Authorisation Session Management in Complex Resource Provisioning

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

OGSA-AUTHZ WG meeting
7 May 2007, Manchester, UK

Outline

- Background: Use cases and origin projects
- General Complex Resource Provisioning (CRP) model
- AuthZ Service components to support AuthZ session (and dynamic security context) management
- AuthZ ticket format for extended AuthZ session management
- Future developments

gJAF – gLite Java AuthZ Framework

GT4-AuthZ – GT4 AuthZ Framework

GAAA – Generic Authentication, Authorization, Accounting

GAAA-AuthZ – GAAA AuthZ Framework

Background – Origin/Target projects

- AuthZ service for dynamic (distributed) Grid applications
  - Adding extended security context management to Grid oriented AuthZ Frameworks (EGEE gJAF and coordinated with Globus GT-AuthZ)
- Distributed multidomain Authorisation service for network on-demand services and OLPP
  - EU Project PHOSPHORUS and NL national project RoN GP-NG
    - Requires extended AuthZ/provisioning session context management in multidomain scenario
- Central Authorisation service for Grid based Collaborative applications
  - GAAA-AuthZ Architecture and Implementation (Collaboratory.nl, VL-e projects)
    - Domain based resource management and RBAC (RBAC-DM)
    - AuthZ session/ticket for AuthZ service performance optimisation

Complex Resource Provisioning (CRP)

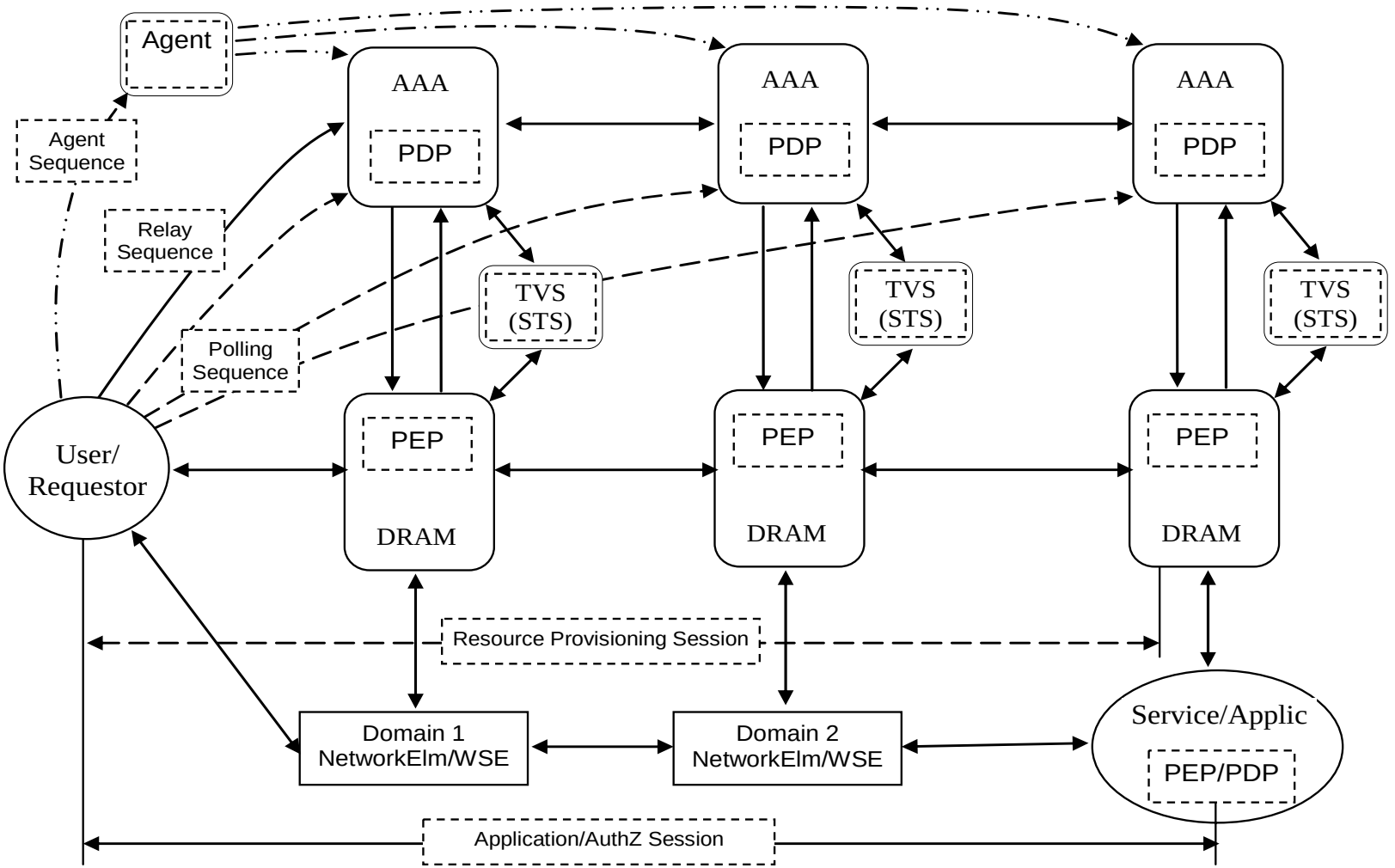
Basic use cases for CRP

- *OLPP and Network on-demand provisioning*
- *Virtual Laboratory - Hierarchical and distributed resources and user attributes*
- *Grid Computing Resource – Virtualised, distributed and heterogeneous*

