Authorisation Session Management in On-Demand Resource Provisioning in Collaborative Applications

Yuri Demchenko, Cees de Laat, Thierry Denys, Christian Toinard
University of Amsterdam, ENSI de Bourges
{demch, delaat}@science.uva.nl
{thierry.denys, christian.toinard}@ensi-bourges.fr

ABSTRACT

Effective use of the resources in modern collaborative environment suggests their sharing between collaborating organisations and user groups and on-demand provisioning for the specific tasks and projects that may involve distributed resources and users from different administrative and security domains. The proposed in earlier authors' work the general Complex Resource Provisioning (CRP) model provides a basis for developing the Authorisation (AuthZ) infrastructure for on-demand multidomain resource provisioning. This paper discusses such important issues as managing authorisation session and security context in multidomain CRP and security mechanisms used for this. The use of AuthZ tokens for AuthZ session management in multidomain network resource provisioning is considered as a particular case for the general CRP. It provides information about practical implementation of AuthZ session management functionality in the GAAA Toolkit library being developed in the framework of the Phosphorus project.

KEYWORDS: Generic Authentication, Authorization and Accounting, Complex Resource Provisioning, AAA Authorisation Framework, Authorisation session, Authorisation Ticket, Pilot Token, Token Validation Service.

1. INTRODUCTION

The modern collaborative environment may include multiple resources (such as data repositories, scientific instruments, analytical and image scanning equipment, object and process controlling systems, visualisation systems, user terminals, and data processing centers) and connecting them networking infrastructure. To achieve effective management and use of such collaborative environment, all the resources should be provisioned on-demand and support dynamic security associations that may include both resources and users. Existing Grid technologies can provide a framework for creating project oriented collaborative environments in a form of the Virtual Organisations (VO) which however have rather static character and doesn't solve a problem of more dynamic on-demand resource provisioning [1].

The general Complex Resource Provisioning (CRP) was proposed in earlier authors' works [2] to address on-demand resources provisioning issues. It was successfully used for developing AuthZ infrastructure for Grid and network resources provisioning in the framework of the Phosphorus project [3]. The proposed infrastructure supports all stages of the CRP process/lifecycle in a consistent way that first of all requires flexible provisioning and user access control sessions management.

In this paper we summarise our recent developments and discuss the proposed mechanisms such as AuthZ tickets and tokens that allow supporting authorisation session and security context management in multidomain CRP.

The paper is organized as follows. Section 2 provides a state of the art about using the policy languages such as eXtensible Access Control Markup Language (XACML) for expressing authorisation policies in Grid based applications and Security Assertion Markup Language (SAML) for expressing authentication and authorisation assertions. It shows that multi-domain authorisation issues including AuthZ session management are poorly addressed. Section 3 updates on the general CRP model and summarises general requirements for AuthZ session management capabilities to support the whole CRP process. Section 4 analyses different types of AuthZ sessions typically present in CRP systems: provisioning session and access session. This section introduces AuthZ tokens and tickets used as session credentials and access credentials. Section 5 provides information about the general AuthZ token format and defines different access and pilot types. Section 6 explains in details the process of the AuthZ context communication during multidomain network resource reservation using pilot tokens. Section 7 provides information about the GAAA Toolkit implementation to support the suggested functionality for AuthZ tickets and tokens handling during the resource reservation and access, including tokens generation, validation and relaying.

2. STATE OF THE ART

Several works discuss issues related to using such security languages as XACML and SAML correspondingly for expressing authorisation policies or expressing authentication and authorisation assertions that can be used for Single-Sign-On or as access control session credentials. The majority of the works consider authentication across a single organisation or inside single trust domain or federation. Other works consider how to enforce protection models using XACML, how to manage the update of the policy or how to combine several policies. Finally, recent state of art studies shows that authorization for distributed systems and applications, which examples are Grid and collaborative systems, is really poorly addressed and does not consider multiple organizations sharing a large range of Grid resources. Below we briefly refer to some particular works and papers.

