Authorisation Infrastructure for On-Demand Network Resource Provisioning

Yuri Demchenko, Alfred Wan, Mihai Cristea, Cees de Laat
University of Amsterdam
{demch, wan, cristea, delaat}@science.uva.nl

Abstract

High performance Grid applications require high speed network infrastructure that should be capable to provide network connectivity service on-demand. This paper presents results of the development of the Authorisation (AuthZ) infrastructure for on-demand multidomain network resource provisioning (NRP). We propose a general Complex Resource Provisioning (CRP) model that can be used as a basis for AuthZ infrastructure development providing a common abstraction for provisioning both network and Grid resources. This model allows common policy expressions, using single user sign-on credentials when requesting and accessing complex Grid-Network resources. The implementation described is based on the generic AAA Authorisation Framework (GAAA-AuthZ) and suggests a number of security mechanisms and components that extends GAAA-AuthZ to achieve consistent policy enforcement and security context management: Token Validation Service (TVS), AuthZ ticket used for AuthZ session management, a special XACML profile for NRP, reference model for policy obligations handling (OHRM). The proposed infrastructure and solutions are being implemented in the framework of the EU project Phosphorus and use authors experiences gained from the major Grid based and Grid oriented projects.

KEYWORDS: Complex Resource Provisioning, Multidomain Network Resource Provisioning, AAA Authorisation Framework, Authorisation session, Token Validation Service, XACML.

1. Introduction

High performance distributed Grid applications that deal with high volume of processing and visualisation data require dedicated high-speed network infrastructure provisioned on-demand. Currently larger Grid projects such as EGEE/LCG and national research networks use their own dedicated network infrastructure that can handle the required data throughput but typically are over-provisioned. Any network upgrade or reconfiguration still requires human interaction to change or negotiate a new Service Level Agreement and involve network engineers to configure the network. Moreover, Grid application developers and users always intended to have control over network characteristics to optimise application performance and network cost.

In Grid applications, Grid middleware allows for dynamic resource allocation and deployment. Recent research and developments to make network resources Grid middleware enabled, like in the Phosphorus project [1], will allow using common tools for combined Grid and Network resources provisioning.

In this paper, we analyse two major use cases on-demand network resource provisioning and Grid-based Collaborative Environments to define the general Complex Resource Provisioning (CRP) model and corresponding requirements for the distributed multidomain AuthZ service.

In general, complex resources and/or services may have different logical organisation and represented as hierarchical structure, ordered or unordered resource collection. CRP operational model should be capable to support different resource organisation and consequently different provisioning and access control models. Most of existing network or Grid resource provisioning frameworks address separately resource scheduling, reservation, and resource or service access and consumption. Security aspects and AuthZ aspects are not addressed in such frameworks.

The proposed CRP model defines three stages: reservation, deployment, access, - that operates different resource management and security models The proposed AuthZ architecture is based on the further development of the generic Authentication, Authorisation, and Accounting (AAA) Authorisation framework (GAAA-AuthZ) [2] that is extended with new security mechanisms and components to support complex AuthZ scenarios in on-demand multidomain network resource provisioning.

GAAA-AuthZ services are designed in such a way that they can be used at all networking layers (dataflow plane, control plane and service plane) and allow easy integration with Grid middleware and application layer security. For this purpose, special mechanisms are proposed to manage inter-layer and inter-domain security context.

The presented research and proposed solutions are specifically oriented for using with the popular Grid middleware such as gLite [3] and Globus Toolkit [4] being developed in the framework of large international projects and consortia such as EGEE Globus Alliance. The authors also have been actively involved into the Grid middleware development and related AuthZ interoperability initiatives such as at OGSA-AuthZ Working Group [5] and joint OSG-EGEE AuthZ interoperability project [6].

The paper is organized as follows. Section 2 describes the proposed general CRP model that separates resource reservation, resource deployment, and resource access or consumption stages. The section summarises common requirements to AuthZ services/infrastructure to support different provisioning and AuthZ scenarios in distributed dynamic environment.

