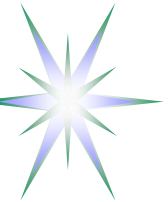
Extending Role Based Access Control Model for Distributed Multidomain Applications

Yuri Demchenko <demch@science.uva.nl>
System and Network Engineering Group
University of Amsterdam

IFIPSEC2007 Conference
14-16 May 2007, Sandton, South Africa



Outline

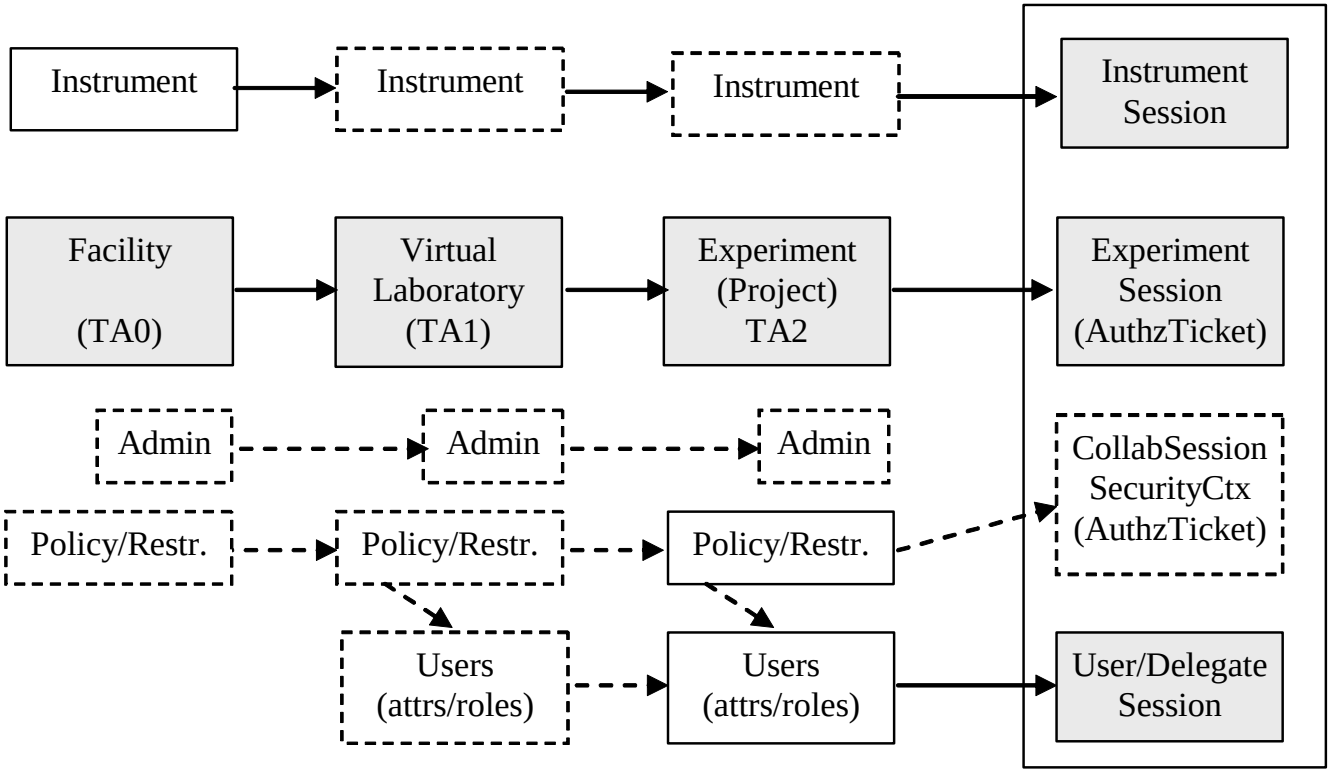
- Background – Origin and target projects
- Domain based Resource management in Collaborative applications
- RBAC Overview
- RBAC extension for multidomain resource organisation
- Implementation suggestions
  - AuthZ session management
  - Using XACML for policy expression
    - XACML policy example
- Future development

Background – Origin/Target projects

- Central Authorisation service for Grid based Collaborative applications
  - GAAA-AuthZ Architecture and Implementation (Collaboratory.nl, VL-e projects)
    - Domain based resource management and RBAC
    - AuthZ session/ticket for AuthZ service performance optimisation
- Distributed multidomain Authorisation service for network on-demand services and OLPP
  - EU Project PHOSPHORUS and NL national project RoN GP-NG
- AuthZ service for (distributed) dynamic Grid applications
  - Extended security context management in Grid oriented AuthZ Frameworks (gLite AuthZ Service, EGEE Project)

- Part of ongoing development of the generic AAA Authorisation Framework (GAAA-AuthZ) for Complex Resource Provisioning (CRP)

Hierarchical Domain based Resource organisation in Grid based Collaborative applications



Domains are defined by common administration and security policy and associated trust anchors

Full Resource URI/ID –

CNL:Facility:VirtualLab:Experiment:InstrModel

Full User Session context –

Facility < Virtual Lab < Experiment < Experiment Session < Collaborative Session

Resource hierarchy and security context

Facility > Virtual Lab > Experiment > Experiment/Collaborative Session

- Provides context for
  - Instrument as an access object (i.e., Resource)
  - User roles/attributes handling
- Access to instruments is based on pool accounts at Facility
- Users must be registered as VL or Experiment members
  - Additionally, a possibility to invite new Collaborative session members
- Access control policies depend on the domain context and experiment stage and/or collaborative session
  - Need to combine multiple policies and/or multiple AuthZ decisions

Two general approaches to flexibly manage dynamic security context

(1) Context aware access control infrastructure

(2) Experiment workflow to manage security context (requires (1))

Domain based Resource Management - Assumptions

- Physically Instruments are located at the Facility but logically they are assigned to the VL and allocated to an Experiment
- Users/members of collaborative sessions are assigned to the Experiment
  - Managerial and operator personnel belongs to VL and Facility and may have specific and limited functions in the Experiment
  - Domain based restrictions/policy can be applied to (dynamic) role assignment
- Administrative rights/functions can be delegated by the superior entity/role in this hierarchical structure
- Trust Anchors (TA) can be assigned to hierarchical domain related entities to enable security associations and support secure communication
  - VL TA1 suggested as minimum required in DM
  - Experiment TA2 may be included into the Experiment description
  - Collaborative session security association can be supported by AuthZ tickets.