The paper [4] discusses security and privacy issues with authentication of individuals in Web Services using a ring signature. It uses SAML but does not consider authorization. The paper [5] analyses applicability of XACML to enforce the classical access control models such as Bell-LaPadula; Biba or Chinese Wall. They do not consider a larger range of protection models and do not address multi-domain issues. Other works address different issues related to expressing and managing XACML based policies, in particular, [6] describes an RBAC implementation using XACML; [7] discusses how to update XACML policies; and [8] analyses policy rules conflict resolution when merging XACML policies. However, all mentioned above works do not address specific access control issues in distributed Grid systems that involve multiple domains. The authors in [9] provide a state of the art study of security for the Grid infrastructures. They found several works related with authorization decision-making for multiple users and resources but none of them includes the notion of domain. Finally, Single Sign On (SSO) across multiple domains can be found in the literature but in most cases the proposed solutions are oriented on web-based applications and using web browser as a user client, they use browser cookies for access session management but don't deal with authorization [10, 11].

3. CRP OPERATIONAL MODELS AND MULTIDOMAIN AUTHORISATION ARCHITECTURE

The two major use cases for the general CRP are on-demand network resource provisioning (NRP) [12] and Grid-based Collaborative Environments (GCE) [13]. Although different in current implementations, they can be abstracted to the same CRP operational model when considering their implementation with the Grid or Web Services. This abstraction is considered as an important step to provide a common basis to define a common access control infrastructure for dedicated optical networks and Grid brokered networks.

The typical on-demand resource provisioning process includes four major stages, as follows:

(1) resource reservation
(2) deployment (or activation)
(3) resource access/consumption, and additionally
(4) resource de-commissioning after it was used.

In its own turn, the reservation stage (1) typically includes three basic steps:

- resource lookup,
- complex resource composition (including alternatives), and
- reservation of individual resources including authorisation of the reservation request.

The reservation stage may require the execution of complex procedures that may also request individual resources authorisation. This process can be controlled by an advance reservation system [14] or a meta-scheduling system [15] and is driven by the provisioning workflow and related authorization (AuthZ) policy. At the deployment stage, the reserved resources are bound to a reservation ID, which we refer as the Global Reservation Identifier (GRI). The decommissioning stage is considered as an important stage in the whole resource provisioning workflow from the provider point of view and should include such important actions as global provisioning/access session termination and user/process logout, log information sealing, accounting and billing.
The rationale behind defining different CRP workflow stages is that they may require and can use different security models for policy enforcement, trust and security context management, but still may need to use common dynamic security context.

In the discussed CRP model, domains are defined (as associations of entities) by a common policy or a single administration, with common namespaces and semantics, shared trust, etc. In this case, the domain related security context may include:

- static security context [16] such as domain based policy authority reference, trust anchors, all bound by the domain ID and/or domain trust anchor;
- dynamic or session security contexts bound to the GRI and optionally to a Local Reservation ID (LRI).

In general, 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 contexts. In the remainder of the document, we will refer to the typical use case of the network domains that are connected as chain (sequentially) providing connectivity between a user and an application.

Figure 1 illustrates major interacting components in the multi-domain on-demand NRP:

- A User/Requestor (represented by User client).
- A Destination end service or application.
- Multiple Network Elements (NE) (related to the Network plane).
- Network Resource Provisioning Systems (NRPS) acting as a Domain Controller (DC).
- Inter-Domain Controller (IDC) and AAA service controlling access to the domain- related resources.
- Policy Enforcement Point (PEP), Policy Decision Point (PDP), and Policy Authority Point (PAP) as major functional components of the AuthZ infrastructure.
- Token Validation Services (TVS) that handles AuthZ tokens used as AuthZ session credentials during reservation and access stages.

