



034115

PHOSPHORUS

Lambda User Controlled Infrastructure for European Research

Integrated Project

Strategic objective:
Research Networking Testbeds

Deliverable reference number D.4.3.1



GAAA toolkit pluggable components and XACML policy profile for ONRP

Due date of deliverable: 31-07-2008
Actual submission date: 31-07-2008
Document code: <Phosphorus-WP4-D.4.3.1>

Start date of project:
October 1, 2006

Duration:
30 Months

Organisation name of lead contractor for this deliverable:
University of Amsterdam

| Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006) | | |
|---|---|---|
| Dissemination Level | | |
| PU | Public | X |
| PP | Restricted to other programme participants (including the Commission Services) | |
| RE | Restricted to a group specified by the consortium (including the Commission Services) | |
| CO | Confidential, only for members of the consortium (including the Commission Services) | |



Abstract

This deliverable describes the implementation of the generic AAA Authorisation framework (GAAA-AuthZ) for Optical Network Resource Provisioning (GAAA-NRP profile) as the pluggable GAAA Toolkit library (GAAA-TK) and the proposed XACML-NRP attributes and policy profile.

The report summarises recent developments and enhancements to the GAAA-NRP resulted from discussions with the project partners and the Grid community. Some additional modifications have been done to the initially proposed GAAA-AuthZ/GAAA-NRP architecture in the project deliverable D4.1.

The report describes the major security mechanisms and functional components that comprise the GAAA-NRP profile such as authorisation tickets and tokens, Token Validation Service (TVS), reference model for policy obligations handling (OHRM).

The document describes the XACML-NRP attributes and the policy profile that proposes a set of attributes to describe a resource, a subject, and an action that are the major components of the authorisation/access control policy definition. The XACML-NRP profile has been proposed to and was discussed within the Grid community and it got positive feedback and useful contribution.

It is one of the design principles of the GAAA-NRP and GAAA-TK that the proposed architecture and implementation should allow an easy integration in other network management/control frameworks and Grid middleware that are currently used and being developed by NREN and Grid communities.

The report describes in detail a set of APIs used to call main the GAAA-TK services: authorisation service requested via the Policy Enforcement Point (PEP) or directly to a Policy Decision Point (PDP), and the TVS API that provides rich functionality for handling token used for access control and signalling. A special section is devoted to the GAAA-TK library setup and configuration.

The current GAAA-TK library implementation provides full functionality needed to support basic testbed scenarios. It has been tested with the WP1 Harmony testbed and expects integration into the WP2 G2MPLS middleware. It is however planned that the library will be updated after feedback received from the implementers and integrators.

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



Table of Contents

| | | |
|-------|---|----|
| 0 | Executive Summary | 5 |
| 1 | Introduction | 7 |
| 2 | Authorisation Infrastructure for Multidomain Network Resource Provisioning (GAAA-NRP) | |
| | – General Design | 9 |
| 2.1 | NRP/CRP operational models and AAA Authorisation service architecture | 9 |
| 2.2 | GAAA-AuthZ access control mechanisms and components | 12 |
| 2.3 | AuthZ Ticket formats for extended AuthZ Session Management | 14 |
| 2.3.1 | AuthZ Ticket use in GAAA-NRP | 14 |
| 2.3.2 | AuthzTicket data model and schema | 14 |
| 2.4 | Using AuthZ Tokens for Access Control and Signalling | 17 |
| 2.5 | Token Validation Service (TVS) | 19 |
| 2.5.1 | Basic TVS Functionality | 19 |
| 2.5.2 | Token handling model | 19 |
| 2.6 | Policy Obligations and Obligations Handling Reference Model (OHRM) | 20 |
| 3 | XACML policy profile for OLPP/CRP | 23 |
| 3.1 | Access Control in NRP – Basic Use Cases | 23 |
| 3.2 | Network topology formats used in multi-domain NRP | 24 |
| 3.2.1 | Phosphorus WP1 topology definition [17] | 24 |
| 3.2.2 | Network Description Language (NDL) by UvA and OGF [18] | 26 |
| 3.2.3 | OSCARS topology description format [19] | 28 |
| 3.3 | XACML Policy and attributes format | 29 |
| 3.4 | Attributes used for Authorisation and XACML policy definition | 31 |
| 3.4.1 | Attributes namespace | 31 |
| 3.4.2 | Network or Resource related attributes | 31 |
| 3.4.3 | Subject related attributes | 33 |
| 3.4.4 | Action related attributes | 34 |
| 3.4.5 | Environment related attributes | 35 |
| 3.4.6 | Policy Obligations used in NRP | 35 |
| 3.5 | Policy Expression Conventions | 36 |
| 3.6 | Policy identification and policy resolution | 37 |

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



GAAA toolkit pluggable components and XACML policy profile for ONRP

| | | |
|-------|---|----|
| 3.6.1 | General suggestions | 37 |
| 3.6.2 | Policy resolution convention in the GAAA-TK library | 38 |
| 3.6.3 | Policy identification | 39 |
| 4 | GAAA Toolkit Library | 40 |
| 4.1 | GAAA-TK library components | 40 |
| 4.2 | General AAA/AuthZ API and programming examples | 42 |
| 4.2.1 | PEP-GAAAPI interface | 42 |
| 4.2.2 | Simple XACML PDP API | 44 |
| 4.2.3 | GAAA-PEP API programming examples | 44 |
| 4.2.4 | Attribute expression conventions | 46 |
| 4.3 | TVS API | 48 |
| 4.3.1 | TVS interface | 48 |
| 4.3.2 | TVS programming examples | 49 |
| 4.4 | Authorisation ticket and token examples | 50 |
| 4.4.1 | Authorisation ticket examples | 50 |
| 4.4.2 | TVS XML Token format and examples | 50 |
| 4.5 | Simple XACML policy generation tools | 52 |
| 5 | GAAA-TK library Installation and configuration | 53 |
| 5.1 | Configuration | 53 |
| 5.2 | Installation | 54 |
| 5.3 | Required external libraries | 54 |
| 6 | Conclusion | 56 |
| 7 | References | 58 |

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



0 Executive Summary

The Authentication, Authorisation and Accounting (AAA) service is an important component of the supporting infrastructure for on-demand Optical Network Resource Provisioning (ONRP) across multiple domains and different target consumer applications. A consistent AAA infrastructure requires interactions of the related AAA components at all networking layers including the network/forwarding elements, the control plane, the reservation and provisioning service, and the user/target application layer.

The report summarises recent developments and enhancements to the GAAA-NRP resulted from discussions with the project partners and the Grid community. Some additional modifications have been done to the initially proposed GAAA-AuthZ/GAAA-NRP architecture in the project deliverable D4.1.

The report describes the major security mechanisms and functional components that comprise the GAAA-NRP profile such as authorisation tickets and tokens, the Token Validation Service (TVS), and a reference model for policy obligations handling (OHRM).

The document describes the XACML-NRP attributes and a policy profile that proposes a set of attributes to describe a resource, a subject, and an action that are the major components of the authorisation/access control policy definition. The XACML-NRP profile has been proposed to and was discussed within the Grid community and it got positive feedback and useful contribution.

It is one of the design principles of the GAAA-NRP and GAAA-TK that the proposed architecture and implementation should allow an easy integration in other network management/control frameworks and Grid middleware as currently used and being developed by NRENs and Grid communities.

The report describes in detail a set of APIs used to call main GAAA-TK services: authorisation service requested via the Policy Enforcement Point (PEP) or directly to a Policy Decision Point (PDP), and the TVS API that provides rich functionality for handling token used for access control and signalling. A special section is devoted to the GAAA-TK library setup and configuration.

The current GAAA-TK library implementation provides full functionality needed to support basic testbed scenarios. It has been tested with the WP1 Harmony testbed and expects integration into the WP2 G2MPLS

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



GAAA toolkit pluggable components and XACML policy profile for ONRP

middleware. It is however planned that the library will be updated after feedback received from the implementers and integrators.

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



1 Introduction

The main objective of the Phosphorus project is to address some of the key technical challenges to enable on-demand e2e network services across multiple administrative and security domains. The Authentication, Authorisation and Accounting (AAA) service(s) is considered as an important component of the supporting infrastructure for on-demand Optical Network Resource Provisioning (ONRP) across multiple domains and different target consumer applications. A consistent AAA infrastructure requires interaction of the related AAA components at all networking layers including network/forwarding elements, control plane, reservation and provisioning service, and user/target applications layer.

This deliverable describes the result of the implementation of the AAA Authorisation infrastructure for multi-domain Optical Network Resources Provisioning (ONRP) as a pluggable GAAA-TK Java library. The proposed library is designed in such a way that it could support the major Phosphorus testbed use cases and can be used at all networking layers: Data plane, Control Plane, Service plane, and can also work with applications. The proposed GAAA-NRP architecture and GAAA-TK library also targets to ensure future compatibility with the Grid and NREN access control solutions and infrastructures.

The report is organised as follows. Section 2 summarises the recent Generic AAA Authorisation framework (GAAA-AuthZ) for ONRP developments. It describes necessary functionalities to support multi-domain ONRP and introduces a number of mechanisms and solutions to support them, in particular: an AuthZ ticket format for extended AuthZ session management, an AuthZ token format for multi-domain access control and signalling, a Token Validation Service (TVS) to enable token based policy enforcement, and a policy Obligation Handling Reference Model (OHRM). The proposed architecture will allow a smooth integration with other AuthZ frameworks as currently used and developed by NRENs and the Grid community.

Section 3 describes in detail the proposed XACML-NRP attributes and the policy profile for general and optical network resource provisioning.

Section 4 describes a set of APIs used to call the main GAAA-TK services. The authorisation service can be requested via the Policy Enforcement Point (PEP) or directly from the XACML Policy Decision Point (PDP). The TVS API provides rich functionality to handle token used for access control and signalling. A separate section 5 is devoted to the GAAA-TK library setup and configuration.

Finally, Section 6 provides a summary and suggests further library developments and updates after receiving feedback from other packages.

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



GAAA toolkit pluggable components and XACML policy profile for ONRP

It was a special decision to release the full functional GAAA-TK library two months before the major WP1 and WP2 deliverables to allow them to integrate AAI functionality into their products and testbeds.

The developed library has been initially tested by WP1 and currently being integrated into the WP1 Harmony application software.

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



2 Authorisation Infrastructure for Multidomain Network Resource Provisioning (GAAA-NRP) – General Design

This chapter describes the GAAA/AuthZ architecture for Optical Network Resource Provisioning, hereafter referred to as GAAA-NRP. The GAAA-NRP extends further the generic AAA Authorisation Framework [1, 2] and provides a basis for defining the major security mechanisms and components to support multi-domain network provisioning process such as an AuthZ ticket format for extended AuthZ session management, an AuthZ token format for multi-domain access control and signalling, a Token Validation Service (TVS) to enable token based policy enforcement, and a policy Obligation Handling Reference Model (OHRM).

2.1 NRP/CRP operational models and AAA Authorisation service architecture

The recent research by the authors showed that the major Network Resource Provisioning (NRP) use cases can be abstracted to the same CRP operational model when considering their implementation with the Grid or Web Services [3, 4]. 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 such networks.

The typical on-demand resource provisioning process includes four major stages: (1) resource reservation, (2) deployment (or activation), (3) the reserved resource access/consumption, and additionally (4) resource de-commissioning after it was used. 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 an advance reservation system or a meta-scheduling system [5] and is driven by the provisioning workflow and related AuthZ policy [6]. At the deployment stage the reserved resources are bound to the reservation ID, which we refer as the Global

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



GAAA toolkit pluggable components and XACML policy profile for ONRP

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 which are again currently considered as separate actions outside of the general provisioning workflow. In this paper we are primarily focused on the first three provisioning stages as the most related to the applications operation.

Defining different CRP workflow stages will allow developing and using different security models for the policy enforcement, trust and security context management.

In the discussed CRP model, domains are defined (as associations of entities) by a common policy of a single administration, with 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 [6]. 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 multi-domain distributed resource management model requires the following functionality from the GAAA-AuthZ infrastructure:

- multiple policies processing and combination.
- attributes/rules mapping/convert 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 2.1 illustrates major interacting components in the multi-domain 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) (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 Service (TVS) that allows efficient authorisation decision enforcement when accessing reserved resources, and additionally can support token based service level signalling at the reservation stage.

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



GAAA toolkit pluggable components and XACML policy profile for ONRP

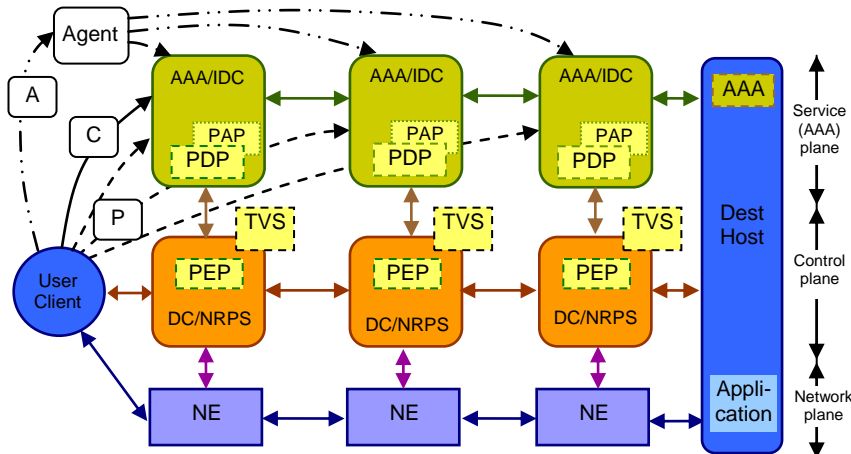


Figure 2.1. Components involved in multi-domain network resource provisioning and basic sequences (agent (A), chain (C), and polling (P))

Figure 2.1 also illustrates different provisioning models or sequences that can be executed when composing a complex resource:

Chain reservation sequence (also referred to as a provider sequence). The user contacts only the local network domain/provider that provides the destination address. Each consecutive domain provides a path to the next domain.

Polling sequence. The user client polls all resources or network domains, builds the path and makes the reservation.

Agent (or tree) sequence. The user delegates the network provisioning negotiation to an agent that will take care of all necessary negotiations to provide the required network path to the user. A benefit of “outsourcing” the resource provisioning is that the agents can maintain their own reservation and trust infrastructure. This can be considered as a basic provisioning sequence for currently used Grid resource management and advance reservation systems.

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) [1], 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 environment that AuthN credentials or assertions are presented as a security context in the AuthZ decision request and are evaluated before sending request to the PDP.

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

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



GAAA toolkit pluggable components and XACML policy profile for ONRP

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. The Token and token key generation and validation model can use either a shared secret, a PKI based trust model, or recently researched by authors the Identity Based Cryptography (IBC) [7, 8].

Using AuthZ tickets during the reservation stage to communicate 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/confirmed as a result of the successful reservation process.

AuthZ ticket and token formats and their use in the proposed AuthZ infrastructure for interdomain AuthZ context management and access control are described in details below.

To avoid significant policy enforcement overhead when handing a 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 [9].

It is an important convention for the consistent CRP operation that the 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.

2.2 GAAA-AuthZ access control mechanisms and components

The proposed GAAA-NRP access control mechanisms and components extend the generic model described in GAAA-AuthZ with the specific functionality for on-demand ONRP, 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 policy enforcement mechanisms that can be used in the G2MPLS Control plane and in-band.

Policy Obligations Handling model to support usable/accountable resource access/usage and additionally global and local user account mapping widely used in Grid based applications and supercomputing.

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



GAAA toolkit pluggable components and XACML policy profile for ONRP

Although the above listed functionalities can be implemented under extended PEP or PDP functionality, such an approach would significantly limit 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.

Figure 2.2 illustrates the major GAAA-NRP/GAAA-AuthZ modules and how they interact when evaluating a service request.

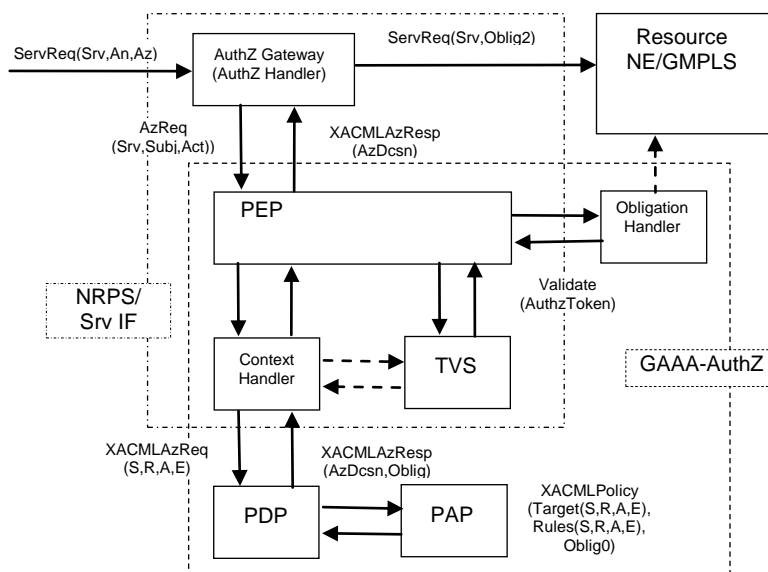


Figure 2.2. GAAA-AuthZ components providing service request evaluation

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. 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 which semantics is actually defined by the used policy. 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, attributes, (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 may reference 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. Separating TVS as a separate function or service allows creating flexible token and/or ticket policy enforcement infrastructures for on-demand network resource provisioning.

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



2.3 AuthZ Ticket formats for extended AuthZ Session Management

2.3.1 AuthZ Ticket use in GAAA-NRP

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.

As discussed above, there are two types of sessions in the proposed CRP model that require a security context management: reservation and provisioning 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 a session or access credentials.

To reduce possible high communication and processing overhead because of a potentially large size of an 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 a global or a 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 Token Based Networking, the AuthzTicket is used for programming a TVS and provides both a reservation ID/reference and detailed information for configuring a TBS (token based ForCES switch) [10].