Section 3 describes the proposed AuthZ mechanisms and components to support multidomain network resource provisioning: AuthZ ticket for AuthZ session management, special XACML profile for network resource provisioning, reference model for policy obligations handling (OHRM). Section 4 describes the Token Validation Service (TVS) model and operation which is considered as an important component of the CRP AuthZ architecture to provide flexibility in policy enforcement when accessing the reserved network resources. Section 5 briefly presents our ongoing implementation, and finally section 6 provides a short summary and suggests future developments.

2. CRP operational models and Multidomain Authorisation service architecture

The two major use cases for the general CRP are on-demand network resource provisioning (NRP) [7] and Grid-based Collaborative Environments (GCE) [8]. 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 resources accessed and brokered over network.

The typical on-demand resource provisioning process includes three major stages: (1) resource reservation, (2) deployment (or activation), and (3) the reserved resource access/consumption. In its own turn, the reservation stage includes three basic steps: (a) resource lookup, (b) complex resource composition (including alternatives), and (c) reservation of individual resources. The reservation stage may require the execution of complex procedures that may also request individual resources authorisation. This process can be controlled by the AAA driving policy that should support the whole provisioning workflow and related AuthZ policy [9], or driven by a Meta Scheduling system [10]. At the deployment stage the reserved resources are bound to the reservation ID, which we refer to as the Global Reservation Identifier (GRI).

The rationale behind defining different CRP stages is that they may require and can use different security models for policy enforcement, trust and security context management.

In the discussed CRP model, domains are defined (as associations of entities) by a common policy under single administration, common namespaces and semantics, shared trust, etc. In this case, the 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 security context at the reservation and access stages [11]. 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 context. In the remainder of the paper 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.

The CRP model for the multidomain distributed resource management model requires the following functionality from the GAAA-AuthZ infrastructure:

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

Figure 1 illustrates major interacting components in the multi-domain CRP using multidomain NRP as a major use case:

- 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) (typically related to the Control plane).
- 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 allow efficient authorisation decision enforcement when accessing reserved resources.

The above described CRP model can be generalized for another typical CRP use case of the Virtual Laboratory (VL) workspace provisioning 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.

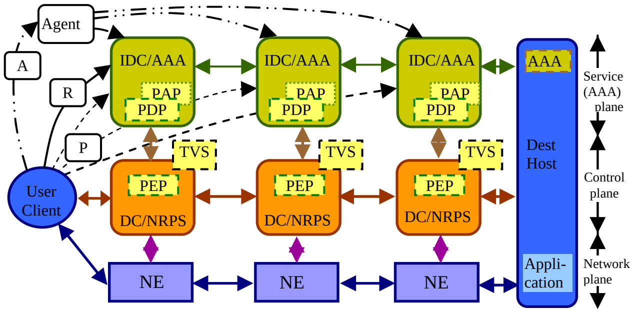

Figure 1. Components involved in multidomain network resource provisioning and basic sequences (agent (A), chain (C), and polling (P))

Figure 1 also illustrates different provisioning models or sequences that can be executed when composing a complex resource:

- **Chain** reservation sequence (also referred to as chaining sequence) when the user contacts only the local network domain/provider providing destination address, and each consecutive domain provides a path to the next domain.
- **Polling** sequence when the user client polls all resource or network domains, builds the path and makes the reservation.
- **Agent** (or tree) sequence when the user delegates network provisioning negotiation to the agent that will take care of all necessary negotiations to provide the 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 NRPS and protected by the AAA service that enforces a resource access control policy. This is achieved by placing a PEP gateway at the NRPS. Depending on the basic GAAA-AuthZ sequence (push, pull or agent) [2], the requestor can send a resource access request to the resource or service (which in our case are represented by NRPS) or an AuthZ decision request to the designated AAA server which in this case will act as a PDP. The PDP identifies the applicable policy or policy set and retrieves them from the PAP, collects the required context information, evaluates the request against the policy, and makes the decision whether to grant access or not.