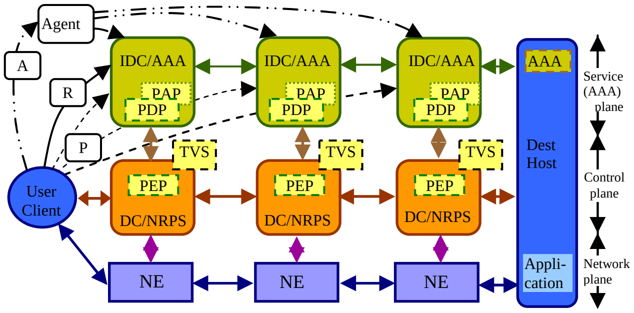

Figure 1. Components Involved in Multidomain Network Resource Provisioning and Basic Sequences (Agent (A), Chain (C), and Polling (P))

The above described CRP model can be generalized for another typical CRP use case of the GCE if we consider collaborative applications as separate resource domains that can be logically organised into different structures and described with the similar set of attributes as traditional network domains. Following [2, 17], in the remainder of this document we will refer to the AuthZ infrastructure described above as GAAA-NRP.

Managing AuthZ session context during the reservation stage is essential to ensure consistent resource protection and effective decision making. At the service access/consumption stage the reserved resource may be simply identified by the assigned GRI created as a result of the successful reservation request authorisation.

It is an important convention for the consistent CRP operation that GRI is created at the beginning and sent to all polled/requested domains when running (advance) reservation process. Then in case of a confirmed reservation, the DC/NRPS will store the GRI and bind it to the committed resources. In addition, a domain can also associate internally the GRI with the LRI.

4. AUTHZ SESSION MANAGEMENT WITH AUTHZ TICKETS AND TOKENS

There are two types of sessions in the proposed CRP model that require a security context management: reservation session, and the reserved resource access session. Although the provisioning session may require wider security context support, both of them are based on the (positive) AuthZ decision, may have a similar AuthZ context and will require a similar functionality when considering distributed multi-domain scenarios. Figure 2 illustrates relationship between provisioning AuthZ session that spans all CRP stages and resource or application access session which starts at the access stage. It is essential that both of these sessions share the same GRI created at the beginning when the resource request has been authorised.

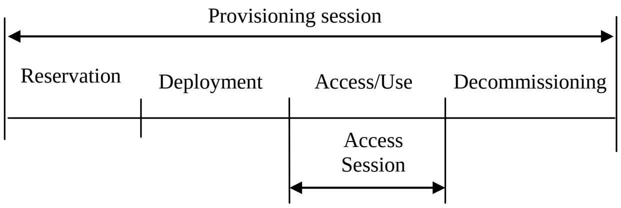

Figure 2. CRP Stages and Session Types.

We distinguish two types of the AuthZ session credentials:

- AuthZ tickets that should allow for expressing and communicating the full AuthZ session context and in this way could be used as access credentials
- AuthZ tokens that should provide more flexible functionality for managing the AuthZ session and the whole provisioning process.

We will discuss in detail the use of AuthZ tokens during multidomain network resources reservation and will refer to the detailed AuthZ ticket format and functionality description given in [16].

5. AUTHZ TOKEN DATAMODEL AND TOKEN TYPES

In the proposed AuthZ architecture the tokens are used for access control and signalling at different CRP stages and considered as a flexible and powerful mechanism for communicating and signalling security context between domains.

The token is defined as an abstract reference to the reservation or the AuthZ session context in domains using an abstract shared token meaning/context that is referenced by the token attributes. Tokens can be used for both access control when accessing the reserved resources and for signalling during reservation and deployment stages. Correspondingly we distinguish the two major types of token in the GAAA-NRP architecture: access tokens and pilot tokens. Access tokens are used in rather traditional manner and described in details in [12]. Pilot tokens functionality and format were proposed and defined as the current development result of the AuthZ infrastructure as an integral component of the NRP infrastructure. Figure 3 illustrates the common data model of both access tokens and pilot tokens. Although the tokens share a common data-model, they are different in the operational model and in the way they are generated and processed. When processed by AuthZ service components they can be distinguished by the presence or value of the token type attribute which is optional for access token and mandatory for pilot token.

Access tokens used in GAAA-NRP have a simple format and contain three mandatory elements: the SessionId attribute that holds the GRI, the TokenId attribute that holds a unique token ID attribute and is used for token identification and authentication, and the TokenValue element, and two optional elements: the Condition element that may contain two time validity attributes notBefore and notOnOrAfter, and the Decision element that holds two attributes ResourceId and Result, and optional element Obligations that may hold policy obligations returned by the PDP.

