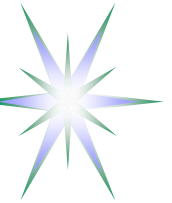


# XACML-Grid and XACML-NRP Attributes and Policy Profiles and Policy Obligations Handling

Overview by Yuri Demchenko  
On behalf of OSG/EGEE AuthZ Interoperability WG and  
Phosphorus project

SNE Group, University of Amsterdam

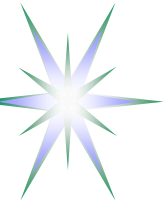
TF-EMC2 Meeting  
3 December 2008, Utrecht



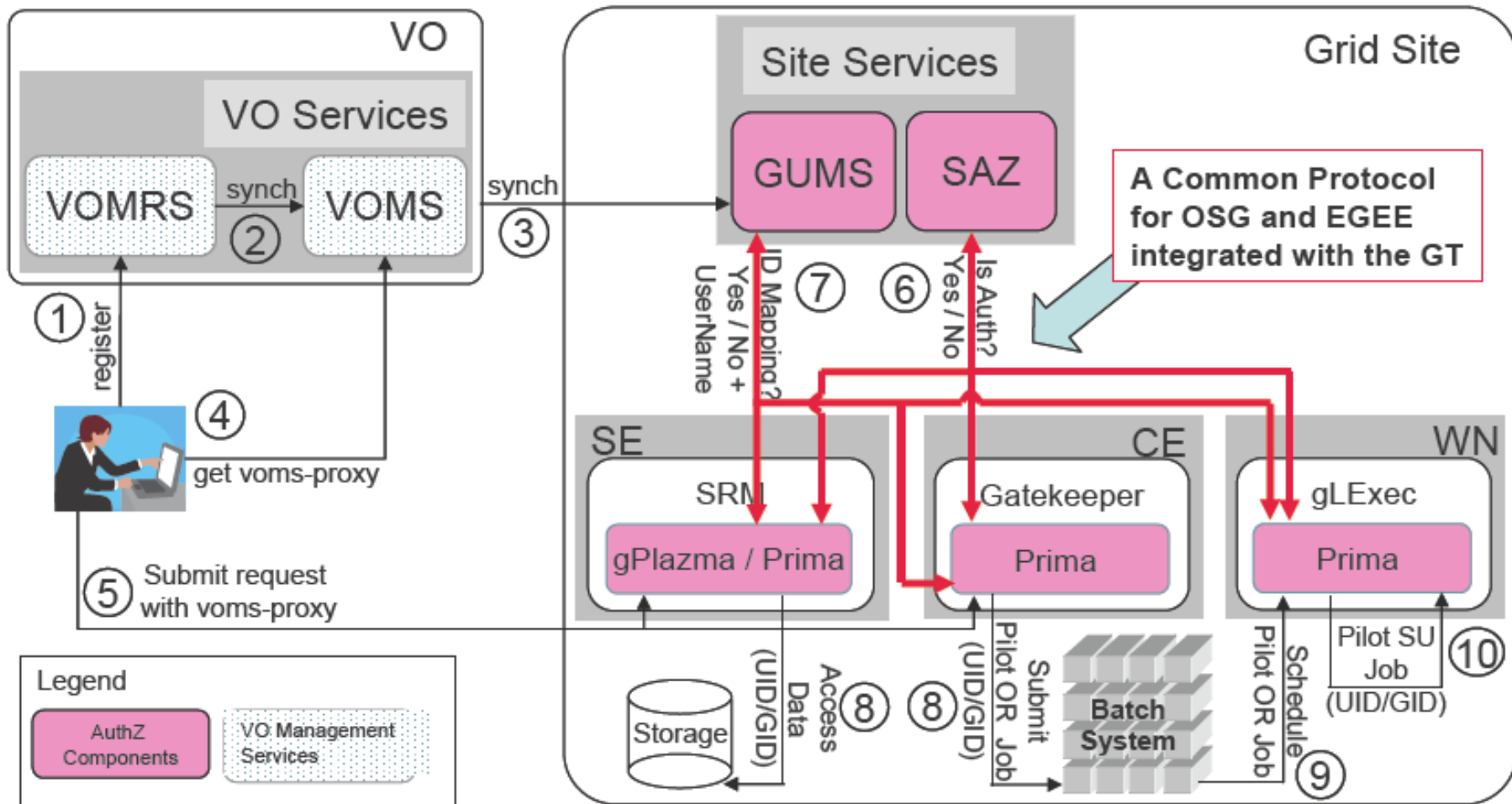
# Outline

---

- OSG/EGEE Grid AuthZ Interop Architecture and Phosphorus Network Resource Provisioning AuthZ infrastructure
- XACML Policy and Policy Obligations
- XACML-Grid attributes – Subject, Resource, Action, Environment, Obligations
- XACML-NRP attributes and examples
- Reference Model for Obligations Handling (OHRM)
- Implementation and experience



# Architecture - OSG view



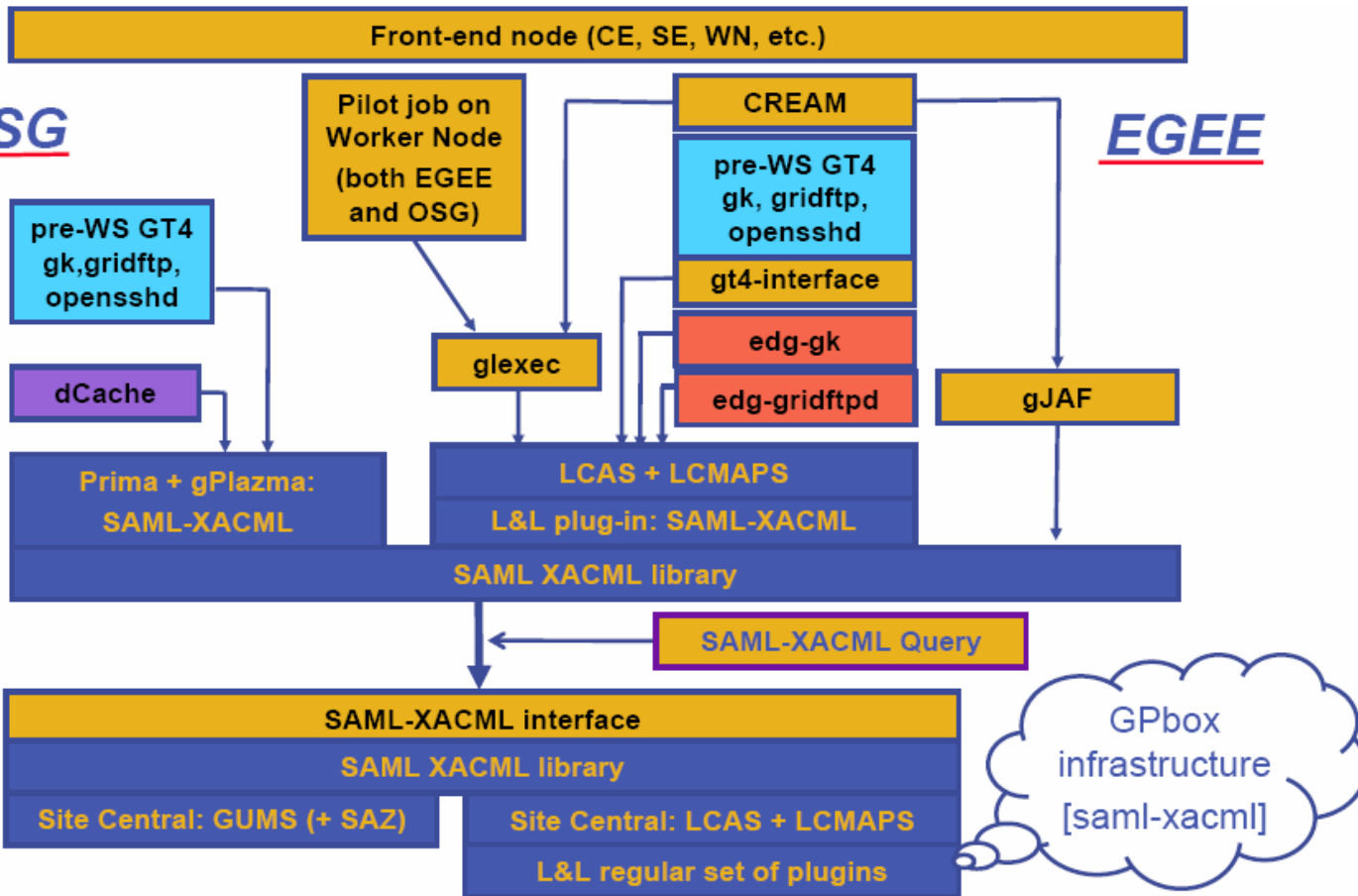
- Mostly based on Globus Toolkit AuthZ framework