2 major stages/phases in CRP operation

- *Provisioning consisting of 4 basic steps*
  - *Resource Lookup*
  - *Resource composition (including options)*
  - *Component resources reservation (including individual resources AuthZ resulted with "global" reservation ID)*
  - *Deployment*
- *Access (to the resource) or consumption (of the consumable resource)*
  - *Reservation/AuthZ decision enforcement*

CRP infrastructure elements and basic sequences



Provisioning sequences

* Polling
* Relay
* Agent

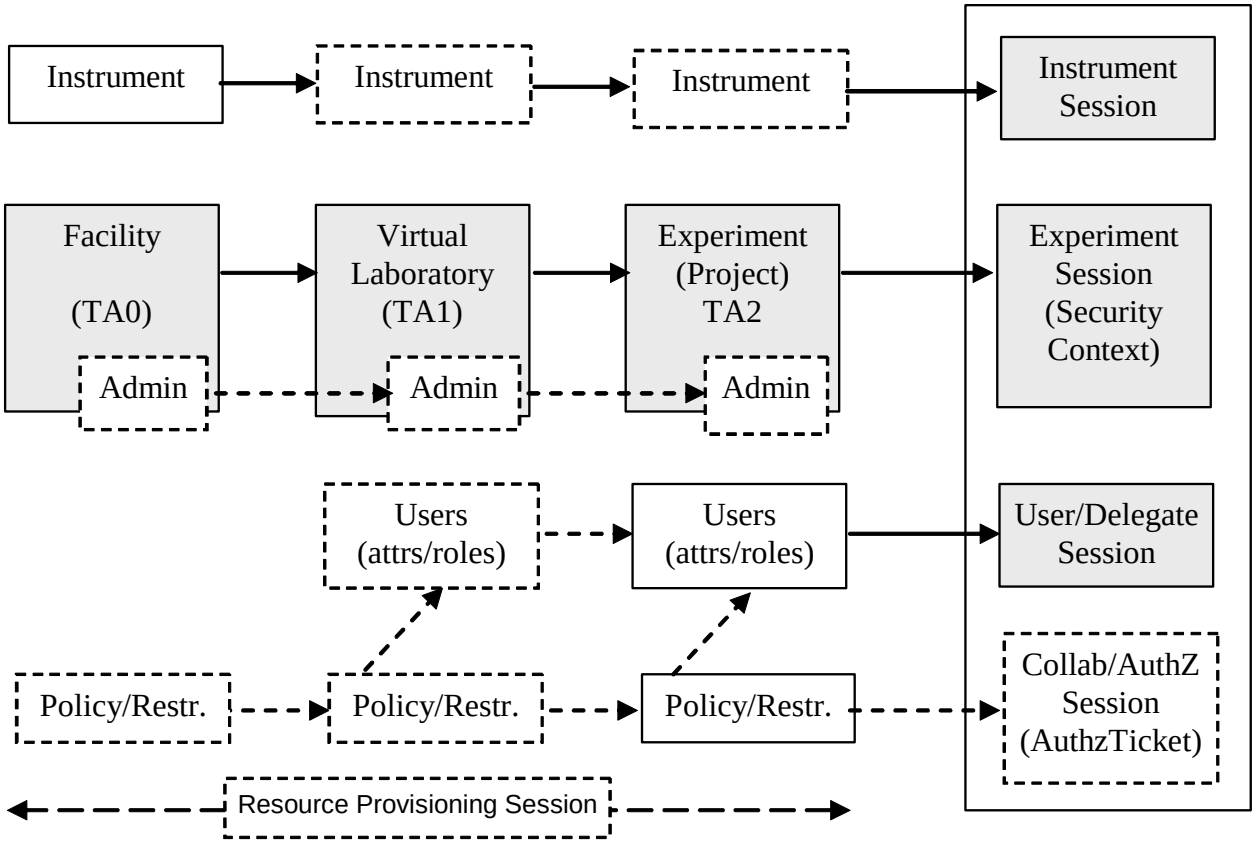
TVS – Token Validation Service

DRAM – Dynamic Resource Allocation and Mngnt

PDP – Policy Decision Point

PEP – Policy Enforcement Point

Domain based Resource management



- Implements RBAC3 model + Experiment AuthZ session management
- Uses XACML RBAC profile and XACML v3.0 administrative policy profile