2.3.2 AuthzTicket data model and schema

An AuthzTicket has a complex but flexible format. The current AuthzTicket format and its implementation in the GAAAPI supports extended functionality for distributed multi-domain hierarchical resources access control and user roles/permissions management, in particular, administrative policy management (as defined in XACML 3.0 Administrative policy profile [11]), capabilities delegation and conditional AuthZ decision assertion (to support XACML policy obligations). It is one of the general design suggestions that an AuthzTicket should be easy to map to the SAML AuthzDecision Assertion [12] or to XACMLAuthzDecision Assertion defined by the SAML profile of XACML [13, 14].

Diagram 2.3 illustrates the top and main AuthzTicket elements. The core AuthZ ticket schema is provided in the Appendix B (for more details see section 6 in the deliverable D4.1). Note, the Resource element is defined as

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



GAAA toolkit pluggable components and XACML policy profile for ONRP

an extendable in the AuthzTicket schema what allows to incorporate with different types of target resources (to which AuthZ decision is applied).

The AuthzTicket contains the following major groups of elements that allow for extended AuthZ session security context management:

- The Root element attributes `TicketID`, `SessionID`, and `Issuer` that allows for the ticket unique identification and defines its binding to the session and domains related processes/authorities.
- The `Decisions/Decision` element that holds the PDP AuthZ decision bound to the requested resource or service expressed as the `ResourceID` attribute.
- The `Resources` extendable element that may hold proprietary description of the reserved resource.
- 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 delegating the capabilities defined by the AuthzTicket to another subject(s) or community. The attributes define restriction on type and depth of delegation
- The `Conditions` element specifies the validity constrains for the ticket, including validity time and the AuthZ session identification and additionally context. The extensible `ConditionAuthzSession` element provides rich possibilities for AuthZ context expression.

The `Obligations/Obligation` element can hold obligations that a PEP/resource should perform in conjunction with the current PDP decision.

The semantics of AuthzTicket elements are defined in such a way that allows an easy mapping to related elements in SAML and XACML. The first three elements the `Decision`, the `Actions/Action`, and the `Subject` have a direct mapping to the related SAML elements. Other AuthzTicket elements the `Delegation`, the `Conditions`, and the `Obligations/Obligation` element, which is originated from XACML, can be implemented as extensible element of the SAML `Condition` element.

The `SessionID` attribute although defined as a general AuthZ session identifier in currently discussed Phosphorus use cases holds either a global or local (to a domain) reservation identifier (GRI/LRI).

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



GAAA toolkit pluggable components and XACML policy profile for ONRP

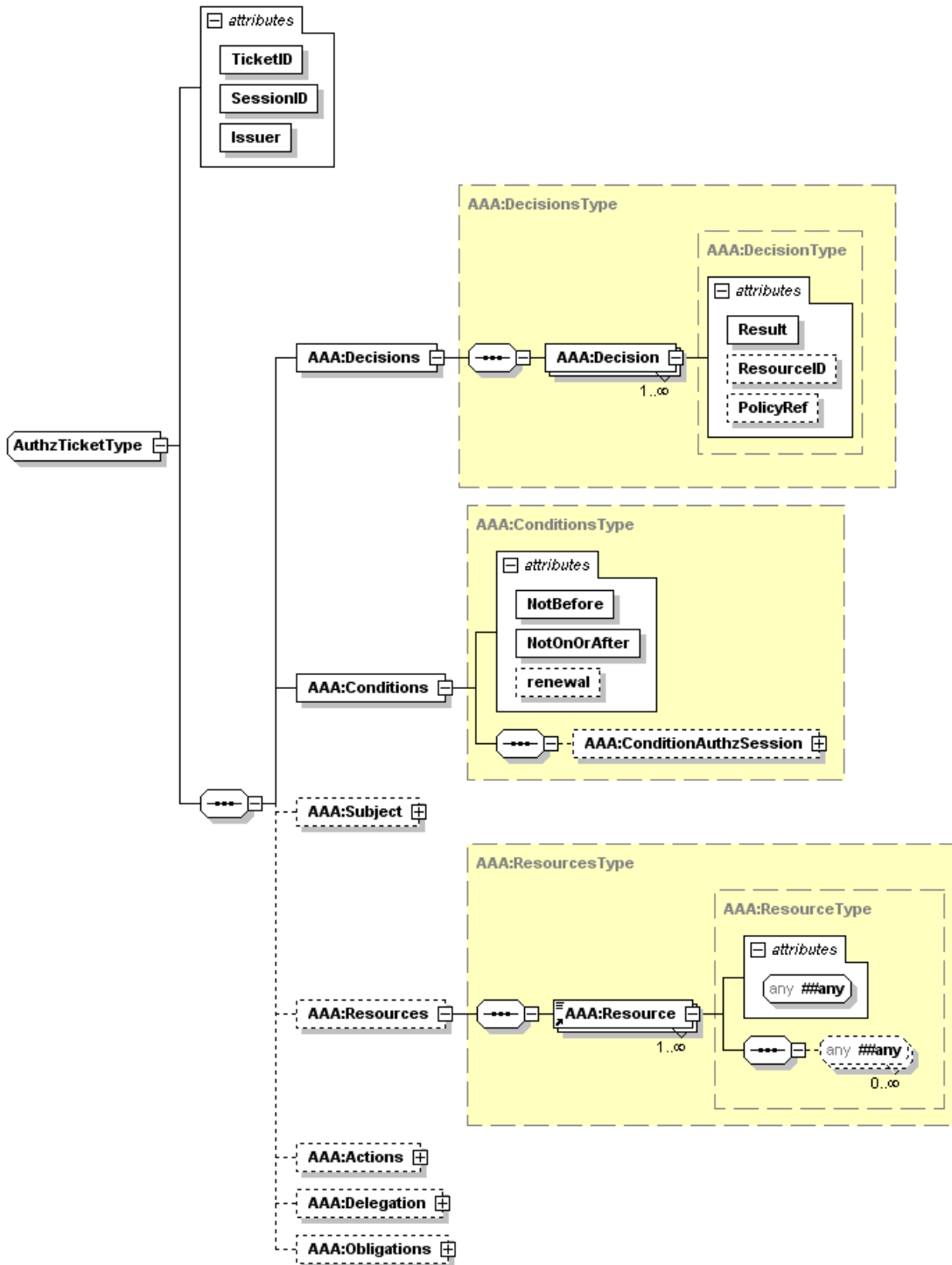


Fig. 2.3 AuthzTicket top and main elements (note, the Resource element is defined as an extendable).

The AuthzTicket is digitally signed (as shown in the example) and cached by the resource's AuthZ service. To reduce communication overhead when using an AuthzTicket for consecutive request validation, the associated AuthZ token (AuthzToken) can be generated from the AuthzTicket. The AuthzToken may contain just two

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



GAAA toolkit pluggable components and XACML policy profile for ONRP

elements: `TokenID = TicketID` and `TokenValue = SignatureValue`, needed for the identification of the cached `AuthZTicket`.

The current `AuthZTicket` functionality is supported by the GAAA-TK library (see chapter 4 for details).

2.4 Using AuthZ Tokens for Access Control and Signalling

The proposed GAAA-NRP architecture uses token extensively for access control and signalling at different NRP stages considering it as a flexible and powerful mechanism for communicating and signalling security context between domains.

We define token 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. This definition is more oriented for the NRP provisioning model/workflow and extends the proposed token definition as shared abstract permission in earlier authors' paper [4].

Tokens can be used for both access control when accessing the reserved resources and for signalling during reservation and deployment stages and 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 [4]. Pilot tokens functionality and format was proposed and defined as a result of the further development of the AuthZ infrastructure as an integral component of the NRP.

Figure 2.4 illustrates the common data model of both access tokens and pilot tokens. Although sharing a common data model they are different in the operational model and the way how 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 has a simple format and contains two mandatory elements: The `SessionId` attribute that holds the GRI, and the `TokenValue` element, and optional a `Condition` element that may contain two token validity time attributes `notBefore` and `notOnOrAfter`.

The GAAA-NRP 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 doesn't 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 of the GRI by applying e.g. HMAC function to the GRI together with the requestor symmetric secret or private key.

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



GAAA toolkit pluggable components and XACML policy profile for ONRP

Type3 – this pilot token type extends the Type2 with a Domains element that allows to collect domains security context information (in the Domains/Domain element) 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 domains security context information 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.

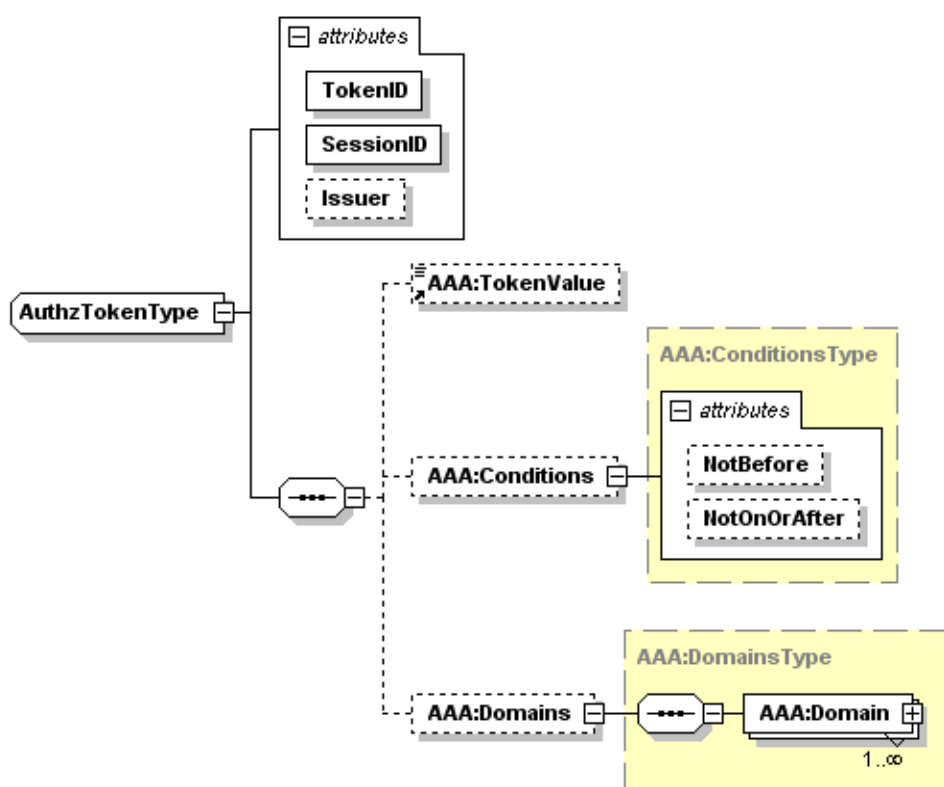


Figure 2.4. The access and pilot tokens data model.

The AuthzToken contains the following elements:

- The Root element attributes `TokenID`, `SessionID`, and `Issuer` that allows for the ticket unique identification and defines its binding to the session and domains related processes/authorities.
- The `Decisions/Decision` element that holds the PDP AuthZ decision bound to the requested resource or service expressed as the `ResourceID` attribute.
- The `Resources` extendable element that may hold proprietary description of the reserved resource.
- The `Actions/Action` complex element contains actions which are permitted for the subject or its delegates.

When used together with an AuthzTicket the ticket and token identification elements `TokenID`, `SessionID`, and `Issuer` can be shared. Examples of different token types are provided in section 4.

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



2.5 Token Validation Service (TVS)

2.5.1 Basic TVS Functionality

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 posses/presents current token, has right/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 resource/service confirms to a reservation condition. It is intended that extended TVS functionality will also support policy enforcement for the consumable (or usable) resource.

In a simple/basic scenario, the TVS operates locally and checks a local reservation table directly or indirectly using a reservation ID (typically a Global Reservation Id - GRI). It is suggested that in a multi-domain scenario each domain may maintain its Local Reservation ID (LRI) and 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 TVS functionality should support three basic use cases: Token Based Networking (TBN) using in-band token based policy enforcement [15], Control Plane token based signalling in VLSR networks, and Service Plane access control and signalling.

2.5.2 Token handling model

The token generation and handling model is based on the shared secret HMAC-SHA1 algorithm [16]. 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-TokenKey-TokenValue => LRI-1-TokenKey-1-Token"
```

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



GAAA toolkit pluggable components and XACML policy profile for ONRP

The key management model is not discussed at this stage of the project. 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) [7, 8] that will allow to replace shared secret token handling model that has know manageability problems.

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

2.6 Policy Obligations and Obligations Handling Reference Model (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 [14], 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 ONRP, 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, mapping a domain path request to a specific VLAN, assigning quotas or usage limits, and others.

Figure 2.5 below illustrates the Obligations Handling Reference Model (OHRM) for processing obligations in the general case of the Domain-Central AuthZ service (DCAS) that can be part of the Domain Controller (DC). The DCAS means that all domain located resources and services use a central AuthZ service that maintains a common set of policies for this domain. The described processing model is compliant to the model used in XACML (refer to XACML2.0 standard or see Appendix C) but adds Web services and AuthZ callout protocol details and specifically focuses on the obligations handling dataflow.

The 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. 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.

A number of assumptions are made to reflect possible options in an AuthZ service infrastructure implementation and different types of obligations both stateful and stateless that are concerned with assigning pool accounts, enforcing quotas, controlling usable resource (e.g., number of resource access, purchased video/music listening time, etc.), logging and accounting.

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



GAAA toolkit pluggable components and XACML policy profile for ONRP

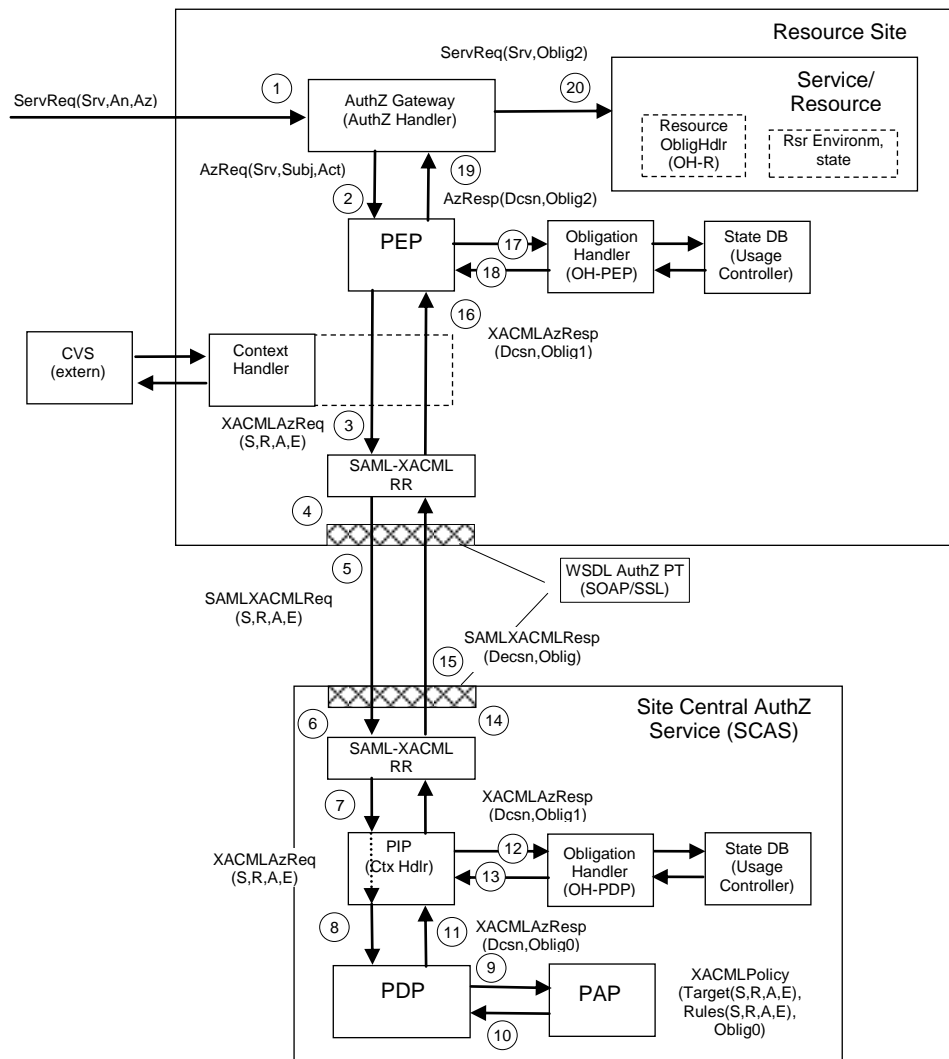


Figure 2.5. Generic authorisation dataflow and obligations handling in distributed AuthZ service.

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 the manageability purpose, policy is 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 state 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)
```

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



GAAA toolkit pluggable components and XACML policy profile for ONRP

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. (Important to mention that due to security reason obligations format and semantics should not use executable code or reference to locally executed commands).

2) **Obligation1** or **Obligation2** – obligations have been handled by the obligation handler at the DCAS/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) 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” or “sink”. This is done by the resource itself or by services managed/controlled by the resource.

In the proposed model, option with Obligation1 handling stage at the DCAS or PDP side is introduced to illustrate a case when we need to implement a stateful PDP/DCAS what is important for the general CRP and ONRP use cases in particular. However, this should not be considered as XACML specification violation as distinguishing between PEP and PDP functions in the generic obligations handling model is based on what module actually makes policy based request evaluation.

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



3 XACML policy profile for OLPP/CRP

This section provides information about the proposed and implemented in the GAAA-TK library the XACML-NRP attributes and policy profile for network resource provisioning. The section starts with the definition of the two basic use cases for access control and policy definition in the Phosphorus testbed. It provides a short overview of the existing network topology description formats that has relations to the Phosphorus testbed. The next section describes a set of resource, subject, actions and environment attributes that are considered important/relevant to the policy definition in more general NRP use cases. Additionally, the section provides suggestions about policy obligations that can be used in more complex policy definition and enforcement for advance reservations and multi-domain and multi-provider use cases.

Policy examples and corresponding Request and Response messages are provided in Appendix D.

3.1 Access Control in NRP – Basic Use Cases

The access control system and infrastructure protects a resource to allow/ensure that only authorised users can have access or execute specific actions on the resource. In general, the access control policy comprises of rules and conditions that specify what user with what attributes may access or execute what action on the resource with what attributes.