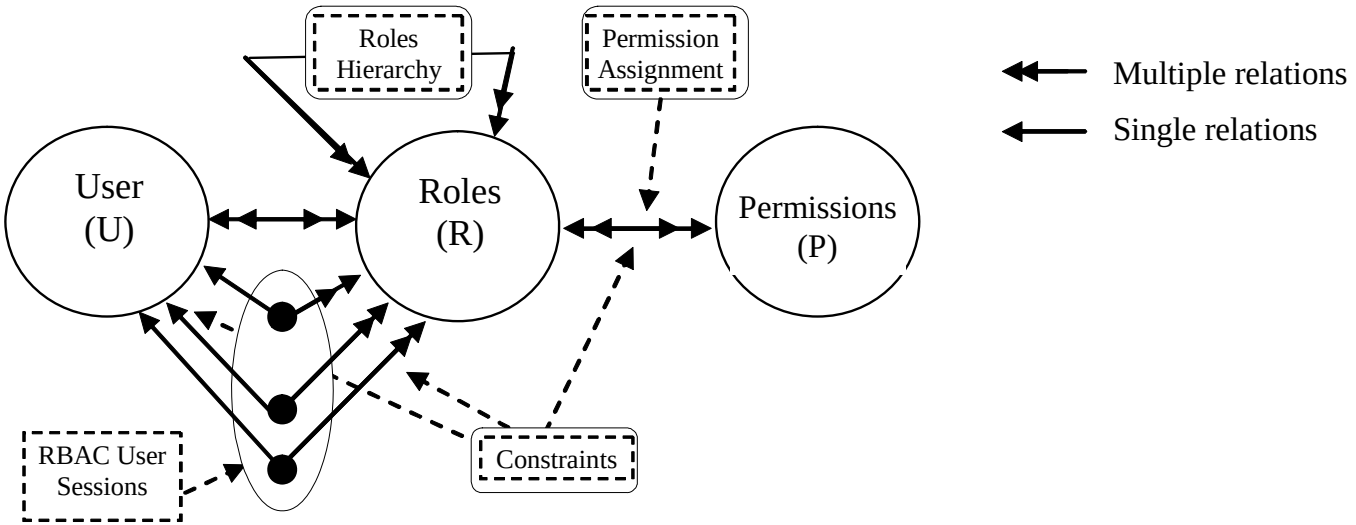
Generic RBAC models - Overview

Proposed by Sandhu (1997) and specified by ANSI/INCITS 359-2004

- Focus on roles management and static vs dynamic separation of duties

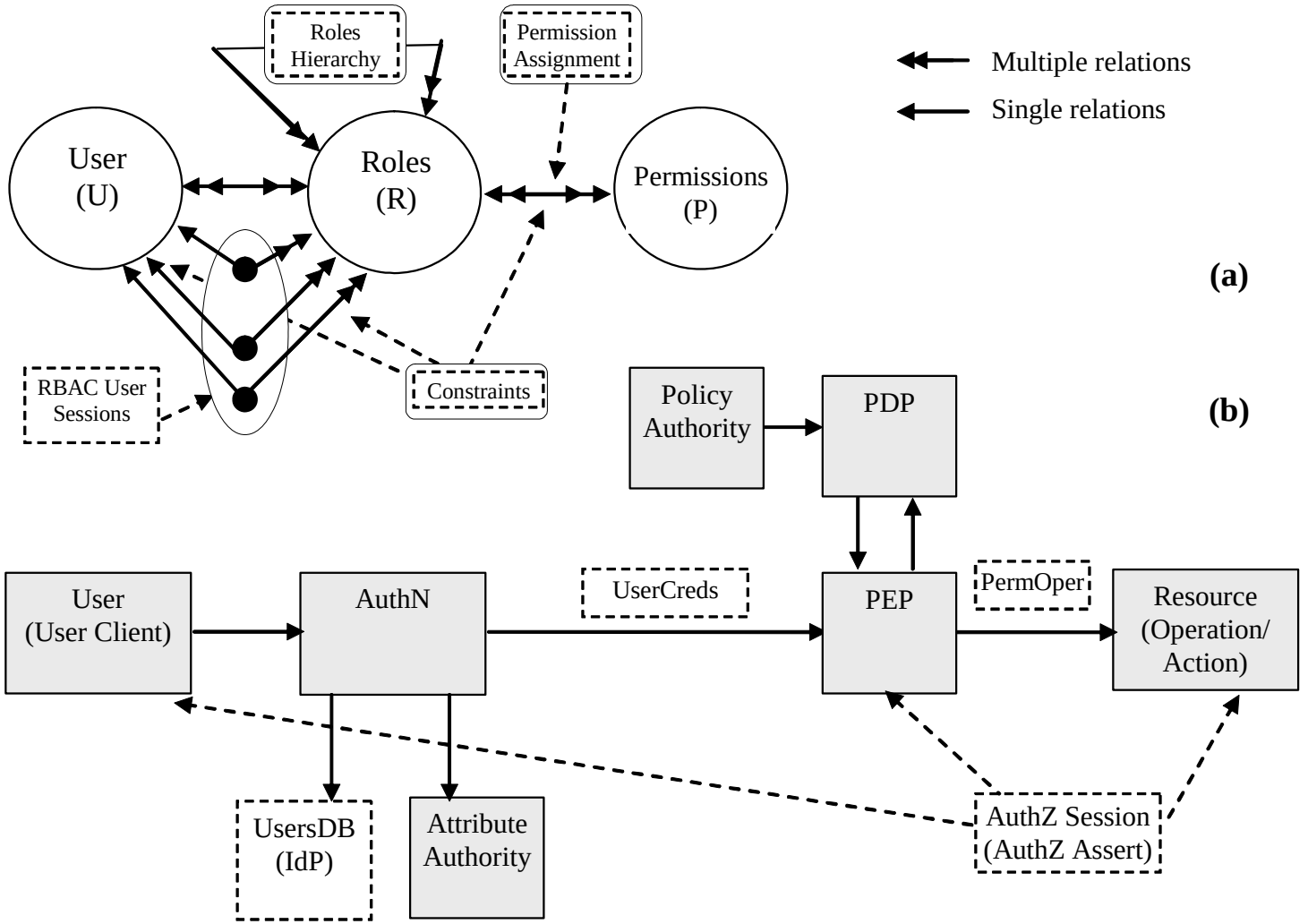
- RBAC0 – flat role-permissions model
  - One user per session (single or multiple roles)
  - One user can have multiple sessions
- RBAC1 – roles hierarchy and capabilities inheritance
  - One user per session (dominant roles can be added)
- RBAC2 = RBAC0 + constraints
  - Enforces high-level (local) policies
  - Decentralised security model and context -dependent
- RBAC3 = RBAC1 + constraints