Full Resource URI/ID –

CNL:Facility:VirtualLab:Experiment:InstrModel

Full User Session context –

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

Required AAA/Service plane functionality for OLPP/CRP

Authentication and Identity management

- Federated Identity and Federated Resource Access
- Attribute management (issue, validation, mapping, delegation)

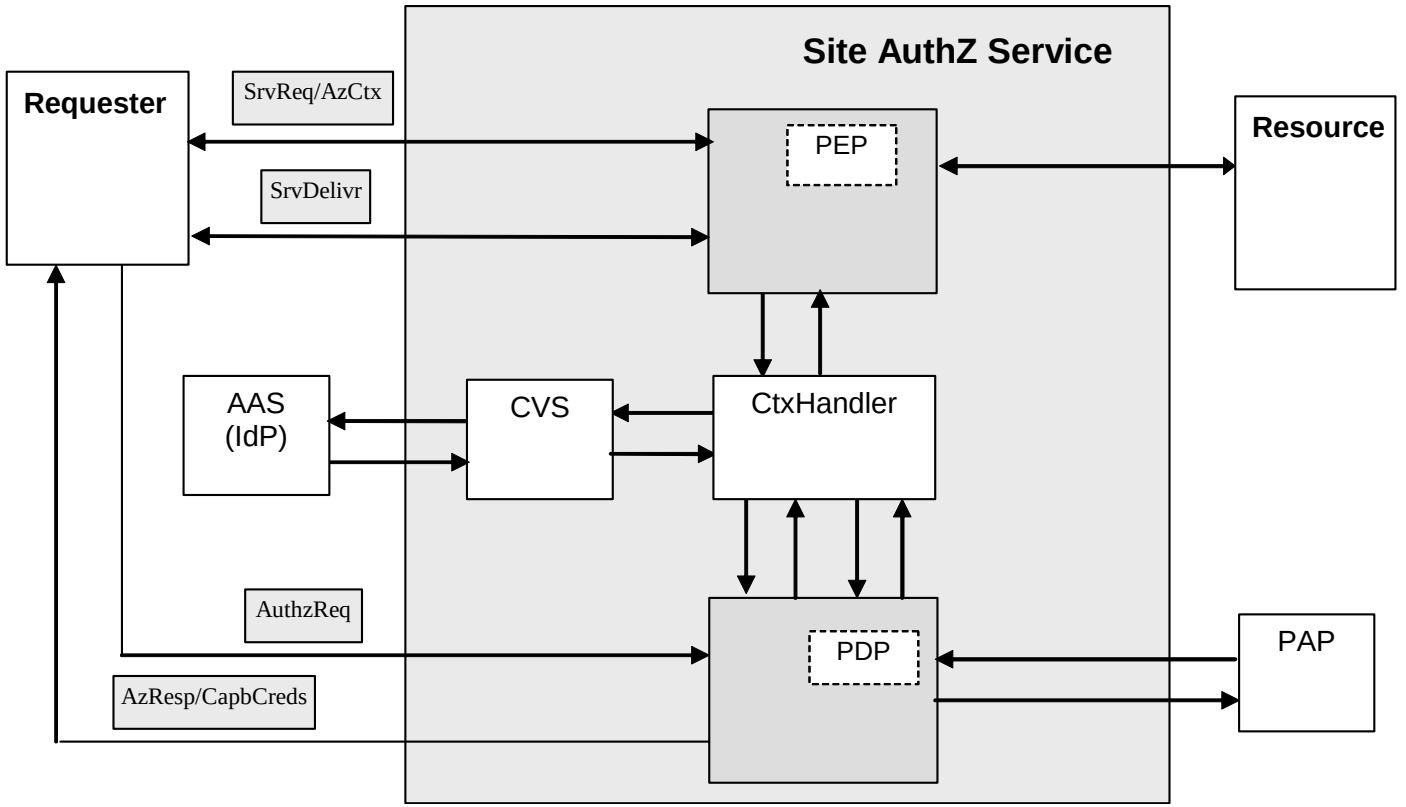
Authorisation

- Multidomain AuthZ policy and/or decisions combination
- AuthZ session Management to convey AuthZ decision between domains

Trust management

- User and Resource based Federations (Shibboleth, NREN/GN2 AAI, VO)
  - Pre-established trust relations
- Trusted Computing Platform (TCG)
  - Hardware rooted trust anchors allowing for initial trusted introduction
- DNSSEC – *for multidomain use cases*
  - Allows for DNS based VO certificates publishing to enable initial trusted introduction

AuthZ Service Components (by OGSA-AUTHZ)



