Using SAML and XACML for Complex Authorisation Scenarios in Dynamic Resource Provisioning

Yuri Demchenko, Leon Gommans, Cees de Laat
System and Network Engineering Group, University of Amsterdam
{demch, lgommans, delaat}@science.uva.nl

Abstract

This paper presents ongoing research and current results on the development of flexible access control infrastructures for complex resource provisioning in Grid-based collaborative applications and on-demand network services provisioning. The paper identifies basic resource provisioning models and specifies major requirements to Authorisation (AuthZ) service infrastructure to support these models and focus on two main issues – AuthZ session support and policy expression for complex resource models. For the practical implementation, we investigate the use of two popular standards SAML and XACML for complex authorisation scenarios in dynamic resource provisioning across multiple administrative and security domains. The paper describes a proposed XML based AuthZ ticket format that is capable of supporting extended AuthZ session context. Additionally, the paper discusses what specific functionality should be added to existing Grid-oriented authorization frameworks to handle dynamic domain-related security context including AuthZ session support. The paper is based on experiences gained from major Grid based and Grid oriented projects such as EGEE, NextGrid, Phosphorus and GigaPort Research on Network.

1. Introduction

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 Laboratory (VL) or Virtual Organisation (VO). Such a virtualisation of resources and users can be created on-demand dynamically using available Grid technologies and middleware, based on experiment or service agreement and terminated once the experiment has been completed or service/resource delivered or consumed. Important component of the distributed VL infrastructure is a dedicated network infrastructure that should also be provisioned on-demand. Both VL workspace and on-demand network infrastructure provisioning can be considered as particular cases of the general Complex Resource Provisioning (CRP).

In general, complex resources may have different logical organisation and are represented as ordered or unordered resource collection, or hierarchical structure. CRP operational model should be capable to support different resource organisation and consequently different provisioning and access control models. Most of existing CRP solutions address separately initial resource reservation and allocation and following resource or service access and consumption.

This paper proposes further development of the generic Authentication, Authorisation, and Accounting (AAA) Authorisation framework (GAAA-AuthZ) [1, 2, 3] to support complex AuthZ scenarios in on-demand multidomain resource provisioning. The paper also explores the possibilities and presents our experiences with such technologies as SAML and XACML that provide rich functionality for the CRP policy expression and dynamic security context management in distributed multidomain access control. As native XML technologies, SAML and XACML allow natural integration with the Grid and Web Services security services infrastructure.

We analyse two major use cases to define required functionality for the distributed multidomain AAA services to support CRP: Optical Light Path Provisioning (OLPP) [4] and Grid-based Collaborative Environments (GCE) [5].

Approaches and technical solutions proposed in this paper are based on an extended gap analysis undertaken in the framework of the SURFnet GigaPort Research on Network (GigaPort-RoN)[1] project to identify general and specific requirements to access control infrastructure for on-demand network services provisioning, in particular, OLPP [6].

[1] http://ron.gigaport.nl/

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[2] and Globus Alliance[3].

The paper is organized as follows. Section 2 describes briefly two basic use cases where the CRP is required: GCE and OLPP, and proposes generalized model for distributed CRP that separates resource reservation, resource allocation, and resource access or consumption stages. Different CRP and AuthZ sequences are discussed.

Section 3 describes the AuthZ ticket format that provides necessary functionality for the extended provisioning and user/application AuthZ session context management. Section 4 discuses what functionality is available in the XACML specification suite for expressing access control policies to complex distributed resources with different logical organisations (multiple, multiple constrained, and hierarchical) and different user access rules that also may require domain based hierarchical user roles and permissions management..

Section 5 describes how the domain related dynamic security context and authorisation session management can be added to the standard Grid and Web Services oriented authorisation frameworks.

2. CRP operational models and AAA Authorisation service requirements

Network on-demand provisioning using OLPP model [4] and Virtual Laboratory in GCE [5] represents two major use cases for the general CRP. Although different in current implementations, they can be abstracted to the same CRP operational model when considering their implementation with the SOA based Grid or Web Services [7, 8].