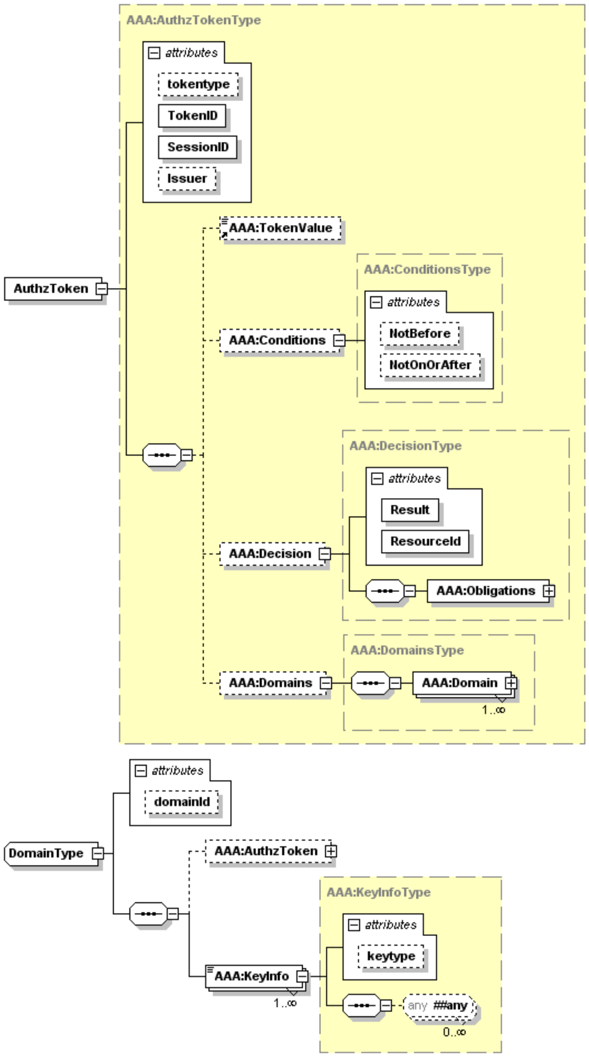

Figure 3. Authorisation Token Data Model.

The GAAA-CRP architecture defines four types of pilot tokens that have different profiles of the common data model and different processing/handling procedure:

Type1 – this pilot token type is used just as a container for communicating the GRI during the reservation stage. It contains the mandatory SessionId attribute and an optional Condition element (it does not contain a TokenValue element).

Type2 – this pilot token type is the origin/requestor authenticating token. Its TokenValue element contains a value that can be used as the authentication value for the token origin. The token value is calculated on the GRI by applying e.g. an HMAC function to the GRI together with the requestor symmetric secret or private key.

Type3 – this pilot token type extends the Type2 with an element, including multiple Domains, that allows to collect the Security Contexts (SecCtx) related to those domains when passing multiple domains during the reservation process. Such information includes the previous token and the domain's trust anchor or public key.

Type4 – this pilot token type is used at the deployment stage and can communicate, between the SecCtx of the crossed domains, about all participating in the provisioned lightpath or network infrastructure resources. This token type can be used for programming/setting up a TVS infrastructure for consistent access control tokens processing at the resource access stage. When used together with an AuthzTicket the ticket and token identification elements TokenId/TicketId, SessionId, and Issuer can be shared.

6. HANDLING INTERDOMAIN SECURITY CONTEXT DURING RESOURCE PROVISIONING

6.1. Interdomain AuthZ/Security Context Communication with the Pilot Tokens

During a multidomain network resource reservation, the pilot token goes from the requestor to the resource throw several domains. In order to collect the previous domains AuthZ and security context, the content of the token will change. The first token is created as a result of positive authorisation and a confirmed reservation in the first domain. Typically it is the pilot token type 2 (PTT2). When the second domain confirms reservation, a new pilot token type 3 (PTT3) is created that now includes a DomainsContext element that contains a Domain element as a child holding context information from the previous domain. The process continues in the next domain and a new Domain element is simply added. Figure 4 provides an example of the pilot token change during a multidomain network resource reservation. The token issued in the current domain contains the local DomainId and the DomainsContext element holds information from all previous domains including related tokens which are chronologically ordered.

