Extending Role Based Access Control Model for Distributed Multidomain Applications

Yuri Demchenko, Leon Gommans, Cees de Laat
University of Amsterdam, System and Network Engineering Group
Kruislaan 403, NL-1098 SJ Amsterdam, The Netherlands
{demch, lgommans, delaat}@science.uva.nl

Abstract. This paper presents the results related to the development of a flexible domain-based access control infrastructure for distributed Grid-based Collaborative Environments and Complex Resource Provisioning. The paper proposes extensions to the classical RBAC model to address typical problems and requirements in the distributed hierarchical resource management such as: hierarchical resources policy administration, user roles/attributes management, dynamic security context and authorisation session management, and others. It describes relations between the RBAC and the generic AAA access control models and defines combined RBAC-DM model for domain-based access control management and suggests mechanisms that can be used in the distributed service-oriented infrastructure for security context management. The paper provides implementation details on the use of XACML for fine-grained access control policy definition for domain based resources organisation and roles assignments in RBAC-DM. The paper is based on experiences gained from the major Grid-based and Grid-oriented projects in collaborative applications and complex resource provisioning.

1 Introduction

Role Base Access Control (RBAC) is an industry recognized and widely accepted access control model that naturally integrates with effective Identity management technologies. However, at the same time its practical implementation in complex research and industry environment for advanced collaborative and resource provisioning scenarios reveals a number of problems. Most of these problems are originated from the industry and research community gradually moving to the Grid and Web Services based Service Oriented Architecture (SOA) [1, 2]. SOA suggests service applications decomposition and decoupling including separation of different component in the traditional access control model such as Authentication, Authorisation, Identity and Attribute management. Although relying on secure network layer, all service oriented security services in SOA are bound to messages exchanged between services representing a user or requestor and a target service or resource. Classic RBAC provides a good model for internal organisational access control and scales bad in distributed and multi-organisational environment.

The generic Authentication, Authorisation, Accounting (GAAA) architecture, described in [3, 4], proposes a general model for the Authentication (AuthN), Authorisation (AuthZ), Accounting services operation and their integration with typical client/server applications. Conceptual Authorisation framework discussed in the OGSA informational document [5] suggests the GAAA-AuthZ as a basic model for the Grid service-oriented environment.

This paper describes our experiences when developing a flexible, customer-driven, security infrastructure for Grid based Collaborative Environment (GCE) and Complex Resource Provisioning (CRP) in general. These two use cases are analysed to explain specific requirements to multidomain access control and suggest RBAC extensions for multidomain applications.

The presented research and proposed solution are specifically oriented for using with the popular Grid middleware being developed in the framework of large international projects such as EGEE (http://public.eu-egee.org/) and Globus Alliance (http://www.globus.org/). The middleware provides a common communication/messaging infrastructure for all resources and services exposed as Grid services, and also allows for a uniform security configuration at the service container or messaging level.

The paper is organized as follows. Section 2 describes the Virtual Laboratory organisation in GCE as a basic use case for domain based resource organisation and management and refers to more general CRP requirements. Section 3 discusses what functionality is currently available in known RBAC implementations and identifies extensions to address specifics in controlling access to distributed hierarchical resources. Section 4 compares RBAC and GAAA access control model and identifies mechanisms to express and convey domain related dynamic security context. Section 5 provides practical suggestions and an example of using XACML for policy expression in hierarchical multidomain access control.

2 Domain Based Resource Management in GCE

The research community and processing industry makes extensive use of advanced computing resources and unique equipment which are associated and virtualised in a form of the Virtual Organisation (VO) or Virtual Laboratory (VL) [6]. VL provides a flexible framework for associating instruments, resources and users into distributed interactive collaborative environment. However, committed to the VL resource still remain in the possession and under direct administration of their original owner enterprises.

The following administrative and security domains can be defined for user, resources, policy and trust management:

1) Facility that provides administrative/legal platform for all further operational associations; may define what kind of technologies, formats, credentials can be used.

2) VL that can be created on the basis of the VL agreement that defines VL resources, common services (first of all, information/registry and security), administrative structure and a VL administrator. Trust relations can be established via PKI and/or VL Certificate population.

3) Experiment/Project defined together with the VL resources allocation, members, task/goals, stages, and additionally workflow. It is perceived that experiment related context may change during its lifetime.