The typical on-demand resource provisioning includes 2 major stages: resource reservation and the reserved resource access or consumption. In its own turn, 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/allocation. The reservation stage may require execution of complex procedures that may also request individual resources authorisation. This process can be controlled by the AAA driving policy or described as combination of the provisioning workflow and related AuthZ policy.

In the discussed CRP model, 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 (TA), authority references, and also dynamic/session related context [9]. For the generality, domains can be hierarchical, flat or organized in the mesh, but all these cases require the same basic functionality for the access control infrastructure to manage domain and session related security context.

CRP for the hierarchical and distributed resources management model requires the following functionality from the GAAA-AuthZ infrastructure:

- multiple policies processing and combination;
- attributes/rules mapping/converting based on interdomain trust management infrastructure;
- hierarchical roles/permissions management, including administrative policies and delegation;
- policy support for different logical organisation of resources, including possible constrains on resource combination and interoperation.

Figure 1 illustrates major interacting components in the multi-domain CRP using OLPP as an example:

- User/Requestor.
- Target end service or application,
- Multiple Network elements (NE) (related to the Network plane).
- Dynamic Resource Allocation and Management (DRAM) service (typically related to the Control plane).
- AAA service controlling access to the domain-related resources that can also operate own communication infrastructure.
- Token Validation Service (TVS) that allows efficient authorisation decision enforcement when accessing reserved resources.

Described above CRP model can be generalized for both discussed usecase if we consider virtual Workspace elements (WSE) in the hierarchical VL organisation as separate resource domains that can be logically organised into different structures and described with the same attribute types as traditional network domains.