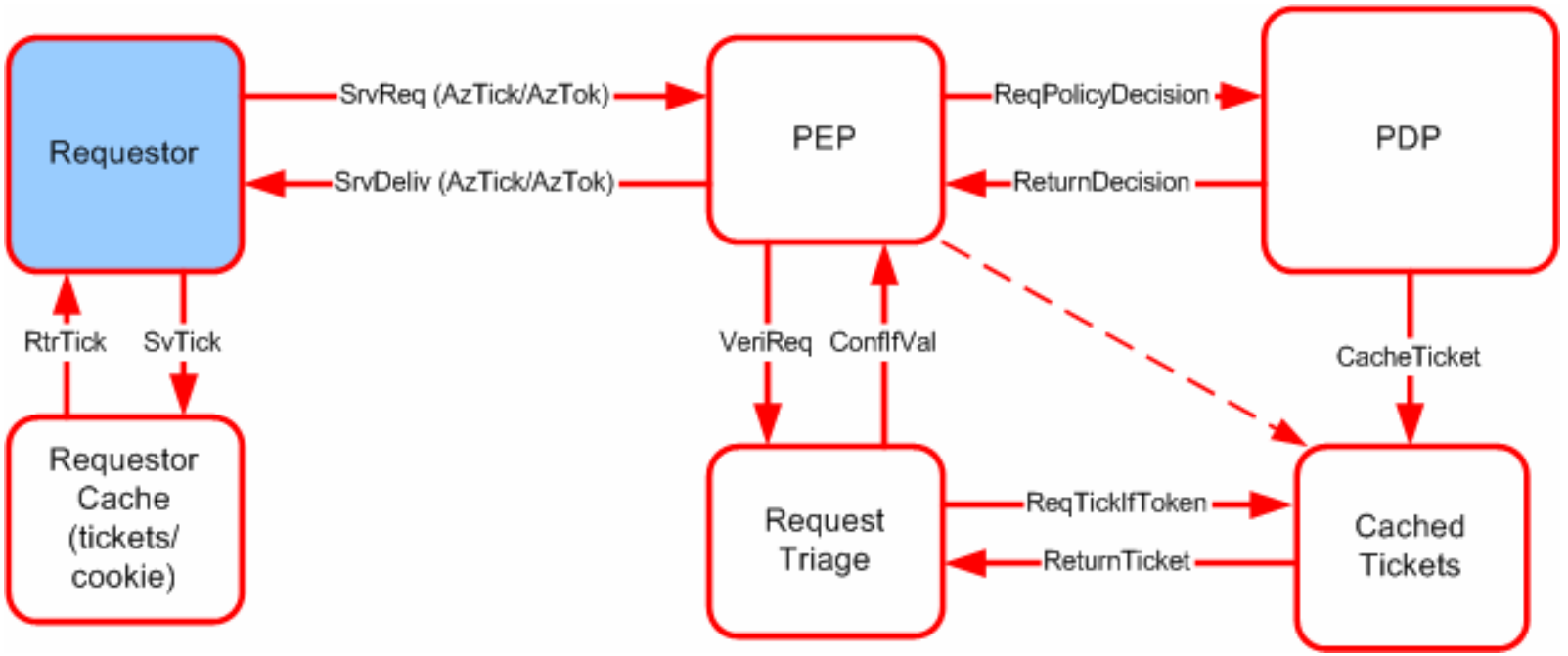
Basic sequences for attributes providing and for AuthZ decisions