RBAC implementation issues



- Practical RBAC implementation requires resolution of many other administration and security issues left out of scope in classical RBAC
  - Policy expression and management
  - AuthZ session management mechanisms
  - Rights/privileges delegation
  - Security context management in multidomain scenarios
  - Scalability issues

Relation between RBAC and GAAA-AuthZ



(a) RBAC0-3

(b) GAAA AuthN and AuthZ

RBAC extension for Domain based Resource organisation

Two major directions

- Multiple and hierarchical policies management that reflect hierarchical resource organisation
  - Domain related access control policy definition and attributes assignment
  - Policy combination or access control decisions combination
- Dynamic domain related security context management
  - To allow for dynamic roles assignment and delegation with the domain defined restrictions

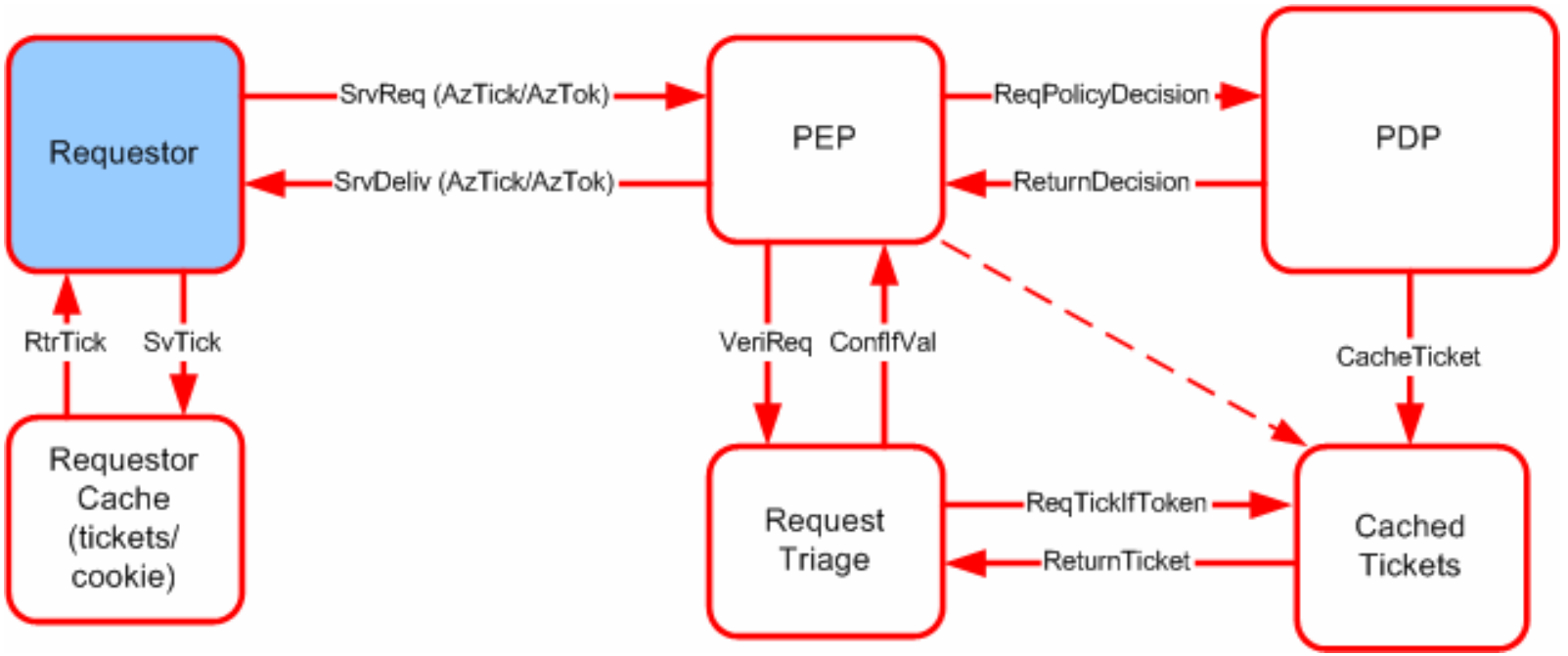
Proposed solutions to address multidomain access control issues