Depending on the used authorisation and attribute management models, some attributes for the policy evaluation can be either provided in the request or collected by the PDP itself. It is essential in the Grid/Web services based service oriented environment that AuthN credentials or assertions are presented as a security context in the AuthZ decision request and are evaluated before sending request to PDP.

Based on a positive AuthZ decision (in one domain) the AuthZ ticket (AuthzTicket) can be generated by the PDP or PEP and communicated to the next domain where it can be processed as a security context for the policy evaluation in that domain.

In order to get access to the reserved resources the requestor needs to present the reservation credentials that can be in a form of an AuthZ ticket (AuthzTicket) or an AuthZ token (AuthzToken) which will be evaluated by the PEP to grant access to the reserved network elements or the resource. In more complex provisioning scenarios the token or credential validation function may be outsourced to the TVS service. The TVS infrastructure can additionally support an interdomain trust management infrastructure for off-band token and token key distribution between the PEP-NRPS and IDC/AAA services that typically takes place at the deployment stage when access credentials or tokens are bound to the confirmed GRI by means of shared or dynamically created interdomain trust infrastructure. Token and token key generation and validation model can use either shared secret, PKI based trust model, or recently researched by authors the Identity Based Cryptography (IBC) [12, 13].

The TVS as a special GAAA-AuthZ component to support token-based enforcement mechanism in the Token Based Networking (TBN) is briefly described below. TVS can be implemented as a proprietary AAA-NRPS solution or it can use one of the existing standard models such as the Credential Validation Services (CVS) [14] or WS-Trust Secure Token Service (STS) [15].

Using AuthZ tickets during the reservation stage for communicating the 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 assigned GRI 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 an NRPS or a TVS in each domain and referred to with the AuthzToken that can be much smaller and even communicated in-band. At the resource PEP it can be compared with the cached AuthzTicket, AuthZ session context or reservation context and will allow local PEP/resource access control decisions. Such an access control enforcement model is being implemented in the Token Based Network (TBN) described in [16].

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 Local Reservation Identifier (LRI). The proposed TVS and token management model allows for hierarchical and chained GRI-LRI generation and validation.

3. GAAA-AuthZ access control mechanisms and components For CRP

The proposed GAAA-AuthZ access control mechanisms and components extend the generic model described in GAAA-AuthZ [2] with the specific functionality for on-demand NRP, in particular:

- AuthZ session management to support complex AuthZ decision and multiple resources access, including multiple resources belonging to different administrative and security domains.
- AuthZ tickets with extended functionality to support AuthZ session management, delegation and obligated policy decisions.
- Authorisation and reservation tokens as part of policy enforcement mechanisms that can be used in the control plane and in-band.
- Reference model for policy obligations Handling (OHRM) to support usable/accountable resource access/usage and additionally global and local user account mapping widely used in Grid based applications and supercomputing.

Although the above listed functionalities can be implemented under extended PEP or PDP functionality, such an approach would significantly limit the AuthZ service flexibility and potentially affect interoperability of different implementations as the discussed functionalities require an agreement on a number of protocol issues, messaging formats and attribute semantics.

The solutions proposed in the GAAA-AuthZ framework are based on using such structural components and solutions as a Token Validation Service (TVS), the Obligation Handling Reference Model (OHRM), and the XACML policy profile for multidomain NRP, being developed in the framework of the Phosphorus project and briefly described in this section.

Figure 2 illustrates the major GAAA-AuthZ modules and how they interact when evaluating a service request.

The authorisation service is called from the service/application interface via the AuthZ gateway (that can be just an interceptor process called from the service or application) that intercepts a service request ServiceRequest (ServiceId, AuthN, AuthZ) that contains a service name (and variables if necessary) and AuthN/AuthZ attributes.