# Authorisation Interoperability – EGEE view

**OSG**

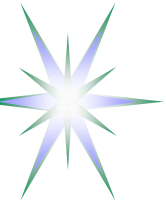
**EGEE**



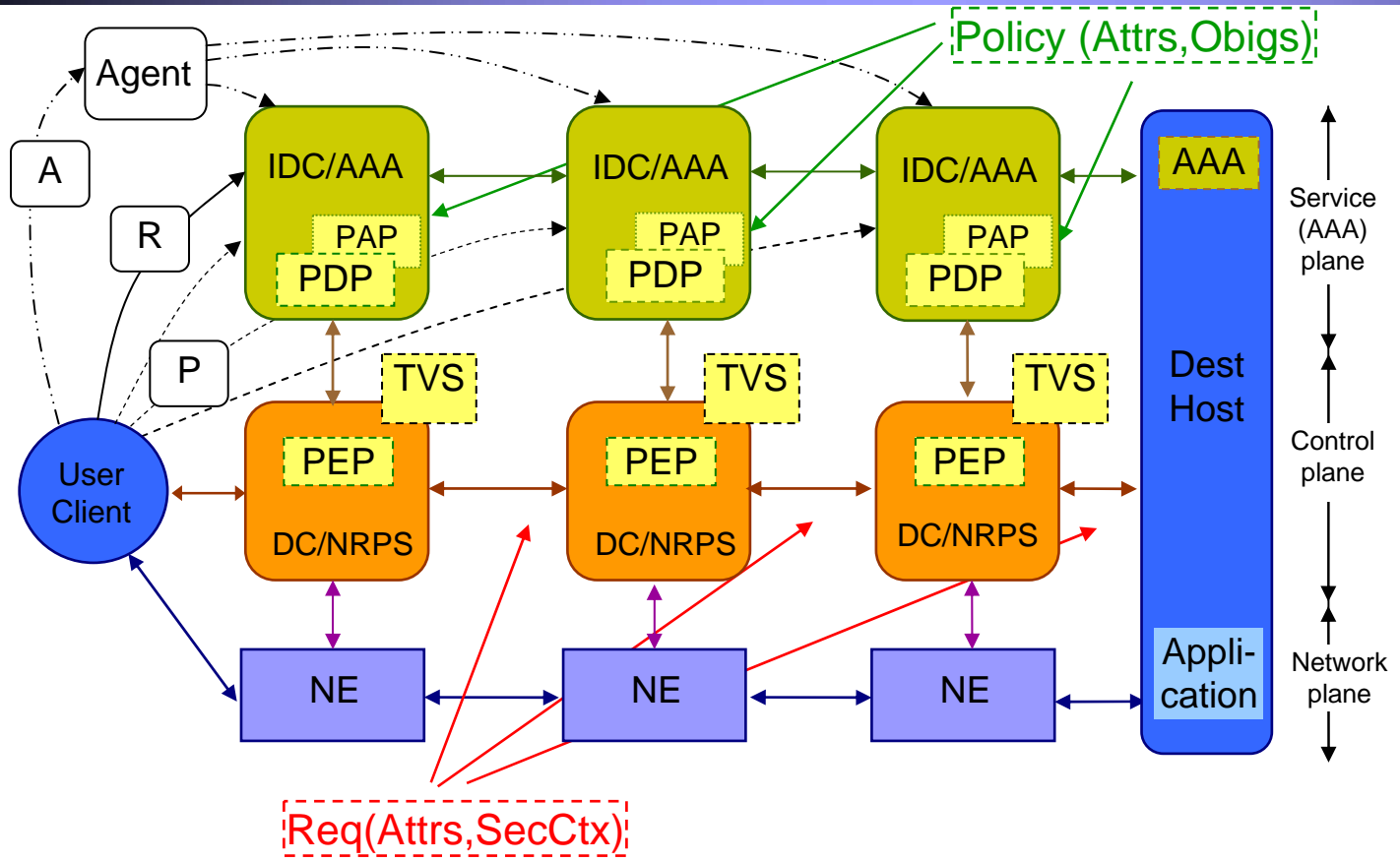
SAML-XACML profile as interoperability framework

Policy Obligation concept/mechanism identified as a solution to allow specific for Grid account mapping and other types of AuthZ decision enforcement (quota, path, priority)

- Introduced Site Central AuthZ Service (SCAS)
- More heterogeneous and LCAS/LCMAPS based
- gLExec as a gateway between Grid environment and CE/WN UNIX execution environment



# Multidomain Network Resource Provisioning (NRP)



## Provisioning sequences

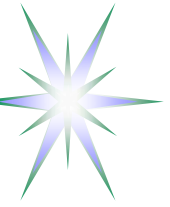
- Agent (A)
- Polling (P)
- Relay (R)

## Token based policy enforcement

GRI – Global Reservation ID  
AuthZ tickets for multidomain context mngnt

IDC – Interdomain Controller  
DC – Domain Controller  
NRPS – Network Resource Provisioning System

AAA – AuthN, AuthZ, Accounting Server  
PDP – Policy Decision Point  
PEP – Policy Enforcement Point  
TVS – Token Validation Service  
KGS – Key Generation Service



# Complex Resource Provisioning (CRP)

Two use case of the general Complex Resource Provisioning (CRP)

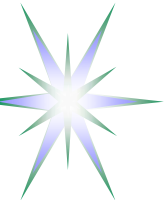
- ONRP and Network on-demand provisioning
- Grid Computing Resource – Distributed and heterogeneous

3 major stages/phases in CRP operation/workflow

- Provisioning consisting of 3 basic steps
  - ◆ Resource Lookup
  - ◆ Resource composition (including options)
  - ◆ Component resources reservation (in advance), including combined AuthZ/policy decision, and assigning a global reservation ID (GRI)
- Deployment – reservation confirmation and distributing components/domain configuration (including trusted keys)
- Access (to the reserved resource) or consumption (of the consumable resource)

Now considering two other stages: “decommissioning” and “relocation”

- Topic for future research and discussions
- Will allows integrating resource provisioning into the upper layer scientific workflow in more consistent way