- AuthZ ticket as used for extended Authorisation session context management
  - Proprietary and SAML2.0 format
- Using core XACML (eXtensible Access Control Markup Language) and its special profiles

AuthZ Session management in GAAA-RBAC-DM

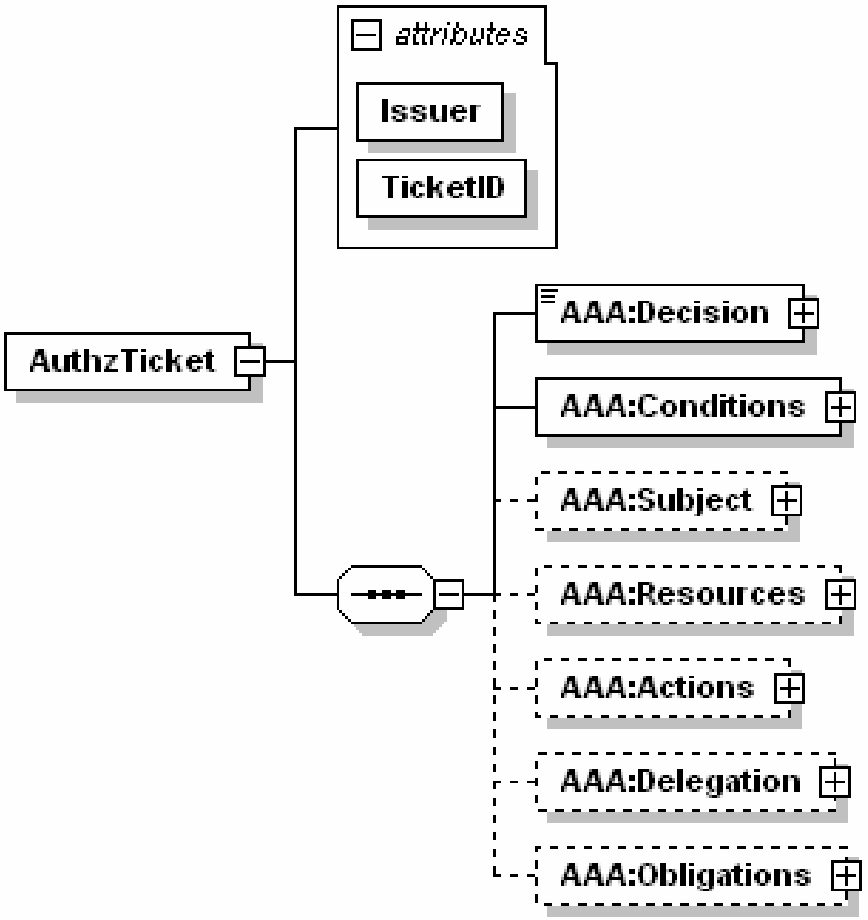
- AuthZ session is a part of the generic RBAC and AAA-AuthZ functionality
- Session can be started only by an authorised Subject/Role
  - Session can be joined by other less privileged users
  - Session permissions/credentials can be delegated to (subordinate) subjects
- Session context includes Request/Decision information and may include any other environment or process data/information
  - AuthZ Session context is communicated in a form of extended AuthZ Ticket (or AuthZ Assertion)
  - SessionID is included into AuthzTicket together with other AuthZ Ctx
  - Signed AuthzTicket is cached by PEP or PDP
- If session is terminated, cached AuthzTicket is deleted
  - Note: AuthzTicket revocation should be done globally for the AuthZ trust domain

AuthZ session Tickets/Tokens handling in AuthZ system



- AuthzTicket is issued by PDP and may be issued by PEP
- AuthzTicket must be signed
- AuthzTicket contains all necessary information to make local PEP-Triage Request verification
- When using AuthzTokens, AuthzTickets must be cached; Resolution mechanism from token to ticket must be provided

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

Using XACML and its special profiles for policy expression

XACML RBAC profile

- defines policies that require multiple Subjects and roles combination to access a resource and perform an action
- implements hierarchical RBAC model when some actions require superior subject/role approval to perform a specific action
- can significantly simplify rights delegation inside the group of collaborating entities/subjects