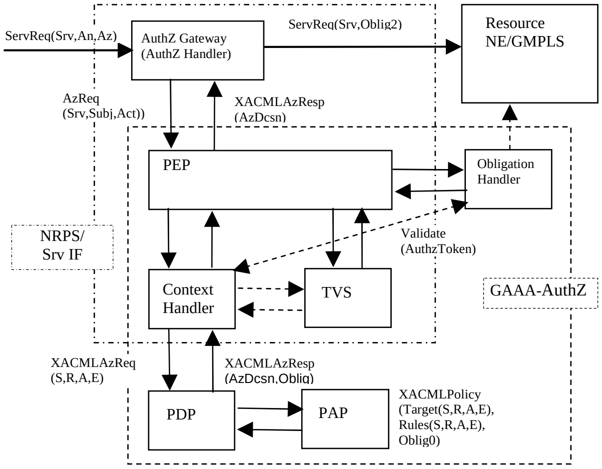

Figure 2. GAAA-AuthZ components providing service request evaluation

The AuthZ Gateway extracts necessary information and sends an AuthZ request AuthzRequest (ServiceId, Subject, Action), that contains a service name ServiceId, the requestor's identification and credentials, and the requested Action(s), to the PEP. The major PEP's task is to convert AuthZ request's semantics into the PDP request of which the semantics is defined by the used policy format. When using an XACML policy and correspondingly an XACML PDP, the PEP will send an XACML AuthZ request to the PDP in the format (Subject, Resource, Action, (Environment)). If in general case the XACML policy contains obligations, they are returned in the XACMLAzResponse (AuthzDecision, Obligations). The PEP calls the Obligation Handler to process obligations which are defined as actions to be taken on the policy decision or in conjunctions with the service access (like account mapping, quota enforcing, logging, or accounting).

If the service request contains an AuthZ token that references a local or global reservation ID, or just identifies an AuthZ session in which context the request is sent, the token validation is performed by the Token Validation Service (TVS). The TVS is typically called from the PEP and returns a confirmation if the token is valid. Defining TVS as a separate function or service allows creating flexible token and/or ticket based policy enforcement infrastructures for on-demand network resource provisioning. Specific details of our TVS implementation are described in Section 4.

3.1. AuthZ Session Management with AuthZ Ticket and AuthZ Token

The authorisation ticket (AuthzTicket) is a part of the GAAA-AuthZ framework functionality and allows the transfer of a full AuthZ decision and policy enforcement context between a requestor and an AuthZ service or between different AuthZ/security domains. More general information about using AuthZ tickets in different AAA/AuthZ operational model is provided in [17].

As discussed above, 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. In this case an AuthZ ticket should provide all necessary context information and will serve as session or access credentials.

To reduce possible high communication and processing overhead because of a potentially large size of AuthZ ticket, an AuthZ token can be used. In this case the AuthZ token should unambiguously reference the original AuthZ ticket or instant AuthZ session context that must be securely stored at the resource or access point. At the time of the authorised or reserved resource access, the original AuthZ ticket or AuthZ session context object will be retrieved and used for the request evaluation. When used together, AuthzTicket and AuthzToken share the SessionId attribute which can be either global or local reservation/session ID and are cryptographically connected, e.g. the token value is a hash value of the ticket content. An AuthzTicket must be digitally signed to keep its integrity.

In a particular use case of the TBN, the AuthzTicket is used for programming TVS and it provides both a reservation ID/reference and detailed information for configuring a token based ForCES switch (TBS) [18].

The proposed AuthzTicket format and its current implementation in the GAAA-AuthZ supports extended functionality for distributed multidomain hierarchical resources access control and user roles/permissions management, session based permissions delegation and conditional AuthZ decision assertion (to support XACML policy obligations). Important AuthzTicket functionality is that it may include any security context that need to be communicated between domains and layers that is included into specially introduced for this purpose the SessionContext element. It is one of the general design suggestions that an AuthzTicket should be easily mapped to the XACMLAuthzDecision Assertion defined by the SAML profile of XACML [19].

We refer to [11, 17] for more detailed information about the AuthzTicket format, of which the report [17] provides also examples and comparative information about the AuthzTicket and AuthzToken size when using proprietary AAA AuthzTicket format and SAML-XACML AuthZ assertion.