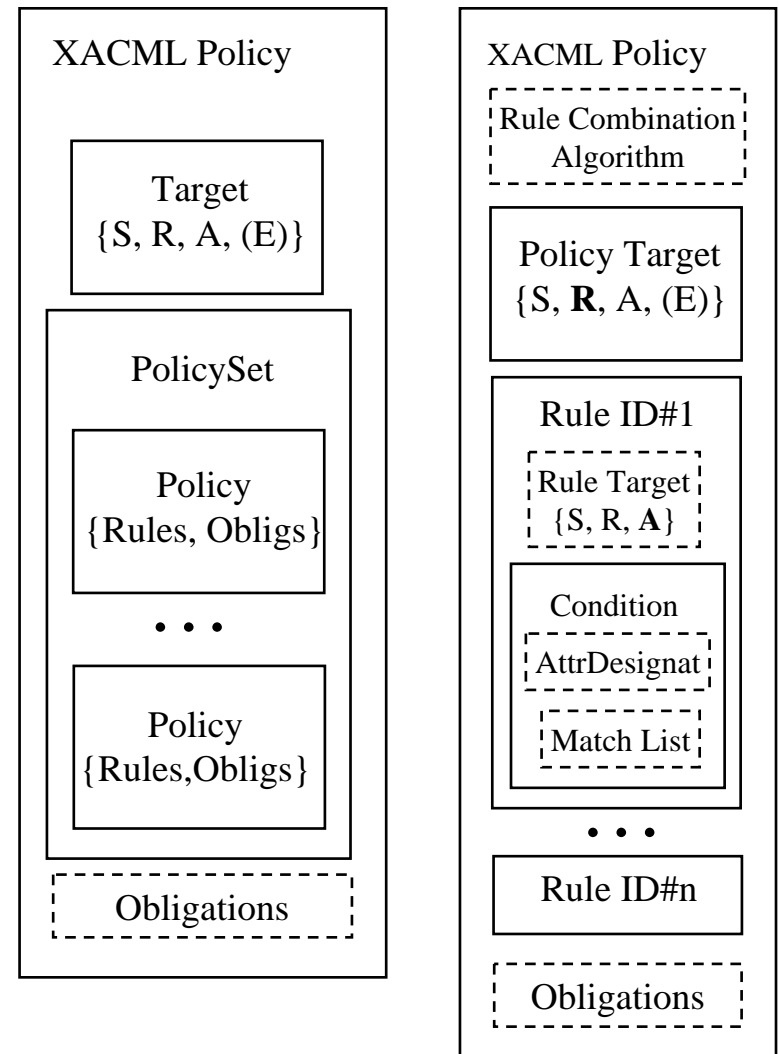
# XACML Policy format

XACML standard specifies XACML policy format and XACML request/response messages

Policy consists of Policy Target and Rules

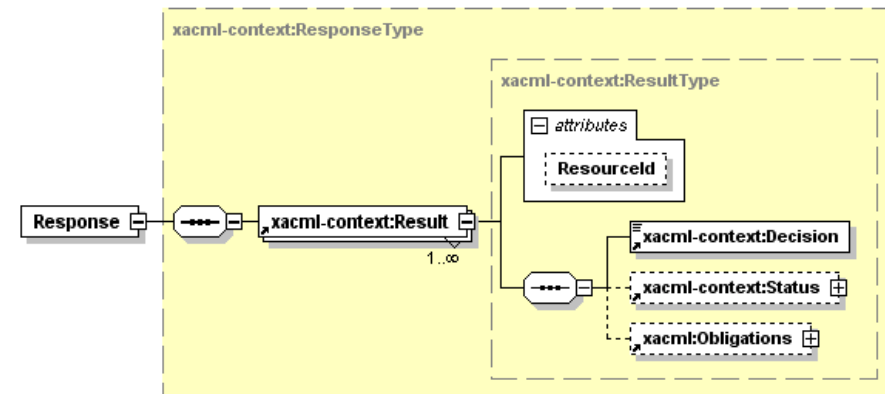
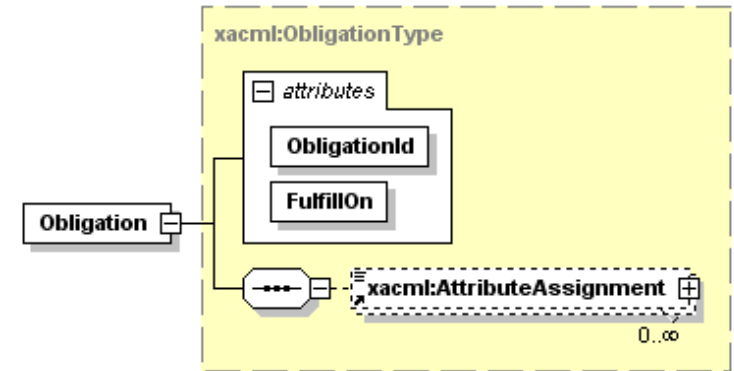
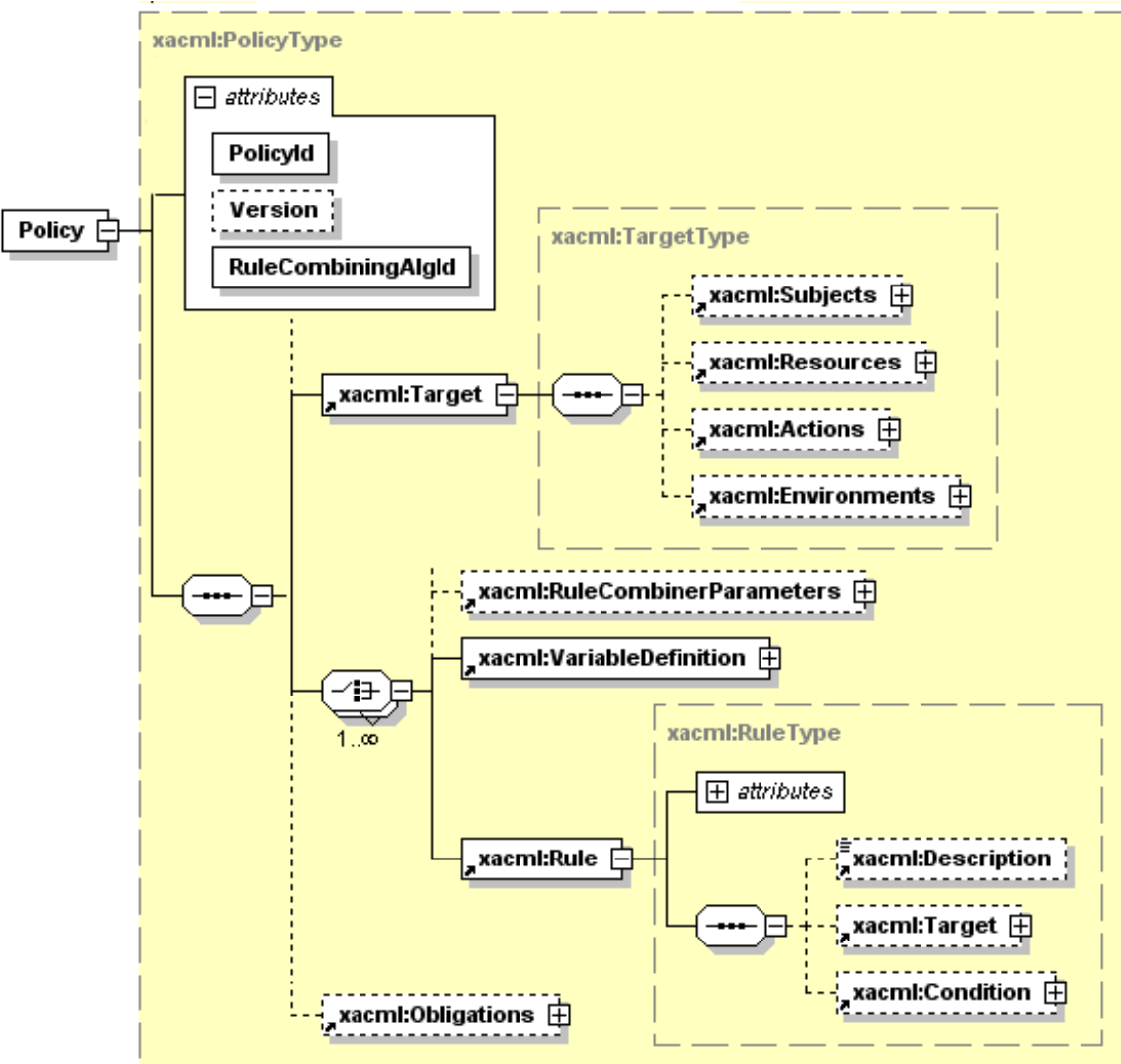
- Policy Target is defined for the tuple Subject-Resource-Action (-Environment)
- Policy Rule consists of Conditions and may contain Obligations
- Obligation defines actions to be taken by PEP on Policy decision by PDP

XACML PDP returns all Obligations that match policy decision (defined by attribute "FulfillOn") from both PolicySet and comprising individual policies

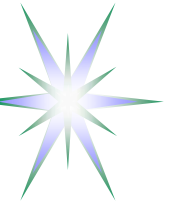




# XACML2.0 Policy Datamodel



XACML Response message contains all Obligations that match policy decision (defined by attribute "FulfillOn") from both PolicySet and comprising individual policies



# XACML Policy Obligations - Definition

Policy Obligation is one of the policy enforcement mechanisms

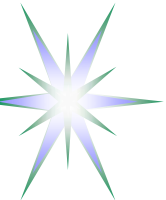
- **Obligations** are a set of operations that must be performed by the **PEP** in conjunction with an **authorization decision** [XACML2.0]

Obligations semantics is not defined in the XACML policy language but left to bilateral agreement between a PAP and the PEP

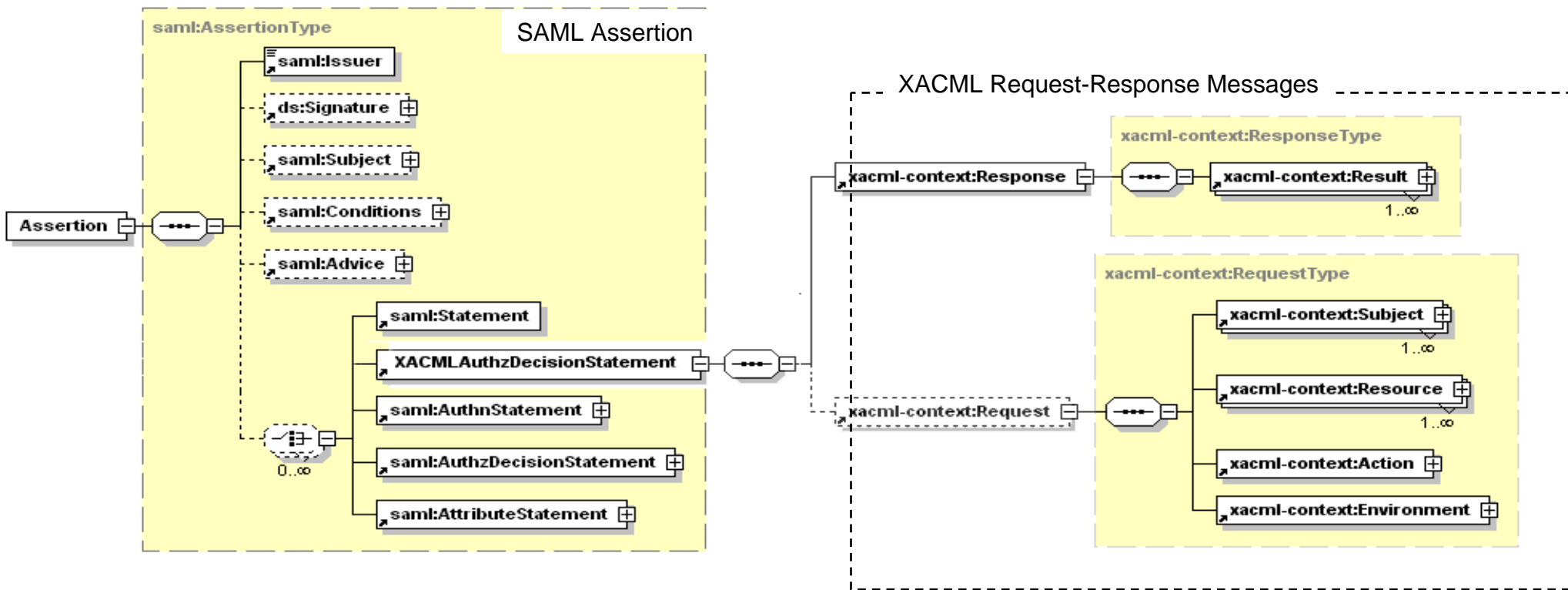
PEPs that conform with XACMLv2.0 are required to deny access unless they understand and can discharge all of the <Obligations> elements associated with the applicable policy

Element <Obligations> / <Obligation>

- The <Obligation> element SHALL contain an **identifier** (in the form of URI) for the obligation and a set of attributes that form arguments of the action defined by the obligation. The FulfillOn attribute SHALL indicate the effect for which this obligation must be fulfilled by the PEP



# SAML2.0 profile of XACML - SAML-XACML Request/Response messages



XACMLRequest (Resource, Subject, Action, Environment)

XACMLResponse (Result (ResourceId, Obligations?))