XACML Hierarchical Resource profile

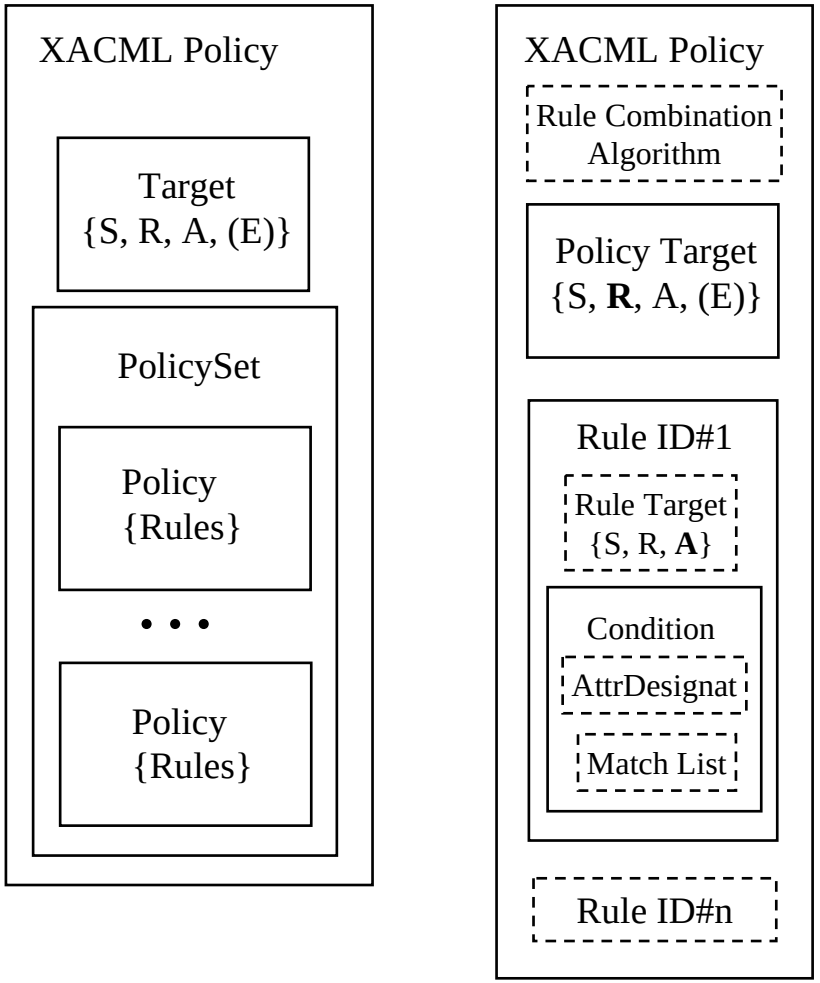
- defines policy format for hierarchically organised resources, e.g. file system or XML-based repositories

XACMLv3.0 Administrative Policy Profile

- Delegation, and Policy Authority

XACML Policy format

- Policy target is defined for the triad {Subject-Resource-Action} and may include Environment
- Environment can be used to transfer AuthZ session context
- Policy may contain Obligation element that defines actions to be taken by PEP on Policy decision by PDP, or define information to be transferred to the next domain



Simple XACML Policy generation conventions

- Policy Target is defined for the Instrument (containing also CNL domain information)
- Policy combination algorithm is “ordered-deny-override” or “deny-override”
- Rule Target is defined for the Action and may include Environment checking
  - Rule’s Condition provides matching of roles which are allowed to perform the Action
- Access rules evaluation
  - Rules are expressed as permissions to perform an Action against Subject role
  - Rule combination algorithm “permit-override”
  - Rules effect is “Permit”
- Subject and Credentials validation – is not supported by current XACML functionality
  - Credential Validation Service (CVS) – proposed OGF-AuthZ WG development
- Environment and Obligation content can be defined using rich XQuery functionality

Example CNL AuthZ policy: Resource, Actions, Subject, Roles

Actions (8)

- StartSession
- StopSession
- JoinSession
- ControlExperiment
- ControlInstrument
- ViewExperiment
- ViewArchive
- AdminTask

Roles (4)

- Analyst
- Customer
- Guest
- Administrator
- (CertifiedAnalyst)

Naming convention

- Resource - http://resources.collaboratory.nl/**<CNLdomain>**/Phillips_XPS1
  CNL:**Facility:VirtualLab:Experiment**:InstrModel
- Subject – "WHO740@**<CNLdomain>.**users.collaboratory.nl"
- Roles - "role" or "role@**<CNLdomain>**.ExperimentID"

Simple Access Control table

Roles	Anlyst	Custm	Guest	Admin
ContrExp	1	0	0	0
ContrInstr	1	0	0	1
ViewExp	1	1	1	0
ViewArch	1	1	0	1
AdminTsk	0	0	0	1
StartSession	1	0	0	0
StopSession	1	0	0	1
JoinSession	1	1	1	0

XACML policy examples

AAAuthreach Project

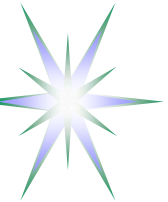
http://staff.science.uva.nl/~demch/projects/aaauthreach/index.html

XACML/XACML3 Policy with DM admin information

```
<PolicySet PolicySetId ="urn:oasis:names:tc:xacml:1.0:cnl:policy-set:CNL2-VL1-test"
     PolicyCombiningAlgId="urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:ordered-deny-overrides">
 <Target/>
 <Policy PolicyId="urn:oasis:names:tc:xacml:1.0:cnl:policy:CNL2-XPS1-test“
     RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:deny-overrides">
   <Description>Permit access for CNL3 users with specific roles</Description>
   <PolicyIssuer>
     <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id“
                          DataType="http://www.w3.org/2001/XMLSchema#string">
       <AttributeValue> urn:oasis:names:tc:xacml:3.0:issuer:cnl:VLab031:trusted</AttributeValue>
     </Attribute>
   </PolicyIssuer>
 <Target><Resources><Resource>
       <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-equal">
         <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#anyURI">
            http://resources.collaboratory.nl/Phillips_XPS1</AttributeValue>
         <ResourceAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
                          DataType="http://www.w3.org/2001/XMLSchema#anyURI"/>
       </ResourceMatch>
     </Resource></Resources></Target>
 <Rule RuleId="urn:oasis:names:tc:xacml:1.0:cnl:policy:CNL2-XPS1-test:rule:ViewExperiment" Effect="Permit">
   <Target><Actions><Action>
         <ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
           <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">ViewExperimen</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">analyst</AttributeValue>
       <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">customer</AttributeValue>
     </Apply>
     <SubjectAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:subject:role"
                          DataType="http://www.w3.org/2001/XMLSchema#string" Issuer="CNL2AttributeIssuer"/>
   </Condition>
  </Rule>
  </Policy></PolicySet>
```

Future developments

- Describing dynamic trust management model for RBAC-DM
- Defining Hierarchical and Multidomain DM profiles (HDM and MDM)
- XACML-RBAC and XACML v3.0 implementation for Policy Authority and permissions delegation
- Contributing AuthZ session management framework to the Open Grid Forum OGSA-AuthZ WG
- Integration with existing access control tools
  - EGEE gLite Java Authorization Framework (gJAF)
  - GAAA-AuthZ toolkit