The next Figure 5 shows an XML pilot token from the example above with two domains: "viola" and "uva".This figure shows the different information related to "viola" and "uva", some conditions (notBefore and NotOnOrAfter) and different elements as explained with Figure 3.

This token also refer to the intermediate type 3 token from Figure 4: two domains have been passed and context information from the first domain is collected in the DomainsContext element.

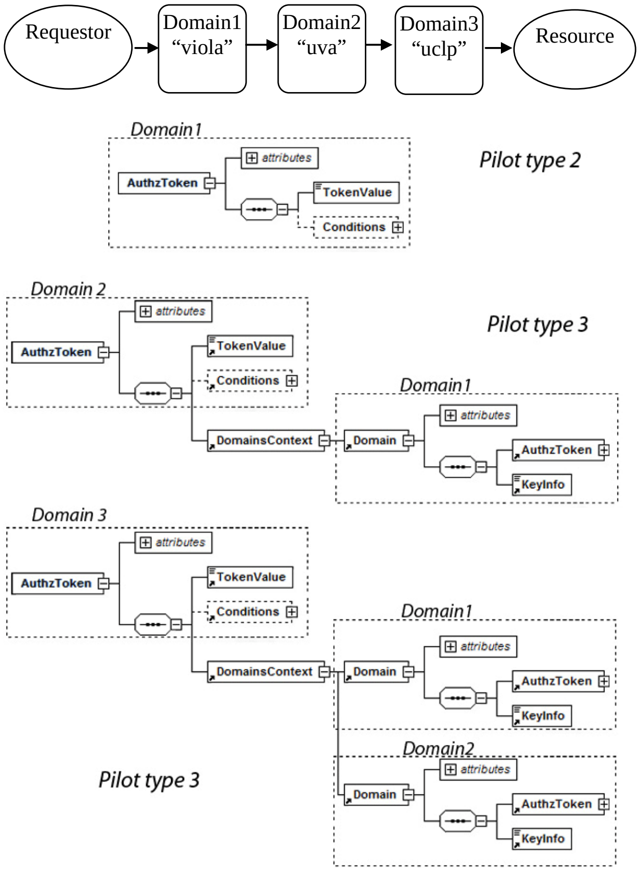

Figure 4. Structure of the Pilot Token type 3 After Passing Three Domains

```
<AAA:AuthzToken
        xmlns:AAA="http://www.aaauthreach.org/ns/AAA"
        Issuer=http://testbed.ist-phosphorus.eu/uva/aaa/TVS/token-pilot
        SessionId="740b241e711ece3b128c97f990c282adcbf476bb"
        TokenId="dc58b505f9690692f7a6312912d0fb4c"
        type="pilot-type3">
 <AAA:TokenValue>190a3c1554a500e912ea75a367c822c09eceaa2f
 </AAA:TokenValue>
 <AAA:Conditions
        NotBefore="2009-01-30T08:57:40.462Z"
        NotOnOrAfter="2009-01-30T09:21:40.462Z"/>
 <AAA:DomainsContext>
  <AAA:Domain domainId="http://testbed.ist-phosphorus.eu/viola">
   <AAA:AuthzToken
        Issuer="http://testbed.ist-phosphorus.eu/viola/aaa/TVS/token-pilot"
        SessionId="2515ab7803a86397f3d60c670d199010aa96cb51"
        TokenId="c44a2f5f70346fdc2a2244fecbcdd244">
    <AAA:TokenValue>dee1c29719b9098b361cab4cfcd086700ca2f414
    </AAA:TokenValue>
    <AAA:Conditions
        NotBefore="2009-01-30T07:57:35.227Z"
        NotOnOrAfter="2009-01-31T07:57:35.227Z"/>
   </AAA:AuthzToken>
   <AAA:KeyInfo>
        http://testbed.ist-phosphorus.eu/viola/_public_key_
   </AAA:KeyInfo>
  </AAA:Domain>
 </AAA:DomainsContext>
</AAA:AuthzToken>
```

Figure 5. XML Pilot Token with Two Domains: "viola" and "uva".

