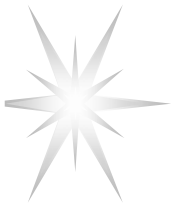
Extending XACML Authorisation Model to Support Policy Obligations Handling in Distributed Applications

Yuri Demchenko
SNE Group, University of Amsterdam

On behalf of Y. Demchenko, C. de Laat, O. Koeroo, H. Sagehaug

MGC 2008 - 6th International Workshop on Middleware for Grid Computing
1 December 2008, Leuven



Outline

- Obligations in Complex Grid and Network Resource Provisioning
  - AAA/AuthZ Architecture for Complex Resource Provisioning (CRP)
- Policy Obligations definition in XACML
- Reference Model for Obligations Handling (OHRM) – proposed extension
- Implementation
  - OSG/EGEE AuthZ Interoperability project and XACML-Grid profile
  - GAAA Toolkit library and XACML-NRP attributes and policy profile
- AAA/AuthZ mechanisms and functional components to support policy obligations handling in multidomain NRP/CRP
- Future developments

Background for this research

- EU funded Phosphorus Project “Lambda User Controlled Infrastructure for European Research” (EC Contract number 034115)
- EGEE project and OSG/EGEE AuthZ Interoperability WG
- University of Amsterdam SNE Group ongoing research on GAAA-AuthZ – Generic Authentication, Authorization, Accounting (GAAA) AuthZ Framework

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)

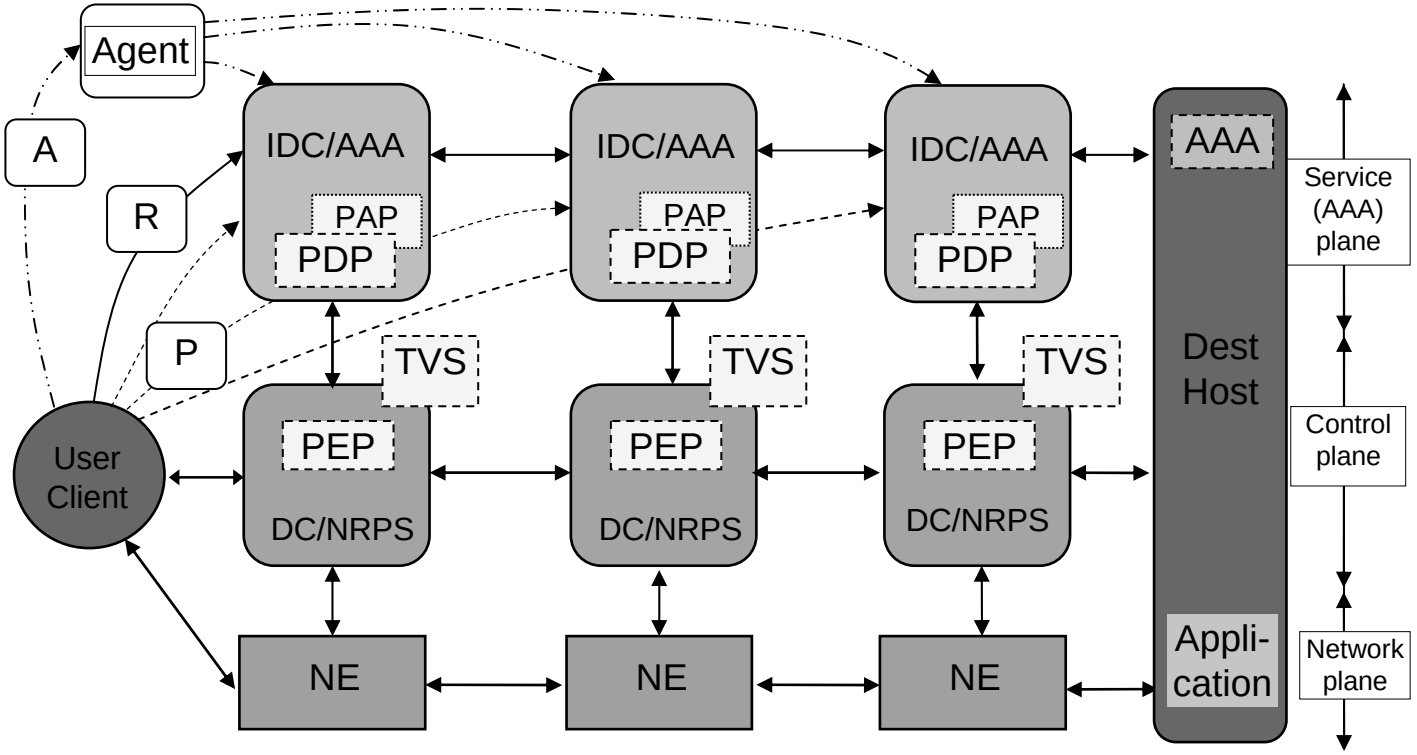
Now considering two other stages: “decommissioning” and “relocation”

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

Obligations in CRP scenarios

- Policy decision is done at the reservation stage (advance reservation stage often requires policy decision) and actual policy enforcement takes place at the access stage
  - Advance resource reservation (ARR) use cases and Usable/Consumable resources
    - ***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
- Policy may contain Obligations and (obligated) policy decision may suggest the following action at later stage
  - Conditional AuthZ decision (e.g. type of service or credentials for multi-domain multi-provider resources)
  - Account mapping
  - Quota assignment
  - Logging and accounting