[2] http://public.eu-egee.org/

[3] http://www.globus.org/

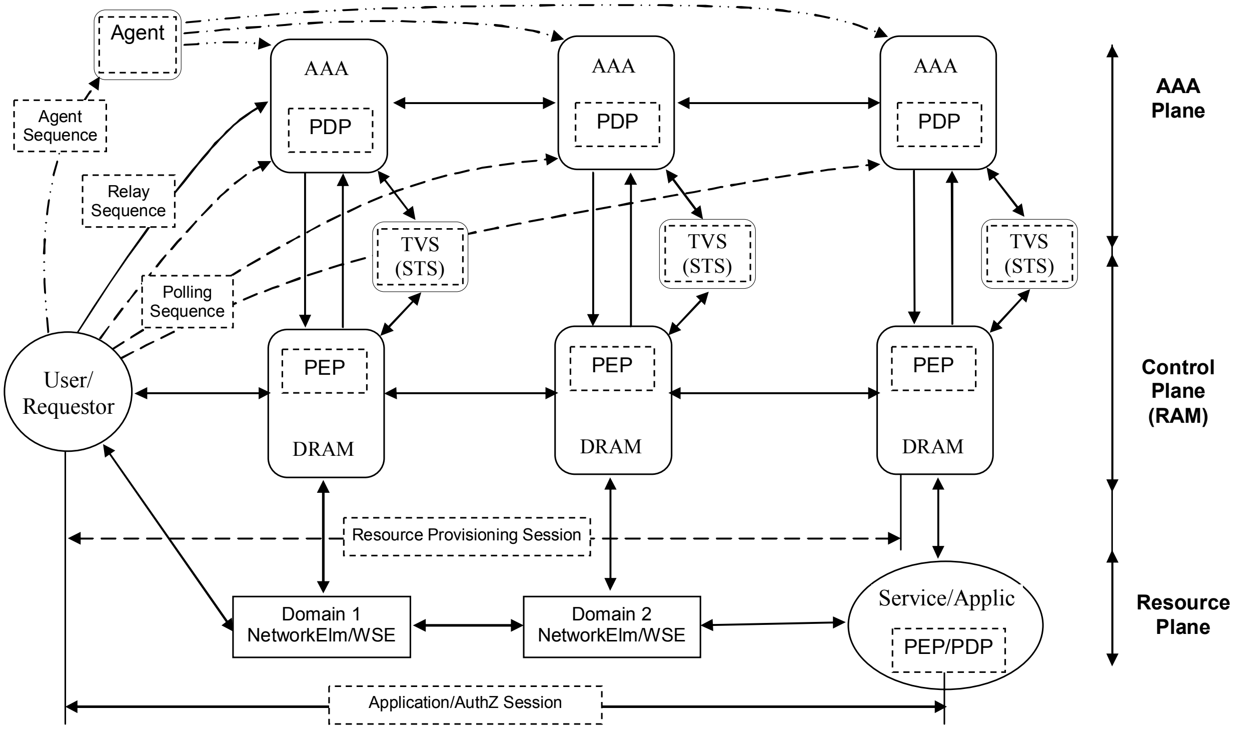

Figure 1. Components involved into complex resource provisioning and basic sequences (agent, relay, and polling)

The figure illustrates different provisioning models or sequences that can be executed when composing a complex resource:

- Polling sequence when the User client polls all resources or network domains, builds the path and makes reservation.
- Relay or hop-by-hop reservation sequence when the user contacts only the local network domain/provider providing destination address, and each consecutive domains provides path to the next domain.
- Agent sequence when the User delegates network provisioning negotiation to the Agent that will take care of all necessary negotiations to provide required network path to the User. A benefit of outsourcing resource provisioning is that the Agents can maintain their own reservation and trust infrastructure.

Access to the Resource or Service is controlled by the DRAM and protected by the AAA service that enforces Resource access control policy by placing Policy Enforcement Point (PEP) gateway at the entrance of DRAM. Depending on the basic AAA-AuthZ sequence (push, pull or agent) [2, 3], the Requestor can send a Resource access request to the Resource or service (which in our case are represented by DRAM) or an AuthZ decision request to the designated AAA server which in this case will act as a Policy Decision Point (PDP). The PDP identifies the applicable policy or policy set and retrieves them from the Policy Authority (PAP), collects the required context information and evaluates the request against the policy.

The User can present as much (or as little) information about the Subject/Requestor, Resource, Action as it decides necessary according to the implemented authorisation model and Resource access control policies. Policy Decision Point (PDP) which is the part of the AAA AuthZ service evaluates request and makes decision whether to grant access or not. Based on the positive AuthZ decision (in one domain) the AuthZ ticket (AzTicket) can be generated by the PDP or PEP and communicated to the next domain where it may be processed as a security context or policy evaluation environment.

It is essential in the Grid/Web Services based service oriented environment that AuthZ decision must rely on both Authentication (AuthN) of the user and/or request message and Authorisation (AuthZ) and AuthN credentials are presented as a security context in the AuthZ decision making.

In order to get access to the reserved resources the

Requestor needs to present the reservation credentials that can be in the form of AuthZ ticket or token (AzTicket or AzToken) which will be evaluated by the PEP to grant access to the reserved network elements or resource. In more complex provisioning scenario token or credentials validation may be outsourced to the TVS service that can additionally support interdomain trust management infrastructure for off-band token and key distribution between DRAM and AAA services. TVS can be implemented as a proprietary AAA-DRAM solution or use one the proposed standard models of the Credential Validation Services (CVS) [10] or WS-Trust Secure Token Service (STS) [11].

Using AuthZ ticket during the reservation stage for communicating interdomain AuthZ context is essential to ensure effective decision making. At the service access/consumption stage the reserved resource may be simply identified by the reservation ID created as a result of the successful reservation process. To avoid significant policy enforcement overhead when handing service reservation context, the ticket can be cached by DRAM or TVS in each domain and referred to with the AzToken that can be much smaller and even communicated in-band. At the Resource PEP it can be compared with the cached AzTicket and will allow for local to the PEP access decision. Such an access control enforcement model is being implemented in the Token Based Network (TBN) and allows for real-time per packet token processing in the packet switched networks up to 1 Gbps [12].

3. Authorisation session ticket format

As discussed in the previous section, there are two types of sessions in the proposed CRP model that require security context management: provisioning session and user or application session. Although provisioning session may require wider security context support, both of them are based on the (positive) AuthZ decision, may have similar AuthZ context and will require similar functionality when considering distributed multi-domain scenarios.