XACML Request-Response messages are enclosed into the SAML2.0 Assertion or SAML2.0 protocol messages

- Implemented as OpenSAML2.0 extension



# Namespace

Two options were discussed and evaluated - URN vs URL

- URL-style doesn't require centralized registration
- Can be established by registering the (relevant) domain name to ensure uniqueness

XACML-Grid uses registered namespace (owned by David Groep)

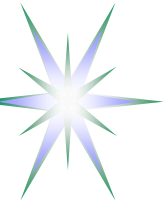
- <http://authz-interop.org/>

Root namespace prefix for all our message elements:

- <http://authz-interop.org/xacml>

XACML Request elements

- Subject: `<ns-prefix>/subject/<subject-attr-name>`
- Action: `<ns-prefix>/<action-attr-name>`
- Resource: `<ns-prefix>/<resource-attr-name>`
- Environment: `<ns-prefix>/environment/<env-type>`



# Subject attributes

---

## Mandatory attributes

- Subject-id  $\Rightarrow$  Subject-X509-id
- Subject-X509-Issuer
- Subject-Condor-Canonical-Name-id
- Subject-VO
- VOMS-signing-subject
- VOMS-signing-issuer
- VOMS-FQAN
- VOMS-Primary-FQAN

## Optional attributes

- Certificate-Serial-Number
- CA-serial-number
- Subject End-Entity X509v3 Certificate Policies OID
- Cert-Chain
- VOMS-dns-port



# Resource attributes

---

## Node-type: (enumerated type)

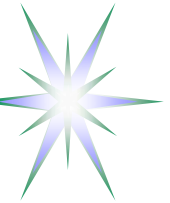
- CE (Computing Element)
- WN (Worker Node)
- SE (Storage Element)

## Host DNS Name

- dns-host-name

## Resource related attributes

- Resource X509 Service Certificate Subject
- Resource X509 Service Certificate Issuer

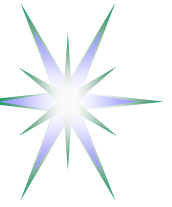


# Action attributes

---

Run-type: expressed as the 'action-id' (enumerated type)

- Queue
  - ◆ Requesting execution to a (remote) queue.
- Execute-Now
  - ◆ Requesting direct execution (remotely)
- Access (file)
  - ◆ Request for (generic) file access
- Resource Specification Language
  - ◆ RSL string



# Environment attributes

---

## PEP-PDP capability negotiation - Supported Obligations

- PEP sends to PDP a list of the supported obligations
- The PDP can choose to return an appropriate set of obligations from this list
- Allows upgradeability of the PEPs and PDPs independently by deploying new functionalities step by step

## Pilot Job context

- To support pull-based job management model
- Policy statement example
  - ◆ “User access to the WM execution environment can be granted only if the pilot job belongs to the same VO as the user VO”
- Pilot job invoker identity
  - ◆ These attributes the the identity of the pilot job invoker



# XACML-Grid Obligations

Uses simplified Obligations expression format

**Obligation = {AttributeAssignment (ObligationId, AttributeValue(Attributeld))}**

ObligationId: *<ns-prefix>/obligation/<obligation-name>*

Attributeld: *<ns-prefix>/attributes/<obligation-attribute-name>*

Supported Obligation types

[T] [S] UID + GID\_(i.e. Unix User ID and Group ID local to the PEP

- Must be consistent with: Username

[T] [S] Multiple secondary GIDs - Requires UID+GID

[T/E] [R] AFS token (type string) - Requires UID+GID

[E] [S] Username (for CE) - Requires UID+GID

[T/E] [R] Path restriction - Root and home path

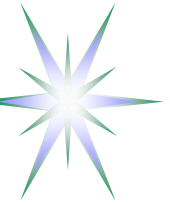
[A] [S] Storage priorities (gPlazma) - Requires UID+GID or Username

[E] [S] Access permission - Requires UID+GID or Username

Legend: [T] – policy may use template Obligation

[E] - policy may use explicit Obligation

[S], [R], [A] – Obligation applied to AuthZ Subject, Resource, Action



# XACML-NRP Profile – Work in Progress

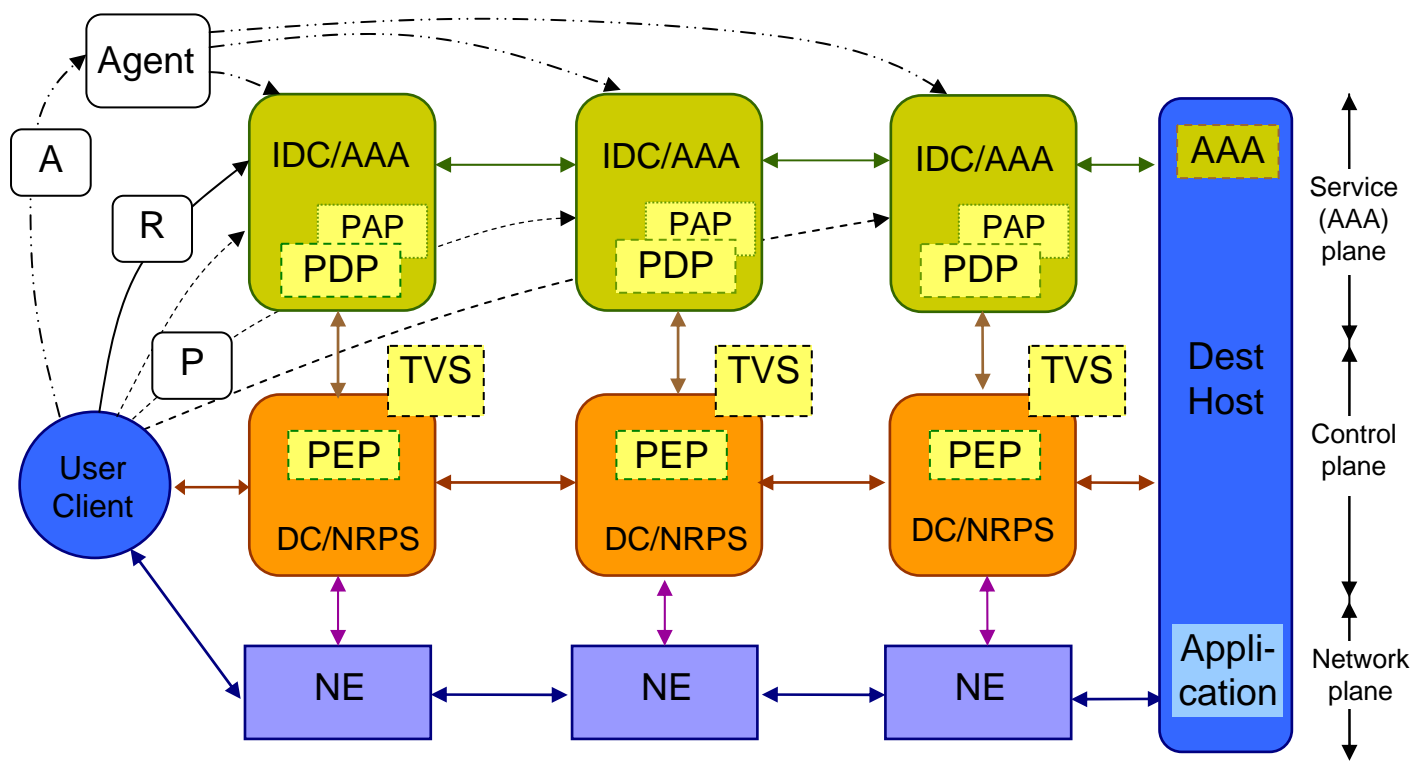
---

## XACML-NRP Authorisation Interoperability profile for Network Resource Provisioning

- Phosphorus technical document. Current draft - <http://staff.science.uva.nl/~demch/projects/aaauthreach/draft-interop-xacml-nrp-profile-012.pdf>
- Also part of the Phosphorus deliverable D.4.3.1 - "GAAA toolkit pluggable components and XACML policy profile for ONRP"  
<http://staff.science.uva.nl/~demch/worksinprogress/Phosphorus-WP4-D4.3.1-GAAA-TK-library-NRP-v04.pdf>
- Incorporate and extends XACML-Grid profile



# Multidomain Network Resource Provisioning (NRP)



## Provisioning sequences

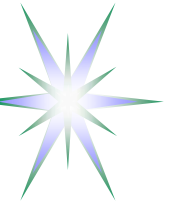
- Agent (A)
- Polling (P)
- Relay (R)

## Token based policy enforcement

GRI – Global Reservation ID  
AuthZ tickets for multidomain context mngnt

IDC – Interdomain Controller  
DC – Domain Controller  
NRPS – Network Resource Provisioning System

AAA – AuthN, AuthZ, Accounting Server  
PDP – Policy Decision Point  
PEP – Policy Enforcement Point  
TVS – Token Validation Service  
KGS – Key Generation Service



# Basic use cases for policy definition in NRP

---

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"

Use case 3: "User/Group A is only allowed to invoke method/action X, Y, and Z"

Use case 4: "User/Group A is only allowed to invoke method X,Y, and Z based on session delegation"



# Example of the Resource attributes expression

Attribute name	Attribute ID	Full XACML attributeld semantics (ns-prefix = http://authz-interop.org/nrp/xacml)
Domain ID	domain-id	{ns-prefix} /resource/domain-id
Subdomain	subdomain	{ns-prefix} /resource/sub-domain
VLAN	vlan	{ns-prefix} /resource/vlan
TNA	tna (+ tna-prefix)	{ns-prefix} /resource/tna-prefix/tna
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
Resource type	resource-type	{ns-prefix} /resource/resource-type ({ns-prefix} /resource/device)
Resource federation	federation	{ns-prefix} /resource/federation

- Domain ID (network domain)
- Subdomain (or relationship)
- Node or TNA and TNA prefix
- Interface ID or Link ID
- Device or resource-type
- 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

3 topology description formats were reviewed

- Phosphorus Harmony/NSP
- NDL by UvA
- OSCARS (currently used)



# Subject related Attributes

<b>Attribute name</b>	<b>Attribute ID</b>	<b>Full XACML attributeld semantics (ns-prefix = http://authz- interop.org/nrp/xacml)</b>
Subject ID	subject-id	{ns-prefix} /subject/subject-id
Subject confirmation	subject-confdata	{ns-prefix} /subject/subject-confdata
Subject Context	subject-context	{ns-prefix} /subject/subject-context
Subject group	subject-group	{ns-prefix} /subject/subject-group
Subject role	subject-role	{ns-prefix} /subject/subject-role
Subject federation	Federation	{ns-prefix} /subject/federation