- Push
- Pull
- Agent

CVS – Credentials Validation Service

PAP – Policy Authority Point

AuthZ session Tickets/Tokens handling in AuthZ system

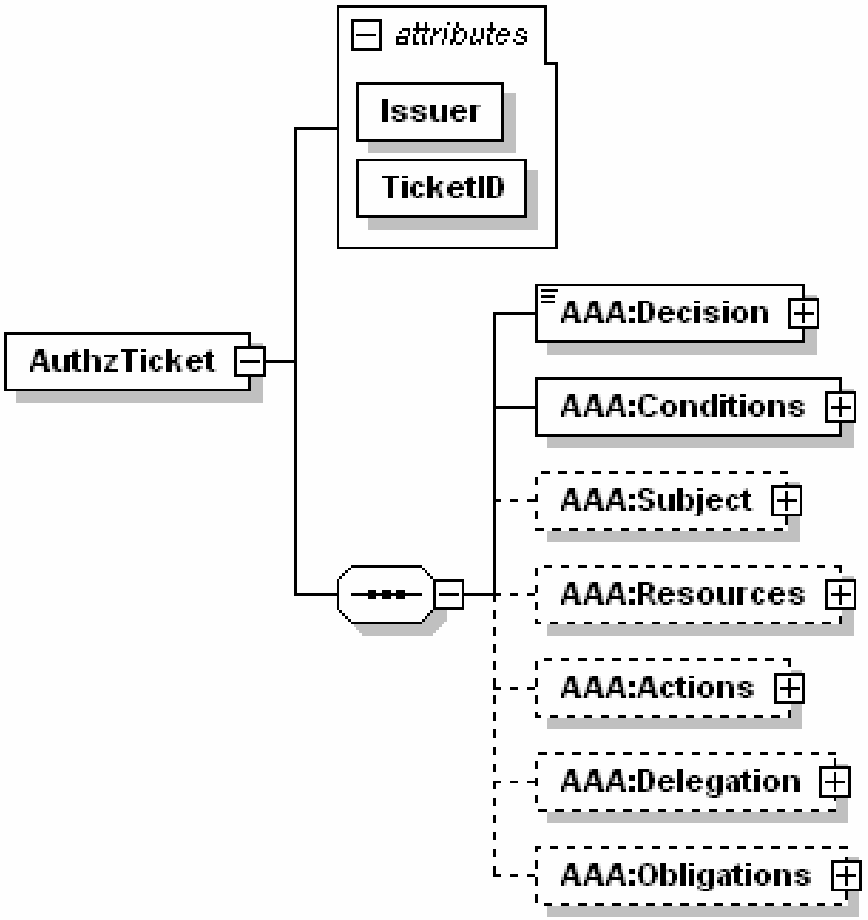


- AuthzTicket is issued by PDP and may be issued by PEP
- AuthzTicket must be signed
- AuthzTicket contains all necessary information to make local PEP-Triage Request verification
- When using AuthzTokens, AuthzTickets must be cached; Resolution mechanism from token to ticket must be provided

AuthZ Session management in GAAA-AuthZ

- 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 Assertion or AuthZ Ticket
  - SessionID is included into AuthzTicket together with other AuthZ Ctx information
  - 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 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

