans: Supporting e-Science and Grid Applications. In *2nd IEEE International Conference on e-Science and Grid Computing (E-Science 2006), Amsterdam, The Netherlands*, 2006.

[26] M. Riedel et al. Visit/gs: Higher level grid services for scientific collaborative online visualization and steering in unicore grids. In *Proc. of 6th International Symposium on Parallel and Distributed Computing 2007 (ISPDC2007), Linz, Austria, ISBN 0-7695-2936-4, on CD*, 2007.

[27] R. Spurzem and E. Khalisi. Nbody6, features of the computer-code. 2003. ftp://ftp.ari.uni-heidelberg.de/pub/staff/spurzem/nb6mpi/nbdoc.tar.gz.

[28] A. Streit et al. UNICORE - From Project Results to Production Grids. In L. Grandinetti, editor, *Grid Computing: The New Frontiers of High Performance Processing, Advances in Parallel Computing 14*, pages 357–376. Elsevier.

[29] A. C. V. Ciachini, V. Venturi. *The VOMS attribute certificate format*. Technical Report, OGSA Authorization Working Group, 2005.

[30] V. Venturi et al. Virtual organisation management across middleware boundaries. In *Proc. of the 1st International Interoperability and Interoperation Workshop (IGIIW) at e-Science 2007, Bangalore*, 2007.