4) Experiment session that may include multiple Instrument sessions and Collaborative sessions that involves experiment members into interactions.

5) Collaborative session – user interactive session.

Experiment session may include multiple Instrument sessions and Collaborative sessions that involves experiment members into interactions.

In the above provided classification domains are defined (as associations of entities) by common policy under single administration, common namespace and semantics, shared trust, etc. In this case, domain related security context may include: namespace aware names and ID's, policy references/ID's, trust anchors, authority references, and also dynamic/session related context. For the generality, domains can be hierarchical, flat or organized in the mesh, but all these cases require the same basic functionality from the access control infrastructure to manage domain and session related security context.

The Domain-based resource management model (DM) closer reflects business practice among cooperating organisations contributing their resources (instruments, other facilities and operator personal) to create a Virtual laboratory that can run complex experiments on request from customers. To become consistent the DM should be supported by corresponding organisation of the access control infrastructure.

Figure 1 illustrates relations between major components in the hierarchical DM resource management and security model. The following suggestions were used when creating this abstraction of the DM [6]:

1) physically Instruments are located at the Facility but logically they are assigned to the VL and next allocation to the Experiment. Full context Instrument name will look like:

```
ResourceDM:Facility:VirtualLab:Experiment:InstrModel
```

2) users/members of collaborative sessions are assigned to the Experiment, managerial and operator personnel belongs to VL and Facility and may have specific and limited functions in the Experiment;

3) particularly, domain based restrictions/policy can be applied to (dynamic) role assignment;

4) additionally, administrative rights/functions can be delegated by the superior entity/role in this hierarchical structure;

5) Trust Anchors (TA) can be assigned to hierarchical domain related entities to enable security associations and support secure communication. VL TA1 is suggested as minimum required in DM, Experiment TA2 may be included into the Experiment description. Collaborative session security association can be supported by AuthZ tickets.

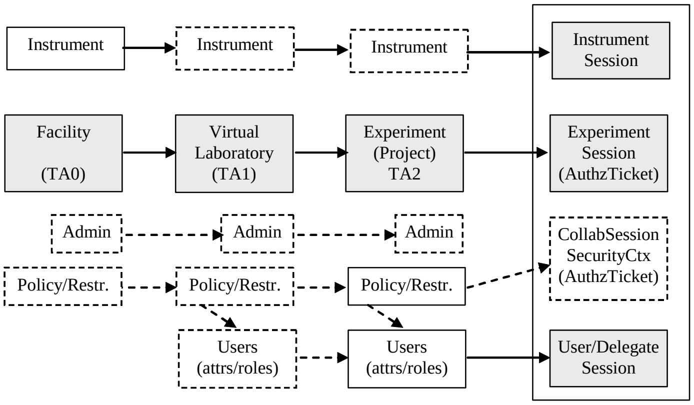

Fig. 1. Hierarchical Domain based Resource management in GCE

The Experiment description plays an important role in the DM security infrastructure, it is created by the experiment owner as a semantic object on the basis of a signed Experiment agreement (and in the context of the overall VL agreement). It contains all the information required to run the analysis, including the Experiment ID, allocated/provisioned instruments, assigned users and roles, and a trust/security anchor(s) in the form of the resource and, additionally, the customer's digital signature(s). The experiment description provides experiment-dependent configuration data for other services to run the experiment and manage the dynamic security context.

VL and Experiment/Project resources can be provisioned dynamically on-demand. In this case the VL/Experiment lifecycle or operation will include resource and service provisioning stage. The recent paper [7] by authors discusses other practical issues of implementing DM for the general CRP in Grid environment. The paper distinguishes 2 major stages in CRP: resource reservation and the reserved resource access or consumption. The reservation and allocation stage includes 4 basic steps: resource lookup, complex resource composition (including alternatives), reservation of individual resources and their association with the reservation ticket/ID, and finally delivery or deployment. The reservation stage may require execution of complex procedures that may also request individual resources authorisation in multiple administrative and access control domains.

3 Generic RBAC and Domain Based Resource Management

Generic RBAC model [8, 9, 10] provides an industry recognised solution for effective user roles/privileges management and policy based access control. It extends Discretional Access Control (DAC) and Mandatory Access Control (MAC) models with more flexible access control policy management adoptable for typical hierarchical roles and responsibilities management in organisations, but at the same time it suggest a full user access control management from user assignment to granting permissions. This can be suitable for internal organisational environment and particularly for human access rights management but reveals problems when applied to distributed service-oriented environment.