Current AzTicket format and its implementation in the GAAA-AuthZ support extended functionality for distributed multidomain hierarchical resources access control and user roles/permissions management, in particular, administrative policy management (as defined in XACML 3.0 Administrative policy profile), capabilities delegation and conditional AuthZ decision assertion (to support XACML policy obligations). The semantics of AzTicket elements is defined in such a way that allows easy mapping to related elements in other XML-based and AuthZ/AuthN related formats, like the Security Assertion Markup Language (SAML) [13] and the eXtensible Access Control Markup Language (XACML) [14].

Figure 2 illustrates the AzTicket data model and shows the top elements. Figure 3 below provides an example of the XML based AzTicket that can be used for extended AuthZ session security context management. The listing also contains comments that explain a suggested mapping to SAML2.0 Authorisation assertion elements, which demonstrates that even for basic AuthZ session data, few extension elements are required for extended security context expression.

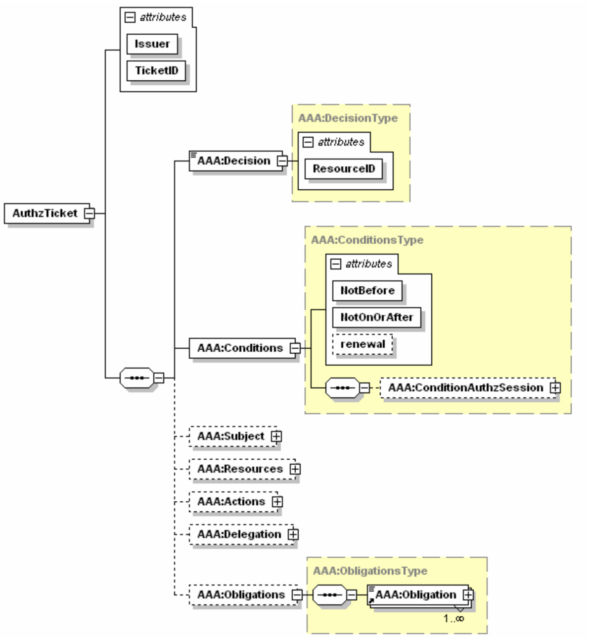

Figure 2. The AzTicket data model and top elements.

The AzTicket contains the following major groups of elements:

- The `Decision` element that holds the PDP AuthZ decision bound to the requested resource or service expressed as the `ResourceID` attribute.
- The `Conditions` element specifies the validity constrains for the ticket, including validity time and AuthZ session identification and additionally context. The extensible `ConditionAuthzSession` element provides rich possibilities for AuthZ context expression.
- The `Actions/Action` complex element contains actions which are permitted for the Subject or its delegates.
- The `Subject` complex element contains all information related to the authenticated Subject

who obtained permission to do the actions, including sub-elements: `Role` (holding subject's capabilities), `SubjectConfirmationData` (typically holding AuthN context), and extendable sub-element `SubjectContext` that may provide additional security or session related information, e.g. Subject's VO, project, or federation.

- The `Delegation` element allows to delegate the capabilities defined by the AzTicket to another Subjects or community. The attributes define restriction on type and depth of delegation
- The `Obligations/Obligation` element can hold obligations that PEP/Resource should perform in conjunction with the current PDP decision.