Two particular use cases for access control in Network Resource Provisioning (NRP) can be expressed in a simple narrative form:

Use case 1: "User A is only allowed to use user endpoints X, Y and Z", or

Use case 2: "User A is only allowed to use endpoints in domain N and M".

The following assumptions are made to satisfy the two above use cases and to ensure effective management of policies, user and resource identities and attributes:

Users and resources are described/identified by their unique ID's and may have also assigned attributes, which for the user may include such attributes as user group, role, or federation, and for the resource may include such attributes as domain/subdomain, resource type, level of service;

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



GAAA toolkit pluggable components and XACML policy profile for ONRP

users and resources (domains and endpoints) may be organised/associated into administrative and or security domains or federations, i.e. a user and a resource can be a member of one or multiple associations; different domains and endpoints participating in network connection (for which the authorisation is requested) may belong to different federations or security associations;

It is important that only authenticated users can access the resources that protected by AuthZ service. User authentication is confirmed by issuing AuthZ assertion by trusted AuthN service or creating user related security context environment of the process/session started at/after the time user logs in.

In general, user authentication may be resulted in the following:

service or process session initiation;

release of the user attributes or credentials;

depending on the user attributes (federations, groups, roles) the user can be assigned specific level of service;

a simple policy format will contain rules that express conditions what user attributes are required to access the resource with specific attributes or execute a specific action on the resource with specific attributes.

3.2 Network topology formats used in multi-domain NRP

The following rationale can be considered for defined a common topology format:

Topology format should provide necessary information about the network resource to allow consistent policy evaluation, and vice versa the policy format may be defined but the network topology to which the policy is applied.

Topology semantics will define the resource attributes semantics, and vice versa.

3.2.1 Phosphorus WP1 topology definition [17]

The network resource description includes the following attributes:

domainId

relationship (peer-domain or sub-domain)

TNA and TNA prefix

Link parameters average delay and maximum bandwidth

ReservationEPR as an attribute that may directly or indirectly define the resource federation or security/administrative domain

This information is presented in the XML based network topology description:

Example NSP-1

```
<ns4:Domains>
  <ns3:DomainId xmlns:ns3="http://ist_phosphorus.eu/nsp">SURFnet</ns3:DomainId>
  <ns3:Relationship
xmlns:ns3="http://ist_phosphorus.eu/nsp">subdomain</ns3:Relationship>
  <ns3:SequenceNumber xmlns:ns3="http://ist_phosphorus.eu/nsp">8279</ns3:SequenceNumber>
  <ns3:Description xmlns:ns3="http://ist_phosphorus.eu/nsp">SURFnet DRAC
domain</ns3:Description>
```

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



GAAA toolkit pluggable components and XACML policy profile for ONRP

```
<ns3:ReservationEPR
xmlns:ns3="http://ist_phosphorus.eu/nsp">http://10.1.4.1:8080/DracNRPS_WSRF/services/Reser
vation</ns3:ReservationEPR>
  <ns3:TNAPrefix xmlns:ns3="http://ist_phosphorus.eu/nsp">10.4.0.0/16</ns3:TNAPrefix>
</ns4:Domains>
```

Example NSP-2

```
<ns4:Domains>
  <ns3:DomainId xmlns:ns3="http://ist_phosphorus.eu/nsp">dummy</ns3:DomainId>
  <ns3:Relationship
xmlns:ns3="http://ist_phosphorus.eu/nsp">subdomain</ns3:Relationship>
  <ns3:SequenceNumber xmlns:ns3="http://ist_phosphorus.eu/nsp">1171</ns3:SequenceNumber>
  <ns3:Description xmlns:ns3="http://ist_phosphorus.eu/nsp">Virtual dummy
domain</ns3:Description>
  <ns3:ReservationEPR
xmlns:ns3="http://ist_phosphorus.eu/nsp">http://localhost:8080/nrpsDummyReservation/servic
es/MyService</ns3:ReservationEPR>
  <ns3:TopologyEPR
xmlns:ns3="http://ist_phosphorus.eu/nsp">http://localhost:8080/nrpsDummyTopology/services/
MyService</ns3:TopologyEPR>
  <ns3:NotificationEPR
xmlns:ns3="http://ist_phosphorus.eu/nsp">http://localhost:8080/nrpsDummyNotification/servi
ces/MyService</ns3:NotificationEPR>
  <ns3:TNAPrefix xmlns:ns3="http://ist_phosphorus.eu/nsp">128.0.0.0/16</ns3:TNAPrefix>
  <ns3:avgDelay xmlns:ns3="http://ist_phosphorus.eu/nsp">50</ns3:avgDelay>
  <ns3:maxBW xmlns:ns3="http://ist_phosphorus.eu/nsp">1111</ns3:maxBW>
</ns4:Domains>
```

Example NSP-3

```
<ns4:getEndpointsResponse xmlns:ns4="http://ist_phosphorus.eu/nsp/webservice/topology">
  <ns4:Endpoints>
    <ns3:EndpointId xmlns:ns3="http://ist_phosphorus.eu/nsp">10.8.1.2</ns3:EndpointId>
    <ns3:Name xmlns:ns3="http://ist_phosphorus.eu/nsp">Endpoint2</ns3:Name>
    <ns3:Description
xmlns:ns3="http://ist_phosphorus.eu/nsp">CRC_To_Viola</ns3:Description>
    <ns3:Interface xmlns:ns3="http://ist_phosphorus.eu/nsp">border</ns3:Interface>
    <ns3:DomainId xmlns:ns3="http://ist_phosphorus.eu/nsp">crc</ns3:DomainId>
    <ns3:Bandwidth xmlns:ns3="http://ist_phosphorus.eu/nsp">1250</ns3:Bandwidth>
  </ns4:Endpoints>
  <ns4:Endpoints>
    <ns3:EndpointId xmlns:ns3="http://ist_phosphorus.eu/nsp">10.8.1.4</ns3:EndpointId>
    <ns3:Name xmlns:ns3="http://ist_phosphorus.eu/nsp">Endpoint4</ns3:Name>
    <ns3:Description
xmlns:ns3="http://ist_phosphorus.eu/nsp">CRC_To_SURFnet</ns3:Description>
    <ns3:Interface xmlns:ns3="http://ist_phosphorus.eu/nsp">border</ns3:Interface>
    <ns3:DomainId xmlns:ns3="http://ist_phosphorus.eu/nsp">crc</ns3:DomainId>
    <ns3:Bandwidth xmlns:ns3="http://ist_phosphorus.eu/nsp">1250</ns3:Bandwidth>
  </ns4:Endpoints>
</ns4:getEndpointsResponse>
```

Example NSP-4

```
<ns4:getLinksResponse xmlns:ns4="http://ist_phosphorus.eu/nsp/webservice/topology">
```

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



GAAA toolkit pluggable components and XACML policy profile for ONRP

```
<ns4:Link>
  <ns4:SourceEndpoint>10.8.1.4</ns4:SourceEndpoint>
  <ns4:DestinationEndpoint>10.4.1.2</ns4:DestinationEndpoint>
  <ns4:Name>VLAN948</ns4:Name>
  <ns4:Description>crc-SURFnet interdomain link "VLAN948"</ns4:Description>
</ns4:Link>
<ns4:Link>
  <ns4:SourceEndpoint>10.7.3.2</ns4:SourceEndpoint>
  <ns4:DestinationEndpoint>10.4.1.1</ns4:DestinationEndpoint>
  <ns4:Name>VLAN947</ns4:Name>
  <ns4:Description>viola-SURFnet interdomain link "VLAN947"</ns4:Description>
</ns4:Link>
</ns4:getLinksResponse>
```

3.2.2 Network Description Language (NDL) by UvA and OGF [18]

NDL topology description uses RDF format for describing relations between network entity and its property. The following attributes or properties are specified:

Device identified by the name
Location describing domain
Interface
port
link
bandwidth

Example NDL-1

```
<!-- TDM3.amsterdam1.netherlight.net -->
<ndl:Device rdf:about="#tdm3.amsterdam1.netherlight.net">
  <ndl:name>tdm3.amsterdam1.netherlight.net</ndl:name>
  <ndl:locatedAt rdf:resource="#amsterdam1.netherlight.net"/>
  <ndl:hasInterface rdf:resource="#tdm3.amsterdam1.netherlight.net:501/1"/>
  <ndl:hasInterface rdf:resource="#tdm3.amsterdam1.netherlight.net:501/2"/>
  <ndl:hasInterface rdf:resource="#tdm3.amsterdam1.netherlight.net:501/3"/>

  <ndl:hasInterface rdf:resource="#tdm3.amsterdam1.netherlight.net:505/3"/>
  <ndl:hasInterface rdf:resource="#tdm3.amsterdam1.netherlight.net:505/4"/>
</ndl:Device>

<!-- all the interfaces of TDM3.amsterdam1.netherlight.net -->

<ndl:Interface rdf:about="#tdm3.amsterdam1.netherlight.net:501/1">
  <ndl:name>tdm3.amsterdam1.netherlight.net:POS501/1</ndl:name>
  <ndl:connectedTo rdf:resource="#tdm4.amsterdam1.netherlight.net:5/1"/>
  <ndl:capacity
rdf:datatype="http://www.w3.org/2001/XMLSchema#float">1.2E+9</ndl:capacity>
</ndl:Interface>
<ndl:Interface rdf:about="#tdm3.amsterdam1.netherlight.net:501/2">
  <ndl:name>tdm3.amsterdam1.netherlight.net:POS501/2</ndl:name>
  <ndl:connectedTo rdf:resource="#tdm1.amsterdam1.netherlight.net:12/1"/>
```

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



GAAA toolkit pluggable components and XACML policy profile for ONRP

```
<ndl:capacity
rdf:datatype="http://www.w3.org/2001/XMLSchema#float">1.2E+9</ndl:capacity>
</ndl:Interface>
```

Example NDL-2

```
<rdf:Description rdf:about="">
  <!-- meta data on this document itself -->
  <rdfs:label xml:lang="en">Configuration of Speculaas</rdfs:label>
  <dc:title xml:lang="en">Configuration of Speculaas</dc:title>
  <dc:description xml:lang="en">Configuration of the Speculaas switch at
Netherlight. This file is semi-dynamically generated by a cron job that logs in the
devices and retrieves all information. You should expect this data to be stale for about 5
minutes. If you really need real-time data, then don't use NDL, but another mechanism
(e.g. a routing protocol).</dc:description>
  <dc:publisher xml:lang="en">glimmerglassconfig.py script on
ndl.uva.netherlight.nl</dc:publisher>
  <dcterms:issued>2007-01-31</dcterms:issued>
  <dcterms:modified>2007-12-12T15:17:11+0100</dcterms:modified>
</rdf:Description>

<ndl:Device rdf:about="http://speculaas.uva.netherlight.nl#Speculaas">
  <rdfs:label>Speculaas</rdfs:label>
  <dc:description xml:lang="en">Speculaas Glimmerglass OXC</dc:description>
  <ndl:hasInterface rdf:resource="http://speculaas.uva.netherlight.nl#intf01"/>
  <ndl:hasInterface rdf:resource="http://speculaas.uva.netherlight.nl#intf02"/>
  <ndl:hasInterface rdf:resource="http://speculaas.uva.netherlight.nl#intf03"/>

  <ndl:hasInterface rdf:resource="http://speculaas.uva.netherlight.nl#intf19"/>
  <capability:hasSwitchMatrix>
  <capability:SwitchMatrix
rdf:about="http://speculaas.uva.netherlight.nl#FiberSwitchMatrix">
  <capability:hasSwitchingCapability
rdf:resource="http://www.science.uva.nl/research/sne/ndl/wdm#FiberNetworkElement" />
  <capability:hasSwappingCapability
rdf:resource="http://www.science.uva.nl/research/sne/ndl/wdm#FiberNetworkElement" />
  <ndl:hasInterface
rdf:resource="http://speculaas.uva.netherlight.nl#intf01"/>
  <ndl:hasInterface
rdf:resource="http://speculaas.uva.netherlight.nl#intf02"/>

  <ndl:hasInterface
rdf:resource="http://speculaas.uva.netherlight.nl#intf12"/>
  </capability:SwitchMatrix>
  </capability:hasSwitchMatrix>
</ndl:Device>

<ndl:Interface rdf:about="http://speculaas.uva.netherlight.nl#intf01">
  <rdfs:type
rdf:resource="http://www.science.uva.nl/research/sne/ndl/wdm#FiberNetworkElement"/>
  <!-- static FiberInterface -->
  <rdfs:label>1</rdfs:label>
  <dc:description xml:lang="en">Rembrandt0 (10GE)</dc:description>
  <wdm:ingressPowerLevel rdf:datatype="http://www.w3.org/2001/XMLSchema#float">-
3.29</wdm:ingressPowerLevel>
  <wdm:egressPowerLevel rdf:datatype="http://www.w3.org/2001/XMLSchema#float">-
53.03</wdm:egressPowerLevel>
</ndl:Interface>
```

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



GAAA toolkit pluggable components and XACML policy profile for ONRP

```
<ndl:Interface rdf:about="http://speculaas.uva.netherlight.nl#intf02">
  <rdf:type
rdf:resource="http://www.science.uva.nl/research/sne/ndl/wdm#FiberNetworkElement"/>
  <!-- static FiberInterface -->
  <rdfs:label>2</rdfs:label>
  <dc:description xml:lang="en">Rembrandt1 (10GE)</dc:description>
  <wdm:ingressPowerLevel rdf:datatype="http://www.w3.org/2001/XMLSchema#float">-
3.76</wdm:ingressPowerLevel>
  <wdm:egressPowerLevel rdf:datatype="http://www.w3.org/2001/XMLSchema#float">-
52.22</wdm:egressPowerLevel>
</ndl:Interface>
```

3.2.3 OSCARS topology description format [19]

OSCARS project uses topology description in a form of XML file that includes set of end points `srcEndpoint` and `destEndpoint` together with the path description as the list of connecting hops/links.

Endpoints and links are identified by ID's in a form of URN consisting of the following parts:

Domain (or LAN)
Node
Port
link

Example of the OSCARS resource URN:

```
urn:ogf:network:domain=blue.pod.lan:node=vlsr1:port=3:link=11.2.1.2
```

Example OSCARS-1

```
<!-- blue-es1 to blue-es2 -->
<staticPathEntry id="blue-es1-blue-es2">
  <srcEndpoint>urn:ogf:network:domain=blue.pod.lan:node=vlsr1:port=3:link=11.2.1.2</srcEndpoint>
  <destEndpoint>urn:ogf:network:domain=blue.pod.lan:node=vlsr3:port=3:link=11.2.5.1</destEndpoint>
  <path id="blue-es1-blue-es2">
    <hop id="1">
      <linkIdRef>urn:ogf:network:domain=blue.pod.lan:node=vlsr1:port=3:link=11.2.1.2</linkIdRef>
    </hop>
    <hop id="2">
      <linkIdRef>urn:ogf:network:domain=blue.pod.lan:node=vlsr1:port=5:link=11.2.3.1</linkIdRef>
    </hop>
    <hop id="3">
      <linkIdRef>urn:ogf:network:domain=blue.pod.lan:node=vlsr3:port=5:link=11.2.3.2</linkIdRef>
    </hop>
    <hop id="4">
      <linkIdRef>urn:ogf:network:domain=blue.pod.lan:node=vlsr3:port=3:link=11.2.5.1</linkIdRef>
    </hop>
  </path>
  <availableVtags></availableVtags> <!-- deprecated: leave blank -->
</staticPathEntry>
```

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



3.3 XACML Policy and attributes format

The XACML policy format provides rich functionality for Network Resource Provisioning in its core specification [14] and special profile for hierarchical resources [20], and for RBAC [21]. Hierarchical policy management and dynamic rights delegation, that are considered as important functionality in multi-domain ONRP, can be solved with the XACML v3.0 administrative policy profile [11]. XACML allows a flexible definition of authorisation rules, conditions and use of different attribute formats.

A XACML policy is defined for the target tuple “Subject-Resource-Action” (S-R-A) which can also be completed with the Environment element (S-R-A-E) to add additional context to instant policy evaluation (refer to Fig. 3.1). 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 the potential integration of the AuthZ system with logging or auditing facilities.

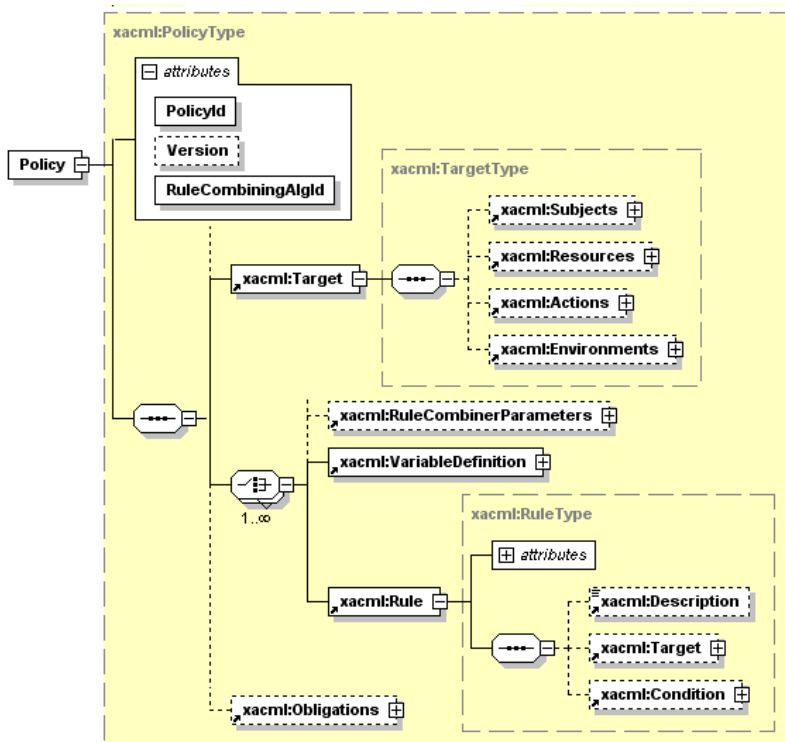


Figure 3.1. XACML Policy data model.

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.

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



GAAA toolkit pluggable components and XACML policy profile for ONRP

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 (internal) 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 can contain any of S-R-A-E attributes and a policy identification attribute IDRef. XACML policy format provides 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.