3.2. XACML policy and Attributes profile for NRP

The XACML policy format supports the required functionality for Complex Resource Provisioning in its core specification [20] and special profiles, in particular, for hierarchical resources [21]. Hierarchical policy management and dynamic rights delegation, that are considered as important functionality in multidomain NRP, can be solved with the XACML v3.0 administrative policy profile [22].

A XACML policy is defined for the target tuple "Subject-Resource-Action" (S-R-A) which can also be completed with the Environment (S-R-A-E) component in order to add additional context to instant policy evaluation. 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.

A decision request sent in a request message provides a context for the policy-based decision. The policy applied 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 the S-R-A-E elements allow an extensible "Attribute/AttributeValue" definition to support different attributes semantics and data types. Additionally, XACML allows referencing context information (from the request message) and external XML document elements by means of XPath functionality.

Two mechanisms can be used to bind the XACML policy to the resource: a Target element that can contain any of S-R-A-E attributes, and a policy identification attribute IDRef. The XACML policy format provides a few mechanisms to add and handle domain or session related context during the policy selection and request evaluation:

- Policy identification that is done based on the Target comprising of the Resource, Action, Subject, and optionally Environment elements.
- Attributes semantics and metadata can be namespace aware and used for attributes resolution during the request processing.
- AuthZ ticket that can be provided as an Environment or Resource attribute.

Such specific use case as multidomain NRP using chain provisioning model (see section 2 for definition) requires that the resource reservation policy in each successive domain will rely on the previous domain positive AuthZ decision, additionally the policy may require implying special conditions for next domain, e.g. type of network service or type of user account. In a simple case, this can be achieved by placing an AuthZ or reservation ticket from the previous domain in the Environment element. When the sequence is important it can be achieved with the ordered rules and policy combination algorithms correspondingly defined for the Policy or PolicySet [20].

Another important functionality that allows specifying requested network path and/or its parameters can be achieved with using XML/RDF based Network Description Language (NDL) [23] provided as a context in the Resource element of the XACML Request. The path parameters can be included into the XACML policy evaluation by using XPath based XACML AttributeSelector functionality [20] and additional attribute identifiers specified in the XACML hierarchical resource profile [21].

In order to use the XACML policy format for AuthZ in NRP, a special XACML-NRP profile for Network Resource Provisioning was proposed to address the following issues [24]:

- Namespace definition for the network resources, user attributes, and GAAA/AuthZ components
- Attribute semantics and expression format, including supported list of enumerated values, if necessary
- Set of basic rules and policy templates (including possible mapping to other currently used policy formats in NRP)

A successful XACML-NRP profile introduction will depend on available reference implementation. This is one of the goals of the Phosphorus project [1] to provide a reference implementation for the GAAA-AuthZ and XACML-NRP in particular..

3.3. Reference Model for Policy Obligations Handling (OHRM)

In many applications, policies may specify actions that must be performed either instead of or in addition to the policy decision. In the XACML specification [20], obligations are defined as actions that must be performed in conjunction with policy evaluation on a positive or negative decision. Obligations are included into the policy definition and returned by PDP to PEP which in its turn should take actions as prescribed in the obligation instructions or statements.

In the context of the GAAA-AuthZ architecture for NRP, obligations provide an important mechanism for policy decision enforcement in the provisioned network resources, in particular, obligations can be used for mapping global user ID/account to local accounts or groups, assigning quotas, usage limits, or specifying requirements for interdomain connectivity, VLAN or link types.

The proposed obligations handling model allows two types of obligations execution: at the time of receiving obligations from the PDP and at the later time when accessing a resource or performing an authorised action. The first type is described below, the second type of handling obligations can be achieved by using AuthZ tickets that hold obligations together with AuthZ decisions.

It is important to notice that obligations are an integral part of the policy and typically included into the policy at the stage of its creation by the policy administrator or resource owner. For manageability purposes, policies are considered stateless and the statefulness of obligations is achieved by the obligation handlers. The obligations enforcement process can be resulted either in modifying the service request (e.g., map from subject to account name/type) or by changing the resource/system sate or environment.