The AzTicket is digitally signed (as shown in the example) and cached by the Resource's AuthZ service. To reduce communication overhead when using AzTicket for consecutive requests validation, the associated AuthZ token (AzToken) can be generated of the AzTicket. The AzToken may contain just two elements: `TokenID = TicketID` and `TokenValue = SignatureValue`, needed for identification of the cached AzTicket.

```
<AAA:AuthzTicket xmlns:AAA="http://www.aaauthreach.org/ns/#AAA" Issuer="urn:cnl:trusted:tickauth:pdp"
TicketID="cba06d1a9df148cf4200ef8f3e4fd2b3">
  <AAA:Decision ResourceID="http://resources.collaboratory.nl/Philips_XPS1">Permit</AAA:Decision>
  <!-- SAML mapping: <AuthorizationDecisionStatement Decision="*" Resource="*"> -->
  <AAA:Actions>
    <AAA:Action>cnl:actions:CtrlInstr</AAA:Action>     <!-- SAML mapping: <Action> -->
    <AAA:Action>cnl:actions:CtrlExper</AAA:Action>
  </AAA:Actions>
  <AAA:Subject Id="subject">
    <AAA:SubjectID>WHO740@users.collaboratory.nl</AAA:SubjectID>
    <!-- SAML mapping: <Subject>/<NameIdentifier> -->
    <AAA:SubjectConfirmationData>crypto-value-here-Nnld84AggaDkOb5WW4U=</AAA:SubjectConfirmationData>
    <!-- SAML mapping:  EXTENDED <SubjectConfirmationData/> -->
    <AAA:Role>analyst</AAA:Role>
    <!-- SAML mapping:
             <Evidence>/<Assertion>/<AttributeStatement>/<Assertion>/<Attribute>/<AttributeValue> -->
    <AAA:SubjectContext>CNL2-XPS1-2005-02-02</AAA:SubjectContext>
    <!-- SAML mapping:
         <Evidence>/<Assertion>/<AttributeStatement>/<Assertion>/<Attribute>/<AttributeValue> -->
  </AAA:Subject>
  <AAA:Delegation MaxDelegationDepth="3" restriction="subjects">
  <!-- SAML mapping:  LIMITED <AudienceRestrictionCondition> (SAML1.1),
                                       or <ProxyRestriction>/<Audience> (SAML2.0)  -->
    <AAA:DelegationSubjects>
      <AAA:SubjectID>team-member-2</AAA:SubjectID>
      <AAA:SubjectID>team-member-1</AAA:SubjectID>
    </AAA:DelegationSubjects>
  </AAA:Delegation>
  <AAA:Conditions NotBefore="2006-06-08T12:59:29.912Z"
                         NotOnOrAfter="2006-06-09T12:59:29.912Z" renewal="no">
  <!-- SAML mapping: <Conditions NotBefore="*" NotOnOrAfter="*"> -->
    <AAA:ConditionAuthzSession PolicyRef="PolicyRef-GAAA-RBAC-test001" SessionID="JobXPS1-2006-001">
    <!-- SAML mapping: EXTENDED <SAMLConditionAuthzSession PolicyRef="*" SessionID="*"> -->
      <AAA:SessionData>put-session-data-Ctx-here</AAA:SessionData>
      <!-- SAML mapping:  EXTENDED <SessionData/> -->
    </AAA:ConditionAuthzSession>
  </AAA:Conditions>
  <AAA:Obligations>
    <AAA:Obligation>put-policy-obligation(2)-here</AAA:Obligation>
    <!-- SAML mapping:  EXTENDED <Advice>/<PolicyObligation> -->
    <AAA:Obligation>put-policy-obligation(1)-here</AAA:Obligation>
  </AAA:Obligations>
</AAA:AuthzTicket>
<ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
  <ds:SignedInfo> ... </ds:SignedInfo>
  <ds:SignatureValue>e4E27kNwEXoVdnXIBpGVjpaBGVY71Nypos...</ds:SignatureValue>
</ds:Signature>
```

Figure 3. Example of XML based AuthZ ticket format with the capability of preserving extended AuthZ session context. (Note. Comments refer to the suggested SAML2.0 mapping)

Current AzTicket functionality is supported by the GAAAPI package (see section 5 for details). Further development will include adding the following additional functionality:

Elements or attributes that can support mutual AuthZ or session negotiation what is desirable to have even if the negotiation protocol will have own messages format, because the User/AuthZ session credentials have to be bound to requestor/subject credentials and their AuthN context.

Supporting consumable resource attributes (e.g., usage time, data transferred, number of access), and additionally collecting accounting data.

4. Using XACML for policy expression in CRP

Different CRP scenarios may require both policies for complex logically organised resources and for user flexible roles/permissions management. Most of such functionality can be supported by XACML core specification [14] and its special profiles for RBAC [15] and for multiple [16] and hierarchical resources [17]. Hierarchical policy management and dynamic rights delegation, that are considered as important functionality in multidomain CRP, can be solved with the XACML v3.0 Administrative policy profile [18].