The XACML hierarchical resource profile [20] 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 also network resources and Grid jobs executing environment. The profile introduces new resource attributes identifiers that may refer to the “resource-ancestor”, “resource-parent”, or “resource-ancestor-or-self”. There may be different matching expressions 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.

Such specific use case as multi-domain ONRP requires that the resource reservation policy in each successive domain will rely on the previous domain positive AuthZ decision and it may also additionally require informing the next domain. 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 policies combination algorithms defined for the Policy Set or Policy [14].

The XACML RBAC profile [22] 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 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, a RBAC profile can significantly simplify rights delegation inside a group of collaborating entities/subjects which normally would require complex credentials management.

The XACMLv3.0 administrative policy profile [11] 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 a related administrative policy. The 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.

The XACMLv3.0 policy profile can indicate if the policy is issued by the trusted PolicyIssuer for the particular domain. In this case the PDP will rely on an 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.

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



GAAA toolkit pluggable components and XACML policy profile for ONRP

In order to use the XACML format for AuthZ in ONRP, a special XACML-NRP profile for Network Resource Provisioning described in this section addresses the following issues:

Namespace definition for the network resources, user attributes, and GAAA/AuthZ components
Attribute semantics and expression format, including support list of enumerated values, if necessary
Set of basic rules and policy templates (including possible mapping to existing policies in this domain)

To facilitate the XACML-NRP profile introduction the GAAA-TK library developed in WP4 provides a reference implementation, in particular it addresses the following issues:

namespace support, attribute formats and semantics, including enumerated values, typically provided by profile related attribute handler or helpers
support of underlying trust infrastructure/fabric.

Some general examples of the XACML policies are provided in the Appendix D.

3.4 Attributes used for Authorisation and XACML policy definition

3.4.1 Attributes namespace

The XACML-NRP attribute profile will use the AuthZ interoperability namespace of the URL style “http://authz-interop.org/” introduced in the recently released XACML-Grid profile [23]:

`http://authz-interop.org/nrp/xacml`

Note: Alternatively it can be considered to use the URN-style for proprietary or potentially to be registered namespace:

`x-urn:authz-interop:nrp:xacml`

3.4.2 Network or Resource related attributes

Network related attributes allow building policy depending on the network topology or other network characteristics. Network related attributes are considered as a part of the XACML Resource definition.

The following resource/network related attributes can be specified and used for authorisation:

Domain ID (network domain)
Subdomain (or relationship)
VLAN
Node or TNA and TNA prefix, or
Interface ID

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



GAAA toolkit pluggable components and XACML policy profile for ONRP

Device or resource-type

Link ID

Link parameters: average delay and maximum bandwidth

ReservationEPR that may directly or indirectly define the resource federation or security/administrative domain

Federation that defines a number of domains or nodes sharing common policy and attributes

| Attribute name | Attribute ID | Full XACML attributeId semantics (ns-prefix = http://authz-interop.org/nrp/xacml) | Notes |
|----------------|--------------------|--|--|
| Domain ID | domain-id | {ns-prefix} /resource/domain-id | Domain ID means full domain identification including resource realm or other information. In this way it is different from resource domain |
| Realm | resource-realm | {ns-prefix} /resource/resource-realm | Resource realm is associated with a project or profile and typically has own sub-set of attributes and policies (e.g. "testbed.ist-phosphorus.eu") |
| Domain | resource-domain | {ns-prefix} /resource/resource-domain | See note to domain ID attribute |
| Subdomain | subdomain | {ns-prefix} /resource/sub-domain | |
| VLAN | vlan | {ns-prefix} /resource/vlan | |
| TNA | tna (+ tna-prefix) | {ns-prefix} /resource/tna-prefix/tna | Consider relation between TNA, EPR and node |
| Node | node | {ns-prefix} /resource/node | |
| Link | link-id | {ns-prefix} /resource/link-id | |
| avrDelay | delay | {ns-prefix} /resource/delay | |
| maxBW | bandwidth-max | {ns-prefix} /resource/bandwidth | Consider splitting on two limits: bandwidth/(min-limit, max-limit) |
| Resource type | resource-type | {ns-prefix} /resource/resource-type ({ns-prefix} /resource/device) | Resource type may be considered as part of the resource ID by adding "resource-type/{type}" |

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



GAAA toolkit pluggable components and XACML policy profile for ONRP

| | | | |
|---------------------|------------|----------------------------------|--|
| | | | prefix/selector |
| Resource federation | federation | {ns-prefix} /resource/federation | This attribute can be considered as common for Resource and Subject because both of them can be members of the same federation of VO |
| | | | |

Note. It is expected that with wider XACML-NRP profile acceptance a number of the resource related attributes will be provided with a list of enumerated values, such as e.g. Resource type (e.g. nsp, nrps, mss) or Resource federation that typically has a limited number of the pre-configured values such as project related federations/VO, or trusted authorities.

3.4.3 Subject related attributes

Subject related attributes allow building policy depending on the properties of the request Subject or user. Subject related attributes are considered as a part of the XACML Subject definition.

The following subject related attributes can be specified:

Subject ID

Subject confirmation that contains AuthN assertion/token or other attribute confirming subject's ID by trusted AuthN authority

Subject Role

Subject Group

Subject Federation (e.g., Virtual Organisation, or Shibboleth AAI federation) or domain

Subject context that can provide additional information about the Subject other than Subject federation e.g. such as Session ID, or project/experiment name

| Attribute name | Attribute ID | Full XACML attributeld semantics (ns-prefix = http://authz-interop.org/nrp/xacml) | Notes |
|----------------------|------------------|---|---|
| Subject ID | subject-id | {ns-prefix} /subject/subject-id | Consider using standard XACML attribute ID: urn:oasis:names:tc:xacml:1.0:subject:subject-id |
| Subject confirmation | subject-confdata | {ns-prefix} /subject/subject-confdata | |
| Subject federation | federation | {ns-prefix} /subject/federation | This attribute should be considered as similar to the |

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



GAAA toolkit pluggable components and XACML policy profile for ONRP

| | | | |
|-----------------|-----------------|--------------------------------------|--|
| | | | "subject-vo" in XACML-Grid profile if network domain or endpoint are members of a VO |
| Subject group | subject-group | {ns-prefix} /subject/subject-group | |
| Subject role | subject-role | {ns-prefix} /subject/subject-role | Consider relation between TNA, EPR and node |
| Subject Context | subject-context | {ns-prefix} /subject/subject-context | |

Subject attributes are provided as Subject credentials which depending on user client implementation and middleware may take a form of X.509 public key and attribute certificates (PKC, AC), SAML Authentication and Attribute assertions, proprietary AuthN system credentials.

Subject may also use attributes defined in the XACML-Grid profile or attributes originally defined in the XACML and SAML specifications. See Appendix for such candidate attributes.

3.4.4 Action related attributes

Action related attributes represent a limited number of the specific actions that requesting party can ask to initiate network resource reservation, access or management. Action related attributes are considered as a part of the XACML Action definition.

The following Action related attributes can be specified:

Action ID
Action type

| Attribute name | Attribute ID | Full XACML attributeId semantics (ns-prefix = http://authz-interop.org/nrp/xacml) | Notes |
|----------------|--------------|---|---|
| Action ID | action-id | {ns-prefix} /action/action-id | Consider using standard XACML attribute ID: urn:oasis:names:tc:xacml:1.0:action:action-id |
| Action type | action-type | {ns-prefix} /action/action-type/{value} | This type of attribute identifier is used only together with the enumerated action identifier |
| | | | |

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



GAAA toolkit pluggable components and XACML policy profile for ONRP

List of action type enumerated values:

| Attribute name | Enumerated value | XACML attribute value (ns-prefix = http://authz-interop.org/nrp/xacml) | Notes |
|----------------|------------------|--|-------|
| Action type | create-path | {ns-prefix} /action/action-type/create-path | |
| | activate-path | {ns-prefix} /action/action-type/activate-path | |
| | cancel | {ns-prefix} /action/action-type/cancel | |
| | access | {ns-prefix} /action/action-type/access | |
| | | | |

3.4.5 Environment related attributes

Environment related attributes allow providing additional information for policy definition and evaluation.

Environment related attributes are considered as a part of the XACML Environment definition.

There is no specific Environment attributes identified for XACML-NRP profile at this moment. Refer to Appendix for list of the Environment attributes specified in XACML2.0 specification and XACML-Grid profile.

3.4.6 Policy Obligations used in NRP

Policy obligation is one of the authorisation policy enforcement mechanisms that allows adding AuthZ decision enforcement components that can not be defined in the policy at the moment of making policy decision by the PDP, or may not be known to the PDP or policy administrator/writer

Suggested functionality that can be achieved with using obligations includes but not limited to:

Intradomain network/VLAN mapping for cross-domain connections, that can be used to map external/interdomain border links/TNA's to internal VLAN and sub-network

Account mapping

Type of service (or QoS) assigned to a specific request or policy decision

Quota assignment

Service combination with implied conditions (e.g., computing and storage resources)

Usable resources/quota

Below text provides current suggestions for the obligations definition. More details will be provided with wider use and acceptance of the XACML-NRP profile.

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



3.4.6.1 Intradomain network/VLAN mapping

This may be needed for defining specific intra-domain mapping of cross-domain connections depending on specific reservation, path or user attributes.

3.4.6.2 Network user identity mapping

This obligation is returned by the PDP in case of positive decision with instruction to what type of or a specific pool account the user identity should be mapped when accessing a requested network resource.

The need of account mapping may exist in cases when domain based Network Resource Provisioning Systems (NRPS) have pre-installed/built-in pool accounts to which are different types or quality of service are assigned. In such situation authorised user need to use one of such accounts, e.g. "silver", "golden", "platinum". A number of different individual accounts of the same type may be limited, consequently a dynamically assigned account should be selected from the pool of available or free accounts. Such dynamic account assignment can not be specified in the typically stateless policy and cannot be done by PDP. However, the access control policy may contain instruction to PEP to do such mapping.

3.4.6.3 Usability and accounting

Usability and accounting obligations allow that some usability attributes (e.g. number of downloads, total time of using network resource, amount of data transferred) assigned or accounting instruction are applied to the specific request decision.

3.5 Policy Expression Conventions

This section describes the current policy expression conventions for the basic Phosphorus use cases described in section 3.1.

The simple policy example for the specified at the beginning use case 1 will evaluate the request to NSP system for one of actions "create-path, activate-path, cancel, access" from the user with attributes "admin, researcher, professor, or student". Access control table has the following view:

Access Control Table for simple NSP AuthZ policy

| Roles | Admin | Researcher | Professor | Student |
|---------------|-------|------------|-----------|---------|
| Create-Path | 1 | 1 | 0 | 0 |
| Activate-Path | 1 | 1 | 1 | 0 |
| Cancel | 1 | 0 | 0 | 0 |
| Access | 1 | 1 | 1 | 1 |

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



GAAA toolkit pluggable components and XACML policy profile for ONRP

The example in the Appendix A illustrates how such access control table can implemented in XACML using the proposed attributes definition.

3.6 Policy identification and policy resolution

3.6.1 General suggestions

When evaluating AuthZ request the ContextHandler or PDP need to find/select an applicable policy. This is typically done based on the request parameters such as Resource or Subject attributes.

The policy selecting/finding comprises of two steps: policy resolution and policy retrieval. Policy resolution means extracting such information from the AuthZ request that can be used for further policy selection in the storage/repository. Based on this information, a repository request or query can be constructed to retrieve necessary policy.

Note, it is a SunXACML implementation convention that only one Policy or PolicySet should be supplied to PDP for evaluation, and only one component Policy must be selected if using PolicySet.

The following components of the XACML-NRP profile can be used for policy resolution:

- resource ID and resource attributes;
- subject attributes defining context in which the request should evaluated, e.g. project or VO (this information is typically a part of the subject attributes);
- attributes and policy profile namespace, which can actually be a part of the resource ID if expressed in Fully Qualified Attribute Name format (FQAN format).

Depending on the policy storage/repository implementation, the following components can be used for policy identification:

- policy file name and directory, if policy is stored as a file;
- PolicyId attribute of the PolicySet or Policy element;
- policy Target element that can include any of Subject, Resource, Action, Environment elements.

Although using basically different ways of storing policies, the first case and second identification methods can be based on similar approach to composing PolicyId attribute and (defining) policy file location path. When using third option, the policy repository should be capable to query policy database by the policy Target content.

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



3.6.2 Policy resolution convention in the GAAA-TK library

Two components are used for policy resolution in GAAA-TK library profile for Network Resource Provisioning (GAAA-NRP):

ResourceId expressed in the URL-style URI format, actually specifying the resource FQAN;
Subject context (SubjectCtx) that specifies subject (and resource) association (e.g., VO, experiment, or project)

General ResourceId expression format:

```
<<ns-type>><Realm>/<ns-domain>/(<device-type> | <resource-context>)/<variables-name-value-pairs>
```

Examples of the URL style:

```
http://testbed.ist-phosphorus.eu/viola/nsp/source=10.1.1.3/target=10.3.1.3
http://testbed.ist-phosphorus.eu/resource-type/harmony
http://testbed.ist-phosphorus.eu/resource-context/phosphorus
```

where

```
"http://" - URL style namespace identifier;
"testbed.ist-phosphorus.eu" - namespace realm;
"viola", "resource-type", "resource-context" - namespace domain;
"nsp", "harmony", "phosphorus" - device type or resource context;
"source", "target" - device variable presented in a form of "name-value".
```

Note: quotation marks are not allowed in URI string.

The same identifier strings expressed in URN style will use URN specific prefix "x-urn:authz-interop:nrp:resource-id" (or its shorter option "x-urn:nrp:resource-id"):

```
x-urn:authz-interop:nrp:resource-id:testbed.ist-phosphorus.eu:viola:nsp:source=10.1.1.3;target=10.3.1.3
x-urn:authz-interop:nrp:resource-id:testbed.ist-phosphorus.eu:resource-type:harmony
x-urn:authz-interop:nrp:resource-id:testbed.ist-phosphorus.eu:resource-context:phosphorus
```

SubjectCtx can be expressed either in a FQAN format or just contain a simple name without namespace prefixes, e.g. Demo001, or PhosphorusVO, EGEE-VO, etc.

Given above examples the AuthZ Request containing ResourceId and SubjectCtx attributes will be resolved into the following policy path/files:

```
<<root-dir>>/policy/nrp/testbed.ist-phosphorus.eu/viola-policy-nsp-demo001.xml
<<root-dir>>/policy/nrp/testbed.ist-phosphorus.eu/harmony-policy-demo001.xml
<<root-dir>>/policy/nrp/testbed.ist-phosphorus.eu/phosphorus-policy-demo001.xml
```

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



3.6.3 Policy identification

It is suggested that the PolicyId or PolicySetId is created in the same way using typical for URL/URN style conventions:

```
PolicyId = <<url-namespace-prefix/>>testbed.ist-phosphorus.eu/viola/harmony/demo001/policy  
PolicyId = <<urn-namespace-prefix:>>testbed.ist-phosphorus.eu:viola:harmony:demo001:policy
```

where

```
<<namespace-prefix>> - can be dropped;  
namespace-prefix = http://authz-interop.org/nrp/xacml  
or namespace-prefix = x-urn:authz-interop.org:nrp:xacml
```

Example PolicyId expression:

URL style:

```
PolicyId = http://authz-interop.org/nrp/xacml/testbed.ist-phosphorus.eu/phosphorus/demo001/policy  
PolicyId = http://testbed.ist-phosphorus.eu/viola/harmony/demo001/policy  
PolicyId = http://testbed.ist-phosphorus.eu/phosphorus/demo001/policy
```

URN style:

```
PolicyId = x-urn:authz-interop.org:nrp:testbed.ist-phosphorus.eu:viola:harmony:demo001:policy  
PolicyId = x-urn:authz-interop.org:nrp:testbed.ist-phosphorus.eu:phosphorus:demo001:policy
```

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



4 GAAA Toolkit Library

This section provides information about the current implementation of the basic GAAA-NRP functionality in the pluggable GAAA Toolkit library.

4.1 GAAA-TK library components

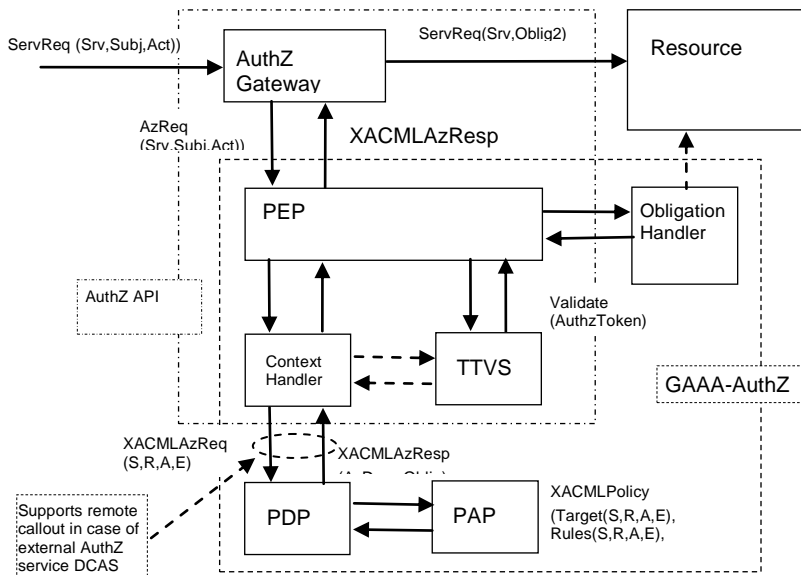
The library provide the major functionality and mechanisms identified and described in the D4.1 report such as a Token Validation Service (TVS), authorisation/provisioning session security context management and access control with AuthZ tickets and tokens, policy obligations handling with obligations handlers supporting the Obligation Handling Reference Model (OHRM), and the XACML policy and attributes profile for OLPP.

Figure 4.1 illustrates the major GAAA-AuthZ modules and how they interact when evaluating a service request, it is based on the Fig. 2.2 but provides more details about internal structure of the ContextHandler module that supports all necessary communications and interaction between PEP and PDP and with external services if additional information is needed for the policy evaluation.

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



GAAA toolkit pluggable components and XACML policy profile for ONRP



Note: This figure will be updated with more details about Context handler and related modules.