# Action related Attributes and Enumerated values

<b>Attribute name</b>	<b>Attribute ID</b>	<b>Full XACML attributeld semantics (ns-prefix = http://authz-interop.org/nrp/xacml)</b>
Action ID	action-id	{ns-prefix} /action/action-id
Action type	action-type	{ns-prefix} /action/action-type/{value}

<b>Attribute name</b>	<b>Enumerated value</b>	<b>XACML attribute value (ns-prefix = http://authz-interop.org/nrp/xacml)</b>
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



# Harmony/NSP input Resource URI and Subject data and XACML Request (1) – Resource and Subject attributes

**ResourceInputURI =**

```
"http://testbed.ist-phosphorus.eu/viola/harmony/source=10.3.1.16/target=10.7.3.13"
```

```
ResMap = resource-id=http://testbed.ist-phosphorus.eu/viola/harmony,  
resource-realm=testbed.ist-phosphorus.eu  
resource-domain=viola  
resource-type=harmony  
source=10.3.1.16  
target=10.7.3.13
```

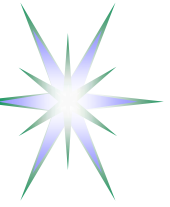
```
SubjMap = subject-id=WHO740@users.testbed.ist-phosphorus.eu, subject-role=researcher,  
subject-context=demo041, subject-confdata="IGhA11...8bUktYh" }
```

```
Policy file = data/policy/nrp/testbed.ist-phosphorus.eu/viola-policy-harmony-demo041.xml
```



# Harmony/NSP input Resource URI and Subject data and XACML Request (2) – XACML Request Message

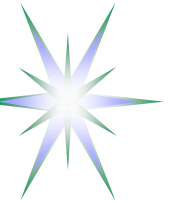
```
<Request>
<Subject SubjectCategory="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject">
<Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
  DataType="http://www.w3.org/2001/XMLSchema#string" Issuer="http://testbed.ist-
  phosphorus.eu/phosphorus/aaa/AttributeIssuer" IssueInstant="2008-12-03T12:10:21.218000000+01:00">
  <AttributeValue>WHO740@users.testbed.ist-phosphorus.eu</AttributeValue></Attribute>
<Attribute AttributeId="http://authz-interop.org/AAA/xacml/subject/subject-role">
  <AttributeValue>researcher</AttributeValue></Attribute>
<Attribute AttributeId="http://authz-interop.org/AAA/xacml/subject/subject-context">
  <AttributeValue>demo041</AttributeValue></Attribute>
<Attribute AttributeId="http://authz-interop.org/AAA/xacml/subject/subject-confdata">
  <AttributeValue>IGhAllvwa8bUk...xtmuCxLldw==</AttributeValue></Attribute>
</Subject>
<Resource>
<Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id">
  <AttributeValue>http://testbed.ist-phosphorus.eu/viola/harmony</AttributeValue></Attribute>
<Attribute AttributeId="http://authz-interop.org/AAA/xacml/resource/resource-realm">
  <AttributeValue>testbed.ist-phosphorus.eu</AttributeValue></Attribute>
<Attribute AttributeId="http://authz-interop.org/AAA/xacml/resource/resource-domain">
  <AttributeValue>viola</AttributeValue></Attribute>
<Attribute AttributeId="http://authz-interop.org/AAA/xacml/resource/target">
  <AttributeValue>10.7.3.13</AttributeValue></Attribute>
<Attribute AttributeId="http://authz-interop.org/AAA/xacml/resource/source">
  <AttributeValue>10.3.1.16</AttributeValue></Attribute>
</Resource>
<Action>
<Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id" >
  <AttributeValue>create-path</AttributeValue></Attribute>
</Action></Request>
```



# Environment related Attributes

---

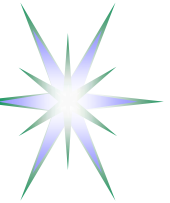
- Last-domain conformation
- Authorisation context
  - ◆ AuthZ session credentials or AuthZ ticket
- Delegation or Obligations from the previous domain
  - ◆ User ID or group to which access is delegated
  - ◆ Actions which need to be taken when processing request or granting access



# XACML-NRP Policy Obligations

## Suggested policy obligations for multidomain NRP

- Intra-domain network/VLAN mapping for cross-domain connections
  - ◆ Can be used to map external/interdomain border links/endpoints 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 e.g. advance resource reservation
  - ◆ **Fixed ARR** that implies strict time/amount constraints
  - ◆ **Deferrable ARR** that allows some degree of freedom in the time domain with fixed amount (or bandwidth)
  - ◆ **Malleable ARR** that allows variable duration and amount for the fixed consumption amount



# XACML Obligations – Implementation suggestions

**Obligation = Apply (TargetAttribute, Operation (Variables)), or**

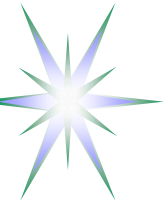
**Obligation = Apply (TargetAttribute, Operation (Variables), Chronicle)**

## Obligations enforcement scenarios

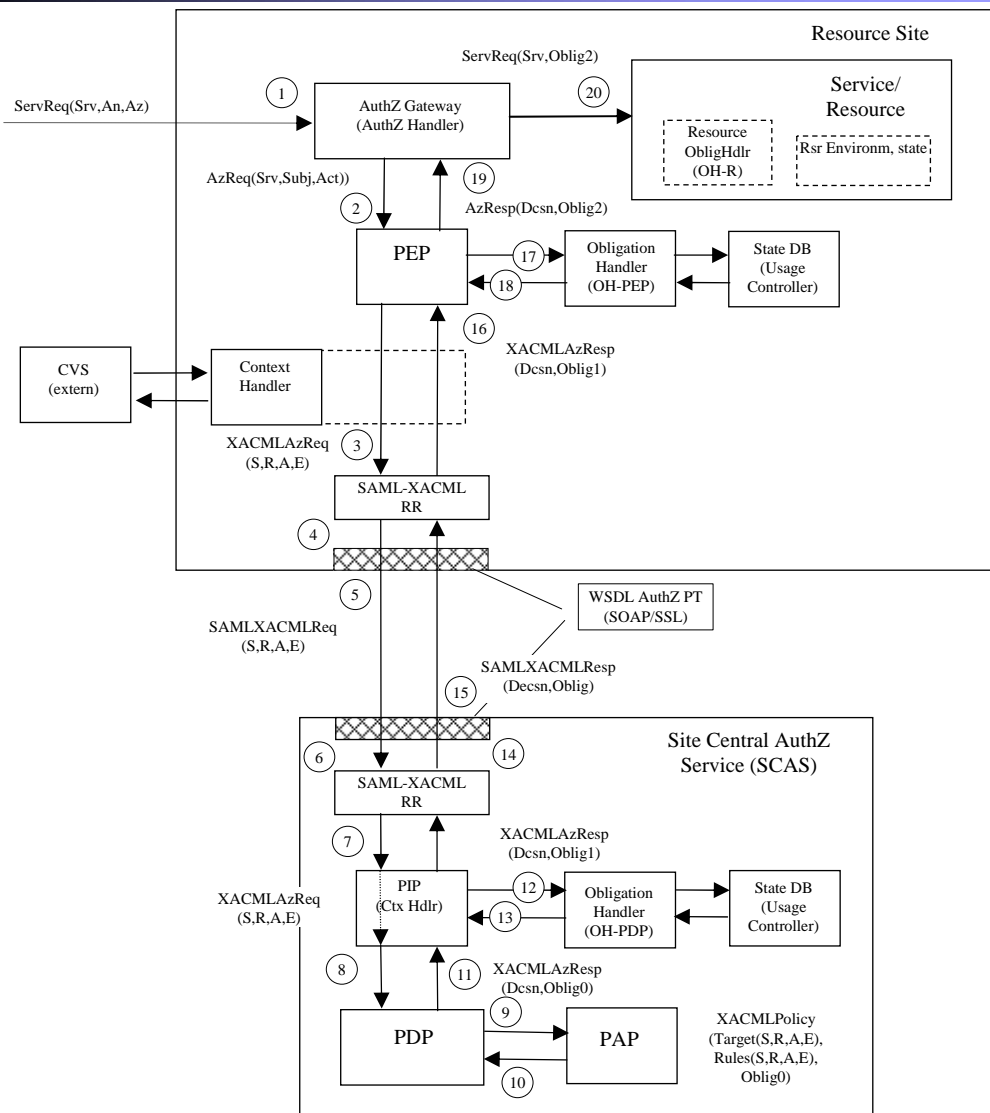
- Obligations are enforced by PEP at the time of receiving obligated AuthZ decision from PDP
- Obligations are enforced at later time when the requestor accesses the resource or service
- Obligations are enforced before or after the resource or service accessed/delivered/consumed

Obligation handling model was proposed as complimentary to XACML-Grid profile developed by OSG, EGEE, and Globus AuthZ interoperability WG

- ObligationId (of type URI) has to be mapped to a specific handler that is called by the PEP
- Obligation parameter values are passed to handler
- Handler returns True/False that determines PEP's Permit/Deny



# Proposed Obligations Handling Reference Model



## Generic AuthZ service model

PEP – Policy Enforcement Point

PDP – Policy Decision Point

PAP – Policy Authority Point

OH – Obligation Handler

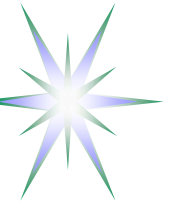
CtxHandler – Context Handler

(S, R, A, E) – components of the AuthZ request

(Subject, Resource, Action, Environment)

## 4 stages in Obligations processing

- Obligation0 – Obligations in the policy
- Obligation1 & Obligation2 – stages of the XACML Response processing
- Obligation3 – may require a kind of the secure container (AuthZ ticket or SAML Assertion)



# Obligations Handling Stages

Obligation0 = tObligation => Obligation1 (“OK?”, (Attributes1 v Environments1))  
=> Obligation2 (“OK?”, (Attributes2 v Environments2))  
=> Obligation3 (Attributes3 v Environments3)

## Obligation0 – (stateless or template)

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.