A XACML policy is defined for the so-called target triad "Subject-Resource-Action" (S-R-A) which can also be completed with the Environment (S-R-A-E) component to add additional context to instant policy evaluation. The Envrionment element can be a place where the proposed in this paper AuthZ ticket can be placed to provide information about AuthZ session context. The XACML policy can also specify actions that must be taken on positive or negative PDP decisions in the form of an optional Obligation element. This functionality is important for potential integration of the AuthZ system with logging or auditing facilities.

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.

Any of S-R-A-E elements allow for extensible "Attribute/AttributeValue" definition to support different attributes semantics and data types. Additionally, XACML allows for referencing internal and external XML documents elements by means of XPath functionality [19].

XACML policy format provides few mechanisms to add and handle domain related context during the policy selection and request evaluation. First of all, this is the policy identification that is done based on the Target comprising of the Resource, Action, Subject, and optionally Environment elements. Next, attributes semantics and metadata can be namespace aware and used for attributes resolution during the request processing.

The XACML RBAC profile [15] 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 a specific action. 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 [17] 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".

Two mechanisms can be used to bind the XACML policy to the Resource: Target elements that can contain any of S-R-A-E attributes and policy identification attribute IDRef.

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 or other logical organisation. When the sequence is important it can be achieved with the ordered rules and policies combination algorithms defined for the Policy Set or Policy [14].

Such specific usecase as multidomain OLPP requires that resource reservation policy in each successive domain may depend on the reservation/authorisation decision in previous domain and in its own turn may create a context/environment for the next domain. This interdomain information can be communicated using AuthZ or reservation ticket that in more extended use can accumulate multiple domains authorisation decisions or resource reservations.

XACMLv3.0 administrative policy profile [18] 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 relevant to the hierarchical resources and user roles management in CRP and currently being investigated.

XACMLv3.0 policy profile allows indicating if the policy is issued by the trusted PolicyIssuer for the particular domain. In this case the PDP will rely on already assigned or default PAP and established trust relations, otherwise when other entity is declared as a PolicyIssuer, the PDP should initiate checking administrative policy and delegation chain what is a suggested functionality of the PIP module.

Examples of XACML policies for different resource models and requests can be found at the AAAuthreach project page [20].

5. Adding security context management to major Authorisation frameworks

To provide described above functionality for domain based security context handling and extended AuthZ session management, a number of features should be added to existing AuthZ frameworks such as Globus Toolkit 4.0 AuthZ Framework (GT4-AuthZ) [21], gLite Java Authorisation Framework (gJAF) [22], or Acegi Security [23]. This functionality is currently being developed as pluggable GAAAPI modules of the GAAA-AuthZ Toolkit that can be called in a standard way from either PEP or PDP [1, 2, 24]. In such a way they can be added as external plugins to other AuthZ frameworks, first of all GT4-AuthZ and gJAF.

GT4 Authorization Frameworks (GT4-AuthZ) is a component of the widely used Grid middleware that provides general and specific functionality to control access to Grid applications using XACML, Grid ACLs, gridmap file, identity or host credentials, calling out to external AuthZ service via OGSA AuthZ PortType.

gLite Java Authorisation Framework (gJAF) is a component of the gLite security middleware. It inherits compatibility with the early versions of the GT4-AuthZ that should ensure their future interoperability and common use of possible application specific modules. Both the GT4-AuthZ and gJAF services can be called from the SOAP based Grid services by configuring the interceptor module which operates in this case as a virtual PEP module.

Acegi Security is the industry recognised security solution with a particular emphasis on applications using Spring framework for J2EE (http://www.springframework.org/). It provides channel security, reach authentication and Single Sign-On (SSO) functionality, and also domain object authorization using ACL. Similar to GT4-AuthZ and gJAF, Acegi security services can be called from the main service using service specific filters.