Figure 4.1. GAAA-AuthZ library components providing service request evaluation.

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. 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 which semantics is actually defined by the used policy. 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, attributes, (environment)). The Policy Decision Point (PDP) evaluates the request and makes a decision whether to grant access or not.

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 may be processed as a security context or policy evaluation 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).

The service request may contain AuthZ ticket that hold extended AuthZ session context or AuthZ token that can just reference a local or global reservation ID, or identify an AuthZ session in which context the request is sent. The AuthZ 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. TVS is introduced as a separate function or

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



GAAA toolkit pluggable components and XACML policy profile for ONRP

service to allow creating flexible token based policy enforcement infrastructures for on-demand network resource provisioning and possibly use it as a pluggable component to other AuthZ frameworks.

4.2 General AAA/AuthZ API and programming examples

GAAAPI package provides all necessary functionality to smoothly integrate AAA/AuthZ services into target application. GAAAPI package is provided together with the PEP implementation and simple XACML PDP implementation.

PEP-GAAAPI is called from the application AuthZ gateway that extracts necessary information from the service request and creates AuthZ request to PEP. GAAAPI functionality supports all necessary communication between PEP and PDP and depending on implementation may include also external callout to such components as PDP, PAP, Attribute Authority Service (AAS), TVS, and Obligation Handlers (OH).

Note: Currently available GAAAPI implementation supports only local call to internal/local components. Further GAAAPI development will include external callouts to domain or site central AAA/AuthZ services.

4.2.1 PEP-GAAAPI interface

PEP-GAAAPI interface provides a few commands/methods to request policy based AuthZ decision depending on the set of provided information:

a) Method #1 should either return a logical value "True" or "False", or throw the appropriate exception

```
Boolean org.aaaarch.gaaapi.PEP.authorizeAction
    (String resourceURI, String actions, HashMap subjmap)
throws java.lang.Exception,
org.aaaarch.gaaapi.NotAuthenticatedException,
/* user subjconfdata (i.e. authenticationToken) is not valid */
org.aaaarch.gaaapi.NotAvailablePDPEXception;
/* PEP could not reach PDP, or other internal PDP error*/
```

where

```
@ resourceURI - Resource ID in a form of URI
@ actions - requested actions (currently supported only one action)
@ {subjmap} set of values (subject-id, subject-confdata, subject-role, subject-context)
@ subject-id - subject Id in form of RFC822
@ subject-confdata - AuthN token or SAML AuthN assertion
@ subject-role - role for the particular request (may be in a form either simple
    attribute or RQAN)
@ subject-context - subject context, e.g. Experiment, VO, or VLab in which
    the subject and resource attributes are defined
```

Note: This method uses complex resource URI that may consist of ResourceId part and additional parameters in a form of "name=value" pairs (see section 4.2.4 for attribute expression conventions).

b) Method #2 should either return a logical value "True" or "False", or throw the appropriate exception

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



GAAA toolkit pluggable components and XACML policy profile for ONRP

```
Boolean org.aaaarch.gaaapi.PEP.authorizeAction
    (HashMap resmap, HashMap actmap, HashMap subjmap)
throws java.lang.Exception,
org.aaaarch.gaaapi.NotAuthenticatedException,
org.aaaarch.gaaapi.NotAvailablePDPEException;
```

where

```
@ resmap - set of the Resource related attributes in a form of HashMap
@ resmap = (resource-id, resource-domain, resource-type) and other resource
related attributes
@ actmap - requested actions (currently supported only one action)
```

c) Method #3 should either return a logical value "True" or "False", or throw the appropriate exception

```
Boolean org.aaaarch.gaaapi.PEP.authorizeAction
    (String resourceId, String actions, String subjectId, String subjconfdata,
String roles, String subjctx)
throws java.lang.Exception,
org.aaaarch.gaaapi.NotAuthenticatedException,
org.aaaarch.gaaapi.NotAvailablePDPEException;
```

Note: This method uses simple resource ID format. All additional parameters will be ignored and not used for policy resolution.

d) Method #4 should either return a logical value "True" or "False", or throw the appropriate exception

```
Boolean org.aaaarch.gaaapi.PEP.authorizeAction
    (String authzToken, HashMap resmap, HashMap actmap, HashMap subjmap)
throws java.lang.Exception,
org.aaaarch.gaaapi.NotAuthenticatedException,
org.aaaarch.gaaapi.NotAvailablePDPEException;
```

where

```
@ authzToken - access token in a form of XMLToken
```

e) Method #5 should either return a valid AuthorisationTicket or AuthorisationToken (refer to section 2 and Appendix for AuthzTicket and AuthzToken format and examples), or throw the appropriate exception

```
String org.aaaarch.gaaapi.PEP.authorizeAction
    (String authzTicketToken, String sessionId, String resourceURI,
String actions)
throws java.lang.Exception,
org.aaaarch.gaaapi.NotAuthenticatedException,
org.aaaarch.gaaapi.NotAuthorizedException,
org.aaaarch.gaaapi.NotAvailablePDPEException;
```

were

```
@ authzTicketToken - AuthZ ticket or token containing all necessary AuthZ session context
@ sessionId - Session ID that can be also a Global or Local reservation ID (LRI/GRI)
```

d) Method #6 should either return a valid AuthorisationTicket or AuthorisationToken (refer to section 2 and Appendix for AuthzTicket and AuthzToken format and examples), or throw the appropriate exception

```
String org.aaaarch.gaaapi.PEP.authorizeAction
    (String authzTicketToken, String sessionId, String resourceURI,
String actions, HashMap subjmap)
throws java.lang.Exception,
org.aaaarch.gaaapi.NotAuthenticatedException,
```

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



GAAA toolkit pluggable components and XACML policy profile for ONRP

```
org.aaaarch.gaaapi.NotAuthorizedException,  
org.aaaarch.gaaapi.NotAvailablePDPEException;
```

4.2.2 Simple XACML PDP API

The main GAAA-TK use suggests that the XACML PDP is requested via PEP that converts the application specific AuthZ request to XACML Request format. However, it is possible to request the XACMLPDPsimple class directly via the following API:

a) Method #1 return XACML Response message as String, or throws an exception

```
String org.aaaarch.gaaapi.impl.pdp.XACMLPDPsimple.requestPDP  
(RequestCtx request, String policyref)  
throws java.lang.Exception
```

a) Method #1 return XACML Response message as String, or throws an exception

```
String org.aaaarch.gaaapi.impl.pdp.XACMLPDPsimple.requestPDP  
(String requestStr, String policyref)  
throws java.lang.Exception
```

were

@ request, or requestStr - XACML request in a form of the XACML RequestCtx or String
@ policyref - path to policy file location

Examples of the XACML Request/Response messages and corresponding XACML policy are provided in Appendix:

4.2.3 GAAA-PEP API programming examples

1) Preparing PEP request data

Two types of data are used as input to PEP.authoriseAction methods – string variables and HashMap attributes set. The example below illustrates how to put String variables into HashMap and how to extract individual attribute from HashMap.

```
HashMap<String, String> subjmap = new HashMap<String, String>();  
HashMap resmap = new HashMap();  
HashMap actmap = new HashMap();  
  
// Obtaining test set of the subject attributes  
subjmap = SubjectSet.getSubjSetTest();  
// extracting subject attrs from the subjmap  
String subjectId = subjmap.get(ConstantsNS.SUBJECT_SUBJECT_ID).toString();  
String subjconfdata = subjmap.get(ConstantsNS.SUBJECT_CONFDATA).toString();  
String roles = subjmap.get(ConstantsNS.SUBJECT_ROLE).toString();  
String subjctx = subjmap.get(ConstantsNS.SUBJECT_CONTEXT).toString();  
// modifying subjctx for experiments  
  
//Example Subject attributes  
String subjectId = "WHO740@users.collaboratory.nl";  
String subjconfdata = "SeDFGVHYTY83ZXxEdsweOP8Iok";
```

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



GAAA toolkit pluggable components and XACML policy profile for ONRP

```
String roles = "researcher";
String subjctx = "demo001";

//Putting/replacing Subject attributes into subjmap
subjmap.put(ConstantsNS.SUBJECT_ROLE, subjrole);}
subjmap.put(ConstantsNS.SUBJECT_SUBJECT_ID, "WHO750@users.testbed.ist-phosphorus.eu");

//Obtaining resource map from the ResourceHelper class
resourceInputURI = "http://testbed.ist-phosphorus.eu/viola/harmony/source=10.3.1.16/target=10.7.3.13";

resmap = ResourceHelper.parseResourceURI(resourceInputURI);

// Putting action attributes into actmap
actmap.put(ConstantsNS.ACTION_ACTION_ID, action);
```

Note this example uses Subject, Resource, Action constants that define attributes name/ID's in correspondence to the XACML-NRP profile

2) Requesting PEP with (Subject, Resource, Action) information (using method#1 and method#2)

The following two examples explain how to call PEP method #1 and method #2 using prepared data like above.

```
// Calling PEP method #1
boolean decision = PEP.authorizeAction (resourceInputURI, actions, subjmap);

// Calling PEP method #2
boolean decision = PEP.authorizeAction (resmap, actmap, subjmap);
```

2) Requesting PEP with (Subject, Resource, Action) information and XMLToken obtained from TVS (using method #4)

To enable XML token operation, the following steps need to be performed (this a combined use of the PEP and TVS functionality explained in details below):

- a) obtain positive decision from PEP-PDP (see examples 1 and 2 above)
- b) save session/reservation context in the TVS table
- c) generate XML token that contains GRI (global reservation identifier) and token value generated cryptographically of GRI and the domain token key
- d) present this token in all consequent AuthZ requests to PEP. In this case PEP will request AuthZ request evaluation with TVS, TVS will retrieve session context from TVS table and compare it with the request context without requesting PDP.

```
// Obtaining or setting domain ID information
String domainViola = ConfigDomainsPhosphorus.DOMAIN_PHOSPHORUS_VIOLA;
String domainId = domainViola;
// Obtaining sessionId if not received with the pilot token
String griprefix = "";
String sessionId = GRIgenerator.generateGRI(32, griprefix);
```

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



GAAA toolkit pluggable components and XACML policy profile for ONRP

```
// Composing session context vector
Vector sessionCtx = TVS.getSessionCtxVector (domainId, gri, resmap, actmap, subjmap);

/* If it is necessary, the TVSTable can be purged completely of for a selected domain
 * Use this method
 * TVSTable.purgeTVSTable(null, 0);
 */

// Adding a new TVS entry
TVS.setEntryTVSTable(domainId, gri, sessionCtx);

// checking TVS table content
String tablefile = TVS.getTVSTableFile ();
Document tabdoc = HelpersXMLsecurity.readFileToDOM(tablefile);
HelpersXMLsecurity.printDOMdoc(tabdoc); // pring TVSTable context if you want

// Generating XMLToken (type 0)
// Set or obtain necessary variable for XMLToken generation
Boolean simple = false; // simple token format doesn't contain time validity

Int validtime = TVS.getConfigValidityTimeDefault();
Int validtime = 1440; // validity time is 24 hrs

String tokenxml = TokenBuilder.getXMLToken(domainId, gri, null, validtime, simple);

//Request PEP with XMLToken (method #4)

boolean decision = PEP.authorizeAction (tokenxml, resmap, actmap, subjmap);
```

4.2.4 Attribute expression conventions

Information provided in the AuthZ request to PEP-PDP contains information about Resource, Subject, Action, and optionally Environment.

a) Resource attributes

Currently the Resource variable in the AuthZ request contains one attribute ResourceId in the form of URI string that includes the network resource identifier and a list of parameter used for policy-based request evaluation. When sending a XACML Request to XACML PDP the input URI string is converted into the HashMap `resmap` that contains a set of resource related attributes. The names for some of the relevant resource attribute identifiers are taken from the XACML-NRP profile, such as “resource-id”, “resource-domain”, “resource-realm”, “resource-type”, “source”, “target”, etc., however it is responsibility of the application developer to correctly format the ResourceId URI string.

The following ResourceId formats are supported:

a) `http://testbed.ist-phosphorus.eu/{domain}/{device | service}/{parameters}`

Example: `http://testbed.ist-phosphorus.eu/viola/harmony/source=10.7.12.2/target=10.3.17.3,`

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



GAAA toolkit pluggable components and XACML policy profile for ONRP

This URI will be converted into the following set of attributes and put into the resmap:

```
resource-id = http://testbed.ist-phosphorus.eu/viola/harmony
resource-realm = http://testbed.ist-phosphorus.eu
resource-domain = viola
resource-type = harmony
source = 10.7.12.2
target = 10.3.17.3
```

b) `http://testbed.ist-phosphorus.eu/resource-type/{resource-type-name}`

Example: `http://testbed.ist-phosphorus.eu/resource-type/harmony`

c) `http://testbed.ist-phosphorus.eu/resource-context/((project | association) name) (optional)`

Example: `http://testbed.ist-phosphorus.eu/resource-context/phosphorus`

b) Subject attributes

The Subject variable of the AuthZ request contains the following attributes (which are either sent to PEP separately or put into the subjmap):

a) SubjectId (attribute identifier “subject-id”) – subject identifier in RFC822 (email) or X.521 (LDAP or X.509 Public Key Certificate) formats (must be the same as used in the SubjectConfirmatioData)

Example: `WHO740@users.testbed.ist-phosphorus.eu`

b) SubjectConfirmatioData (attribute identifier “subject-confdata”) – Authentication assertion or token provided by the trusted AuthN service (can be also SAML AuthN Assertion), or crypto-string provided local AuthN service.

c) SubjectRole (attribute identifier “subject-role”) – subject role, currently supporting single value.

Example: `admin`, or `researcher@project01`, or `admin@viola.testbed.ist-phosphorus.eu`

Future GAAA-TK release will support coma-separated list of roles, SAML Attribute assertion and VOMS Attribute Certificate

d) SubjectContext (attribute identifier “subject-context”) – this attribute is used for providing additional information about a user (and a resource) association like VO, project, experiment/job.

Example: `demo001`; or `VO-Phosphorus`

Potentially this attribute can be extended to provide instant reservation context for dynamically configured AuthZ service.

c) Action attributes

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



GAAA toolkit pluggable components and XACML policy profile for ONRP

The Action element contains single attribute/value that defines requested action in a form of simple string attribute or using fully qualified name for some enumerated action ID's like proposed in the XACML-NRP profile.

Note. It is important to note that corresponding attributes in the AuthZ request and in the policy must use the same attribute names/ID's and format.

4.3 TVS API

4.3.1 TVS interface

a) Token Builder commands

```
public static byte[] TokenBuilder.getBinaryToken(String gri, byte[] tokenkey)

public static String TokenBuilder.getXMLToken(String domainId, String gri, byte[]
tokenKey, int validtime, boolean simple)

public static String TokenBuilder.getXMLTokenPilot(String domainId, String gri, String
domain, int validtime, byte[] tokenKey, int ptokentype, String tokenCtx)
```

b) TVS token validation interface - validates the binary or XML token themselves;

```
public static boolean validateBinaryToken (String token, String gri, byte[] tokenKey)
    throws Exception

public static boolean validateXMLToken (Document aztdoc, byte[] tokenKey)
    throws Exception,
    MalformedXMLTokenException,
    NotValidAuthzTokenException

public static boolean validateXMLToken (String authzToken, byte[] tokenKey)
    throws Exception,
    MalformedXMLTokenException,
    NotValidAuthzTokenException
```

c) PEP-TVS interface: is called from PEP and validates AuthZ Request (resmap, actmap, subjmap) against XML token;

```
public static boolean validateAuthzRequestByToken (String authzToken,
    HashMap resmap, HashMap actmap, HashMap<String, String> subjmap)
    throws Exception,
    MalformedXMLTokenException,
    NotValidAuthzTokenException
```

d) Internal TVS programming interface use the following basic commands:

```
TVS.setEntryTVSTable(String domainId, String gri,
```

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



GAAA toolkit pluggable components and XACML policy profile for ONRP

```
HashMap resmap, HashMap actmap, HashMap subjmap)

TVS.getEntryTVSTable(String domainId, String gri)

TVS.deleteEntryTVSTable(String domainId, String gri)

public static boolean purgeTVSTable (String domainId, int expireTime)
```

Note, that TVS programming calls will be exposed as We Services.

e) External/WS TVS programming interface

External TVS interface will allow programming TVS table by sending particular reservation information in a SOAP message.

4.3.2 TVS programming examples

1) Generating binary token

To request token generation from the calling application, use these commands/methods:

```
// Prepare data for token generation
String gri = "".concat(org.aaaarch.gaaapi.common.IDgenerator.generateID(20).toString());
byte[] tokenkey = TokenKey.generateTokenKey(gri);

byte[] token = TokenBuilder.getBinaryToken(gri, null);
```

2) Generating XML token

```
// Set parameter for token generation
boolean simple = true; // simple token doesn't contain Conditions element
int validtime = 0; // this variable sets token validity in minutes; default is 24 hrs
String domainId = "http://testbed.ist-phosphorus.eu/viola/harmony";
//
String gri = GRIgenerator.generateGRI(20).toString();

String tokenxml = TokenBuilder.getXMLToken(domainId, gri, null, validtime, simple);
```

3) Validating binary token

```
boolean valid = TVS.validateBinaryToken (token, gri, null);
```

4) Validating XML token itself. Two methods can be used to validate full token validity and time validity:

```
XMLTokenType token = new XMLTokenType (tokendoc);
// Checking token validity time
boolean timevalid = token.isTimeValid(token);

// checking token validity
boolean valid = TVS.validateXMLToken (tokendoc, null);
```

4) Validating AuthZ request context against XML token

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



GAAA toolkit pluggable components and XACML policy profile for ONRP

```
//Prepare input data for requesting TVS

// If XMLToken is received as String
Document tokendoc = HelpersXMLSecurity.readStringToDOM(String) (tokenString);

XMLTokenType token = new XMLTokenType (tokendoc);

// Simple token time validity check
boolean timevalid = token.isTimeValid(token);

// Validating token against stored in TVS table session context
TVS.validateXMLToken(tokendoc, null);

// Validating AuthZ request against XML token and stored session context
boolean confirmed = TVS.validateAuthzRequestByToken (aztstr, resmap, actmap, subjmap);
```

4.4 Authorisation ticket and token examples

4.4.1 Authorisation ticket examples

GAAAPI supports AuthZ tickets (and additionally AuthN tickets) generation in a proprietary XML format and by using the SAML assertion format. AuthZ ticket format was discussion in section 2.3. Examples of AuthZ tickets are provided in the Appendix B.