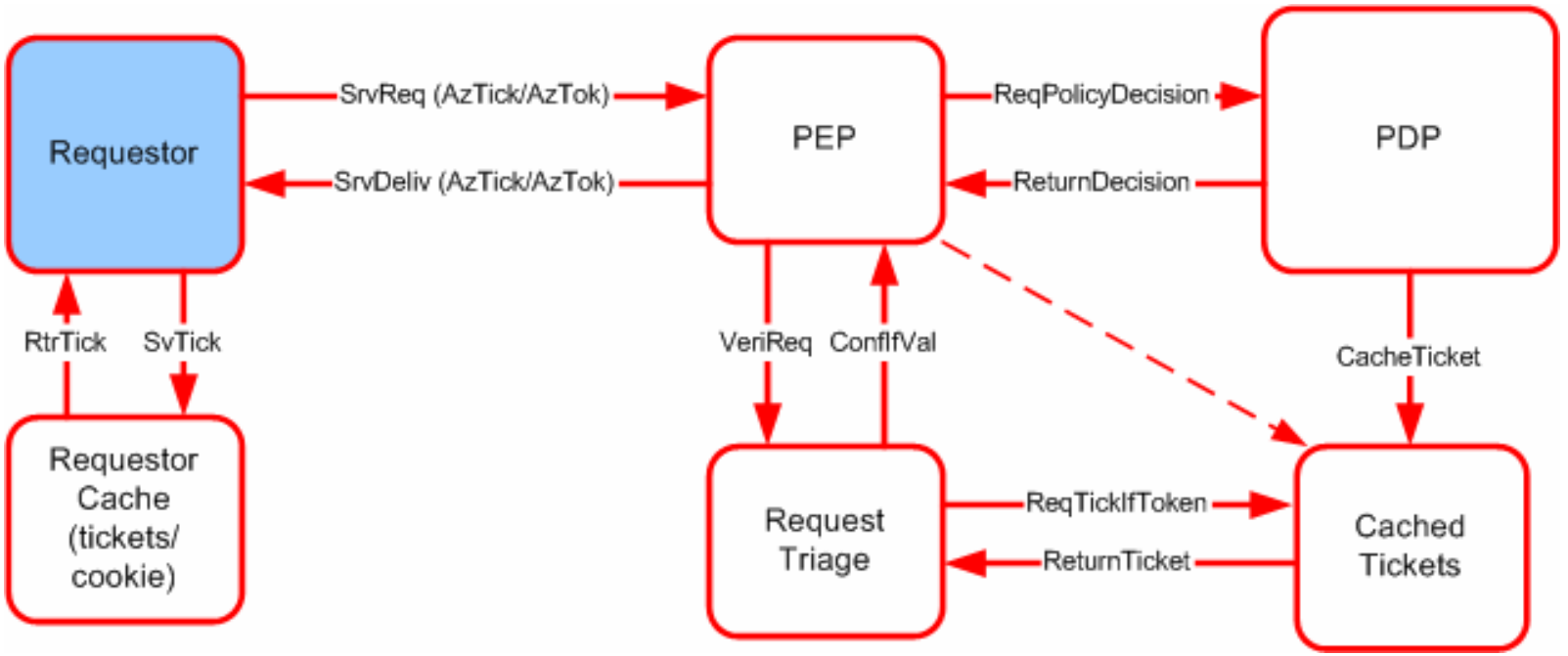
Additional information

- AuthZ session management in GAAA-AuthZ
- Detailed AuthZ ticket and token examples

AuthZ Session management in GAAA-RBAC-DM

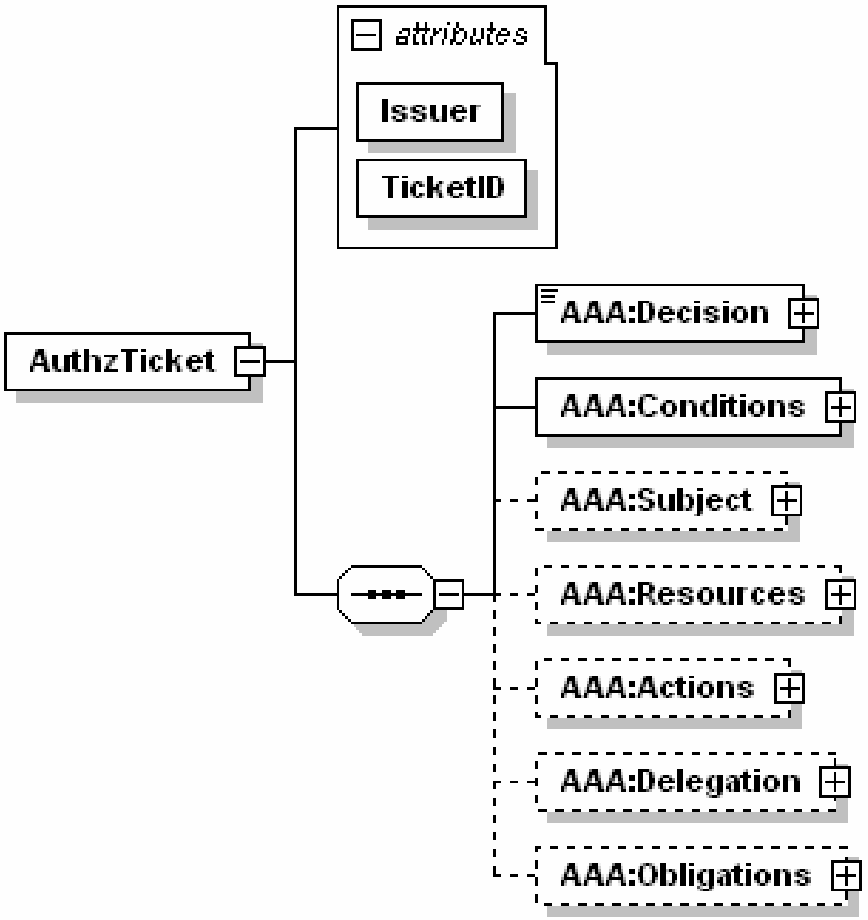
- AuthZ session is a part of the generic AAA-AuthZ (and RBAC) functionality
- Session can be started only by an authorised Subject/Role
  - Session can be joined by other less privileged users
  - Session permissions/credentials can be delegated to (subordinate) subjects
- Session context includes Request/Decision information and may include any other environment or process data/information
  - AuthZ Session context is communicated in a form of extended AuthZ Ticket (or AuthZ Assertion)
  - SessionID is included into AuthzTicket together with other AuthZ Ctx
  - Signed AuthzTicket is cached by PEP or PDP
- If session is terminated, cached AuthzTicket is deleted
  - Note: AuthzTicket revocation should be done globally for the AuthZ trust domain

