

Security Architecture

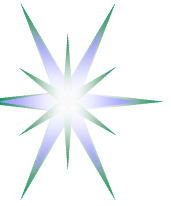
for

Open Collaborative Environment

Yuri Demchenko <demch@science.uva.nl>
AIRG, University of Amsterdam

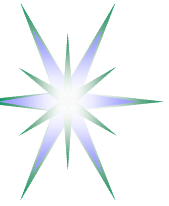
on behalf of

Yuri Demchenko, Leon Gommans, Cees de Laat, Bas Oudenaarde
University of Amsterdam
Andrew Tokmakoff, Martin Snijders, Rene van Buuren
Telematika Institute



Outline

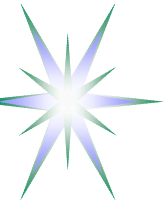
- General Security architecture for Web Services and Grid
- Security architecture and Job-centric security model for Open Collaborative Environment (OCE)
- Using Generic AAA Authorisation framework and RBAC for fine grained access control
 - ◆ Combined/optimised push-pull-agent model using AuthZ tickets and tokens
- GAAAPI and implementation details – Collaboratory.nl project
- Prospective integration with GT4 and gLite Authorisation Frameworks
 - ◆ Integration problems and strategy
- Acknowledgements



Common and specific problems/concerns addressed

- Access control service performance
 - ◆ Ticket/token system – integrity and secure context mngnt
- Key management and Trust relations in distributed access control infrastructure
- Compatibility and integration with existing access control tools
 - ◆ Common format for policy exchange and combination
- Critical and optimistic look at WS technology

- Open Collaborative Environment specifics
 - ◆ Dynamic and multidomain
 - ◆ Customer driven
 - ◆ Human controlled and interactive



General Web Services and Grid Security Architecture

Site Security Services

- Intrusion Detection and Incident Response
- Secure Logging
- Policy Management (AuthZ, Privacy, Trust, Federation)
- User Management
- Key Management

Services/Operational Security

- Authentication and Identity Management
- Authorisation (Access Control)
- Secure Context Management
- Auditing and Notarisation

Policy Expression and Exchange

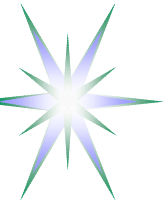
- Identity/Federation Policy
- Authorisation Policy
- Trust Management Policy
- Confidentiality and Privacy Policy

Messaging Security: SOAP/WS- Security

Communication/Transport Security: SSL/TLS, VPN, IPsec, Stunnel, etc.

XML Security (XMLSig, XMLEnc, XKMS)
 SAML, XACML
 WS-Security (SOAP Security), WS-Addressing, WS-ReliableMessaging
 WS-Trust, WS-Policy, WS-PolicyAssertion, WS-PolicyAttachment, WS-SecureConversation, WS-Federation
 WS-Agreement, WS-Coordination, WS-AtomicTransaction, WS-BusinessActivity, WSBPEL
 Device Profile for WS, WS-I Basic Security Profile

Most of above:
 Thanks to (the guerrilla activity of) Microsoft, IBM, BEA, VeriSign, RSA



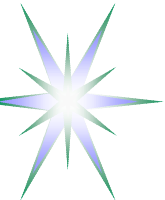
So, any WS-Alternative to WS-EveryWhere ?

Quoted from “The Loyal WS-Opposition” (2004/09/18) at
<http://tbray.org/ongoing/When/200x/2004/09/18/WS-Oppo>

So here’s what I’m going to do. I’m going to stay out of the way and watch the WS-visionaries and WS-dreamers and WS-evangelists go ahead and WS-build their WS-future. Because I’ve been wrong before, and maybe they’ll come up with something that WS-works and people want to WS-use. And if they do that, I’ll stand up and say “I was WS-wrong.”

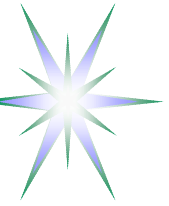
BUT do we have WS-Alternative to:

- Services and runtime decoupling and integration?
- End-to-end and Message/Document/Data centric security model?
- Customer driven or provider independent security model?
- Ontologies/Semantics/Namespace context management?