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

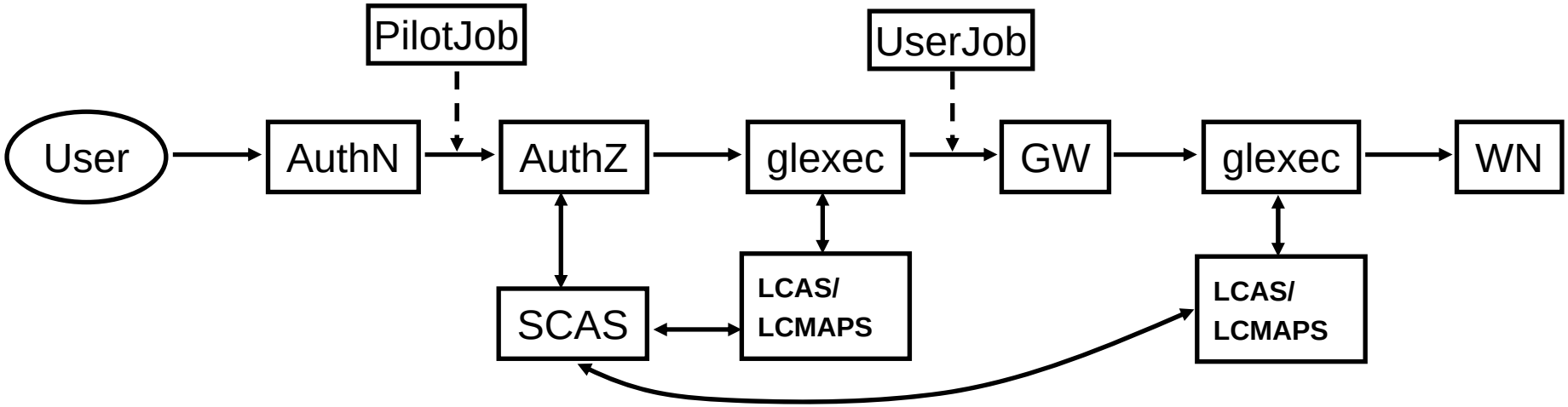
IDC – Interdomain Controller

DC – Domain Controller

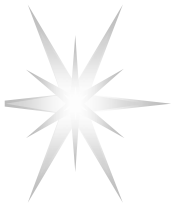
NRPS – Network Resource Provisioning System

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

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



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

Obligations in other AuthZ and policy based management frameworks

XACML policy obligations definition is originated from the two other concepts

- Provisional Authorisation model by Kudo (implemented in the IBM's XACL)
  - Includes Provisional AuthZ Module (PAM) and Request Execution Module (REM)
  - PAM can authorise a request provided the requestor or system (actually REM) will take some security actions, defined as "provisional actions" prior to the request execution, e.g. presenting additional credentials, signing privacy statements, logging events, etc.
- Obligation policies (by Sloman) are defined together with Authorisation policies as part of the policy based management in distributed systems
  - Obligation policies provide simpler way of enforcing state-based policies over managed objects
    - Stateful part of the management policies can be implemented as obligation policies
  - Requires trusted manager (that can be treated similar to the Reference Monitor concept in the Trusted Computing Base (TCB))
  - Provisions and obligations concepts have been further developed by Bettini et al

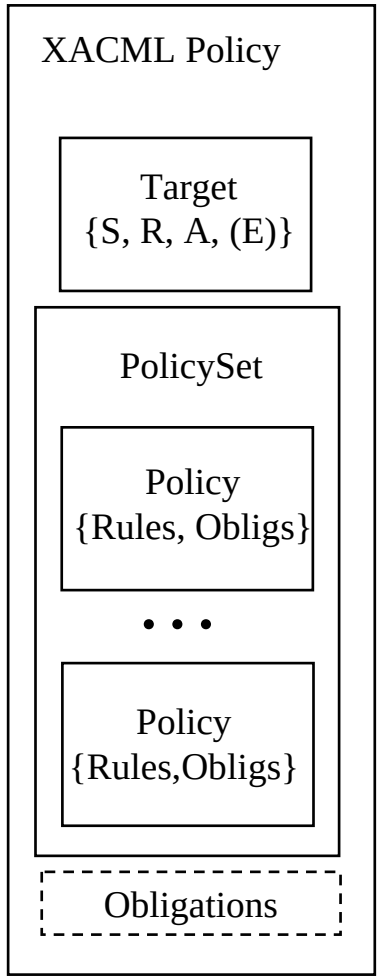
XACML Policy format

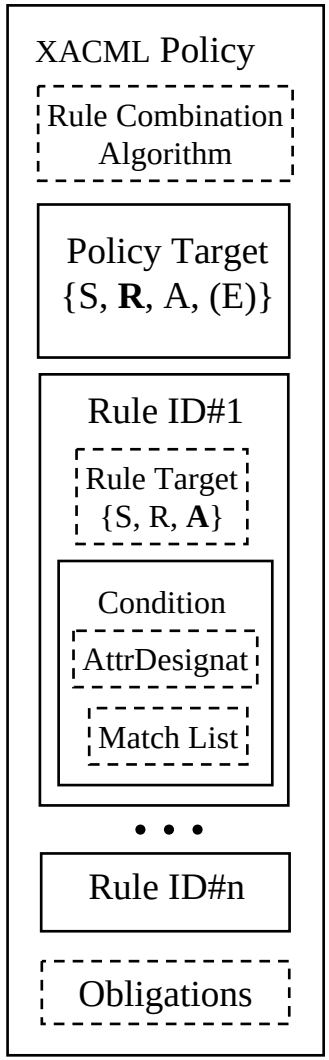
XACML standard specifies XACML policy format and XACML request/response messages