4.4.2 TVS XML Token format and examples

Refer to the token data model and token types definition in section 2.4

a) Access token (type 0)

XML token format uses a special “TVS/TBN” profile of the more general AuthzToken format. Example of the full TVS XML token is shown below:

```
<AAA:AuthzToken xmlns:AAA="http://www.aaauthreach.org/ns/#AAA"
  Issuer="urn:aaa:gaaapi:token:TVS"
  SessionId="a9bcf23e70dc0a0cd992bd24e37404c9e1709afb"
  TokenId="d1384ab54bd464d95549ee65cb172eb7">
  <AAA:TokenValue>ebd93120d4337bc3b959b2053e25ca5271a1c17e</AAA:TokenValue>
  <AAA:Conditions NotBefore="2007-08-12T16:00:29.593Z"
    NotOnOrAfter="2007-08-13T16:00:29.593Z"/>
</AAA:AuthzToken>
```

where the element <TokenValue> and attributes SessionId and TokenId are mandatory, and the element <Conditions> and attributes Issuer, NotBefore, NotOnOrAfter are optional.

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



GAAA toolkit pluggable components and XACML policy profile for ONRP

Minimum token format is illustrated in the following example:

```
<AAA:AuthzToken xmlns:AAA="http://www.aaauthreach.org/ns/#AAA"  
  SessionId="a9bcf23e70dc0a0cd992bd24e37404c9e1709afb"  
  TokenId="d1384ab54bd464d95549ee65cb172eb7">  
  <AAA:TokenValue>ebd93120d4337bc3b959b2053e25ca5271a1c17e</AAA:TokenValue>  
</AAA:AuthzToken>
```

Attributes "Issuer" allow for distinguishing different AuthzToken profiles. The TVS profile is identified by the URN "urn:aaa:gaaapi:token:TVS".

b) Pilot token type 1

The pilot token type 1 is used just as a container for communicating GRI during the reservation stage. It contains mandatory SessionId attribute and optional Condition element (it doesn't contain TokenValue element).

```
<AAA:AuthzToken xmlns:AAA="http://www.aaauthreach.org/ns/AAA"  
  Issuer="http://testbed.ist-phosphorus.eu/phosphorus/aaa/TVS"  
  SessionId="7172533c4e83ae7f19c13e015e07e244bb986dee"  
  TokenId="cc99a687df8ef6aeb661e6579f8f209b">  
</AAA:AuthzToken>
```

c) Pilot token type 2

The pilot token type 2 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 of GRI by applying e.g. HMAC function to the GRI together with the requestor symmetric secret or private key.

```
<AAA:AuthzToken xmlns:AAA="http://www.aaauthreach.org/ns/AAA"  
  Issuer="http://testbed.ist-phosphorus.eu/phosphorus/aaa/TVS/token-pilot"  
  SessionId="7f841d4ec3e802fd7852a8db35906abaff53f79a"  
  TokenId="b5e4f3386bdfef7c9ff9f76e67d30957" type="pilot-type2">  
  <AAA:TokenValue>2e31717173e63002360294a9175388c3138299f0</AAA:TokenValue>  
  <AAA:Conditions NotBefore="2008-07-19T22:59:40.281Z"  
    NotOnOrAfter="2008-07-20T22:59:40.281Z"/>  
</AAA:AuthzToken>
```

d) Pilot token type 3

The pilot token type 3 extends the type2 with Domains element that allows to collect domains security context information (in the Domains/Domain element) when passing multiple domains during the reservation process. Such information includes the previous token and the domain's trust anchor or public key.

```
<AAA:AuthzToken xmlns:AAA="http://www.aaauthreach.org/ns/AAA"  
  Issuer="http://testbed.ist-phosphorus.eu/phosphorus/aaa/TVS/token-pilot"  
  SessionId="0912182e7f9c7d156028e77e3d6b460de8e4937c"  
  TokenId="a99b91e70307bdd329c9a0aec18bb4a3" type="pilot-type3">  
  <AAA:TokenValue>3923c7ecb979e7078ab8745191a7b25348cdcb48</AAA:TokenValue>  
  <AAA:Conditions NotBefore="2008-07-25T09:38:39.890Z"  
    NotOnOrAfter="2008-07-26T09:38:39.890Z"/>  
  <AAA:DomainsContext>
```

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



GAAA toolkit pluggable components and XACML policy profile for ONRP

```
<AAA:Domain domainId="http://testbed.ist-phosphorus.eu/viola">
  <AAA:AuthzToken Issuer="http://testbed.ist-phosphorus.eu/viola/aaa/TVS/token-pilot"
    SessionId="b0b6202d7bd7fb7b591b5de29950d21fdb8bf375"
    TokenId="e7c88fad8cff42d7faaa961b96411ae6">
    <AAA:TokenValue>f09194bbddeef95bc4acb187f71b0bb20b2d0b44</AAA:TokenValue>
    <AAA:Conditions NotBefore="2008-07-18T21:55:15.296Z"
      NotOnOrAfter="2008-07-18T21:55:15.296Z"/>
  </AAA:AuthzToken>
  <AAA:KeyInfo>http://testbed.ist-phosphorus.eu/viola/_public_key_</AAA:KeyInfo>
</AAA:Domain>
</AAA:DomainsContext>
</AAA:AuthzToken>
```

4.5 Simple XACML policy generation tools

GAAA-TK library provides simple XACML policy generation tool that allows to automatically generate XACML policy for few predefined logical models. Currently XACMLPolicyMaker class provided as a part of the org.aaaarch.policy.utils package supports two basic policy models and their combination: RBAC policy model and policy controlling TNA range. More complex policies will require manual policy writing e.g. using any text or XML editor such as XMLSpy.

It is suggested that when more policy enforcement use cases will be defined, new policy models will be included into the XACMLPolicyMaker options. Further XACMLPolicyMaker development will allow reading and editing policies in a simple way. However it is not a scope of current GAAA-TK development to create a full policy editor.

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



5 GAAA-TK library Installation and configuration

This section provides basic information about configuration parameters and the GAAA-TK library can be installed and integrated into the main application that needs to be protected by the AuthZ service.

5.1 Configuration

GAAAPI/TVS installation requires configuration of a few folders that contain a keystore or used as a temporal directories when processing AuthZ session credentials.

The following directories are used in current implementation and can be configured via the ConfigSecurity.java class (currently hard coded):

```
LOCAL_DIR_ROOT = "" - GAAAPI installation directory

LOCAL_DIR_SECURITYCONFIG = LOCAL_DIR_ROOT + "data/config/";
LOCAL_DIR_KEYSTORE = LOCAL_DIR_ROOT + "etc/security/keystore/";
LOCAL_DIR_KEYSTORE_TRUSTED = LOCAL_DIR_KEYSTORE + "trusted/";
LOCAL_DIR_SYMKEYSTORE = LOCAL_DIR_KEYSTORE + "cnlsec/symkeystore/";
LOCAL_DIR_KEYSTORE_IBC = LOCAL_DIR_KEYSTORE + "ibc/";
LOCAL_DIR_POLICY = LOCAL_DIR_ROOT + "data/policy/";
LOCAL_DIR_SCHEMAS = LOCAL_DIR_ROOT + "data/schemas/";
LOCAL_DIR_AAADATA_CACHE_AZTICKETS = LOCAL_DIR_ROOT + "_aaadata/cache/aztickets/";
LOCAL_DIR_AAADATA_TMP = LOCAL_DIR_ROOT + "_aaadata/tmp/";
```

Note. Provided GAAA-TK package contains all necessary directories structure and also crypto keys. TVS shared secret is hard coded into the token building classes.

```
<installation-root>
+-- data
|   +-- config
|   +-- docs
|   +-- policy
|       +-- nrp
|           +-- testbed.ist-phosphorus.eu
|   +-- schemass
```

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



GAAA toolkit pluggable components and XACML policy profile for ONRP

```
| +-- sql
+-- +-- wsdl
+-- etc
| +-- security
|     +-- keystore
|     +-- ibc
|     +-- trusted
|     +-- xmlsec
|     +-- symkeystore
+-- gaaa-bin
| +-- gaaapi-nrp-v0*-release-date*.jar
+-- gaaa-lib
| +-- endorsed
| +-- lib-ibc
+-- x-output
+-- _aadata
    +-- cache
        | +-- aztickets
        | +-- sessions
    +-- tmp
```

where `gaaapi-nrp-v0*-release-date*.jar` is the GAAA-TK library of the recent release (should be checked at the WP4 wikipedia http://www.ist-phosphorus.eu/wiki/index.php/Pluggable_GAAA-TK_library).

5.2 Installation

Current GAAA-TK library requires manual installation.

The installation package consists of the 3 archives:

`gaaa-tk-lib-external-libraries.zip` – all required libraries including GAAA-TK library itself.

`gaaa-tk-lib-directories.zip` – all necessary supporting directories

`gaaa-tk-lib-test-caesses.zip` – test classes that contains examples how to call the library functions.

Installation procedure is simple. To install GAAA-TK library, you need to unpack provided archives into the selected `<root-directory>` from which the GAAA-TK functions will be run.

To run GAAA-TK library functions, use programming examples described in section 4.

5.3 Required external libraries

The list of currently used libraries to support core GAAAPI and TVS functionality:

```
bcprov-jdk15-130.jar
commons-codec-1.3.jar
commons-logging-1.0.3.jar
commons-logging-api.jar
dom3-xercesImpl-2.5.0.jar
dom3-xml-apis-2.5.0.jar
jaxrpc-1_1-fr-spec-api.jar
jaxrpc-sec.jar
```

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



GAAA toolkit pluggable components and XACML policy profile for ONRP

```
joda-time-1.4.jar  
junit-3.8.1.jar  
log4j-1.2.8.jar  
opensaml-2.0-TP1-jdk-1.5.jar  
openws-1.0-alpha1-jdk-1.5.jar  
resolver.jar  
saaj-api.jar  
saaj-impl.jar  
soapprocessor.jar  
xmldsig.jar  
sunxacml-cvs1.6.jar  
sunxacml-support-cvs1.6.jar  
xmlsec-1.4.1.jar  
xmlsecSamples.jar  
xalan-2.6.jar  
xercesImpl.jar  
xml-apis.jar
```

The following libraries must be placed into "endorsed" directory:

```
endorsed/resolver.jar  
endorsed/xalan-2.6.jar  
endorsed/xercesImpl.jar  
endorsed/xercesSamples.jar  
endorsed/xml-apis.jar  
endorsed/xmlParserAPIs.jar
```

The following libraries are required to support use of Identity Based Cryptography for token key distribution in multi-domain environment:

```
lib-ibc/IdentityBasedEncryptionJCA.1.0.38.jar  
  
lib-ibc/library/jakarta-regexp-1.4.jar  
lib-ibc/library/bcel-head.jar  
lib-ibc/library/FieldTracker.jar  
lib-ibc/library/artima/suiterunner-1.0beta6.jar  
lib-ibc/library/nuimcscg/tender-dev.jar  
lib-ibc/library/nuimcscg/ArtimaSuiteRunnerAntTask.1.1.3.jar  
lib-ibc/library/nuimcscg/blitz-dev.jar  
lib-ibc/library/nuimcscg/fault-dev.jar
```

Full set of libraries is provided next to the GAAA-TK jar-file at the WP4 wiki page http://www.ist-phosphorus.eu/wiki/index.php/Pluggable_GAAA-TK_library

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



6 Conclusion

The report describes the results of the development and implementation of the Generic AAA Authorisation framework (GAAA-AuthZ) for ONRP developments. It describes the key functionalities to support multi-domain ONRP and introduces a number of mechanisms and solutions to support them, in particular: an AuthZ ticket format for extended AuthZ session management, an AuthZ token format for multi-domain access control and signalling, the Token Validation Service (TVS) to enable token based policy enforcement, and a policy Obligation Handling Reference Model (OHRM).

The developed pluggable GAAA-TK Java library is designed in a such way that it could support the major Phosphorus testbed use cases and can be used at all networking layers: Data plane, Control Plane, Service plane, and can be also integrated with applications. The proposed architecture will allow a smooth integration with other AuthZ frameworks as currently used and developed by NRENs and the Grid community.

The report provides an update on the proposed XACML-NRP attributes and policy profile for general and optical network resource provisioning.

The deliverable describes a set of APIs used to call the main GAAA-TK services. The Authorisation service can be requested via the Policy Enforcement Point (PEP) or directly from the XACML Policy Decision Point (PDP). The TVS API provides rich functionality for handling token used for access control and signalling. A separate section 5 is devoted to the GAAA-TK library setup and configuration.

The following are suggested further GAAA-TK developments: adding an AuthZ request input format and content checking, moving the library configuration to an XML configuration file, providing an automatic installation procedure together with the domain related keys installation (or generation), adding IBC support for managing token keys distribution, adding support for pilot token type 4 (i.e. allowing GRI bound security context distribution between domains), adding a Web Services interface and SAML-XACML protocol to support external AuthZ request calls, adding XML database adaptors for storing XACML policies and TVS session context in XML databases, adding SAML subject attribute validation (when information is received from WP3).

Based on the current GAAA-NRP architecture development and the GAAA-TK library implementation, WP4 will focus on extending the GAAA Toolkit functionality and assist other WP's in integrating AAA/AuthZ services into

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



GAAA toolkit pluggable components and XACML policy profile for ONRP

their NRPS's. This should also provide a feedback for further architecture updates and GAAA Toolkit developments.

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



7 References

- [1] RFC2903 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] Demchenko Y, A. Wan, M. Cristea, C. de Laat, "Authorisation Infrastructure for On-Demand Network Resource Provisioning", The 9th IEEE/ACM International Conference on Grid Computing (Grid 2008), Tsukuba, Japan, Sept. 29 - Oct. 1, 2008. Accepted paper.
- [4] Gommans, L., L. Xu, Y. Demchenko, A. Wan, M. Cristea, R. Meijer, C. de Laat, "Multi-domain Lightpath Authorization using Tokens", Future Generation Computer Systems, Special issue on OptiPuter. Accepted paper.
- [5] Viola Meta Scheduling Service Project. [Online]. Available <http://packcs-e0.scai.fhg.de/viola-project/>
- [6] 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.
- [7] 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.
- [8] 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.
- [9] "Token-based authorization of connection oriented network resources", by Leon Gommans, Franco Travostino, John Vollbrecht, Ceas de Laat, and Robert Meijer, in Proceedings of GRIDNETS, San Jose, CA, USA, Oct 2004.

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



GAAA toolkit pluggable components and XACML policy profile for ONRP

- [10] "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>
- [11] "XACML 3.0 administrative policy," OASIS Draft, 10 December 2005. [Online]. Available from http://docs.oasis-open.org/access_control
- [12] 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>
- [13] 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>
- [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] "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.
- [16] Menezes A., P. van Oorschot, S. Vanstone, "Handbook of Applied Cryptography". - ISBN: 0-8493-8523-7, October 1996, 816 pages
- [17] WP1 NSP topology format definition.
- [18] 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
- [19] ESnet On-Demand Secure Circuits and Advance Reservation System (OSCARS). - <http://www.es.net/oscars/>
- [20] "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
- [21] 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
- [22] XACML Attribute and Obligation Profile for Authorization Interoperability in Grids. [Online] Available <https://edms.cern.ch/document/929867/1>

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



Appendix A Acronyms

| | |
|-------------------|---|
| AAA | Authentication, Authorisation, Accounting |
| AAI | Authentication, Authorization Infrastructure |
| ACL | Access Control List |
| AuthZ | Authorization |
| AuthN | Authentication |
| CRP | Complex Resource Provisioning |
| CVS | Credential Validation Services |
| DCAS | Domain Site Central Authorisation Service |
| e2e | end to end |
| EGEE | Enabling Grids for E-science (European Grid Project) |
| GAAA-AuthZ | Generic AAA Authorisation Framework |
| GAAAPI | Generic Authentication/Authorisation Application Programming Interface |
| GEANT2 | Pan-European Gigabit Research Network |
| gLite | EGEE Grid middleware |
| GMPLS | Generalized MPLS (MultiProtocol Label Switching) |
| GSI | Grid Security Infrastructure |
| GT4 | Globus Toolkit Version 4 (Web-Service based) |
| IdM | Identity Manager |
| IdP | Identity Provider |
| NREN | National Research and Education Network |
| NRP | Network Resource Provisioning |
| OLPP | Optical LightPath Provisioning |
| NRPS | Network Resource Provisioning System |
| OHRM | Obligation Handling Reference Model |
| OSCARS | On-demand Secure Circuits and Advance Reservation System |
| PAP | Policy Authority Point |
| PDP | Policy Decision Point |
| PEP | Policy Enforcement Point |
| PIP | Policy Information Point |
| PKC | X.509 Public Key Certificate |
| PKI | Public Key Infrastructure |
| QoS | Quality of Service |
| SAAS | Shibboleth Attribute Authority Service |

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



GAAA toolkit pluggable components and XACML policy profile for ONRP

| | |
|-------------------|---|
| SAML | Security Assertion Markup Language |
| SCAS | Site Central Authorisation Service |
| S-R-A (-E) | Subject – Resource – Action (- Environment) in relation to the XACML policy and context definition |
| SSO | Single Sign-On |
| TBN | Token Based Networking |
| TBS | Token Based Switch |
| TB | Token Builder |
| TVS | Token Validation Service |
| VO | Virtual Organisation |
| VOMS | Virtual Organization Membership Service |
| UNICORE | European Grid Middleware (UNiform Access to COmpute REsources) |
| VLAN | Virtual LAN (as specified in IEEE 802.1p) |
| VIOLA | A German project funded by the German Federal Ministry of Education and Research (Vertically Integrated Optical Testbed for Large Applications in DFN) |
| VPN | Virtual Private Network |
| XACML | eXtensible Access Control Markup Language |
| XML | eXtensible Markup Language |

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



Appendix B AuthZ Ticket XML Schema and Examples

B.1 Current AuthZ ticket schema (version 0.3)