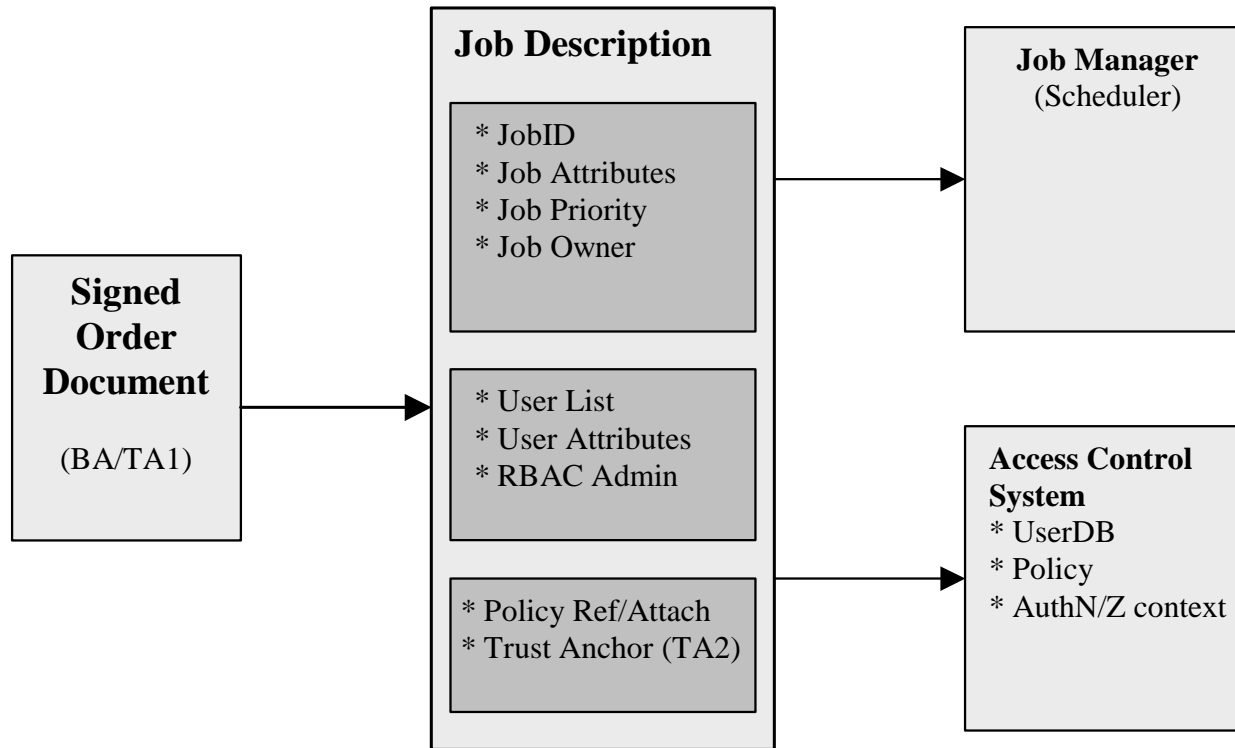


Distributed Security Architecture for OCE and used technologies

- Based on the Job-centric security model
 - ◆ Job description format – to be compatible with WS-Agreement and GGF JSDL (Job Submission Description Language)
- Extended RBAC functionality including RBAC administration tool (using GAAA Toolkits)
- GAAA RBE and AAA policy expression
 - ◆ XACML Request/Response messaging
 - ◆ Migration to XACML based policy exchange and combination
- SAML 2.0 based AuthzTicket format
- XML Signature and XML Encryption for JobDescription and AuthzTicket security
- Policy binding to WSDL and AuthZ portType definition
 - ◆ Using WS-Security Framework and OGSA/WSRF
- TODO: Adding VO and VOMS functionality - for user and resource attributes management



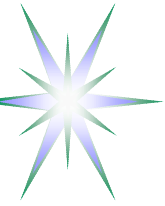
OCE Security built around Job description