AuthZ session Tickets/Tokens handling in AuthZ system



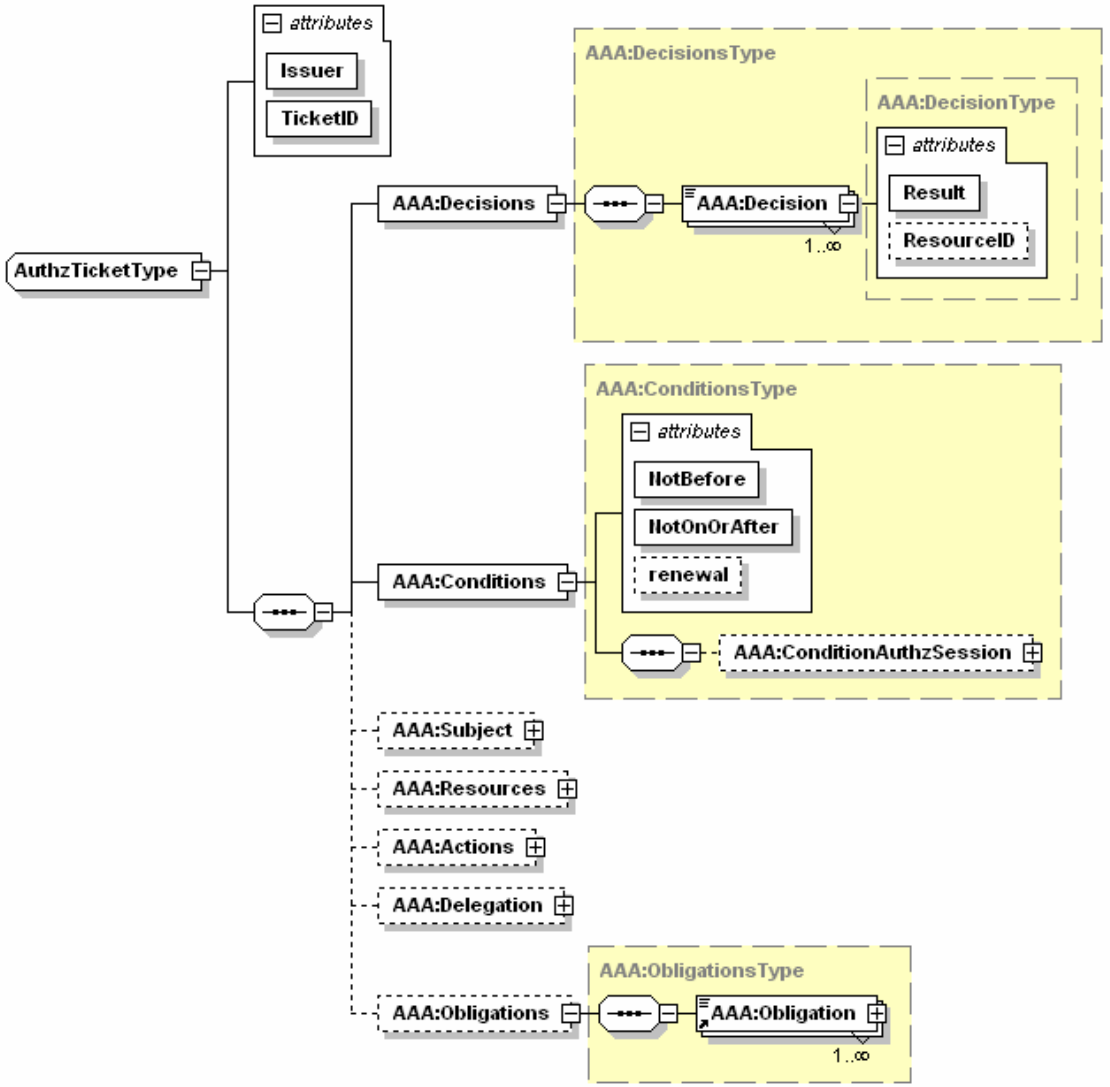
- AuthzTicket is issued by PDP and may be issued by PEP
- AuthzTicket must be signed
- AuthzTicket contains all necessary information to make local PEP-Triage Request verification
- When using AuthzTokens, AuthzTickets must be cached; Resolution mechanism from token to ticket must be provided