Similarity in interaction with the main services and applications provides a good basis for developing common modules/library to support dynamic and resource/application domain related context.

Figure 4 shows the GAAA-RBAC structure that contains the following functional components provided as a GAAAPI package to support all the necessary security context processing and communication between a PEP and a PDP:

- A Context Handler (CtxHandler) that calls to a namespace resolver (NS Resolver) and attribute resolver (AttrResolver), which in its own can call to external Credential Validation Service (CVS) or Attribute Authority Service (AAS) to validate presented attributes or obtain new ones.
- A Policy Information Point (PIP) that provides resolution and call-outs to related authoritative Policy Authority Points (PAP);
- Triage and Cache jointly used to support AuthZ session that uses AuthzTicket as session credentials. Triage provides an initial evaluation of the request against assertions contained in the AuthzTicket.
- A Ticket Authority generates and validates AuthZ tickets or tokens on the requests from PEP or PDP. To support AuthZ session tickets are cached by PEP/PDP or by TickAuth itself.

Additionally, the GAAAPI provides an advanced configuration management capability to support dynamic security context changes (including policies, roles and security associations). In particular, when the PEP function is invoked, during AuthZ request processing, it is dynamically configured with context aware modules NSResolver, Triage, TickAuth, and TrustDMngr. Such functionality can allow easy AuthZ services integration with the multi-stage provisioning workflow.

An AzTicket is generated as the result of a positive PDP decision. It contains the decision and all necessary information to identify the requested service. When presented to the PEP, its validity can be verified and in the case of a positive result, access will be granted without requesting a new PDP decision. Such a specific functionality is provided in the GAAAPI package with the Triage module.

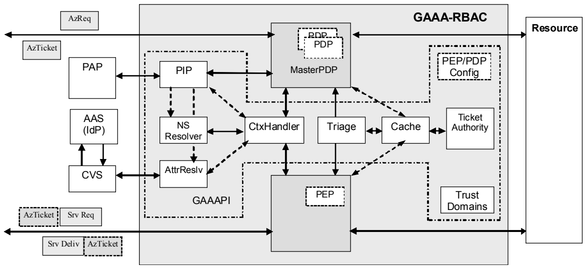

Figure 4. GAAAPI functional components supporting dynamic security context handling

The current GAAAPI implementation supports both SAML-based and proprietary XML-based AzTicket formats (see section 3 for details on the ticket format).

The AuthZ ticket and token handling functionality allows for performance optimisation and supports authorization session management.

Further GAAAPI development includes extended AuthZ ticket format (both proprietary and SAML-based) to support multidomain provisioning scenarios and hierarchical resource and policy administration. Additional features include delegation and extended session context.

6. Conclusion and Summary

The results presented in this paper are the part of the ongoing research and development of the generic AAA Authorization framework and its application to user-controlled service provisioning and collaborative resource sharing. This work is being conducted by the System and Network Engineering (SNE) Group in cooperation with other project/research partners in the framework of different EU and Dutch nationally-funded projects including EGEE, Phosphorus[4], NextGRID, and GigaPort Research on Network. All of these projects deal with the development, deployment or use of Grid technologies and middleware infrastructure platforms whilst also providing a broad scope of different use cases for the GAAA AuthZ Framework development.

The use cases discussed in the paper allowed us to identify the major required functionality to support dynamic security context. The paper identifies basic resource provisioning models and specifies major requirements to AuthZ service infrastructure to support these models.

In the course of practical implementation, we investigate the use of two popular standards SAML and XACML for complex authorisation scenarios in dynamic resource provisioning across multiple administrative and security domains. The paper describes proposed XML based AzTicket format that is designed to support complex AuthZ scenarios and communicate extended AuthZ session context. The paper provides an example of the proprietary ticket format and suggests its mapping to to SAML format. Described AzTicket format is implemented in the GAAAPI package. Further development will extend AzTicket functionality to support dynamic interdomain trust management required for secure CRP and to allow for mutual AuthZ between the requestor and the resource or service.

The paper provides practical analysis what functionality is available in the XACML specification suite for expressing access control policies for complex distributed resources with different logical organisations (multiple, multiple constrained, and hierarchical) and different user access rules that also may require domain based hierarchical user roles and permissions management..