For the general (stateful) obligations handling process we can distinguish the following stages (note: not all stages are necessary to be implemented in a simple use case but they may exist in different cases):

```
   Obligation0 = tObligation => Obligation1
(“OK?”, (Attributes1 V Environment1))
=> Obligation2 (“OK?”, (Attributes2 V
Environment2)) => Obligation3 (Attributes3 V
Environment3)
```

1) Obligation0 – (stateless) obligations are returned by the PDP in a form as they are written in the policy. These obligations can be also considered as a kind of templates or instructions, tObligation.

2) Obligation1 or Obligation2 – obligations have been handled by the obligation handler at the PDP side or at the PEP side, depending on implementation. In this case templates or instructions of the Obligation0 are replaced with the real attributes in Obligation1, e.g. in a form of "name-value" pair. During this stage, the obligation handler can actually enforce obligations or modify obligations and send them further for enforcement by the resource. The result of obligations processing/enforcement, can be returned in a form of modified AuthzResponse (Obligation1) or in a form of global resource environment changes that will be taken into account at the time when the requested service/resource are provided or delivered. In both cases (and specifically in the last case), the obligation handler should return notification about fulfilled obligated actions, e.g. in a form of boolean value "False" or "True", which will be taken into account by PEP or other processing module to finally permit or deny service request by PEP.

3) Obligation3 – this is the final stage when obligations actually take effect, which can be defined as obligations "termination". This can be done by the resource itself or by services managed/controlled by the resource.

In the proposed model, an option with the Obligation1 handling stage at the PDP side is introduced to reflect a case when we need to implement a stateful domain or site central AuthZ service what is considered important for the general CRP and NRP use cases in particular.

Another important aspects of the general obligations handling model is logical or time wise sequence of enforcing obligations before, at, or after the requested action is performed. This aspect of the obligations enforcement and coordinated decision making is discussed in [25] and a solution is proposed in the PERMIS framework [26], but such functionality is not required for currently being developed GAAA-NRP infrastructure.

4. Token Validation Service (TVS)

The Token Validation Service (TVS) is a component of the GAAA-AuthZ infrastructure supporting token based policy enforcement mechanism during the user access of the reserved service or network. Basic TVS functionality allows checking if a service/resource requesting subject or other entity, that presents a token, has the right/permission to access/use a resource based on advance reservation to which this token refers. During its operation TVS checks if a presented token has reference to a previously reserved resource and a request conforms with the reservation conditions. It is intended that extended TVS functionality will also support policy enforcement for the consumable resources (also called usable resources) [27].

In a simple/basic scenario, TVS operates locally and checks the local reservation table directly or indirectly using the GRI. It is anticipated that in multidomain scenarios each domain may maintain a Local Reservation ID (LRI) and its mapping to the GRI.

In more advanced scenario, TVS should allow creation of a TVS infrastructure to support tokens and token related keys distribution in order to enable dynamic resources, users or providers federations.

The general TVS functionality supports two basic use cases: Token Based Networking (TBN) using in-band token based policy enforcement, and Service/Control Plane token based signalling in GMPLS networks. In the first case of TBN, the TVS functions and components are hardware accelerated by using network processors. Details on the implementation of TVS and PEP for in-band token enforcement are available in [18].

The proposed token handling model allows for integration of the circuit provisioning and application flow provisioning as different layers of the token based enforcement model.

In current TVS implementation, the token generation and validation model is based on the shared secret HMAC-SHA1 algorithm [28]. The TokenKey is generated in the following way:

```
TokenKey = HMAC(GRI, tb_secret)
```

where

GRI – global reservation identifier,
tb_secret – shared Token Builder secret.