Sandhu in his two research papers [8, 9] describes 4 basic RBAC models:

- Core RBAC (RBAC0) that associates Users with Roles (U-R) and Roles with Permissions (R-P);
- Hierarchical RBAC (RBA1) that adds hierarchy to roles definition;
- Constrained RBAC (RBAC2) that extends RBAC0 with the constrains applied to U-R and R-P assignment;
- Consolidated RBAC (RBAC3) that adds role hierarchy to RBAC2.

Further RBAC development took place with publishing ANSI INCITS 359-2004 standard [10] that actually re-defined first three basic RBAC models in the context of static or dynamic separation of duties (SSD vs DSD). The standard also proposes RBAC functional specification that can be used for developing generic RBAC API.

In both models, initial Sandhu's and ANSI RBAC, there is a notion of the user session which is invoked by a user and provides instant session-based U-R association. Final result/stage of the RBAC functionality are permissions assigned to the user based on static or dynamic U-R and R-P assignment. RBAC doesn't consider (user) permissions enforcement on the resource or access object. This functionality can be attributed to other more service-oriented frameworks such as ISO/ITU PMI [11] or generic AAA [3, 4, 5].

Many studies suggest RBAC as a natural method to model the security requirements in service oriented environment but at the same time they argue for application specific extensions, e.g., for user group organisation including additional group/team defined restrictions on separation of duties, roles/attributes combinations, etc. [12, 13].

The papers [14, 15] propose an extension of the generic RBAC model the usage control (UCON) based authorisation framework for collaborative application that specifically addresses access control to the consumable resources or which access should be coordinated among a group of users. This is achieved by using obligations, resource/environmental conditions, introducing mutable resource and user attributes, and applying ongoing control. The proposed implementation uses XACML as a policy expression language with proprietary defined the Obligation element. However, detailed analysis of the proposed UCON publications and implementations revealed that the UCON framework uses centralised policy management, environment and attributes control that may have a principal problem of races when using conditions/obligations on mutable attributes. Proposed usage session doesn't allow full functionality required for generic authorisation session management in a multi-domain environment.

Generic RBAC historically was designed for centralized and autonomous access control management and inherits the following problems when applied to typical service-oriented security infrastructure:

- it is not directly applicable and integrated with/to service-oriented applications, although it is well applicable for such use cases as enterprise database/facility access control;

- doesn’t separate basic functional components that have place in typical Enterprise Identity management and Access control infrastructure such as AuthN and AuthZ service, Attribute Authority, Policy Authority;
- User session, as defined in RBAC, is not present in typical PMI and AAA.

But at the same time it defines/specifies generic functional components that can be used in more service oriented access control models such as generic AAA. Practical RBAC implementation requires resolution of many other administration and security related issues left out of scope in classical RBAC such as:

- policy expression and management,
- rights/privileges delegation,
- AuthZ session management mechanisms,
- security context management in distributed dynamic scenario
- scalability in distributed and multidomain applications.

The two basic implementations of the generic RBAC model are Access Control Lists (ACL) that can be rather applications/implementation specific, and an emerging industry standard eXtensible Access Control Markup Language (XACML) that defines a rich policy expression format and simple Request/Response messages format for PEP-PDP communication [16]. XACML extensions and special profiles address most of mentioned above issues at the standard level. However, there are no widely used practical implementations for this new functionality.

The RBAC-DM (note, in most cases we will use abbreviations DM and RBAC-DM as equivalents) that combine the generic RBAC with domain based resource and roles management can address most of above mentioned issues at the practical level by introducing domain related security context that actually reflects natural for cooperating entities/enterprises administration model and separation of duties. Use of Experiment and Collaborative session allows to implement delegations and minimum privileges principle in access control management but in its own turn requires consistent authorisation session context handling. Using AuthZ ticket with full session context in DM allows for distributed access control management and decoupling access control infrastructure components in a distributed environment.

In summary, DM provides the following benefits:

1) reflects distributed hierarchical management model natural in distributed cooperative business environment;

2) multiple and hierarchical policies management that reflects hierarchical resource organisation;

3) allows for dynamic roles assignment with the domain defined restrictions;

4) supports dynamic security context management;

5) provides mechanisms for supporting multidomain authorisation sessions.