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



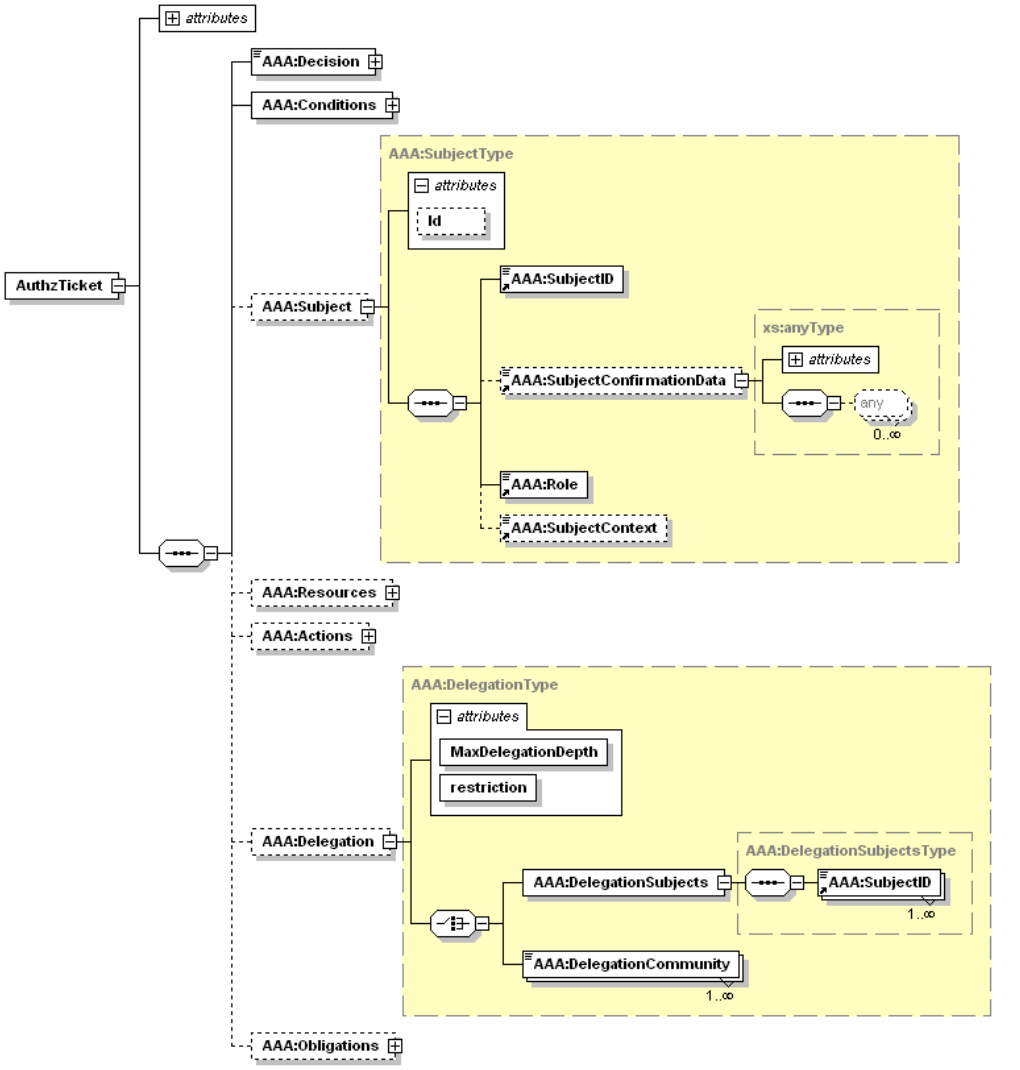
- Required functionality to support multidomain provisioning scenarios
  - Allows easy mapping to SAML and XACML related elements
- Allows multiple Attributes format (semantics, namespaces)
- Establish and maintain Trust relations between domains
  - Including Delegation
- Ensure Integrity of the AuthZ decision
  - Keeps AuthN/AuthZ context
  - Allow Obligated Decisions (e.g. XACML)
- Confidentiality
  - Creates a basis for user-controlled Secure session

AuthZ ticket Data model (2) - Mandatory elements



- TicketID attribute
- Decisions element and ResourceID attribute
- Conditions Element and validity attributes
- Extensible element ConditionAuthzSession
  - Any AuthZ session related data

AuthZ ticket Data model (3) – Subject and Delegation elements



- Subject element to keep AuthN security context and Subject Attributes
- Delegation element to allow permissions/AuthZ decision delegation to other Subjects or groups/community

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 – 3-10KB

```
<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>
```

AuthzToken example – 293 bytes

```
<AAA:AuthzToken TokenID="c24d2c7dba476041b7853e63689193ad">
<AAA:TokenValue>
0IZt9WsJT6an+tIxhhTPtiztDpZ+iynx7K7X2Cxd2iBwCUTQ0n61Szv81DKllWsq75IsHfusnm56
zT3fhKU1zEUsob7p6oMLM7hb42+vjfvNeJu2roknhIDzruMrr6hMDsIfaotURepu7QCT0sADm9If
X89Et55EkSE9oE9qBD8=
</AAA:TokenValue>
</AAA:AuthzToken>
```

AuthzToken is constructed of the AuthzTicket TicketID and SignatureValue

AuthzToken use suggests caching AuthzTicket's

AuthzToken can be used as cookie

Security context management: Context dependent information and existing mechanisms

Context dependent information/attributes:

- Policy
- Trust domains and authorities
- Attributes namespaces
- Service/Resource environment/domain
- Credential semantics and formats

Mechanisms to transfer/manage context related information

- Service and requestor/user ID/DN format that should allow for both using namespaces and context aware names semantics
- Attribute format (either X.509/X.521 or URN/SAML2.0 presentation)
- Context aware XACML policy definition using the Environment element of the policy Target element
- Security tickets and tokens used for AuthZ session management and for provisioned resource/service identification
- Security federations for users and resources, e.g. VO membership credentials