Policy consists of Policy Target and Rules

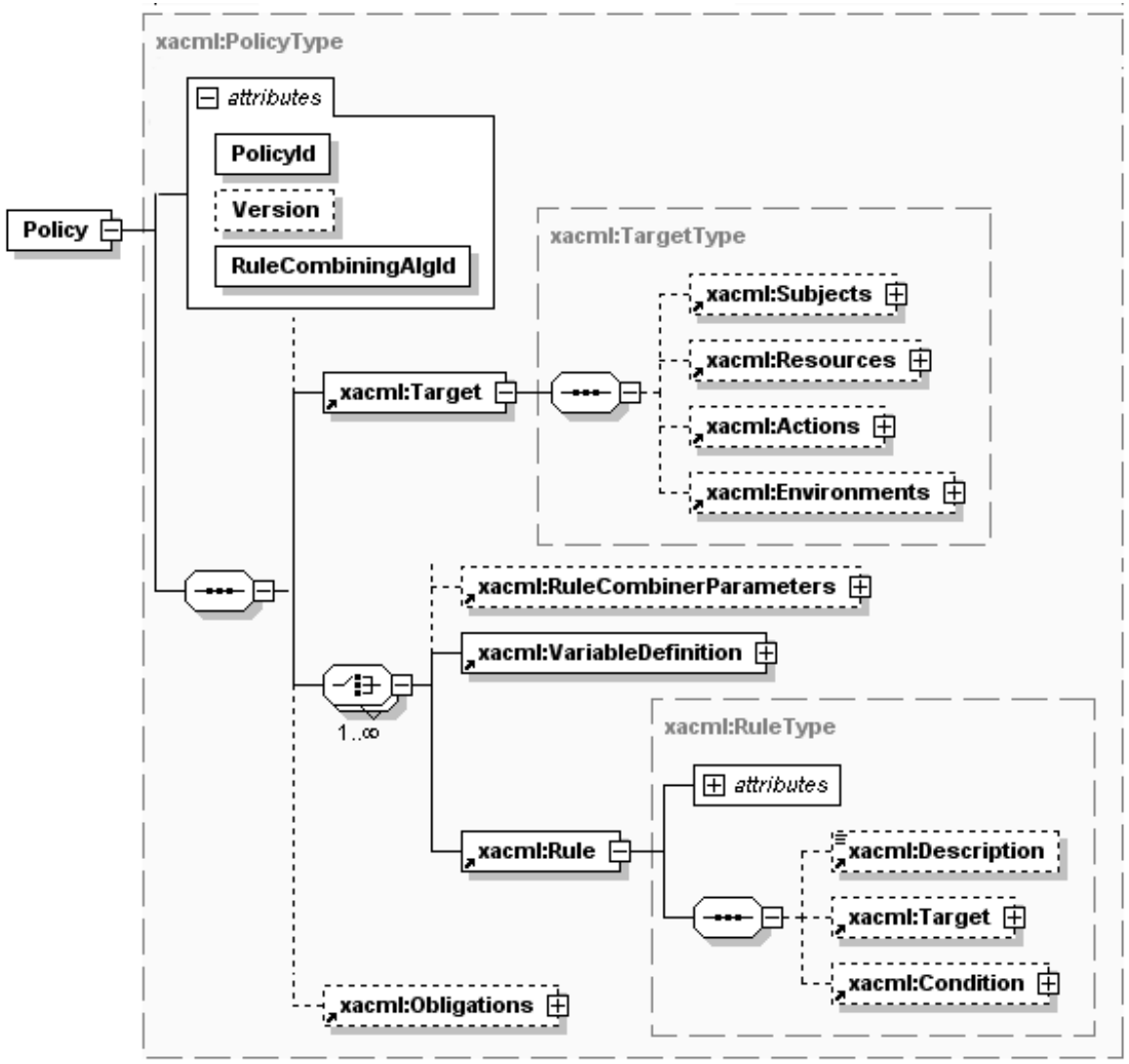
- Policy Target is defined for the tuple Subject-Resource-Action (-Environment)
- Policy Rule consists of Conditions and may contain Obligations
- Obligation defines actions to be taken by PEP on Policy decision by PDP

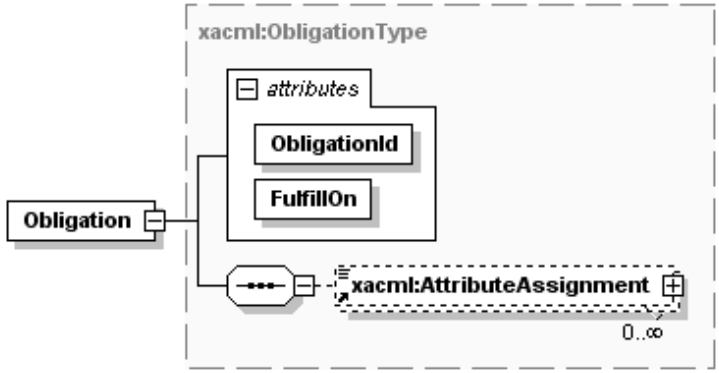
XACML PDP returns all Obligations that match policy decision (defined by attribute “FulfillOn”) from both PolicySet and comprising individual policies