4 Relation between RBAC and GAAA Access Control Models

A Resource or Service in GCE is protected by the site access control system that relies on both AuthN of the user and/or request message and AuthZ that applies access control policies against the service request. It is essential in a service-oriented model that AuthN credentials are presented as a security context in the AuthZ request and that they can be evaluated by calling back to the AuthN service and/or Attribute Authority (AttrAuth). This also allows for loose coupling of services (providing domain independency even for hierarchical DM).

The GAAA AuthZ model includes such major functional components as: Policy Enforcement Point (PEP), Policy Decision Point (PDP), Policy Authority Point (PAP). It is naturally integrated with the RBAC separated User-Role and Role-Privilege management model that can be defined and supported by separate policies.

The Requestor requests a service by sending a service request ServReq to the Resource's PEP providing information about the Subject/Requestor, Resource, Action according to the implemented authorisation model and (should be known) service access policies.

In a simple scenario, the PEP sends the decision request to the (designated) PDP and after receiving a positive PDP decision relays a service request to the Resource. The PDP identifies the applicable policy or policy set and retrieves them from the Policy Authority, collects the required context information and evaluates the request against the policy.

In order to optimise performance of the distributed access control infrastructure, the AuthZ service may also issue AuthZ assertions in the form of AuthzTicket that are based on the positive AuthZ decision and can be used to grant access to subsequent similar requests that match the AuthzTicket. To be consistent, AuthzTicket must preserve the full context of the authorisation decision, including the AuthN context/assertion and policy reference.

A typical DM access control use-case may require a combination of multiple policies and also multi-level access control enforcement, which may take place when combining newly-developed and legacy access control systems into one integrated access control solution. The GCE experiments may apply different policies and require different user credentials depending on the stage of the experiment.

DM can improve overall services manageability but requires additional/corresponding mechanisms for dynamic security context management. It is also suggested that using AuthZ ticket with full session context will simplify distributed access control management in a hierarchical DM and allow for decoupling access control infrastructure components in a distributed environment.

Figure 2 illustrates relations between classical conceptual RBAC model and GAAA AuthN/AuthZ services. The User-Role assignment (defined in RBAC by User session) in GAAA is provided at the stage of the user authentication when a set of role are assigned to the authenticated user. It is important that the user provides sufficient identity credentials which will next define a set of assigned to his/her roles. Mapping between user Roles and Permissions in general/total are defined by the access control policy that is used to evaluate a User request to the Resource. Permitted actions relayed to the Resource by PEP and may be confirmed by the AuthZ assertion that can be used for further access during AuthZ session duration. Figure 2 helps also to understand why many authors and implementers criticise that conceptual RBAC model doesn't fit into majority of enterprise and organisational applications that actually implement another service-oriented access control model that separates AuthN, AuthZ and IdP/Attribute Authority services. The picture also illustrates difference between RBAC User session and AuthZ session.

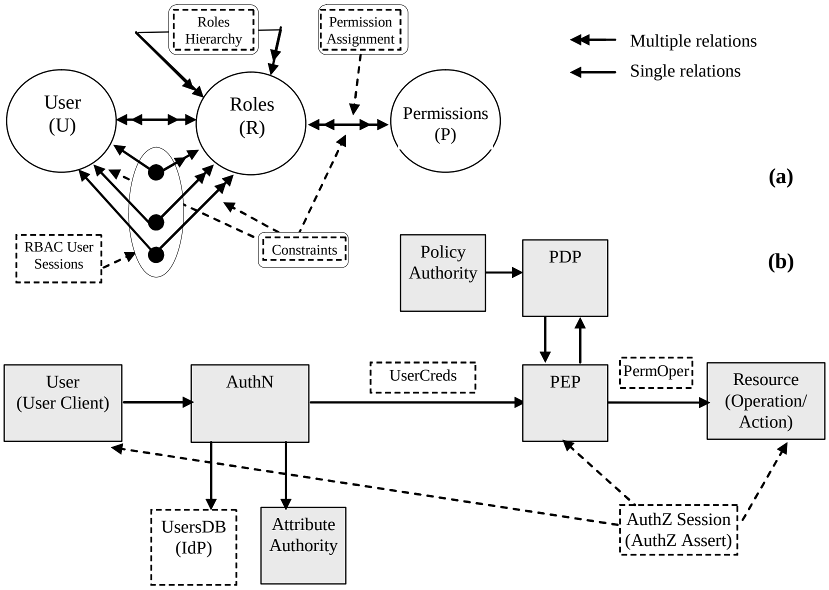

Fig. 2. Relation between (a) RBAC [9] and (b) GAAA-AuthZ/AuthN services

Detailed analysis of how dynamic security context can be managed in Grid based applications is discussed in the paper [17] that identifies the following mechanisms and components to mediate a dynamic security context:

- Service and requestor/user ID/DN format that should allow for both using namespaces and context aware names semantics.
- Attribute format (either X.509/X.521 or URN/SAML2.0 presentation).
- Context aware XACML policy definition using the Environment element of the policy Target element (see next section for detailed discussion).
- Security assertions (e.g., tickets or tokens) used for User and AuthZ session management and for provisioned resource/service identification.
- Workflow as primarily used for complex/combined services orchestration can be also used for managing dynamic security context.

5 Using XACML for Policy Expression in RBAC-DM

A XACML policy is defined for the so-called target triad “Subject-Resource-Action” which can also be completed with the Environment element to add additional context to instant policy evaluation. The XACML policy format can also specify actions that must be taken on positive or negative PDP decisions in the form of an optional Obligation element. The Environment and Obligation elements can be used for multidomain AuthZ decision combination in DM.

A decision request sent in a Request message provides context for the policy-based decision. The policy applicable to a particular decision request may be composed of a number of individual rules or policies. Few policies may be combined to form a single policy that is applicable to the request. XACML specifies a number of policy and rule combination algorithms. The Response message may contain multiple Result elements, which are related to individual Resources.

XACML policy format provides few mechanisms of adding and handling context during the policy selection and request evaluation, in particular: the policy identification using the Target element, the Environment element both in the Target and in the rules definition, and the namespace aware attributes semantics.

The DM makes extensive use of both XACML core specification and its special profiles for RBAC [18] and hierarchical resources [19]. Hierarchical policy management and dynamic rights delegation, that is considered as an important functionality in DM, can be solved with the XACML v3.0 administrative policy [20].

The XACML RBAC profile [19] provides extended functionality for managing user/subject roles and permissions by defining separate Permission <PolicySet>, Role <PolicySet>, Role Assignment <Policy>, and HasPrivilegeOfRole <Policy>. It also allows for using multiple Subject elements to add hierarchical group roles related context in handling RBAC requests and sessions, e.g., when some actions require superior subject/role approval to perform them. In such a way, RBAC profile can significantly simplify rights delegation inside the group of collaborating entities/subjects which normally requires complex credentials management.

The XACML hierarchical resource profile [19] specifies how XACML can provide access control for a Resource that is organized as a hierarchy. Examples include file systems, data repositories, XML documents and organizational resources which example is the DM. The profile introduces new Resource attributes identifiers that may refer to the “resource-ancestor“, “resource-parent“, or “resource-ancestor-or-self“.

XACMLv3.0 administrative policy profile [20] introduces extensions to the XACML v2.0 to support policy administration and delegation. This is achieved by introducing the PolicyIssuer element that should be supported by related administrative policy. Dynamic delegation permits some users to create policies of limited duration to delegate certain capabilities to others. Both of these functionalities are important for the proposed DM and currently being investigated.