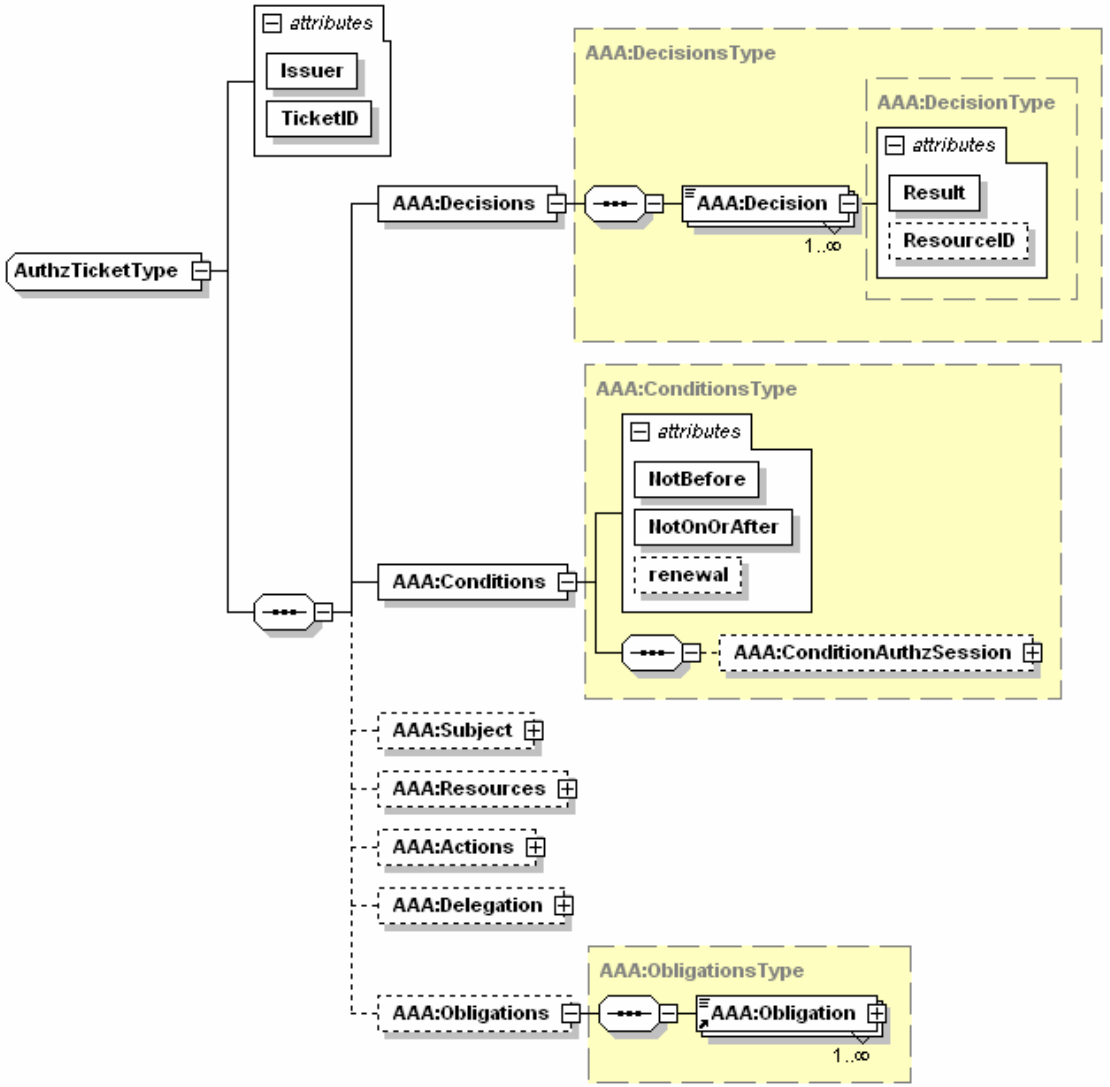
Ensure Integrity of the AuthZ decision

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

Confidentiality

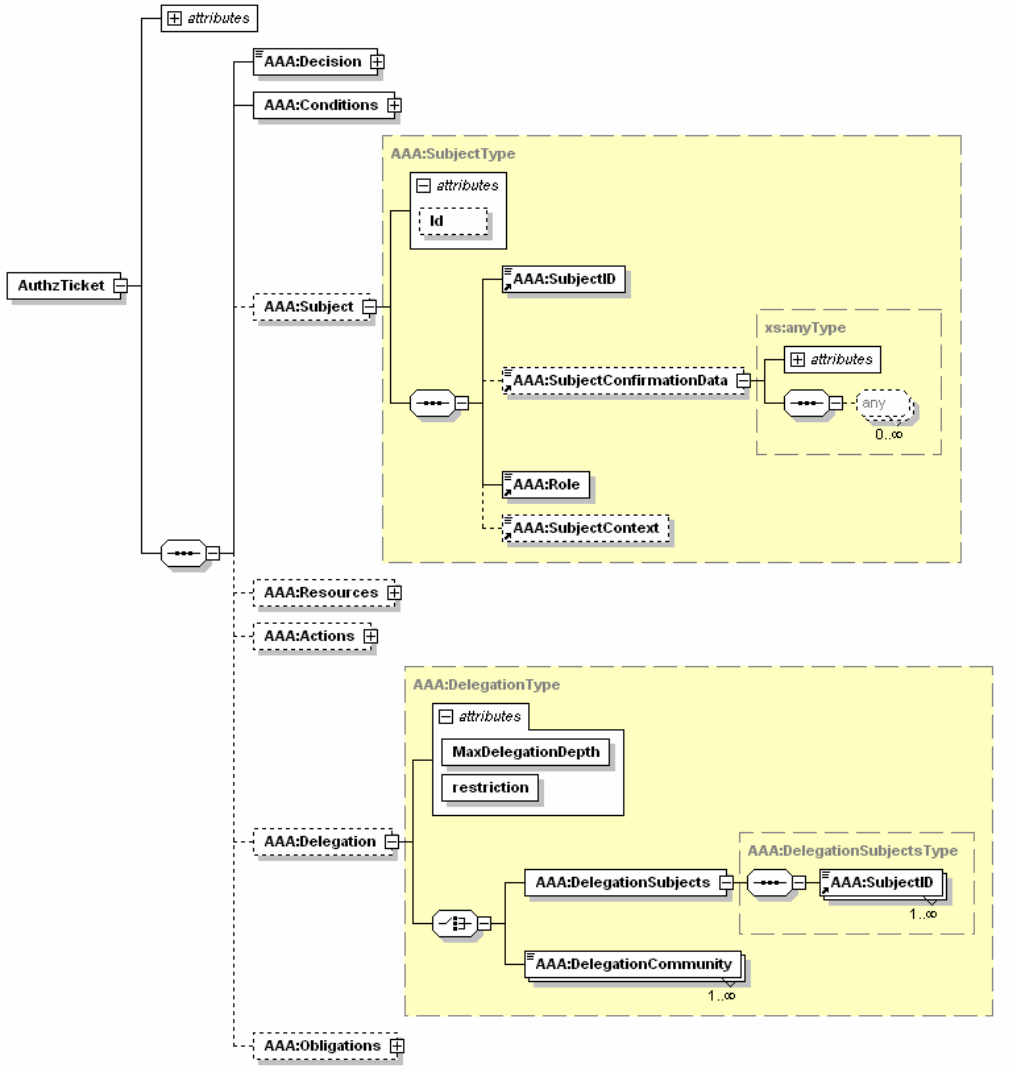
- Creates a basis for user-controlled Secure session