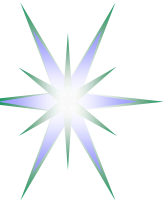
## Obligation1 and Obligation 2

Obligations have been handled by Obligation handler at the SCAS/PDP side or at the PEP side, depending on implementation. Templates or instructions of the Obligation0 are replaced with the real attributes in Obligation1/2, e.g. in a form of “name-value” pair.

- The result of Obligations processing/enforcement is returned in a form of modified AuthzResponse (Obligation1) or global Resource environment changes
- 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.
- Note. Obligation1 handling at the SCAS or PDP side allows stateful PDP/SCAS.

## Obligation3

Final stage when an Obligation actually takes effect (Obligations “termination”). This is done by the Resource itself or by services managed/controlled by the Resource.



# XACML Obligations – Examples of expression for pool account mapping in Grid – Option 1 (used in XACML-Grid v1.0)

```
<!-- Obligations format option 1 (simple): UID, GID explicitly mentioned as separate XML
elements inside AttributeAssignment element -->
<xacml:Obligations>
  <xacml:Obligation
    ObligationId=http://authz-interop.org/xacml/obligation/uidgid
    FulfillOn="Permit">
    <xacml:AttributeAssignment
      AttributeId=http://authz-interop.org/xacml/attribute/posix-uid
      DataType="http://www.w3.org/2001/XMLSchema#integer">
      2501</xacml:AttributeAssignment>
    <xacml:AttributeAssignment
      AttributeId=http://authz-interop.org/xacml/attribute/posix-gid
      DataType="http://www.w3.org/2001/XMLSchema#integer">
      2101</xacml:AttributeAssignment>
    </xacml:Obligation>
  </xacml:Obligations>
```



# XACML Obligations – Examples of expression for pool account mapping in Grid – Option 2

```
<Obligations>
<Obligation ObligationId="http://authz-interop.org/xacml/obligation/map.poolaccount/t0"
  FulfillOn="Permit">
  <!-- Specifies to what kind of attribute the next 'map.to' action is applied to -->
  <AttributeAssignment
AttributeId="urn:oasis:names:tc:xacml:2.0:example:attribute: requesting-subject"
DataType="http://www.w3.org/2001/XMLSchema#string">
  &lt;SubjectAttributeDesignator
    AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
    DataType="http://www.w3.org/2001/XMLSchema#string"/&gt;
  </AttributeAssignment>

  <!-- This is actual account attribute name/value to which it should be mapped -->
  <AttributeAssignment
    AttributeId="http://authz-interop.org/xacml/obligation/attribute/uidgid/t0"
    DataType="http://www.w3.org/2001/XMLSchema#string">
    &lt;UnixId DataType="http://www.w3.org/2001/XMLSchema#string"&gt;
      okoeroo&gt;UnixId&gt;
    &lt; GroupPrimary DataType="http://www.w3.org/2001/XMLSchema#string"&gt;
      computergroup&gt;GroupPrimary&gt;
    &lt;GroupSecondary DataType="http://www.w3.org/2001/XMLSchema#string"&gt;
      datagroup&gt;GroupSecondary&gt;
  </AttributeAssignment>
</Obligation>
</Obligations>
```

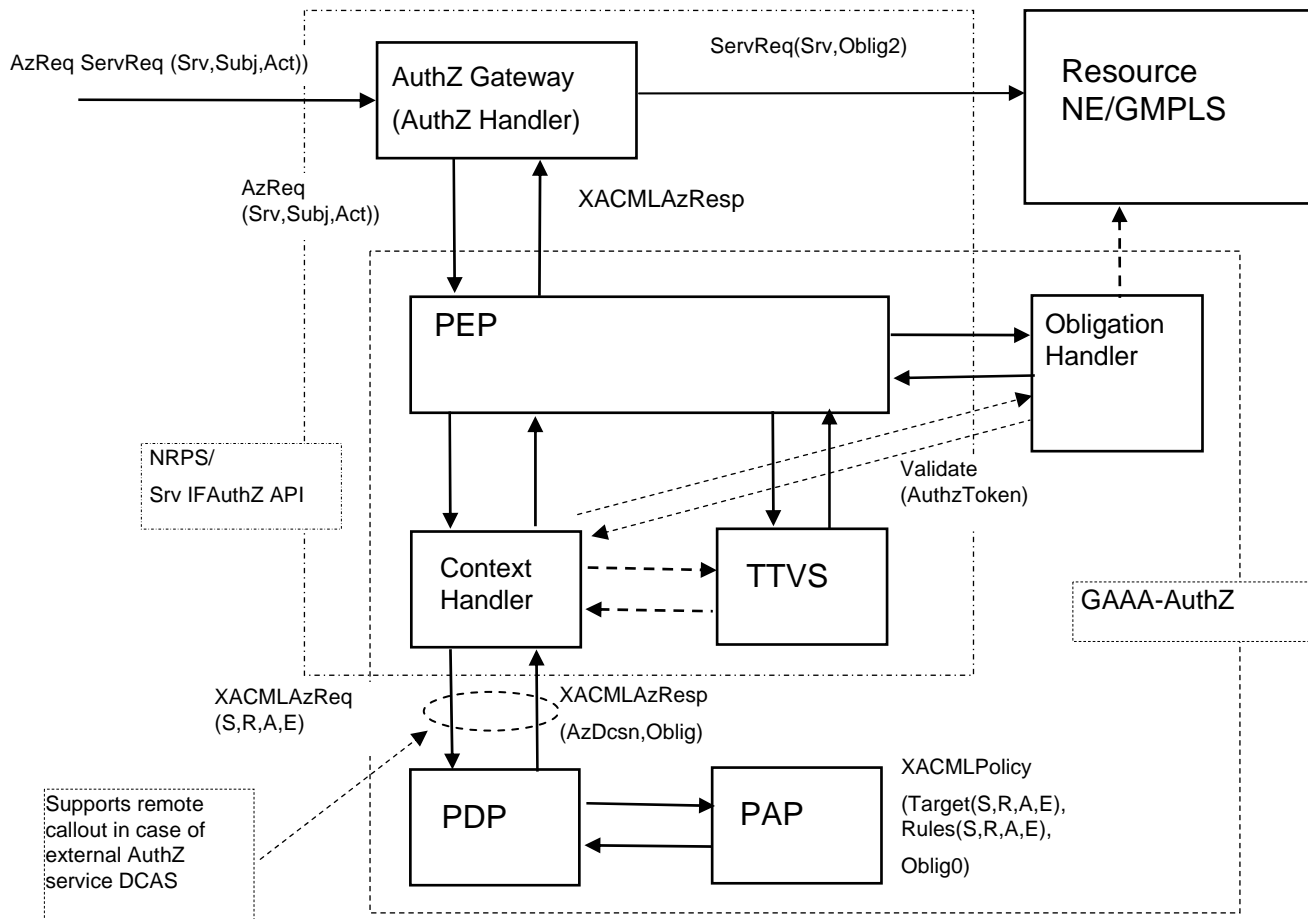


# XACML-NRP Implementation – GAAA-TK Java library

- XACML-NRP profile is implemented in the GAAA-TK Java library
  - ◆ Intended to be compatible with Globus Toolkit AuthZ framework
- GAAA-TK library provides all necessary AuthZ mechanisms and service components to support AuthZ sessions context and Obligations handling
  - ◆ Supports SAML2.0 profile of XACML – protocol and request/response messages
- AuthZ ticket format for extended AuthZ session management
  - ◆ To allow extended AuthZ decision/security context communication between domains
- Access token and pilot tokens used for access control and signalling
  - ◆ Supported by the Token Validation Service (TVS) functionality
  - ◆ Can be used transparently at all Networking layers (Service, Control and Data planes)
- Integrated into the Phosphorus project Network Service Plane (NSP) test-bed and uses simple XACML policy model
  - ◆ Part of the Phosphorus project deliverable D.4.3.1 - "GAAA toolkit pluggable components and XACML policy profile for ONRP"  
<http://staff.science.uva.nl/~demch/worksinprogress/Phosphorus-WP4-D4.3.1-GAAA-TK-library-NRP-v04.pdf>