Figure 3 below provides an example of the XACML policy which Target and IDRef bind the policy to the Resource. There may be different matching expression for the Resource/Attribute/AttributeValue when using XACML hierarchical resource profile what should allow to create a policy for the required resource hierarchy in DM. The example also contains the PolicyIssuer element that is related to the policy administration. In our example the PolicyIssuer is declared as “cnl:VLab031:trusted”, and the PDP will rely on already assigned PAP and established trust relations. In case, when other entity is declared as a PolicyIssuer, the PDP should initiate checking administrative policy and delegation chain.

```
<PolicySet>
 <Target/>
 <Policy PolicyId="urn:oasis:names:tc:xacml:1.0:cnl:policy:CNL2-XPS1-test"
   RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-
     algorithm:deny-overrides">
  <Description>Permit access for CNL3 users with specific roles</Description>
  <PolicyIssuer>
```

```
    <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
        DataType="http://www.w3.org/2001/XMLSchema#string">
      <AttributeValue> urn:oasis:names:tc:xacml:3.0:issuer:cnl:VLab031:trusted
        </AttributeValue>
    </Attribute>
  </PolicyIssuer>
  <Target>
   <Resources><Resource>
    <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-equal">
      <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#anyURI">
           http://resources.collaboratory.nl/Phillips_XPS1</AttributeValue>
      <ResourceAttributeDesignator
          AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
             DataType="http://www.w3.org/2001/XMLSchema#anyURI"/>
    </ResourceMatch>
   </Resource></Resources>
  </Target>
 <Rule RuleId="urn:oasis:names:tc:xacml:1.0:cnl:
               policy:CNL2-XPS1-test:rule:ViewExperiment" Effect="Permit">
  <Target>
   <Actions><Action>
    <ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
      <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
              ViewExperiment</AttributeValue>
      <ActionAttributeDesignator
          AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
               DataType="http://www.w3.org/2001/XMLSchema#string"/>
     </ActionMatch>
    </Action></Actions>
  </Target>
  <Condition FunctionId="urn:oasis:names:tc:xacml:1.0:
                   function:string-at-least-one-member-of">
    <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
     <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
         analyst</AttributeValue>
     <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
         customer</AttributeValue>
    </Apply>
    <SubjectAttributeDesignator DataType=http://www.w3.org/2001/XMLSchema#string
       AttributeId="urn:oasis:names:tc:xacml:1.0:subject:role"
       Issuer="CNL2AttributeIssuer"/>
   </Condition>
  </Rule>
 </Policy>
</PolicySet>
```

Fig. 3. XACML PolicySet containing PolicyIssuer element as defined by XACML3.0.

6 Conclusion and Summary

The results presented in this paper are part of the ongoing research and development of the security infrastructure for user controlled multidomain services and its application to complex resource provisioning. This work is being conducted by the System and Network Engineering (SNE) Group in the framework of different EU and Dutch nationally and industry funded projects including EGEE, Phosphorus and GigaPort Research on Network.

The definition of the Domain based access control model RBAC-DM and proposed solutions described in this paper are based on practical experience we have gained whilst designing and developing an open collaborative environment within the Collaboratory.nl and VL-e projects. RBAC-DM reflects distributed hierarchical management model typical for industrial collaborative infrastructure and has additional features for domain related security context management. Use of Experiment and Collaborative sessions, supported by relevant session's security context management, allows for dynamic roles assignment with the domain defined restrictions, including delegation and minimum privileges principle.

The paper identifies major mechanisms that can be used for expressing and transferring dynamic security context in Grid and Web Services applications using of XML technologies. The proposed solutions are being implemented in the GAAA Toolkit [21] as a GAAAPI package that can be also used with other popular AuthZ frameworks such as GT4-AuthZ and gLite AuthZ frameworks.

Proposed RBAC-DM and its suggested implementation in GAAAPI make extensive use of XACML core specification and its special profiles for RBAC and hierarchical resources, and also XACML v3.0 administrative policy. Provided XACML policy example illustrates most of the discussed features. Practical implementation of this additional functionality will require special extension to the popular Open Source SunXACML library that is being developed as a part of the GAAAPI package.

Another important component that requires additional research and wider potential use cases analysis is the AuthZ ticket definition as a key mechanism and a component of the AuthZ session management functionality. Initial modelling with the GAAAPI package demonstrated effectiveness and sufficient increase of the AuthZ service performance when controlling remote instruments. AuthZ session support in Grid/OGSA applications was recognised as an important functionality and accepted as a work item by the OGF OGSA-AuthZ working group [22].

The authors believe that the proposed RBAC-DM access control architecture for GCE/CRP and related technical solutions will also be useful to the wider community that has similar problems with managing access control to distributed hierarchically organised resources in dynamic/on-demand services provisioning.