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:AAA="http://www.aaathreach.org/ns/#AAA"
targetNamespace="http://www.aaathreach.org/ns/#AAA" elementFormDefault="qualified">
  <xs:element name="AuthzTicket" type="AAA:AuthzTicketType"/>
  <xs:complexType name="AuthzTicketType">
    <xs:sequence>
      <xs:element name="Decisions" type="AAA:DecisionsType"/>
      <xs:element name="Conditions" type="AAA:ConditionsType" minOccurs="0"/>
      <xs:element name="Subject" type="AAA:SubjectType" minOccurs="0"/>
      <xs:element name="Resources" type="AAA:ResourcesType" minOccurs="0"/>
      <xs:element name="Actions" type="AAA:ActionsType" minOccurs="0"/>
      <xs:element name="Delegation" type="AAA:DelegationType" minOccurs="0"/>
      <xs:element name="Obligations" type="AAA:ObligationsType" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="TicketID" type="xs:hexBinary" use="required"/>
    <xs:attribute name="SessionID" type="xs:string" use="required"/>
    <xs:attribute name="Issuer" type="xs:anyURI" use="optional"/>
  </xs:complexType>
  <xs:complexType name="DecisionsType">
    <xs:sequence>
      <xs:element name="Decision" type="AAA:DecisionType" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="DecisionType">
    <xs:attribute name="Result" type="xs:anyURI" use="required"/>
    <xs:attribute name="ResourceID" type="xs:anyURI" use="optional"/>
    <xs:attribute name="PolicyRef" type="xs:string" use="optional"/>
  </xs:complexType>
  <xs:complexType name="ConditionsType">
    <xs:sequence>
      <xs:element name="ConditionAuthzSession" type="AAA:ConditionAuthzSessionType"
minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="NotBefore" type="xs:dateTime" use="optional"/>
    <xs:attribute name="NotOnOrAfter" type="xs:dateTime" use="optional"/>
    <xs:attribute name="renewal" type="xs:string" use="optional"/>
  </xs:complexType>
  <xs:complexType name="ConditionAuthzSessionType">
    <xs:sequence>
```

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



GAAA toolkit pluggable components and XACML policy profile for ONRP

```
        <xs:element ref="AAA:SessionData" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="SessionID" type="xs:string" use="required"/>
</xs:complexType>
<xs:element name="SessionData" type="xs:anyType"/>
<xs:complexType name="SubjectType">
    <xs:sequence>
        <xs:element ref="AAA:SubjectID"/>
        <xs:element name="SubjectConfirmation" type="AAA:SubjectConfirmationType"
minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="SubjectAttribute" type="AAA:SubjectAttributeType"
minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="AAA:SubjectContext" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="Id" type="xs:anyURI" use="optional"/>
</xs:complexType>
<xs:element name="SubjectID" type="xs:string"/>
<xs:complexType name="SubjectConfirmationType">
    <xs:sequence>
        <xs:element ref="AAA:SubjectConfirmationData" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="Method" type="xs:anyURI" use="optional"/>
</xs:complexType>
<xs:element name="SubjectConfirmationData" type="xs:anyType"/>
<xs:element name="SubjectAttributeType" type="xs:string"/>
<xs:complexType name="SubjectAttributeType">
    <xs:simpleContent>
        <xs:extension base="xs:string">
            <xs:attribute name="AttributeId" type="xs:double" use="required"/>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
<xs:element name="SubjectContext" type="xs:string"/>
<xs:complexType name="ActionsType">
    <xs:sequence>
        <xs:element ref="AAA:Action" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:element name="Action" type="xs:anyURI"/>
<xs:complexType name="ResourcesType">
    <xs:sequence>
        <xs:element ref="AAA:Resource" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:element name="Resource" type="AAA:ResourceType"/>
<xs:complexType name="ResourceType" mixed="true">
    <xs:sequence>
        <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute namespace="##any" processContents="lax"/>
</xs:complexType>
<!--
***** Implementation for TBN simple message
*****
***** Contains only obligatory elements AzTicket/Decisions/Decision/Resources/Resource *****
***** and proprietary Resource attributes (as extensible) ResourceID, Key, Port *****
*****

    <xs:complexType name="ResourceType" mixed="true">
        <xs:sequence>
            <xs:element name="LRI" type="AAA:LRIType"/>
            <xs:element name="TokenKey" type="xs:string"/>
            <xs:element name="Ports" type="AAA:Ports"/>
            <xs:element name="ApplicationFlow" type="AAA:ApplicationFlowType"/>
        </xs:sequence>
        <xs:attribute name="ResourceID" type="xs:anyURI" use="required"/>
    </xs:complexType>
<xs:complexType name="LRIType" mixed="true">
```

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



GAAA toolkit pluggable components and XACML policy profile for ONRP

```
        <xs:attribute name="purpose" type="xs:string" use="required"/>
    </xs:complexType>
    <xs:complexType name="PortsType" mixed="true">
        <xs:sequence>
            <xs:element name="port1" type="xs:string"/>
            <xs:element name="port2" type="xs:string"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="ApplicationFlowType" mixed="true">
        <xs:sequence>
            <xs:element name="IPpacketMask" type="xs:string"/>
            <xs:element name="IPsource" type="xs:string"/>
            <xs:element name="IPdestination" type="xs:string"/>
            <xs:element name="PortSource" type="xs:string"/>
            <xs:element name="PortDestination" type="xs:string"/>
        </xs:sequence>
        <xs:attribute name="ResourceID" type="xs:anyURI" use="required"/>
    </xs:complexType>
-->
<xs:complexType name="DelegationType">
    <xs:choice>
        <xs:element name="DelegationSubjects" type="AAA:DelegationSubjectsType"/>
        <xs:element name="DelegationCommunity" type="xs:string" maxOccurs="unbounded"/>
    </xs:choice>
    <xs:attribute name="MaxDelegationDepth" type="xs:decimal" use="required"/>
    <xs:attribute name="restriction" use="required">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:enumeration value="disallowed"/>
                <xs:enumeration value="subjects"/>
                <xs:enumeration value="community"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:attribute>
</xs:complexType>
<xs:complexType name="DelegationSubjectsType">
    <xs:sequence>
        <xs:element ref="AAA:SubjectID" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:element name="Obligation" type="xs:anyType"/>
<xs:complexType name="ObligationsType">
    <xs:sequence>
        <xs:element ref="AAA:Obligation" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
</xs:schema>
```

B.2 AuthzTicket example – Proprietary Format

The listing below provide an example with the Resource element as required for TBS programming generated with the GAAAPI package. 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.

```
<AAA:AuthzTicket xmlns:AAA="http://www.aaauthreach.org/ns/#AAA" Issuer="urn:cnl:trust:tickauth:pep"
TicketID="cba06d1a9df148cf420ef8f3e4fd2b3" SessionID=" f8f3e4fd2b3 df148cf4200">
  <AAA:Decisions><AAA:Decision ResourceID="http://resources.collaboratory.nl/Philips XPS1"
result="Permit"/>
</AAA:Decisions>
<!-- SAML mapping: <AuthorizationDecisionStatement Decision="*" PolicyRef="*" Resource="*"> -->
```

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



GAAA toolkit pluggable components and XACML policy profile for ONRP

```
<AAA:Resources><AAA:Resource ResourceID="http://www.aaaarch.org/TBS/ForceG">
  <AAA:LRI purpose="String">text</AAA:LRI>
  <AAA:TokenKey>String</AAA:TokenKey>
  <AAA:Ports>
    <AAA:port1>String</AAA:port1>
    <AAA:port2>String</AAA:port2>
  </AAA:Ports>
  <AAA:ApplicationFlow>
    <AAA:IPpacketMask>String</AAA:IPpacketMask>
    <AAA:IPsource>String</AAA:IPsource>
    <AAA:IPdestination>String</AAA:IPdestination>
    <AAA:PortSource>String</AAA:PortSource>
    <AAA:PortDestination>String</AAA:PortDestination>
  </AAA:ApplicationFlow>
</AAA:Resource></AAA:Resources>
<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>
    IGhA11vwa8YQomTgB9Ege9JRNnld84AggaDkOb5WW4U=</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>
```

B.3 AuthzTicket example – SAML AuthZ Assertion

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



GAAA toolkit pluggable components and XACML policy profile for ONRP

```
<Assertion xmlns="urn:oasis:names:tc:SAML:1.0:assertion"
xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion" xmlns:sampl="urn:oasis:names:tc:SAML:1.0:protocol"
AssertionID="d9805da9af718909ba6dd2e0e49b9bc4" IssueInstant="2005-03-27T23:58:00.534Z"
Issuer="cnl:subject:CNLAAAauthority" MajorVersion="1" MinorVersion="1">
  <Conditions NotBefore="2004-12-04T23:00:00.000Z" NotOnOrAfter="2004-12-22T21:22:22.000Z"/>
  <Advice>
    <Obligation FulfillOn="Permit" ObligationId="urn:oasis:names:tc:xacml:1.0:obligation">Policy
obligation (1): Action cost = 100 EUR</Obligation>
    <Obligation FulfillOn="Permit" ObligationId="urn:oasis:names:tc:xacml:1.0:obligation">Policy
obligation (2): Request data logging</Obligation>
  </Advice>
  <AuthorizationDecisionStatement Decision="@Resource;Permit"
Resource="http://resources.collaboratory.nl/Phillips XPS1">
    <Subject>
      <NameIdentifier Format="urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress"
NameQualifier="cnl:subject:customer">WHO740@users.collaboratory.nl</NameIdentifier>
      <SubjectConfirmation>
        <ConfirmationMethod>email</ConfirmationMethod>
        <ConfirmationMethod>callback</ConfirmationMethod>
        <ConfirmationMethod>email</ConfirmationMethod>
        <ConfirmationMethod>callback</ConfirmationMethod>
      </SubjectConfirmation>
    </Subject>
    <Action Namespace="urn:oasis:names:tc:SAML:1.0:action:cnl:action">CNLaction02: zoom</Action>
    <Action Namespace="urn:oasis:names:tc:SAML:1.0:action:cnl:action">CNLaction01: 2Dscan</Action>
    <Evidence>
      <AssertionIDReference>2355789adcebb</AssertionIDReference>
      <Assertion AssertionID="a5de72c3b0fa4df31f288d318dcbd0e4" IssueInstant="2005-03-
27T23:58:00.454Z" Issuer="cnl:subject:CNLAAAauthority" MajorVersion="1" MinorVersion="1">
        <Conditions NotBefore="2004-12-04T23:00:00.000Z" NotOnOrAfter="2004-12-22T21:22:22.000Z"/>
        <AttributeStatement>
          <Subject>
            <NameIdentifier Format="urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress"
NameQualifier="cnl:subject:customer">HEIS007@staff.collaboratory.nl</NameIdentifier>
            <SubjectConfirmation>
              <ConfirmationMethod>email</ConfirmationMethod>
              <ConfirmationMethod>callback</ConfirmationMethod>
            </SubjectConfirmation>
          </Subject>
          <Attribute xmlns:typens="urn:cnl" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" AttributeName="AttributeSubject"
AttributeNamespace="urn:cnl">
            <AttributeValue xsi:type="typens:subject">@cnl:subject:role:manager</AttributeValue>
            <AttributeValue xsi:type="typens:subject">cnl:subject:role</AttributeValue>
            <AttributeValue xsi:type="typens:subject">jobID</AttributeValue>
          </Attribute>
        </AttributeStatement>
      </Assertion>
      <AssertionIDReference>@resourceId;Permit</AssertionIDReference>
    </Evidence>
  </AuthorizationDecisionStatement>
</Assertion>
```

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



Appendix C AuthZ Token XML Schema

Current AuthZ token schema.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:AAA="http://www.aaauthreach.org/ns/#AAA"
targetNamespace="http://www.aaauthreach.org/ns/#AAA" elementFormDefault="qualified">
  <xs:element name="AuthzToken" type="AAA:AuthzTokenType"/>
  <xs:complexType name="AuthzTokenType">
    <xs:sequence>
      <xs:element ref="AAA:TokenValue" minOccurs="0"/>
      <xs:element name="Conditions" type="AAA:ConditionsType" minOccurs="0"/>
      <xs:element name="Domains" type="AAA:DomainsType" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="TokenID" type="xs:hexBinary" use="required"/>
    <xs:attribute name="SessionID" type="xs:string" use="required"/>
    <xs:attribute name="Issuer" type="xs:anyURI" use="optional"/>
  </xs:complexType>
  <xs:complexType name="ConditionsType">
    <xs:attribute name="NotBefore" type="xs:dateTime" use="optional"/>
    <xs:attribute name="NotOnOrAfter" type="xs:dateTime" use="optional"/>
  </xs:complexType>
  <xs:element name="TokenValue" type="xs:anyURI"/>
  <xs:complexType name="DomainsType">
    <xs:sequence>
      <xs:element name="Domain" type="AAA:DomainType" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="DomainType">
    <xs:sequence>
      <xs:element name="AuthzToken" type="AAA:AuthzTokenType"/>
      <xs:element name="KeyInfo" type="AAA:KeyInfoType" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="domainId" type="xs:anyURI" use="optional"/>
  </xs:complexType>
  <xs:complexType name="KeyInfoType" mixed="true">
    <xs:sequence>
      <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="keytype" type="xs:string" use="optional"/>
  </xs:complexType>
</xs:schema>
```

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



Appendix D XACML Policy examples

D.1 Example 1 - XACML policy and corresponding request/response messages

b) Policy example evaluating user roles and network source and target TNA's.

```
<Policy PolicyId="http://testbed.ist-phosphorus.eu/viola/harmony/demo001/policy"
RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:permit-overrides">
  <Description>Permit actions for Phosphorus testbed users with specific roles</Description>
  <Target>
    <Subjects>
      <AnySubject/>
    </Subjects>
    <Resources>
      <Resource>
        <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-
equal">
          <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#anyURI">http://testbed.ist-
phosphorus.eu/viola/harmony</AttributeValue>
          <ResourceAttributeDesignator
AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
DataType="http://www.w3.org/2001/XMLSchema#anyURI"/>
          </ResourceMatch>
        </Resource>
      </Resources>
    <Actions>
      <AnyAction/>
    </Actions>
  </Target>
  <Rule RuleId="http://testbed.ist-phosphorus.eu/viola/harmony/demo001/policy/rule/action-
type/create-path" Effect="Permit">
    <Target>
      <Subjects>
        <AnySubject/>
      </Subjects>
      <Resources>
        <AnyResource/>
      </Resources>
      <Actions>
        <Action>
          <ActionMatch
MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">create-path</AttributeValue>
            <ActionAttributeDesignator
AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
DataType="http://www.w3.org/2001/XMLSchema#string"/>
            </ActionMatch>
          </Action>
        </Actions>
      </Target>
      <Condition FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-
member-of">
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
          <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">admin</AttributeValue>
```

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



GAAA toolkit pluggable components and XACML policy profile for ONRP

```

        <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">researcher</AttributeValue>
        <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">professor</AttributeValue>
    </Apply>
    <SubjectAttributeDesignator AttributeId="http://authz-
interop.org/AAA/xacml/subject/subject-role" DataType="http://www.w3.org/2001/XMLSchema#string"/>
    </Condition>
</Rule>
<Rule RuleId="http://testbed.ist-phosphorus.eu/viola/harmony/demo001/policy/rule/action-
type/activate-path" Effect="Permit">
    <Target>
        <Subjects>
            <AnySubject/>
        </Subjects>
        <Resources>
            <AnyResource/>
        </Resources>
        <Actions>
            <Action>
                <ActionMatch
MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
                    <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">activate-path</AttributeValue>
                    <ActionAttributeDesignator
AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
DataType="http://www.w3.org/2001/XMLSchema#string"/>
                </ActionMatch>
            </Action>
        </Actions>
    </Target>
    <Condition FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-
member-of">
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
            <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">researcher</AttributeValue>
            <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">professor</AttributeValue>
        </Apply>
        <SubjectAttributeDesignator AttributeId="http://authz-
interop.org/AAA/xacml/subject/subject-role" DataType="http://www.w3.org/2001/XMLSchema#string"/>
    </Condition>
</Rule>
<Rule RuleId="http://testbed.ist-phosphorus.eu/viola/harmony/demo001/policy/rule/action-
type/cancel" Effect="Permit">
    <Target>
        <Subjects>
            <AnySubject/>
        </Subjects>
        <Resources>
            <AnyResource/>
        </Resources>
        <Actions>
            <Action>
                <ActionMatch
MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
                    <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">cancel</AttributeValue>
                    <ActionAttributeDesignator
AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
DataType="http://www.w3.org/2001/XMLSchema#string"/>
                </ActionMatch>
            </Action>
        </Actions>
    </Target>
    <Condition FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-
member-of">
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">

```

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



GAAA toolkit pluggable components and XACML policy profile for ONRP

```
<AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">admin</AttributeValue>
</Apply>
<SubjectAttributeDesignator AttributeId="http://authz-
interop.org/AAA/xacml/subject/subject-role" DataType="http://www.w3.org/2001/XMLSchema#string"/>
</Condition>
</Rule>
<Rule RuleId="http://testbed.ist-phosphorus.eu/viola/harmony/demo001/policy/rule/action-
type/access" Effect="Permit">
<Target>
<Subjects>
<AnySubject/>
</Subjects>
<Resources>
<AnyResource/>
</Resources>
<Actions>
<Action>
<ActionMatch
MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
<AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">access</AttributeValue>
<ActionAttributeDesignator
AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
DataType="http://www.w3.org/2001/XMLSchema#string"/>
</ActionMatch>
</Action>
</Actions>
</Target>
<Condition FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-
member-of">
<Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
<AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">admin</AttributeValue>
<AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">professor</AttributeValue>
<AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">researcher</AttributeValue>
<AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">student</AttributeValue>
</Apply>
<SubjectAttributeDesignator AttributeId="http://authz-
interop.org/AAA/xacml/subject/subject-role" DataType="http://www.w3.org/2001/XMLSchema#string"/>
</Condition>
</Rule>
</Policy>
```

b) XACML request message example:

```
<Request>
<Subject SubjectCategory="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject">
<Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
DataType="http://www.w3.org/2001/XMLSchema#string" Issuer="http://testbed.ist-
phosphorus.eu/phosphorus/aaa/AttributeIssuer" IssueInstant="2008-07-29T14:34:44.796000000+02:00">
<AttributeValue>WH0740@users.testbed.ist-phosphorus.eu</AttributeValue>
</Attribute>
<Attribute AttributeId="http://authz-interop.org/AAA/xacml/subject/subject-context"
DataType="http://www.w3.org/2001/XMLSchema#string" Issuer="http://testbed.ist-
phosphorus.eu/phosphorus/aaa/AttributeIssuer" IssueInstant="2008-07-29T14:34:44.796000000+02:00">
<AttributeValue>demo041</AttributeValue>
</Attribute>
<Attribute AttributeId="http://authz-interop.org/AAA/xacml/subject/subject-confdata"
DataType="http://www.w3.org/2001/XMLSchema#string" Issuer="http://testbed.ist-
phosphorus.eu/phosphorus/aaa/AttributeIssuer" IssueInstant="2008-07-29T14:34:44.796000000+02:00">
```