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

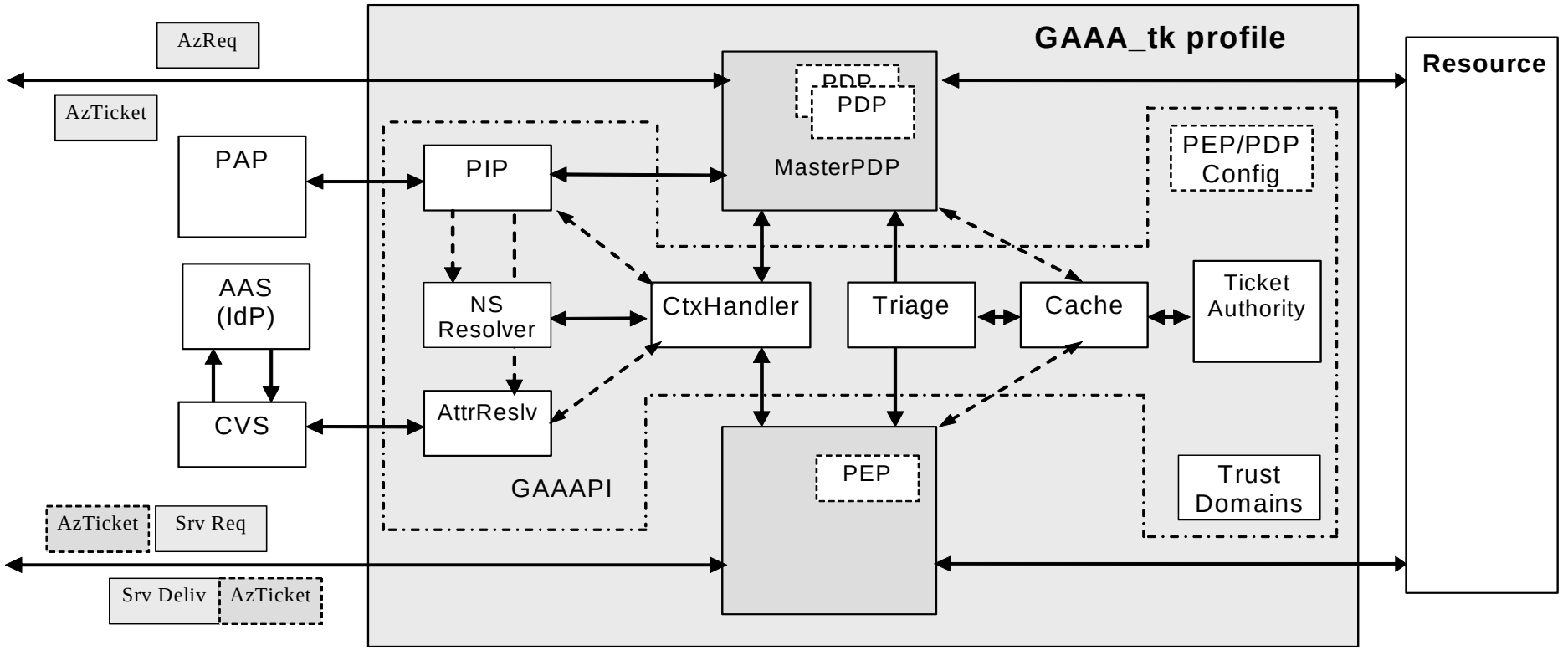
```
<AAA:AuthzToken TokenID="c24d2c7dba476041b7853e63689193ad">
<AAA:TokenValue>
0IZt9WsJT6an+tIxhhTPtiztDpZ+iynx7K7X2Cxd2iBwCUTQ0n61Szv81DKllWsq75IsHfusnm56
zT3fhKU1zEUsob7p6oMLM7hb42+vjfvNeJu2roknhIDzruMrr6hMDsIfaotURepu7QCT0sADm9If
X89Et55EkSE9oE9qBD8=
</AAA:TokenValue>
</AAA:AuthzToken>
```

AuthzToken is constructed of the AuthzTicket TicketID and SignatureValue

AuthzToken use suggests caching AuthzTicket's

AuthzToken can be used as cookie

GAAA-AuthZ/GAAAPI components to support dynamic security context management (1)