6.2. Storing Domain Session Contexts

Tokens' security contexts can be saved either in a TVS table or in XML database in each domain. In the first case, security context is saved in a HashMap table for each domainId using GRI as a key and a vector of attributes as a value. Attributes are saved in following order: notBefore, notOnOrAfter, actionId, subjectId, subjectRole, subjectContext, resourceId, resourceSource, resourceTarget, keyinfo.This structure allows querying security contexts by domainId and GRI.

In the second case, different XML database schemas and different query methods can be used to store security context the XML database.There is one database for each domain, and security contexts are ordered by GRI. With this data structure, security contexts can be queried using domainId and GRI similar to TVStable.

Using XML database allows also storing domain related policy files. Policy files can be queried using PolicyId and ResourceId using XQuery language via internal database adaptor or externally using Web Services.

7. AUTHZ SESSION MANAGEMENT SUPPORT IN GAAA-TK

7.1. GAAA Toolkit Java Library

All required functionality to support GAAA-NRP authorisation infrastructure is currently being implemented in the GAAA Toolkit (GAAA-TK) pluggable Java library in the framework of the Phosphorus project [2]. The library can provide also a basis for building AAA/AuthZ server that can act as Domain Central AuthZ Service (DCAS) or operates as a part of the Inter-Domain Controller (IDC) to enable complex policy driven resource reservation scenarios. The library allows for AuthZ request evaluation with the local XACML based PDP or calling out to the external DCAS using the SAML-XACML protocol. One of the key functional components to support token based AuthZ session management is the Token Validation Service (TVS). It is implemented as a part of the general GAAA-TK library but can also be used separately and integrated into other AuthZ frameworks.

7.2. Token Generation and Validation with TVS

Basic TVS functionality allows checking if a service/resource requesting subject or other entity, that posses current token, has permission to access/use a resource based on advance reservation to which this token refers. During its operation the TVS checks if a presented token has reference to a previously reserved resource and a request information conforms to a reservation condition.

When using pilot tokens for signalling during interdomain resource reservation, TVS can combine token validation from the previous domain and generation of a new token with local domain attributes and credentials. This scenario is supported by a special method “Validate&Relay”. This method requires checking incoming pilot token’s authenticity, which should be a part of the validation process.

In a basic scenario, the TVS operates locally and checks a local reservation table directly or indirectly using a reservation ID (e.g. in a form of GRI). It is also suggested that in a multi-domain scenario each domain may maintain its Local Reservation ID (LRI) and provides its mapping to the GRI. In more advanced scenario the TVS should allow creation of a TVS infrastructure to support tokens and token related keys distribution to support dynamic resource, users or providers federations.

The token generation and handling model can use both shared secret cryptography and public key cryptography and uses HMAC-SHA1 algorithm for calculating token value [18]. Current implementation uses shared secret, which for the sake of simplicity of testbed implementation is provided as a part of the TVS/GAAA-TK library distribution. The TokenKey is generated in the following way:

`TokenKey = HMAC(GRI, tb_secret)`

where
GRI – global reservation identifier,
tb_secret – shared secret.

A token value is computed in a similar way but using TokenKey as a HMAC secret. However it is different for the access token and pilot token (of types 2 and 3). For purpose of authenticating token origin, the pilot token value is calculated of concatenated DomainId, GRI, and TokenId. This approach provides a simple protection mechanism against the pilot token duplication in the framework of the same reservation/authorisation session.

The following expressions are used to calculate the TokenValue for the access token and pilot token:

`TokenValue = HMAC(GRI, TokenKey)` `– access token`

`TokenValue = HMAC(concat(DomainId, GRI, TokenId), TokenKey)` – pilot token type 2 and 3

This algorithm allows for chaining token generation and validation process, in particular when using GRI and LRI in some domains:

```
“GRI-TokenKey-TokenValue =>
       LRI-l_TokenKey-l_Token”
```

The key management model is not discussed at this stage of the research. The token handling model relies on the shared secret that is installed at all participating NRPS nodes. It is being investigated that current model can be replaced with the IBC (Identity Based Cryptography) [19, 20] that will allow to replace shared secret token handling model that has known manageability problems.