References

1. Foster, I. et al (2006). The Open Grid Services Architecture, Version 1.5. Global Grid Forum. Retrieved October 30, 2006, from http://www.ggf.org/documents/GFD.80.pdf

2. "Web Services Architecture". W3C Working Draft 8, August 2003. - http://www.w3.org/TR/ws-arch/

3. Vollbrecht, J., P. Calhoun, S. Farrell, L. Gommans, G. Gross, B. de Bruijn, C. de Laat, M. Holdrege, D. Spence, "AAA Authorization Framework," Informational RFC 2904, Internet Engineering Task Force, August 2000. ftp://ftp.isi.edu/in-notes/rfc2904.txt

4. RFC2903 – "Generic AAA Architecture", C. de Laat, G. Gross, L. Gommans, J. Vollbrecht, D. Spence, IETF Aug 2000, ftp://ftp.isi.edu/in-notes/rfc2903.txt

5. GFD.38 Conceptual Grid Authorization Framework and Classification. M. Lorch, B. Cowles, R. Baker, L. Gommans, P. Madsen, A. McNab, L. Ramakrishnan, K. Sankar, D. Skow, M. Thompson - http://www.ggf.org/documents/GWD-I-E/GFD-I.038.pdf

6. Demchenko, Y., L. Gommans, C. de Laat, A., van Buuren, R. Domain Based Access Control Model for Distributed Collaborative Applications". Accepted, The 2nd IEEE International Conference on e-Science and Grid Computing.

7. Using SAML and XACML for Complex Authorisation Scenarios in Dynamic Resource Provisioning, by Demchenko Y., L. Gommans, C. de Laat. The Second International Conference on Availability, Reliability and Security (ARES 2007), April 10-13, 2007, Vienna. Accepted paper.

8. Sandhu, R. & Samarati, P., 1994. “Access Control: Principles and Practice”, IEEE Communication Magazine, September 1994, pp. 40-48.

9. Sandhu, R., Coyne, E. J., Feinstein, H. L. & Youman, C.E. 1996, "Role-Based Access Control Models", IEEE Computer, February 1996, pp. 38-47.

10. Information Technology - Role Based Access Control, Document Number: ANSI/INCITS 359-2004, InterNational Committee for Information Technology Standards, 3 February 2004, 56 p.

11. ITU-T Rec. X.812(1995) | ISO/IEC 10181-3:1996, Information technology - Open systems interconnection - Security frameworks in open systems: Access control framework.

12. Caelli W., Rhodes A., "Implementation of active role based access control in a collaborative environment", http://www.isi.qut.edu.au/research/publications/technical/ qut-isrc-tr-1999-005.pdf

13. Thomas, R. K. 1997, "Team-based Access Control (TMAC): A Primitive for Applying Role-based Access Controls in Collaborative Environments", Proceeding of the Second ACM Workshop on Role-Based Access Control, ACM, November 1997, pp. 13-19.

14. Park J.S., R Sandhu, “The UCONabc usage control model”, ACM Transaction on Information and System Security, 7(1), February 2004.

15. Xinwen Zhang, Masayuki Nakae, Michael J. Covington, and Ravi Sandhu, A Usage-based Authorization Framework for Collaborative Computing Systems, in the proceedings of ACM Symposium on Access Control Models and Technologies (SACMAT), 2006.

16. Godik, S. et al, “eXtensible Access Control Markup Language (XACML) Version 2.0”, OASIS Working Draft 04, 6 December 2004, available http://docs.oasis-open. org/xacml/access_control-xacml-2_0-core-spec-cd-04.pdf

17. Demchenko, Y., L. Gommans, C. de Laat, A. Taal, A. Wan, O. Mulmo, "Using Workflow for Dynamic Security Context Management in Complex Resource Provisioning", 7th IEEE/ACM International Conference on Grid Computing (Grid2006), Barcelona, September 28-30, 2006, pp.72-79.

18. “Core and hierarchical role based access control (RBAC) profile of XACML v2.0”, OASIS Standard, 1 February 2005, available from http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-rbac-profile1-spec-os.pdf

19. “Hierarchical resource profile of XACML 2.0”, OASIS Standard, 1 February 2005, available from http://docs.oasis-open.org/xacml/access_control-xacml-2.0-hier_profile-spec-cd-01.pdf

20. “XACML 3.0 administrative policy,” OASIS Draft, 10 December 2005. [Online]. Available from http://docs.oasis-open.org/access_control.

21. Generic Authorization Authentication and Accounting. [Online]. Available: http://www.science.uva.nl/research/ air/projects/aaa/

22. OGSA Authorization WG (OGSA-AUTHZ-WG). [Online]. Available: http://www.ogf.org/gf/group_info/view.php?group=ogsa-authz-wg