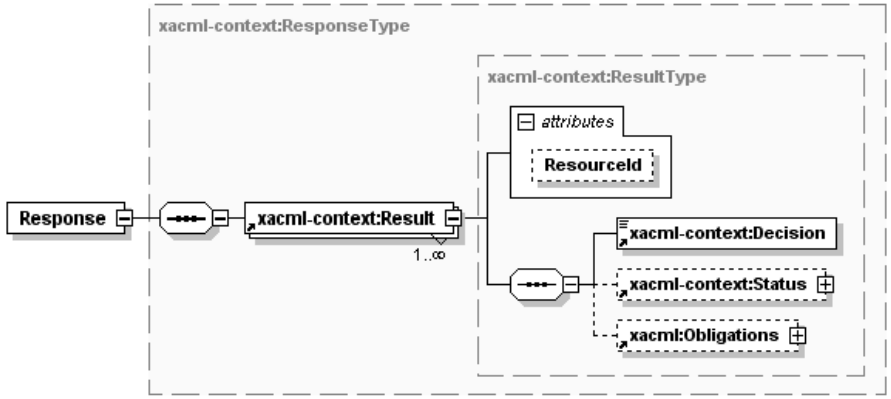




XACML2.0 Policy Datamodel

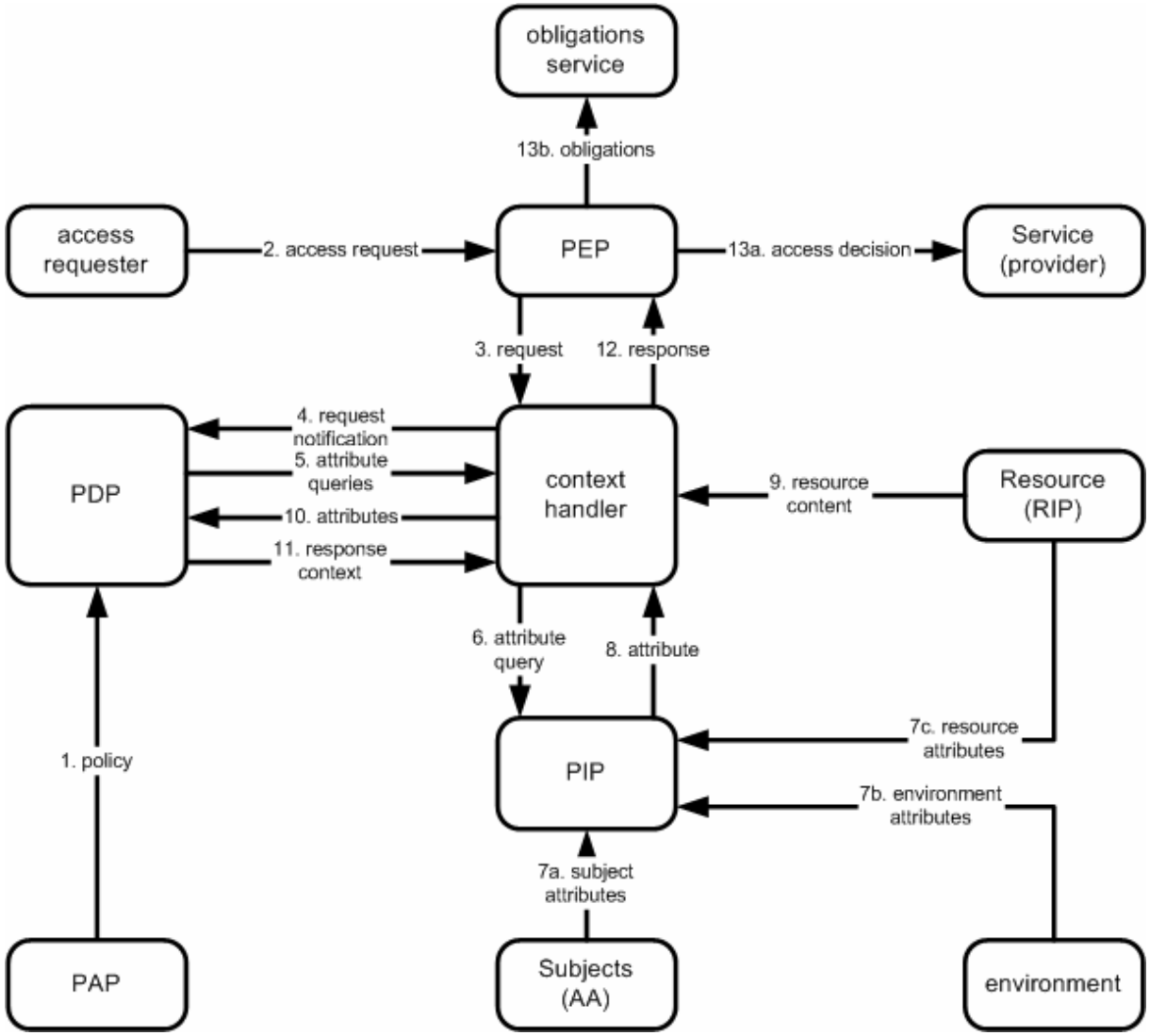






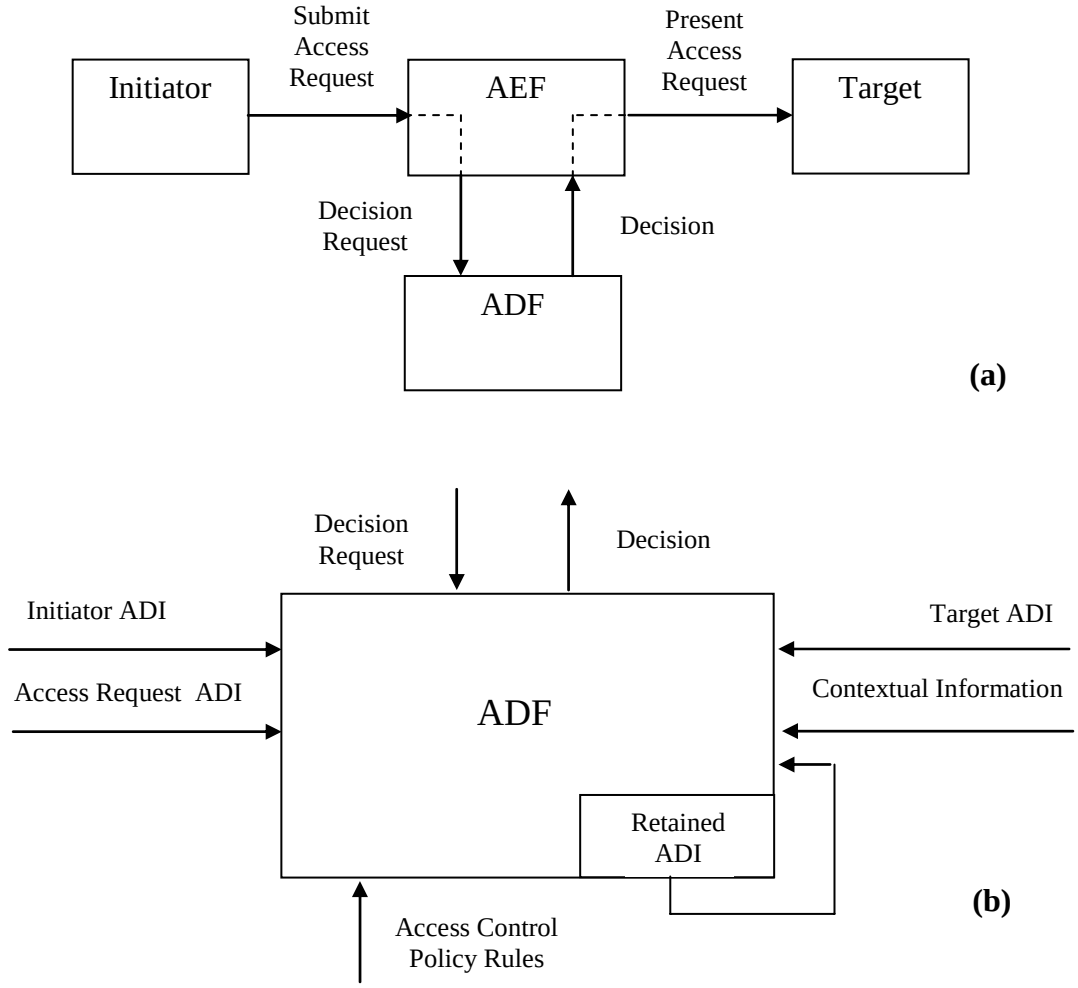
XACML Response message contains all Obligations that match policy decision (defined by attribute “FulfillOn”) from both PolicySet and comprising individual policies

XACML2.0 Authorisation Process Dataflow



- Introduces ContextHandler functionality
- Assumes stateless PDP as actually XACML policy is stateless as well
- Obligations service is called from PEP
  - Compliant with the Obligations definition but put much restriction in distributed environment

Retrospective: X.812 Access Control Framework



- Defined in the ITU X.812 std for Open Systems Security
- Historically one of the first formalised AuthZ model
- Further developed in the IETF Generic AAA AuthZ framework
- Includes a notion of the AuthZ session and AuthZ context management
  - By introducing retained ADI
  - ADF with retained ADI can be treated as stateful ADF
- XACML AuthZ model fits into X.812 AuthZ model