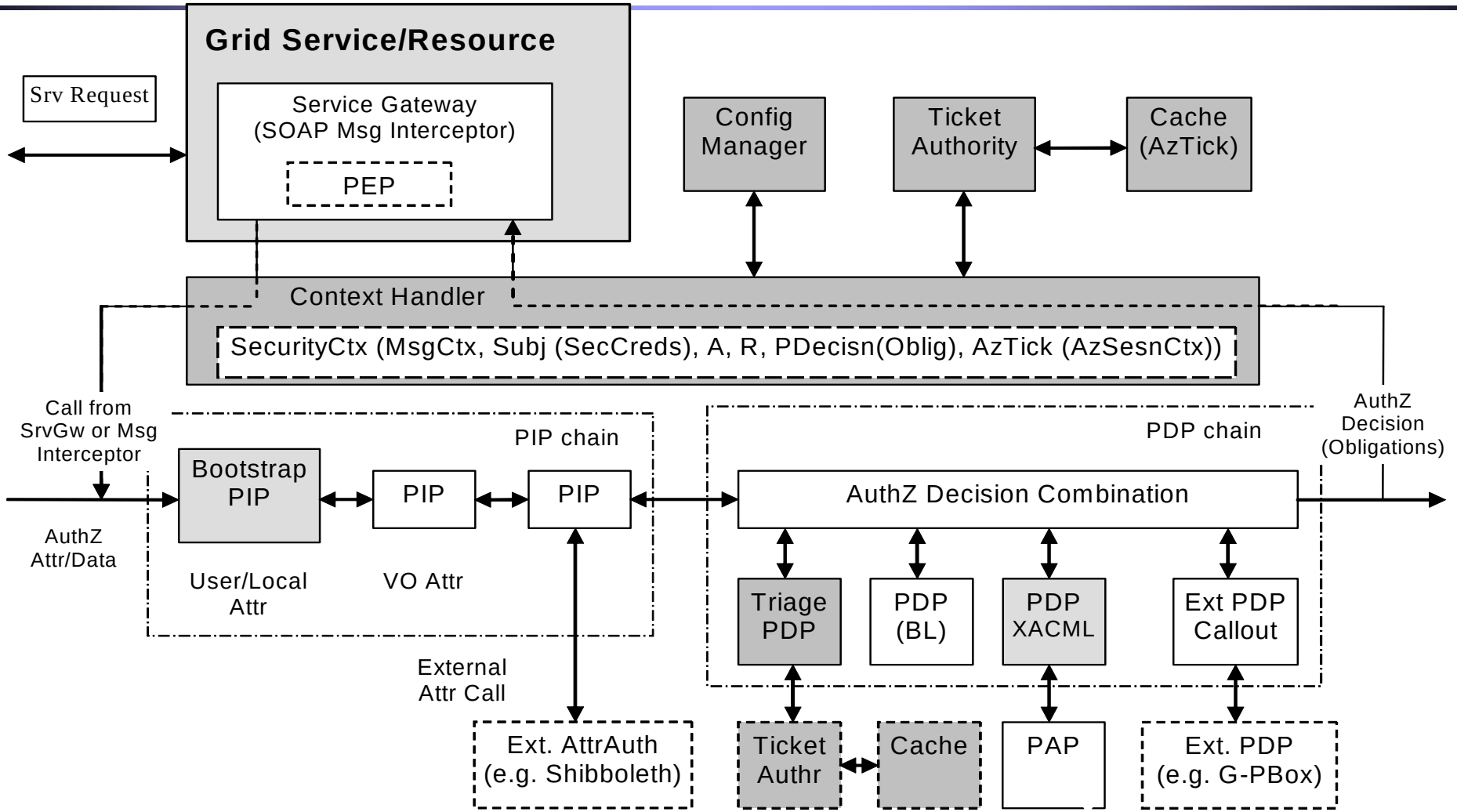


- GAAAPI is a collection of components to support PEP and PDP interaction, implemented in Java
- Needs Trust Anchor configuration in a distributed multidomain infrastructure

GAAAPI components to support dynamic security context management (2)

- Context Handler (CtxHandler) that calls to a namespace resolver (NS Resolver) and attribute resolver (AttrResolver), which in its own can call to external CVS or Attribute Authority Service (AAS) *to validate* presented attributes or obtain new ones
- Triage and Cache to provide an initial evaluation of the request, including the validity of the provided credentials
  - Used for handling AuthZ tickets/tokens, and also for AuthZ session management by evaluating service requests versus the provided AuthZ ticket/token claims
- Ticket Authority (TickAuth) generates and validates AuthZ tickets or tokens on the requests from PEP or PDP
  - to support AuthZ session, tickets are cached by TickAuth directly or by PEP/PDP
- Policy Information Point (PIP) that provides resolution and call-outs to related authoritative Policy Authority Points (PAP)

gJAF – Proposed Extensions



Future developments

- Proposing AuthZ session management framework to OGSA-AUTHZ
- Adding AuthZ session/dynamic security context management to gJAF (in coordination with GT4-AuthZ)
- AuthZ session management with the extended AuthZ ticket functionality
  - Including delegation and complex and obligated policy decisions
  - Needs more discussion on Delegation use cases and scenarios
- *Dynamic Trust management in multidomain CRP*

- Defining XACML policy profiles for
  - Grid applications (mapping between Grid specific policy formats – gridmap, ACL, GACL)
  - Different Resource models (hierarchical, ordered, mesh, etc.)