# GAAA Toolkit pluggable AAA/AuthZ components



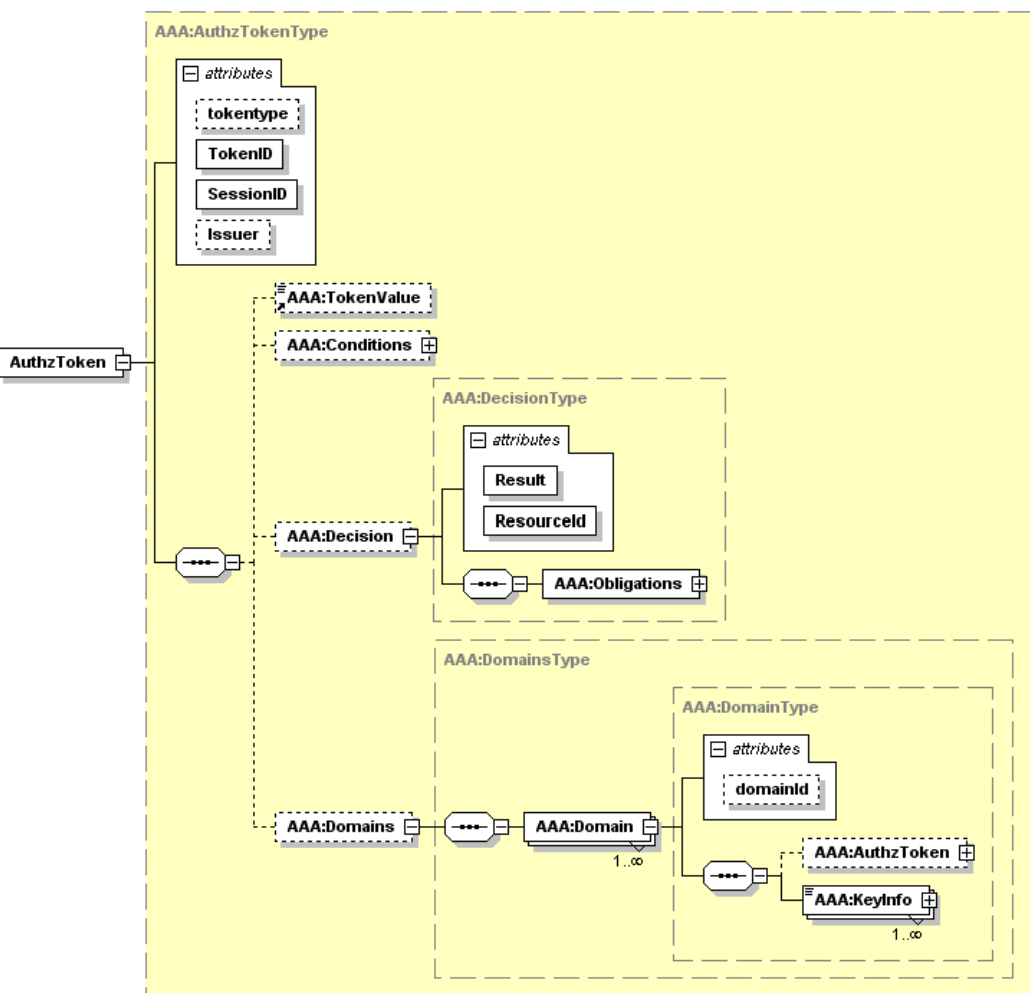
The proposed model intends to comply with both the generic AAA-AuthZ framework and XACML AuthZ model

- ContextHandler functionality can be extended to support all communications between PEP-PDP and with other modules

TTVS – Ticket and token validation and handling service



# General XML Token Format – Access and Pilot Tokens



Generated by XmlSpy

www.altova.com

Required functionality to support multidomain provisioning scenarios

- Allows easy mapping to SAML and XACML related elements

Allows multiple Attributes format (semantics, namespaces)

Establish and maintain Trust relations between domains

- Including Delegation

Ensure Integrity of the AuthZ decision

- Keeps AuthN/AuthZ context
- Allow Obligated Decisions (e.g. XACML)



# Pilot Token Types

**Type 0** – Access token (refers to the reserved resources context)

**Type 1** – Container for communicating the GRI during the reservation stage

- Contains the mandatory SessionId=GRI attribute and an optional Condition element

**Type 2** – Origin/requestor authenticating token

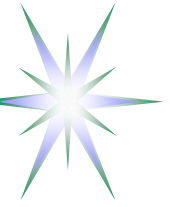
- TokenValue element contains a value that can be used as the authentication value for the token origin
- TokenValue may be calculated of the (GRI, IssuerId, TokenId) by applying e.g. HMAC function with the requestor's symmetric or private key.

**Type 3** – Extends Type 2 with the Domains element that allows collecting domains security context information when passing multiple domains during the reservation process

- Domains' information may include the previous token and the domain's trust anchor or public key.

**Type 4** – Used at the deployment stage and can communicate between domains security context information about all participating in the provisioned lightpath or network infrastructure resources

- Can be used for programming/setting up a TVS infrastructure for consistent access control tokens processing at the resource access stage



# XML access token format - Example

```
<AAA:AuthzToken xmlns:AAA="http://www.aaauthreach.org/ns/#AAA"
  Issuer="urn:aaa:gaaapi:token:TVS" type="access-type1"
  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

SessionId = GRI (Global Reservation Id)

TokenId – unique identifier (serving for logging and accountability)

TokenValue – generated securely from GRI or AuthzTicket (digital SignatureValue)

- The element <TokenValue> and attributes SessionId and TokenId are mandatory, and the element <Conditions> and attributes Issuer, NotBefore, NotOnOrAfter are optional
- Binary token contains just two values – TokenValue and GRI



# PEP Methods supporting token-based access control and signalling

## Method #7 - simple intra-domain delegation

```
boolean authorizeActionSession (String authzToken, String griReq, int delegtype, HashMap resmap, HashMap actmap, HashMap subjmap)
```

- This method allows for flexible session based access control and delegation
  - ◆ AuthzToken is used as session credential intra-domain and supports basic delegation scenarios
    - session (i.e. path creation) can be started privileged use e.g. researcher
    - if token is valid, all other users can perform their allowed actions
    - different scenarios may limit scope of session based delegation, e.g. only own domain, etc.

## Method#8 - intra-domain session initiation and simple delegation (can issue session credentials of different token/ticket types)

```
String authorizeActionSession (String authzToken, String grireq, int delegtype, int sescred, HashMap resmap, HashMap actmap, HashMap subjmap)
```

## Method #9 - Extends method #8 for inter-domain reservation/access control scenario (including simple delegation)

```
String authorizeActionSession (String authzToken, String griReq, int delegtype, int sescredtype, boolean renew, HashMap resmap, HashMap actmap, HashMap subjmap)
```

- Returns:
  - ◆ renewed session/AuthzToken if renew = (1,2) or token=null and requested sescred supported
  - ◆ or string "Permit" or "Deny" depending on PDP decision



# TVS functionality – Access control and signalling

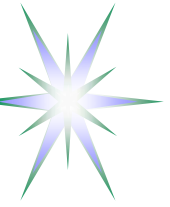
---

Basic TVS functionality is checking validity of an access token received from the PEP or AuthZ gateway/service

- Extended TVS functionality allow token re-building when processing request from the previous domain and relaying to the next domain
  - ◆ Special method to Validate&Relay pilot tokens
- Additionally, TVS may be used for security context token, e.g. token key(s) distribution at the reservation stage or at the stage of the reserved resource deployment

TVS supports pilot tokens handling functionality used during the reservation stage

- Can be used for building dynamic security association of the reserved resources



# Future developments

---

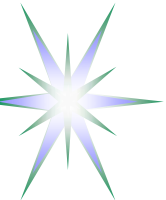
- Obligations Handling API (OH-API) implementing OHRM in GAAA-TK
- GAAA-TK library interoperability and integration with Globus Toolkit AuthZ Framework, in particular OH-API
- OHRM and restricted delegation to support multidomain reservation process and resource access
- Moving XACML-Grid and XACML-NRP profile to the OGF standardisation process
- Conformance test for XACML-Grid profile and XACML-NRP



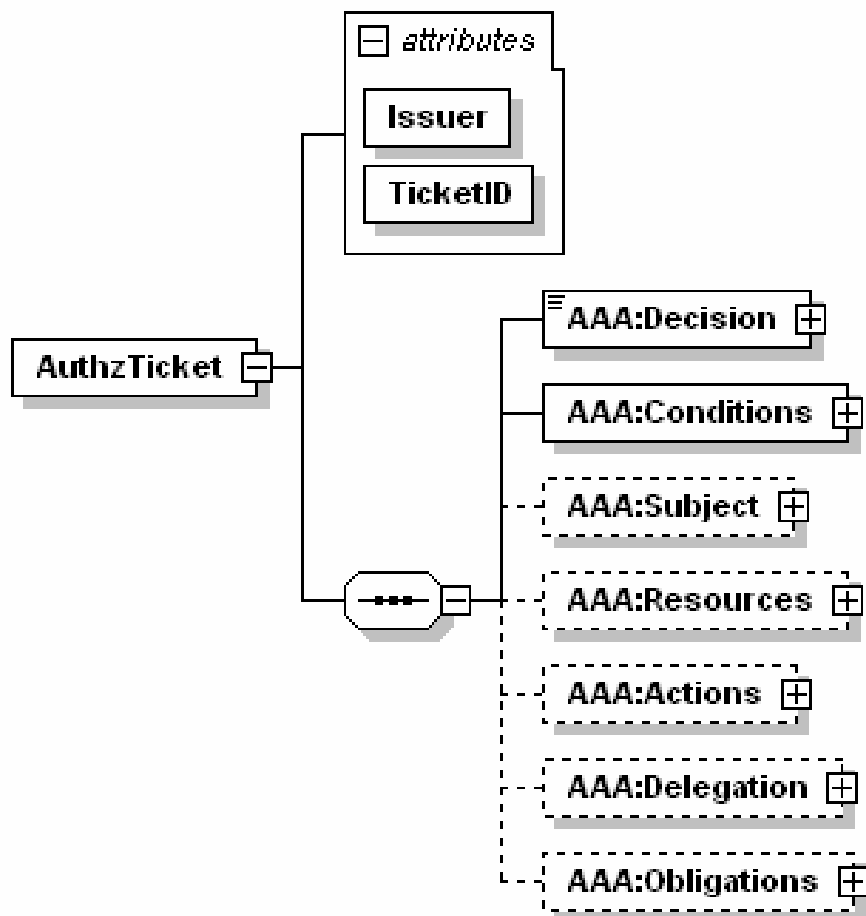
# Additional Information

---

- AuthZ ticket format and example
- OpenSAML SAML-XACML Extension Library
  
- For XACML-Grid profile details
  - ◆ <http://cd-docdb.fnal.gov/cgi-bin/ShowDocument?docid=2685>
  - ◆ <https://edms.cern.ch/document/929867/1>
- For XACML-NRP profile details
  - ◆ <http://staff.science.uva.nl/~demch/projects/aaauthreach/draft-interop-xacml-nrp-profile-012.pdf>



# AuthZ ticket/assertion for extended security context management – Data model (1) - Top elements



Required functionality to support multidomain provisioning scenarios

- Allows easy mapping to SAML and XACML related elements

Allows multiple Attributes format (semantics, namespaces)

Establish and maintain Trust relations between domains

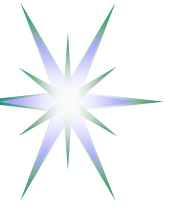
- Including Delegation

Ensure Integrity of the AuthZ decision

- Keeps AuthN/AuthZ context
- Allow Obligated Decisions (e.g. XACML)

Confidentiality

- Creates a basis for user-controlled Secure session



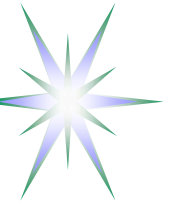
# AuthZ ticket main elements

- <Decision>** element - holds the PDP AuthZ decision bound to the requested resource or service expressed as the ResourceID attribute.
- <Conditions>** element - specifies the validity constraints for the ticket, including validity time and AuthZ session identification and additionally context
- <ConditionAuthzSession>** (extendable) - holds AuthZ session context
- <Subject>** complex element - contains all information related to the authenticated Subject who obtained permission to do the actions
- <Role>** - holds subject's capabilities
  - <SubjectConfirmationData>** - typically holds AuthN context
  - <SubjectContext>** (extendable) - provides additional security or session related information, e.g. Subject's VO, project, or federation.
- <Resources>/<Resource>** - contains resources list, access to which is granted by the ticket
- <Actions>/<Action>** complex element - contains actions which are permitted for the Subject or its delegates
- <Delegation>** element – defines who the permission and/or capability are delegated to: another **DelegationSubjects** or **DelegationCommunity**
- attributes define restriction on type and depth of delegation
- <Obligations>/<Obligation>** element - holds obligations that PEP/Resource should perform in conjunction with the current PDP decision.



# AuthZ ticket format (proprietary) for extended security context management