AEF/ADF - AuthZ Enforcement/Decision Function

ADI – AuthZ Decision Information

XACML Obligations – Implementation suggestions

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

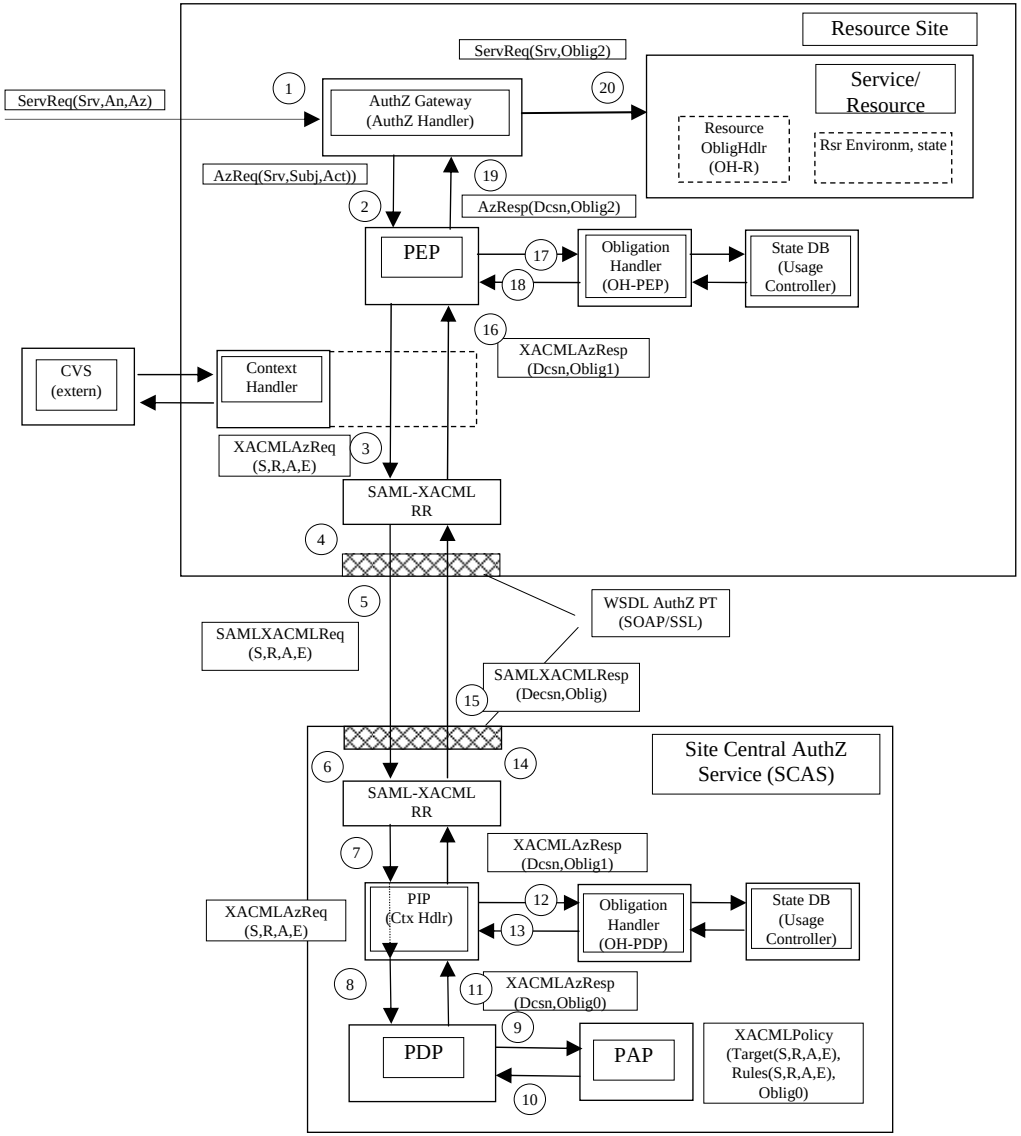
Obligations enforcement scenarios

- 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 (OHRM)



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)

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 AuthzResponce (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.