The implementation suggestions are given for how required context handling functionality can be added to popular AuthZ frameworks such as GT4-AuthZ and gLite AuthZ frameworks. Proposed extension modules are being developed as a GAAAPI package of the GAAA-AuthZ toolkit.

The authors believe that the proposed access control architecture for 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.

7. References

[1] Laat de, C., G. Gross, L. Gommans, J. Vollbrecht, D. Spence, "Generic AAA Architecture," Experimental RFC 2903, Internet Engineering Task Force, August 2000. ftp://ftp.isi.edu/in-notes/rfc2903.txt

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

[3] 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

[4] Gommans, L. et al, "Applications Drive Secure Lightpath Creation across Heterogeneous Domains", Special Issue "*IEEE Communications Magazine, Feature topic Optical*

[4] http://www.ist-phosphorus.eu/

Control Planes for Grid Networks: Opportunities, Challenges and the Vision", March 2006.

[5] Demchenko, Y., L. Gommans, C. de Laat, B. Oudenaarde, A. Tokmakoff, R. van Buuren, "Policy Based Access Control in Dynamic Grid-based Collaborative Environment," in *Proc. The 2006 International Symposium on Collaborative Technologies and Systems*, Las Vegas, NV, USA, May 14-18, 2006. IEEE Computer Society, ISBN: 0-9785699-0-3, pp. 64-73.

[6] Demchenko, Y., L. Gommans, B. van Oudenaarde, "Filling the Gap with GAAA-P: Gap Analysis of Authorization technologies and solutions for Optical Light Path Provisioning", Gigaport-NG RoN Technical report. [Online]. Available: http://staff.science.uva.nl/~demch/analytic/airg-gp6-ron-gap-aaa-12.pdf

[7] 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

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

[9] Demchenko, Y., Leon Gommans, Cees de Laat, Rene van Buuren, "Domain Based Access Control Model for Distributed Collaborative Applications", in Proc. *The 2nd IEEE International Conference on e-Science and Grid Computing*, December 4-6, 2006, Amsterdam.

[10] Credential Validation Service Requirements. Version 1.1. [Online]. Available: https://forge.gridforum.org/sf/docman/do/downloadDocument/projects.ogsa-authz/docman.root.authz_service/doc13948

[11] Web Services Trust Language (WS-Trust). [Online]. Available: ftp://www6.software.ibm.com/software/developer/library/ws-trust.pdf

[12] Gommans, L., B. van Oudenaarde, A. Wan, C.T.A.M. de Laat, R. Meijer, F. Travostino and I. Monga, "Token Based Networking: Experiment NL101", iGrid2005 special issue, Future Generation Computer Systems, volume 22 issue 8, pp. 1025-1031 (2006).

[13] "Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0", OASIS Standard, 15 March 2005. [Online]. Available: http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf

[14] "eXtensible Access Control Markup Language (XACML) Version 2.0", OASIS Standard, 1 February 2005. [Online]. Available: http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf

[15] "Core and hierarchical role based access control (RBAC) profile of XACML v2.0", OASIS Standard, 1 February 2005. [Online]. Available: http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-rbac-profile1-spec-os.pdf

[16] "Multiple resource profile of XACML 2.0", OASIS Standard, 1 February 2005, available from http://docs.oasis-open.org/xacml/access_control-xacml-2.0-mult_profile-spec-cd-01.pdf

[17] "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

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

[19] XPathXML Path Language (XPath)- [Online]. Available: http://www.w3.org/TR/xpath

[20] AAAuthreach Project Information Page [Online]. Available: http://staff.science.uva.nl/~demch/projects/aaauthreach/index.html

[21] GT 4.0: Security: Authorization Framework. [Online]. Available: http://www.globus.org/toolkit/docs/4.0/security/authzframe/

[22] gLite Security Subsystem. [Online]. Available: http://glite.web.cern.ch/glite/security/

[23] Acegi Security. [Online]. Available: http://www.acegisecurity.org/

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