```
<AAA:AuthzTicket xmlns:AAA="http://www.aaauthreach.org/ns/#AAA" Issuer="urn:cnl:trust:tickauth:pep"
  TicketID="cba06d1a9df148cf4200ef8f3e4fd2b3">
  <AAA:Decision ResourceID="http://resources.collaboratory.nl/Philips_XPS1">Permit</AAA:Decision>
  <!-- SAML mapping: <AuthorizationDecisionStatement Decision="*" Resource="*"> -->
  <AAA:Actions>
  <AAA:Action>cnl:actions:CtrlInstr</AAA:Action>      <!-- SAML mapping: <Action> -->
  <AAA:Action>cnl:actions:CtrlExper</AAA:Action>
  </AAA:Actions>
  <AAA:Subject Id="subject">
  <AAA:SubjectID>WHO740@users.collaboratory.nl</AAA:SubjectID>      <!-- SAML mapping: <Subject>/<NameIdentifier> -->
  <AAA:SubjectConfirmationData>IGhA11vwa8YQomTgB9Ege9JRNld84AggaDkOb5WW4U=</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: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 EXTENDED: <SessionData/> -->
  </AAA:ConditionAuthzSession>
  </AAA:Conditions>
  <AAA:Obligations>
  <AAA:Obligation>put-policy-obligation(2)-here</AAA:Obligation>      <!-- SAML EXTENDED: <Advice>/<PolicyObligation> -->
  <AAA:Obligation>put-policy-obligation(1)-here</AAA:Obligation>
  </AAA:Obligations>
</AAA:AuthzTicket>
<ds:Signature> <ds:SignedInfo/> <ds:SignatureValue>e4E27kNwEXoVdnXIBpGVjpaBGVY71Nypos...</ds:SignatureValue></ds:Signature>
```



# Basic use cases for policy definition in NRP

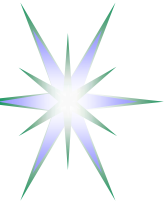
---

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"

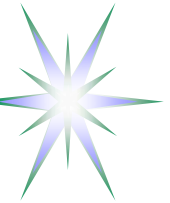
Use case 3: "User/Group A is only allowed to invoke method/action X, Y, and Z"

Use case 4: "User/Group A is only allowed to invoke method X,Y, and Z based on session delegation"



# Policy definition assumptions for NRP

- Users and resources are described/identified by their unique ID's and may have also assigned attributes, e.g.
  - ◆ User attrs: user group, role, federation
  - ◆ Resource attrs: domain/subdomain, resource type, level of service
- Users and resources (domains and endpoints) may be organised/associated into administrative and/or security domains or federations
  - ◆ 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
- Only authenticated user may have access to protected resources
  - ◆ User authentication is confirmed by issuing AuthZ assertion by trusted AuthN service or creating user related security context environment of the started process
- 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
  - ◆ To access a network resources a user identity may need to be mapped to a specific (pool) account



# XACML-Grid - Implementation

---

- In C – actually SAML-XACML front-end for LCAS/LCMAPS based Site Central AuthZ Services (SCAS)
  - ◆ C-based SCAS released and undergoing certification
- In Java
  - ◆ SAML2-XACML profile implemented as part of the recent OpenSAML2.0 library
  - ◆ Programming guidelines and examples
    - <http://www.bccs.uib.no/~hakont/SAMLXACMLExtension/>
  - ◆ Java-based SCAS is being developed as part of the Privilege project



**Implements SAML2.0 profile of XACML2.0 Version 1 (with errata)**

**Builds upon the source of OpenSAML**

**Every XML-element/object in OpenSAML and the extension consists of**

- An interface
- The implementation
- Builder for creating it
- Marshaller, Java->XML
- Unmarshaller, XML->Java

**Supplementary code contains**

- Helper class for making a XACML Request context from a SAML Assertion
- Examples/templates for creating SAML-XACML assertions and queries and extracting attributes and obligations



# Administrative vs Security domain vs Security Association

---

- Domains can be considered as network, administrative or security
  - ◆ Network domains are more static
  - ◆ Administrative domain is managed by the resource owner (or user administration)
  - ◆ Security domain is defined by common trusted identity or attribute management authority
- Security association
  - ◆ Security association can be created dynamically, e.g. for managing project, resource provisioning agreement
    - VO or Shibboleth federation are two examples
  - ◆ Authorisation session as a kind of security association



# Multi-domain NRP – Domain definition

Domains are defined (as associations of entities) by a common policy under single administration, common namespaces and semantics, shared trust, etc.

Domain related security context may include

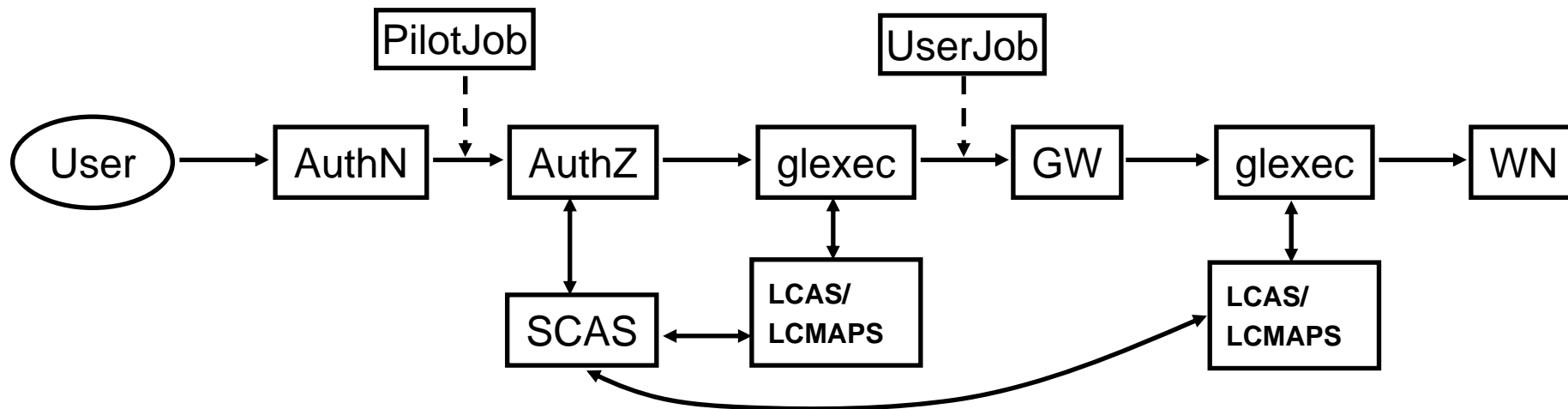
- namespace aware names and ID's
- policy references/ID's
- trust anchors
- authority references
- Additionally, each domain may have/create own dynamic/session related security context (at the reservation and access stages)

Multi-domain NRP AuthZ infrastructure

- Multiple policies processing and combination, including obligated/conditional policy decisions and delegation
- Attributes/rules mapping/converting based on inter domain trust management infrastructure
- Policy support for different logical organisation of resources, including possible constraints on resource combination and interoperation



# Obligations and Pilot Job use case



- Pilot Job is submitted on behalf of a user in advance with PJ submitter account/credentials, User Job is submitted at later stage with real User Job credentials
- Site Central AuthZ Service (SCAS) allows policy enforcement consistency but requires special mechanisms for security context management
  - ◆ SCAS is verified to be compatible with the XACML policy and PDP
- gLExec operates as a gateway between (open) Grid world and executive environment of the Computer Element (CE) and/or cluster Worker Node (WN)
  - ◆ gLExec maps user account to one of available pool accounts