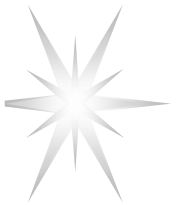
XACML Obligations – Examples of expression for pool account mapping in Grid – Option 1 (simple, ref XACML-Grid v1.0)

```
<!-- 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-Grid and XACML-NRP profiles - Attributes and Obligations definition for Grid and Network Resource Provisioning

Both profiles define a set of attributes and basic policy models

- Common namespace - *http://authz-interop.org/xacml*
- Subject, Resource, Action, Environment attributes semantics and format
- Obligations semantics and expression format
- XACML-Grid profile is implemented in two major middleware frameworks gLite/EGEE (LCAS/LCMAPS) and Globus/Privilege
  - "An XACML Attribute and Obligation Profile for Authorization Interoperability in Grids" (Joint project by EGEE, OSG, GT). Version 1.0, May 16, 2008 -
    - http://cd-docdb.fnal.gov/cgi-bin/ShowDocument?docid=2685
    - https://edms.cern.ch/document/929867/1
  - Implements SAML-XACML call-out interface to Site Central AuthZ Service (SCAS)
  - SAML2-XACML profile is implemented as a part of the OpenSAML2.0 library
- XACML-NRP supports all XACML-Grid attributes and obligations but extends them with NRP-specific and uses different policy models
  - Recent update (July 2008) - http://staff.science.uva.nl/~demch/projects/aaauthreach/draft-interop-xacml-nrp-profile-012.pdf

XACML-Grid Obligations

Uses simplified Obligations expression format

Obligation = {AttributeAssignment (ObligationId, AttributeValue(AttributeId))}

ObligationId: *<ns-prefix>/obligation/<obligation-name>*