7.3. PEP and TVS Methods to Support AuthZ Session Management

The GAAA TK library provides few PEP and TVS methods that support extended AuthZ session management and provide necessary AuthZ tokens and tickets handling functionality (refer to the GAAA-TK release documentation [21] for the complete API description). The two basic PEP methods provide simple AuthZ session management and allow using AuthZ tickets or access tokens as session credentials, however differ in either requiring complete request information or using AuthZ ticket or token as only access credentials. Both of these methods can either return a valid AuthZ ticket or token, or “Deny” value.

The two other methods support more flexible session based AuthZ scenarios that allow also simple intra-domain and inter-domain delegation. The first extended method allows for flexible session based access control and delegation using AuthzToken as a session credential. It supports the following simple delegation scenarios where the session permissions obtained by a privilege user (e.g. researcher, principal investigator) can be delegated to other user depending on session-delegation modes.

The delegation type attribute defines the following session delegation scopes:

0 - strict session based delegation (only authorised roles for only authorised actions - PDP/policy based evaluation)

1 - full session delegation (all actions for all role, i.e. just checking validity of token)

2 - allowed actions for all legitimate roles

3 - controlled delegation (require extended AuthzTicket format; delegation defined by AuthzTicket context).

The second extended method supports either local domain session based access control or can be used for “chained” AuthZ decisions request like in case of multidomain network path creation. This method calls the TVS method validateAndRelayPilotToken to re-generate a token. Most of the PEP methods use two major TVS methods that either simply validate the provided AuthZ token or accept the token from the previous domain and generate a token for the local domain that can be next send to the next domain. TVS methods are called from the PEP interface, however they can be also called directly from the TVS interface.

8. SUMMARY AND FUTURE RESEARCH

This paper presented the results of the ongoing research and development of the generic AAA AuthZ architecture in application to two inter-related research domains: on-demand optical network resource provisioning and Grid based Collaborative Environment that can use the same Complex Resource Provisioning model.

The proposed AuthZ infrastructure will allow easy integration with the Grid middleware and applications what is ensured by using common Grid/network resource provisioning model that defines specific operational security models for three major stages in the general resource provisioning: reservation, deployment or activation, and access or use. The current implementation of the GAAA-NRP authorisation infrastructure that uses the GAAA-TK library in the Phosphorus project multidomain networking testbed provides a good basis for further research on improving efficiency of the provisioning and authorisation sessions management and extending functionality of the session management mechanisms such as discussed in this paper AuthZ tokens and tickets. The future research will also investigate different ways to ensure integrity and validity when using tickets and tokens for access control and for signaling based on the analysis of typical vulnerabilities and threats in multidomain scenarios. Moreover, specific Web Services can be developed to ease the management of the security policies. Finally, we will study how to support advanced Security Properties such as defined in [22]. Thus, the solution will prevent from the information flows that would violate the requested integrity, confidentiality and availability properties.

The authors believe that the proposed solutions for AuthZ session management in on-demand resource provisioning will provide a good basis for further discussion among CTS and networking specialists.

REFERENCES

[1] I. Foster, H. Kishimoto, A. Savva, D. Berry, A. Grimshaw, B. Horn, F. Maciel, F. Siebenlist, R. Subramaniam, J. Treadwell, J. Von Reich. GFD.80 “The Open Grid Services Architecture, Version 1.5,” Open Grid Forum, Sept. 5, 2006.

[2] Y. Demchenko, A. Wan, M. Cristea, C. de Laat, “Authorisation Infrastructure for On-Demand Network Resource Provisioning”. In *Proceedings of the 9th IEEE/ACM International Conference on Grid Computing (Grid 2008)*, Tsukuba, Japan, Sept. 29 - Oct. 1, 2008. IEEE Catalog Number CFP08GRI-CDR, ISBN 978-1-4244-2579-2.

[3] Phosphorus Project. [Online]. Available: http://www.ist-phosphorus.eu/

[4] V. C. Hu, Evan Martin, JeeHyun Hwang, Tao Xie „Conformance Checking of Access Control Policies Specified in XACML“. In *Proceedings of the 31st Annual International Computer Software and Applications Conference (COMPSAC2007)*, Volume 2, Issue , 24-27 July 2007.