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



GAAA toolkit pluggable components and XACML policy profile for ONRP

```
<AttributeValue>IGhA1lvwa8bUktYhuU9que+d4XLUvJFHrtDC/OE3UilbxtmuCxLldw==</AttributeValue>
</Attribute>
<Attribute AttributeId="http://authz-interop.org/AAA/xacml/subject/subject-role"
DataType="http://www.w3.org/2001/XMLSchema#string" Issuer="http://testbed.ist-
phosphorus.eu/phosphorus/aaa/AttributeIssuer" IssueInstant="2008-07-29T14:34:44.796000000+02:00">
  <AttributeValue>researcher</AttributeValue>
</Attribute>
</Subject>
<Resource>
  <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
DataType="http://www.w3.org/2001/XMLSchema#anyURI" Issuer="http://testbed.ist-
phosphorus.eu/phosphorus/aaa/AttributeIssuer" IssueInstant="2008-07-29T14:34:44.765000000+02:00">
  <AttributeValue>http://testbed.ist-phosphorus.eu/viola/harmony</AttributeValue>
</Attribute>
</Resource>
<Action>
  <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
DataType="http://www.w3.org/2001/XMLSchema#string" Issuer="http://testbed.ist-
phosphorus.eu/phosphorus/aaa/AttributeIssuer" IssueInstant="2008-07-29T14:34:44.765000000+02:00">
  <AttributeValue>create-path</AttributeValue>
</Attribute>
</Action>
</Request>
```

c) Example Response message with returned decision "Permit"

```
<Response>
  <Result ResourceId="http://testbed.ist-phosphorus.eu/viola/harmony">
    <Decision>Permit</Decision>
    <Status>
      <StatusCode Value="urn:oasis:names:tc:xacml:1.0:status:ok"/>
    </Status>
  </Result>
</Response>
```

D.2 Example 2 - XACML policy and corresponding request/response messages

b) Policy example evaluating user roles and network source and target TNA's.

```
<Policy PolicyId="http://testbed.ist-phosphorus.eu/viola/harmony/demo010/policy2:tna"
RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:deny-overrides">
  <Description>Permits reservation for a selected set of TNA address ranges: 10.3.*, 10.4.*,
10.7.*, 10.8.* (10.3.1.*, 10.4.1.*, 10.7.3.*, 10.7.12.*, 10.7.12.*, 10.7.13.*, 10.8.1.*)
and Permit actions for Phosphorus testbed users with specific roles</Description>
  <Target>
    <Subjects>
      <AnySubject/>
    </Subjects>
    <Resources>
      <Resource>
        <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-
equal">
          <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#anyURI">http://testbed.ist-
phosphorus.eu/viola/harmony</AttributeValue>
          <ResourceAttributeDesignator
AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
DataType="http://www.w3.org/2001/XMLSchema#anyURI"/>
        </ResourceMatch>
```

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



GAAA toolkit pluggable components and XACML policy profile for ONRP

```
        </Resource>
    </Resources>
    <Actions>
        <AnyAction/>
    </Actions>
</Target>

    <Rule RuleId="http://testbed.ist-phosphorus.eu/viola/harmony/demo010/policy/rule/action-
type/create-path/tna-check-create-path" Effect="Permit">
    <Target>
        <Subjects>
            <Subject>
                <SubjectMatch
MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
                    <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">researcher</AttributeValue>
                    <SubjectAttributeDesignator AttributeId="http://authz-
interop.org/AAA/xacml/subject/subject-role" DataType="http://www.w3.org/2001/XMLSchema#string"/>
                </SubjectMatch>
            </Subject>
            <Subject>
                <SubjectMatch
MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
                    <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">professor</AttributeValue>
                    <SubjectAttributeDesignator AttributeId="http://authz-
interop.org/AAA/xacml/subject/subject-role" DataType="http://www.w3.org/2001/XMLSchema#string"/>
                </SubjectMatch>
            </Subject>
            <Subject>
                <SubjectMatch
MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
                    <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">admin</AttributeValue>
                    <SubjectAttributeDesignator AttributeId="http://authz-
interop.org/AAA/xacml/subject/subject-role" DataType="http://www.w3.org/2001/XMLSchema#string"/>
                </SubjectMatch>
            </Subject>
        </Subjects>
    </Resources>
        <AnyResource/>
    </Resources>
    <Actions>
        <Action>
            <ActionMatch
MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
                <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">create-path</AttributeValue>
                <ActionAttributeDesignator
AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
DataType="http://www.w3.org/2001/XMLSchema#string"/>
            </ActionMatch>
        </Action>
    </Actions>
</Target>
    <Description>
        Checks if TNA address (in a form of string) belongs to special range (10.3.*, 10.4.*,
10.7.*, 10.8.*)
    </Description>
    <Condition FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
        <!-- source TNA -->
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:or">
            <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-regexp-
match">
                <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">10.3.</AttributeValue>
                <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-
one-and-only">
```

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



GAAA toolkit pluggable components and XACML policy profile for ONRP

```
interop.org/AAA/xacml/resource/source" DataType="http://www.w3.org/2001/XMLSchema#string"/>
  </Apply>
</Apply>
<Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-regexp-
match">
  <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">10.4.</AttributeValue>
  <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-
one-and-only">
    <ResourceAttributeDesignator AttributeId="http://authz-
interop.org/AAA/xacml/resource/source" DataType="http://www.w3.org/2001/XMLSchema#string"/>
    </Apply>
  </Apply>
  <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-regexp-
match">
    <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">10.7.</AttributeValue>
    <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-
one-and-only">
      <ResourceAttributeDesignator AttributeId="http://authz-
interop.org/AAA/xacml/resource/source" DataType="http://www.w3.org/2001/XMLSchema#string"/>
      </Apply>
    </Apply>
    <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-regexp-
match">
      <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">10.8.</AttributeValue>
      <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-
one-and-only">
        <ResourceAttributeDesignator AttributeId="http://authz-
interop.org/AAA/xacml/resource/source" DataType="http://www.w3.org/2001/XMLSchema#string"/>
        </Apply>
      </Apply>
      <!-- target TNA -->
      <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:or">
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-regexp-
match">
          <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">10.3.</AttributeValue>
          <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-
one-and-only">
            <ResourceAttributeDesignator AttributeId="http://authz-
interop.org/AAA/xacml/resource/target" DataType="http://www.w3.org/2001/XMLSchema#string"/>
            </Apply>
          </Apply>
          <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-regexp-
match">
            <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">10.4.</AttributeValue>
            <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-
one-and-only">
              <ResourceAttributeDesignator AttributeId="http://authz-
interop.org/AAA/xacml/resource/target" DataType="http://www.w3.org/2001/XMLSchema#string"/>
              </Apply>
            </Apply>
            <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-regexp-
match">
              <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">10.7.</AttributeValue>
              <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-
one-and-only">
                <ResourceAttributeDesignator AttributeId="http://authz-
interop.org/AAA/xacml/resource/target" DataType="http://www.w3.org/2001/XMLSchema#string"/>
                </Apply>
              </Apply>
              <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-regexp-
match">
```

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



GAAA toolkit pluggable components and XACML policy profile for ONRP

```

        <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">10.8.</AttributeValue>
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-
one-and-only">
                <ResourceAttributeDesignator AttributeId="http://authz-
interop.org/AAA/xacml/resource/target" DataType="http://www.w3.org/2001/XMLSchema#string"/>
                </Apply>
        </Apply>
</Condition>
</Rule>

<Rule RuleId="http://testbed.ist-phosphorus.eu/viola/harmony/demo010/policy/rule/action-
type/create-path/tna-check-cancel" Effect="Permit">
<Target>
<Subjects>
<Subject>
<SubjectMatch
MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
<AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">admin</AttributeValue>
<SubjectAttributeDesignator AttributeId="http://authz-
interop.org/AAA/xacml/subject/subject-role" DataType="http://www.w3.org/2001/XMLSchema#string"/>
</SubjectMatch>
</Subject>
</Subjects>
<Resources>
<AnyResource/>
</Resources>
<Actions>
<Action>
<ActionMatch
MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
<AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">cancel</AttributeValue>
<ActionAttributeDesignator
AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
DataType="http://www.w3.org/2001/XMLSchema#string"/>
</ActionMatch>
</Action>
</Actions>
</Target>
<Description>
Checks if TNA address (in a form of string) belongs to special range (10.3.*, 10.4.*,
10.7.*, 10.8.*)
</Description>

<Condition>
<!-- Repeating Condition content removed for easier readability -->
</Condition>

</Rule>

<Rule RuleId="http://testbed.ist-phosphorus.eu/viola/harmony/demo010/policy/rule/action-
type/create-path/tna-check-access" Effect="Permit">
<Target>
<Subjects>
<Subject>
<SubjectMatch
MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
<AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">researcher</AttributeValue>
<SubjectAttributeDesignator AttributeId="http://authz-
interop.org/AAA/xacml/subject/subject-role" DataType="http://www.w3.org/2001/XMLSchema#string"/>
</SubjectMatch>
</Subject>
<Subject>
<SubjectMatch
MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
```

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



GAAA toolkit pluggable components and XACML policy profile for ONRP

```

        <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">professor</AttributeValue>
        <SubjectAttributeDesignator AttributeId="http://authz-
interop.org/AAA/xacml/subject/subject-role" DataType="http://www.w3.org/2001/XMLSchema#string"/>
        </SubjectMatch>
    </Subject>
    <Subject>
        <SubjectMatch
MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
        <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">student</AttributeValue>
        <SubjectAttributeDesignator AttributeId="http://authz-
interop.org/AAA/xacml/subject/subject-role" DataType="http://www.w3.org/2001/XMLSchema#string"/>
        </SubjectMatch>
    </Subject>
    <Subject>
        <SubjectMatch
MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
        <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">admin</AttributeValue>
        <SubjectAttributeDesignator AttributeId="http://authz-
interop.org/AAA/xacml/subject/subject-role" DataType="http://www.w3.org/2001/XMLSchema#string"/>
        </SubjectMatch>
    </Subject>
</Subjects>
<Resources>
    <AnyResource/>
</Resources>
<Actions>
    <Action>
        <ActionMatch
MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
        <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">access</AttributeValue>
        <ActionAttributeDesignator
AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
DataType="http://www.w3.org/2001/XMLSchema#string"/>
        </ActionMatch>
    </Action>
</Actions>
</Target>
<Description>
    Checks if TNA address (in a form of string) belongs to special range (10.3.*, 10.4.*,
10.7.*, 10.8.*)
</Description>

    <Condition>
    <!-- Repeating Condition content removed for easier readability -->
</Condition>

</Rule>

    <Rule RuleId="http://testbed.ist-phosphorus.eu/viola/harmony/demo010/policy/rule/action-
type/create-path/tna-check-activate-path" Effect="Permit">
    <Target>
        <Subjects>
            <Subject>
                <SubjectMatch
MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
                <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">researcher</AttributeValue>
                <SubjectAttributeDesignator AttributeId="http://authz-
interop.org/AAA/xacml/subject/subject-role" DataType="http://www.w3.org/2001/XMLSchema#string"/>
                </SubjectMatch>
            </Subject>
            <Subject>
                <SubjectMatch
MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">

```

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



GAAA toolkit pluggable components and XACML policy profile for ONRP

```

        <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">professor</AttributeValue>
        <SubjectAttributeDesignator AttributeId="http://authz-
interop.org/AAA/xacml/subject/subject-role" DataType="http://www.w3.org/2001/XMLSchema#string"/>
        </SubjectMatch>
    </Subject>
</Subjects>
<Resources>
    <AnyResource/>
</Resources>
<Actions>
    <Action>
        <ActionMatch
MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">activate-path</AttributeValue>
            <ActionAttributeDesignator
AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
DataType="http://www.w3.org/2001/XMLSchema#string"/>
        </ActionMatch>
    </Action>
</Actions>
</Target>
<Description>
    Checks if TNA address (in a form of string) belongs to special range (10.3.*, 10.4.*,
10.7.*, 10.8.*)
</Description>
    <Condition>
    <!-- Repeating Condition content removed for easier readability -->
    </Condition>
</Rule>
</Policy>
```

b) XACML request message example:

```
<Request>
  <Subject SubjectCategory="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject">
    <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
DataType="http://www.w3.org/2001/XMLSchema#string" Issuer="http://testbed.ist-
phosphorus.eu/phosphorus/aaa/AttributeIssuer" IssueInstant="2008-07-29T14:34:44.796000000+02:00">
      <AttributeValue>WHO740@users.testbed.ist-phosphorus.eu</AttributeValue>
    </Attribute>
    <Attribute AttributeId="http://authz-interop.org/AAA/xacml/subject/subject-context"
DataType="http://www.w3.org/2001/XMLSchema#string" Issuer="http://testbed.ist-
phosphorus.eu/phosphorus/aaa/AttributeIssuer" IssueInstant="2008-07-29T14:34:44.796000000+02:00">
      <AttributeValue>demo041</AttributeValue>
    </Attribute>
    <Attribute AttributeId="http://authz-interop.org/AAA/xacml/subject/subject-confdata"
DataType="http://www.w3.org/2001/XMLSchema#string" Issuer="http://testbed.ist-
phosphorus.eu/phosphorus/aaa/AttributeIssuer" IssueInstant="2008-07-29T14:34:44.796000000+02:00">
      <AttributeValue>IGhA11vwa8bUktYhuU9que+d4XLUVJFHrtDC/OE3UilbxtmuCxLldw==</AttributeValue>
    </Attribute>
    <Attribute AttributeId="http://authz-interop.org/AAA/xacml/subject/subject-role"
DataType="http://www.w3.org/2001/XMLSchema#string" Issuer="http://testbed.ist-
phosphorus.eu/phosphorus/aaa/AttributeIssuer" IssueInstant="2008-07-29T14:34:44.796000000+02:00">
      <AttributeValue>researcher</AttributeValue>
    </Attribute>
  </Subject>
  <Resource>
    <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
DataType="http://www.w3.org/2001/XMLSchema#anyURI" Issuer="http://testbed.ist-
phosphorus.eu/phosphorus/aaa/AttributeIssuer" IssueInstant="2008-07-29T14:34:44.765000000+02:00">
      <AttributeValue>http://testbed.ist-phosphorus.eu/viola/harmony</AttributeValue>
```

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |



GAAA toolkit pluggable components and XACML policy profile for ONRP

```
</Attribute>
<Attribute AttributeId="http://authz-interop.org/AAA/xacml/resource/resource-realm"
DataType="http://www.w3.org/2001/XMLSchema#string" Issuer="http://testbed.ist-
phosphorus.eu/phosphorus/aaa/AttributeIssuer" IssueInstant="2008-07-29T14:34:44.765000000+02:00">
  <AttributeValue>testbed.ist-phosphorus.eu</AttributeValue>
</Attribute>
<Attribute AttributeId="http://authz-interop.org/AAA/xacml/resource/target"
DataType="http://www.w3.org/2001/XMLSchema#string" Issuer="http://testbed.ist-
phosphorus.eu/phosphorus/aaa/AttributeIssuer" IssueInstant="2008-07-29T14:34:44.765000000+02:00">
  <AttributeValue>10.7.2.13</AttributeValue>
</Attribute>
<Attribute AttributeId="http://authz-interop.org/AAA/xacml/resource/resource-domain"
DataType="http://www.w3.org/2001/XMLSchema#string" Issuer="http://testbed.ist-
phosphorus.eu/phosphorus/aaa/AttributeIssuer" IssueInstant="2008-07-29T14:34:44.765000000+02:00">
  <AttributeValue>viola</AttributeValue>
</Attribute>
<Attribute AttributeId="http://authz-interop.org/AAA/xacml/resource/resource-type"
DataType="http://www.w3.org/2001/XMLSchema#string" Issuer="http://testbed.ist-
phosphorus.eu/phosphorus/aaa/AttributeIssuer" IssueInstant="2008-07-29T14:34:44.765000000+02:00">
  <AttributeValue>harmony</AttributeValue>
</Attribute>
<Attribute AttributeId="http://authz-interop.org/AAA/xacml/resource/source"
DataType="http://www.w3.org/2001/XMLSchema#string" Issuer="http://testbed.ist-
phosphorus.eu/phosphorus/aaa/AttributeIssuer" IssueInstant="2008-07-29T14:34:44.765000000+02:00">
  <AttributeValue>10.3.1.16</AttributeValue>
</Attribute>
</Resource>
<Action>
  <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
DataType="http://www.w3.org/2001/XMLSchema#string" Issuer="http://testbed.ist-
phosphorus.eu/phosphorus/aaa/AttributeIssuer" IssueInstant="2008-07-29T14:34:44.765000000+02:00">
    <AttributeValue>create-path</AttributeValue>
  </Attribute>
</Action>
</Request>
```

c) Example Response message with returned decision "Permit"

```
<Response>
<Result ResourceId="http://testbed.ist-phosphorus.eu/viola/harmony">
<Decision>Permit</Decision>
<Status>
<StatusCode Value="urn:oasis:names:tc:xacml:1.0:status:ok"/>
</Status>
</Result>
</Response>
```

| | |
|---------------------|--------------------------|
| Project: | Phosphorus |
| Deliverable Number: | D.4.3.1 |
| Date of Issue: | 31/07/2008 |
| EC Contract No.: | 034115 |
| Document Code: | <Phosphorus-WP4-D.4.3.1> |