AttributeId: *<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 Policy Obligations

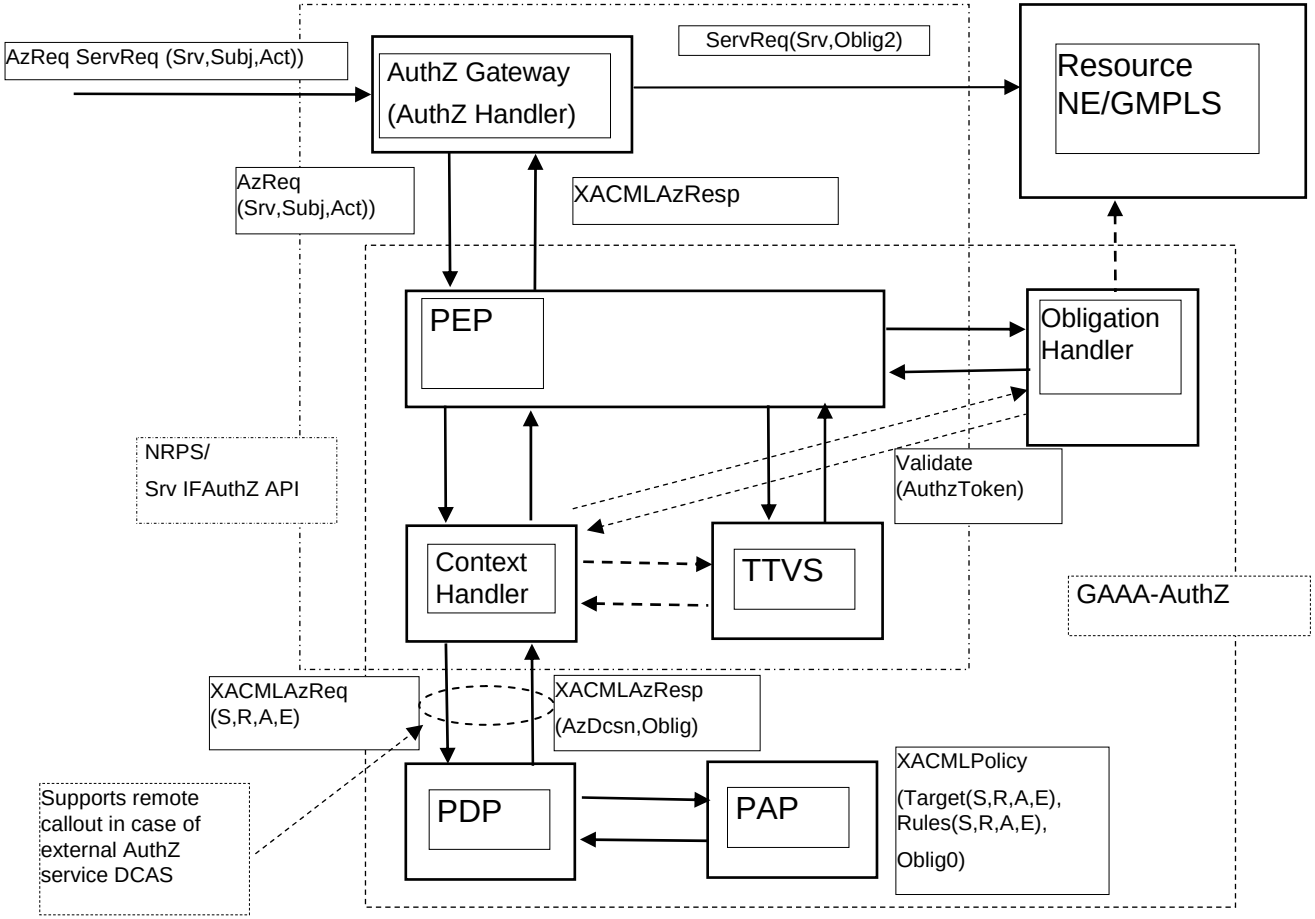
Suggested policy obligations for multidomain NRP

- 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/quota

XACML-NRP Implementation – GAAA-TK Java library

- XACML-NRP profile is implemented in 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



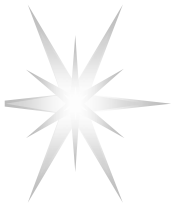
The proposed model intends to comply with both the generic AAA-AuthZ framework and XACML AuthZ model

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

TTVS – Ticket and token validation and handling service

Future developments

- OHRM implementation in the next GAAA-TK release
- GAAA-TK library interoperability and integration with gLite/Globus Toolkit AuthZ Framework, in particular OHRM module
- OHRM and Pilot Job AuthZ workflow definition and implementation with SCAS
- OHRM and restricted delegation to support multidomain reservation process and resource access
- Moving XACML-Grid and XACML-NRP profile to the OGF standardisation process



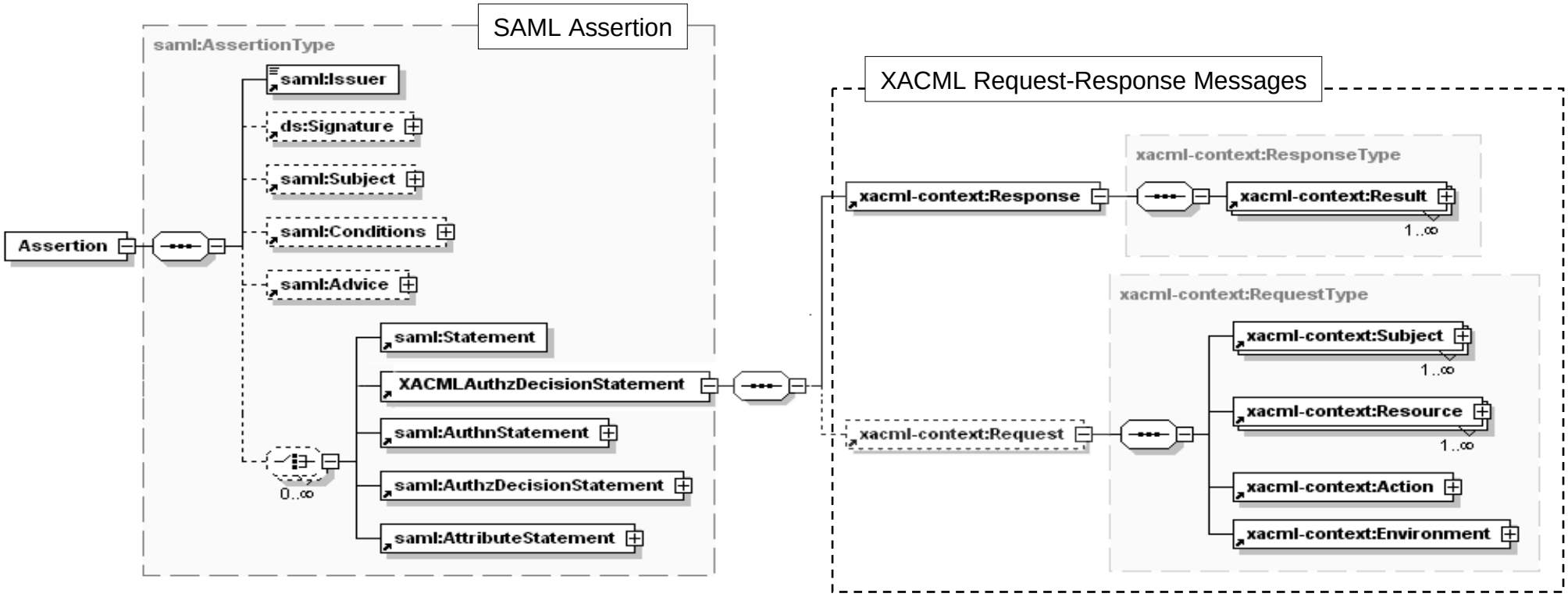
Questions and Discussion

Additional information

- SAML-XACML Request-Response format
- SAML-XACML Extension library for OpenSAML2.0
- AuthZ ticket data model

- 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

SAML-XACML Request/Response messages

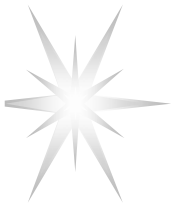


XACMLRequest (Resource, Subject, Action, Environment)

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

XACML Request message - Example