A token is created in a similar way but using TokenKey as a HMAC secret:

```
TokenValue = HMAC(GRI, TokenKey)
```

This algorithm allows for chaining token generation and validation process, e.g.:

```
"GRI-TokenKey0-TokenValue0 =>
     => LRIl-TokenKeyl-Token1"
```

where TokenValue0 in one domain is used as **LRI1=TokenValue0** for generating TokenKey1 in other domain.

The key management model is not discussed at this stage of the TVS implementation. 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) [12, 13] that will allow to replace the currently used shared secret token handling model that has known manageability and scalability problems.

The current TVS implementation allows handling two types of tokens in binary and in XML formats. In both cases the reservation token is a tuple of GRI and TokenValue that should be included into the service request or AuthZ request.

5. GAAA Toolkit implementation

All proposed GAAA-AuthZ functionality is currently being implemented in the GAAA Toolkit (GAAA-TK) pluggable Java library in the framework of the Phosphorus project [29]. The library provides 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) and allows for complex policy driven resource reservation and scheduling scenarios.

The library allows for AuthZ request evaluation with local XACML based PDP or calling out to the external DCAS using the SAML-XACML protocol. For the convenience of application developers, the GAAA-TK provides simple XACML policy generation tools.

Currently, the TVS component is implemented as a part of the general GAAA-TK library but can also be used separately. It provides all required functionality to support token based policy enforcement mechanism that can be used at each networking layer and in particular for token based networking. All basic TVS functions are accessible and requested via a Java API. Further TVS development will extend WS interface to allow all TVS functions be accessible via Web services. Current TVS implementation supports shared secret and PKI based token key distribution, future release will implement IBC based interdomain trust management.

6. 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 User-Programmable Virtual Network (UPVN). 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.

Having current GAAA-TK implemented in the Phosphorus testbed and local University of Amsterdam AAA testbed will provide a basis for testing and further development both the NRP/CRP model and the proposed GAAA-AuthZ mechanisms and supporting functional components.

Further development of the proposed XACML-NRP policy and attributes profile for NRP will require wider Grid and network community discussion to define basic set of network and user related attributes that should allow flexible definition of the topology aware XACML policies and easier integration with Grid applications. As a first step, the XACML-NRP profile was presented to the Network Mark-up Language Working Group (NML-WG) at Open Grid Forum (OGF) [30]. The XACML-NRP profile will re-use where possible the recently released “An XACML Attribute and Obligation Profile for AuthZ Interoperability in Grids” [6].

The authors will continue ongoing research into supporting AAA/AuthZ architecture for UPVN where the proposed CRP model and TVS functionality are considered as security enabling concepts and require more general security model for GRI/LRI management, token key distribution and validation. Currently proposed and being implemented TVS infrastructure uses a shared secret security model that provides limited functionality for flexible network programming and has known manageability problems. To reduce key distribution and management problem when dynamically deploying reserved network resources, we consider investigating and testing the IBC technology for interdomain trust management.

The authors believe that the proposed AuthZ infrastructure and security mechanisms for general and network resource provisioning will provide a good basis for further discussion among Grid and networking specialists.

7. Acknowledgements

The authors would like to acknowledge the FP6 EU funded Integrated project PHOSPHORUS (Lambda User Controlled Infrastructure for European Research) IST-034115 for providing support for this work.

8. References

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

[2] 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

[3] gLite Lightweight Middleware for Grid Computing. [Online]. Available: http://glite.web.cern.ch/glite/

[4] The Globus Toolkit. [Online]. Available: http://www.globus.org/toolkit/

[5] OGSA-AuthZ Working Group. [Online]. Available: https://forge.gridforum.org/sf/projects/ogsa-authz

[6] An XACML Attribute and Obligation Profile for Authorization Interoperability in Grids. [Online] Available: https://edms.cern.ch/document/929867/1

[7] Gommans, L. et al. "Applications Drive Secure Lightpath Creation across Heterogeneous Domains", Special Issue “IEEE Communications Magazine, Feature topic Optical Control Planes for Grid Networks: Opportunities, Challenges and the Vision”, March 2006.