[5] Y. Yang, J. Yang “Towards Unconditional Anonymity: Privacy Enforcement Model in Web Services”. In *Proceedings Congress on Services Part II, 2008. SERVICES-2*, 23-26 Sept. 2008, Beijing.

[6] L. Seitz, E. Rissanen, T. Sandholm, B. Sadighi Firozabadi and O. Mulmo, “Policy Administration Control and Delegation using XACML and Delegent”. In *Proceedings of the 6th IEEE/ACM International Workshop on Grid Computing Workshop 2005*, 13-14 Nov. 2005.

[7] K.I. Kim, H.J. Ko, W.G. Choi, E.J. Lee, U.M. Kim, “A Collaborative Access Control based on XACML in Pervasive Environments”. In *Proceedings the IEEE, International Conference on Convergence and Hybrid Information Technology*, 28-30 Aug. 2008.

[8] J. Alqatawna, E. Rissanen, B. Sadighi, “Overriding of Access Control in XACML“. In *Proceedings of the 8th IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY'07)*, 2007.

[9] A. Chakrabarti, A. Damodaran, and S. Sengupta, “Grid Computing Security”. *IEEE Security & Privacy*, IEEE Computer Society 2007.

[10] L. Wei, S. Sengupta, “Session management for web-based healthcare applications”. In *Proceedings AMIA Symposium*, 1999

[11] V. Samar, “Single Sign-On Using Cookies for Web Applications”. In *Proceedings of the 8th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, 1999.

[12] L. Gommans, L. Xu, Y. Demchenko, A. Wan, M. Cristea, R. Meijer, C. de Laat, “Multi-Domain Lightpath Authorization using Tokens”, *Future Generations Computer Systems*, Vol 25, issue 2, February 2009, pages 153-160

[13] Y. Demchenko, L. Gommans, C. de Laat, Rene van Buuren, “Domain Based Access Control Model for Distributed Collaborative Applications”. *In Proceedings of the 2nd IEEE International Conference on e-Science and Grid Computing*, December 4-6, 2006, Amsterdam.

[14] “Support for advance reservations in scheduling”, Phosphorus Project Deliverable D5.4, September 2007. [Online] Available: http://www.ist-phosphorus.eu/files/ deliverables/Phosphorus-deliverable-D5.4.pdf

[15] Viola Meta Scheduling Service Project. [Online]. Available: http://packcs-e0.scai.fhg.de/viola-project/

[16] Y. Demchenko, L. Gommans, C. de Laat, F. Wan, O. Mulmo, “Dynamic security context management in Grid-based applications”, *Future Generation Computer Systems (2007)*, doi:10.1016/j.future.2007.07.015

[17] “AAA Architectures for multi-domain optical networking scenario's”, Phosphorus Project Deliverable D4.1. – September 30, 2008. [Online]. Available: http://www.ist-phosphorus.eu/files/ deliverables/Phosphorus-deliverable-D4.1.pdf

[18] A. Menezes, P. van Oorschot, S. Vanstone, “Handbook of Applied Cryptography”. ISBN: 0-8493-8523-7, October 1996.

[19] A. Shamir. “Identity-based cryptosystems and signature schemes In G.R. Blakley and D. Chaum, editors, Advances in Cryptology”. In *Proceedings of CRYPTO'84 on Advances in cryptology*. Springer-Verlag LNCS 196, 1985.

[20] H. Tanaka. “A realization scheme for the identity-based cryptosystem”. In *Proceedings of CRYPTO'87 Advances in Cryptology*. Springer-Verlag LNCS 293, 1988.

[21] “GAAA toolkit pluggable components and XACML policy profile for ONRP”, Phosphorus Project Deliverable D4.3.1. [Online]. Available: http://www.ist-phosphorus.eu/files/ deliverables/Phosphorus-deliverable-D4.3.1.pdf

[22] M. Blanc, J. Briffaut, J.F. Lalande, C. Toinard. “Distributed control enabling consistent mac policies and IDS based on a meta-policy approach”, In *Proceedings of the 7th IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY’06)*, 2006.