```
<xacml-context:Request xmlns:xacml="urn:oasis:names:tc:xacml:1.0:policy" xmlns:xacml-
    context="urn:oasis:names:tc:xacml:1.0:context" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="urn:oasis:names:tc:xacml:1.0:context aaa-msg-xacml-01.xsd">
  <xacml-context:Subject Id="subject"    SubjectCategory="urn:oasis:names:tc:xacml:1.0:subject-category:access-
    subject">
    <xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
    DataType="http://www.w3.org/2001/XMLSchema#string" Issuer=" admin@gaaa.virtlab.nl ">
      <xacml-context:AttributeValue>WHO740@users.project.organisation.nl</xacml-context:AttributeValue>
    </xacml-context:Attribute>
    <xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subjconfdata"
    DataType="http://www.w3.org/2001/XMLSchema#string" Issuer=" admin@gaaa.virtlab.nl ">
      <xacml-context:AttributeValue>2SeDFGVHYTY83ZXxEdsweOP8Iok)yGHxVfHom90</xacml-context:AttributeValue>
    </xacml-context:Attribute>
    <xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:subject:role"
    DataType="http://www.w3.org/2001/XMLSchema#string" Issuer=" admin@gaaa.virtlab.nl ">
      <xacml-context:AttributeValue>Analyst</xacml-context:AttributeValue>
    </xacml-context:Attribute>
  </xacml-context:Subject>
  <xacml-context:Resource>
    <xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
    DataType="http://www.w3.org/2001/XMLSchema#string" Issuer="admin@gaaa.virtlab.nl">
      <xacml-context:AttributeValue>Resource-ID-here</xacml-context:AttributeValue>
    </xacml-context:Attribute>
  </xacml-context:Resource>
    <xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
    DataType="http://www.w3.org/2001/XMLSchema#string" Issuer="admin@gaaa.collaboratory.nl">
      <xacml-context:AttributeValue>assign-time</xacml-context:AttributeValue>
    </xacml-context:Attribute>
  </xacml-context:Action>
</xacml-context:Request>
```



OpenSAML SAML-XACML extension library

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

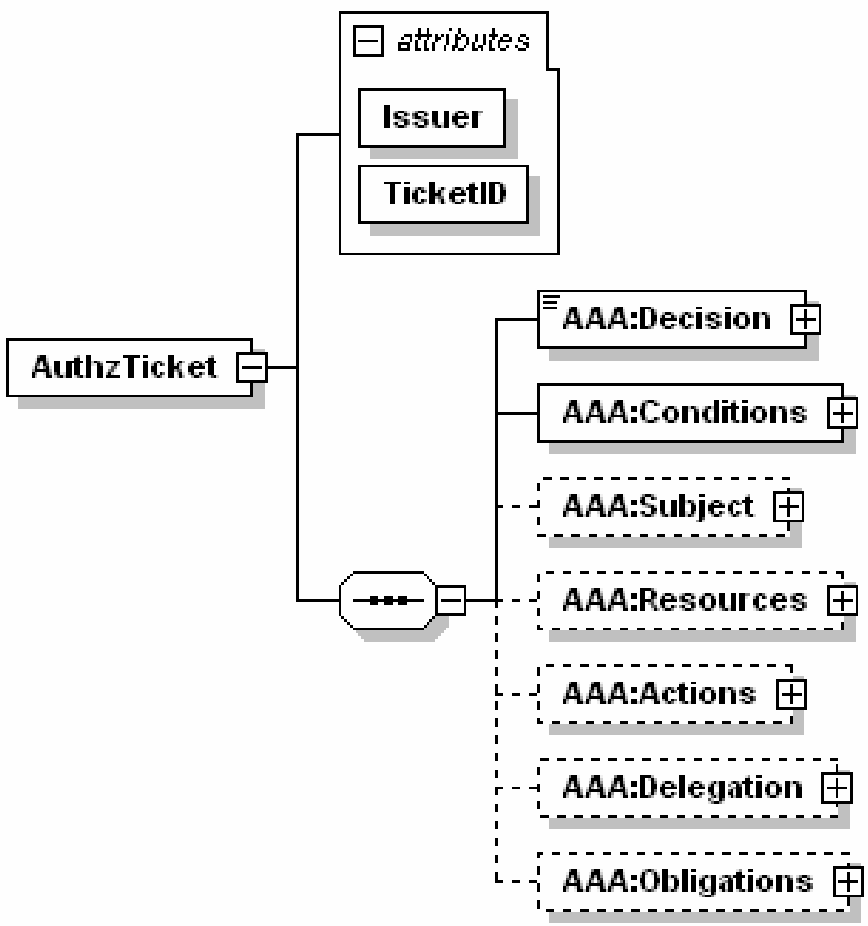
- An interface
- The implementation
- Builder for creating it
- Marshaller, Java->XML
- Unmarshaller, XML->Java

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

OpenSAML2.0 Extension Library to Support SAML2.0 profile of XACML2.0 - http://www.bccs.uib.no/~hakont/SAMLXACMLExtension/

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 constrains 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 capbilities

<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>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: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>
```