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

[9] Demchenko, Y., L. Gommans, C. de Laat, A. Taal, A. Wan, O. Mulmo, "Using Workflow for Dynamic Security Context Management in Grid-based Applications," Grid2006 Conf. Barcelona, Sept. 28-30, 2006.

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

[11] Demchenko Y, L. Gommans, C. de Laat, A. Wan, O. Mulmo, "Dynamic security context management in Grid-based applications", Future Generation Computer Systems (2007), The International Journal of Grid Computing: Theory, Methods and Applications, doi:10.1016/j.future.2007.07.015

[12] A. Shamir. Identity-based cryptosystems and signature schemes. In G.R. Blakley and D. Chaum, editors, Advances in Cryptology - Proceedings of CRYPTO'84, pages 47{53. Springer-Verlag LNCS 196, 1985.

[13] H. Tanaka. A realization scheme for the identity-based cryptosystem. In C. Pomerance, editor, Advances in Cryptology - Proceedings of CRYPTO'87, pages 340{349. Springer-Verlag LNCS 293, 1988.

[14] Chadwick, D., "Use of WS-TRUST and SAML to access a CVS". OGSA-AUTHZ WG Draft. [Online]. Available: https://forge.gridforum.org/sf/docman/do/downloadDocument/projects.ogsa-authz/docman.root.authz_service/doc9011/1

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

[16] "Token-based authorization of connection oriented network resources", by Leon Gommans, Franco Travostino, John Vollbrecht, Cees de Laat, and Robert Meijer, in Proceedings of GRIDNETS, San Jose, CA, USA, Oct 2004.

[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] "The Token Based Switch: Per-Packet Access Authorisation to Optical Shortcuts", by Mihai-Lucian Cristea, Leon Gommans, Li Xu, and Herbert Bos, in Proceedings of IFIP Networking, Atlanta, GA, USA, May 2007.

[19] SAML 2.0 Profile of XACML 2.0, Version 2. Working Draft 2, 26 June 2006. [Online]. Available: http://docs.oasis-open.org/xacml/2.0/xacml-2.0-profile-saml2.0-v2.zip

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

[21] "Hierarchical resource profile of XACML 2.0", OASIS Standard, 1 February 2005, available from http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-hier-profile-spec-os.pdf

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

[23] Grosso, P., F. Dijkstra, J. van der Ham, C. de Laat, "Network Description Language – Semantic Web For Hybrid Networks", Proceedings of TNC2007. [Online]. Available: http://tnc2007.terena.org/programme/presentations/show.php?pres_id=61

[24] XACML Authorisation Interoperability profile for Network Resource Provisioning: Attributes used for authorisation in network resource provisioning, Work in progress, version 0.1, June 20, 2008. [Online]. Available: http://staff.science.uva.nl/~demch/projects/aaauthreach/draft-interop-xacml-nrp-profile-01.pdf

[25] Zhao, G., D. Chadwick, S. Otenko, "Obligations for Role Based Access Control", Proc. "Advanced Information Networking and Applications Workshops", 2007, Advanced Information Networking and Applications Workshops (AINAW), 21st International Conference on, 21-23 May 2007, Pp. 424 - 431.

[26] D.Chadwick and A.Otenko. The PERMIS X.509 Role Based Privilege Management Infrastructure. Future Generation Computer System, 19(2):277--289, 2003.

[27] Zhang, X., M. Nakae, M. J. Covington, R. Sandhu, A Usage-based Authorization Framework for Collaborative Computing Systems, in the proceedings of ACM Symposium on Access Control Models and Technologies (SACMAT), 2006.

[28] Menezes A., P. van Oorschot, S. Vanstone, "Handbook of Applied Cryptography". - ISBN: 0-8493-8523-7, October 1996, 816 pages

[29] Aaauthreach Java project. [Online]. Available from http://staff.science.uva.nl/~demch/projects/aaauthreach/index.html

[30] Network Mark-up Language Working Group (NML-WG). [Online]. http://forge.gridforum.org/sf/projects/nml-wg/