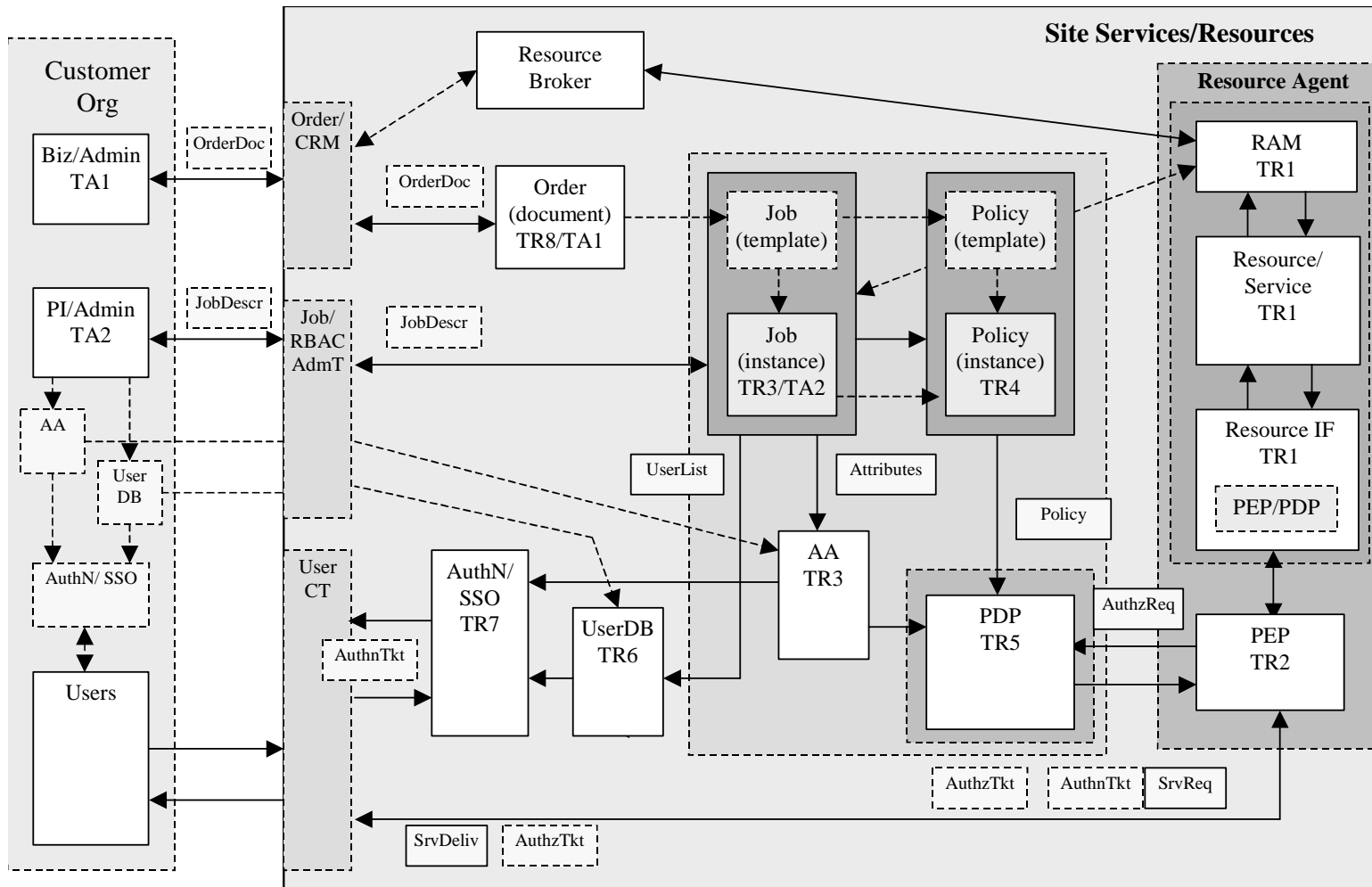
Job Description as a semantic object defining Job attributes and User attributes

- Requires document based or semantic oriented Security paradigm

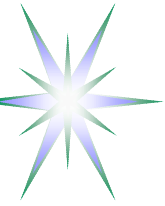
Trust domain based on Business Agreement (BA) or Trust Agreement (TA) via PKI



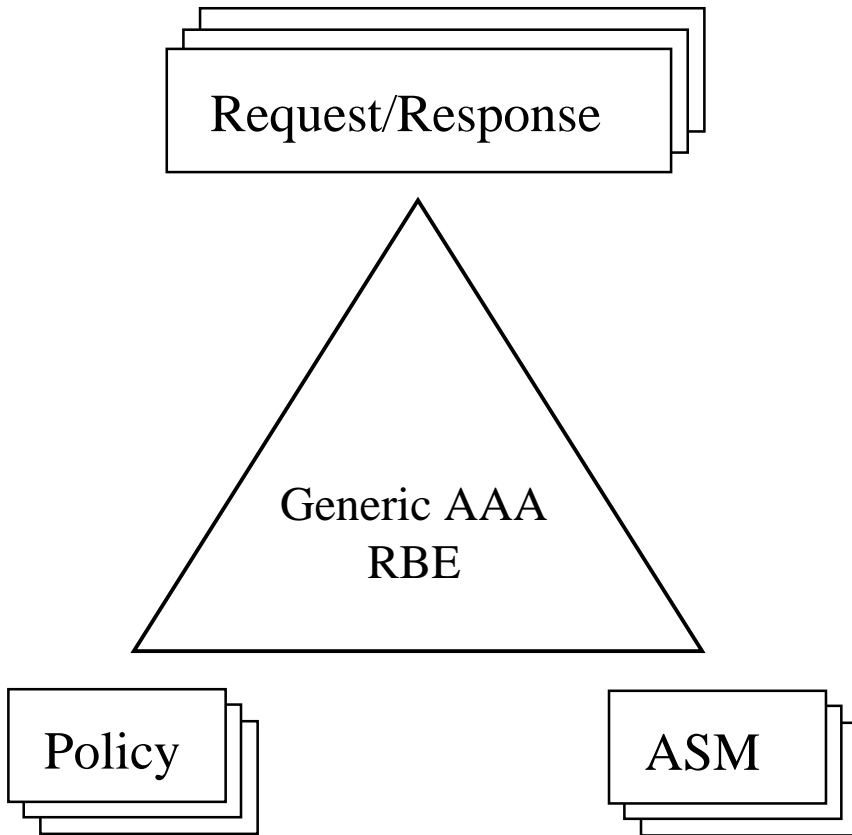
Major interacting components and entities in the Job-centric security model



TA – Trust Anchor
TR# - trust path from root (resource)
RAM – Resource Allocation and Management
UserCT – User Collaborative Tools



(1) Generic AAA Architecture by AIRG (UvA)



Policy based Authorization decision

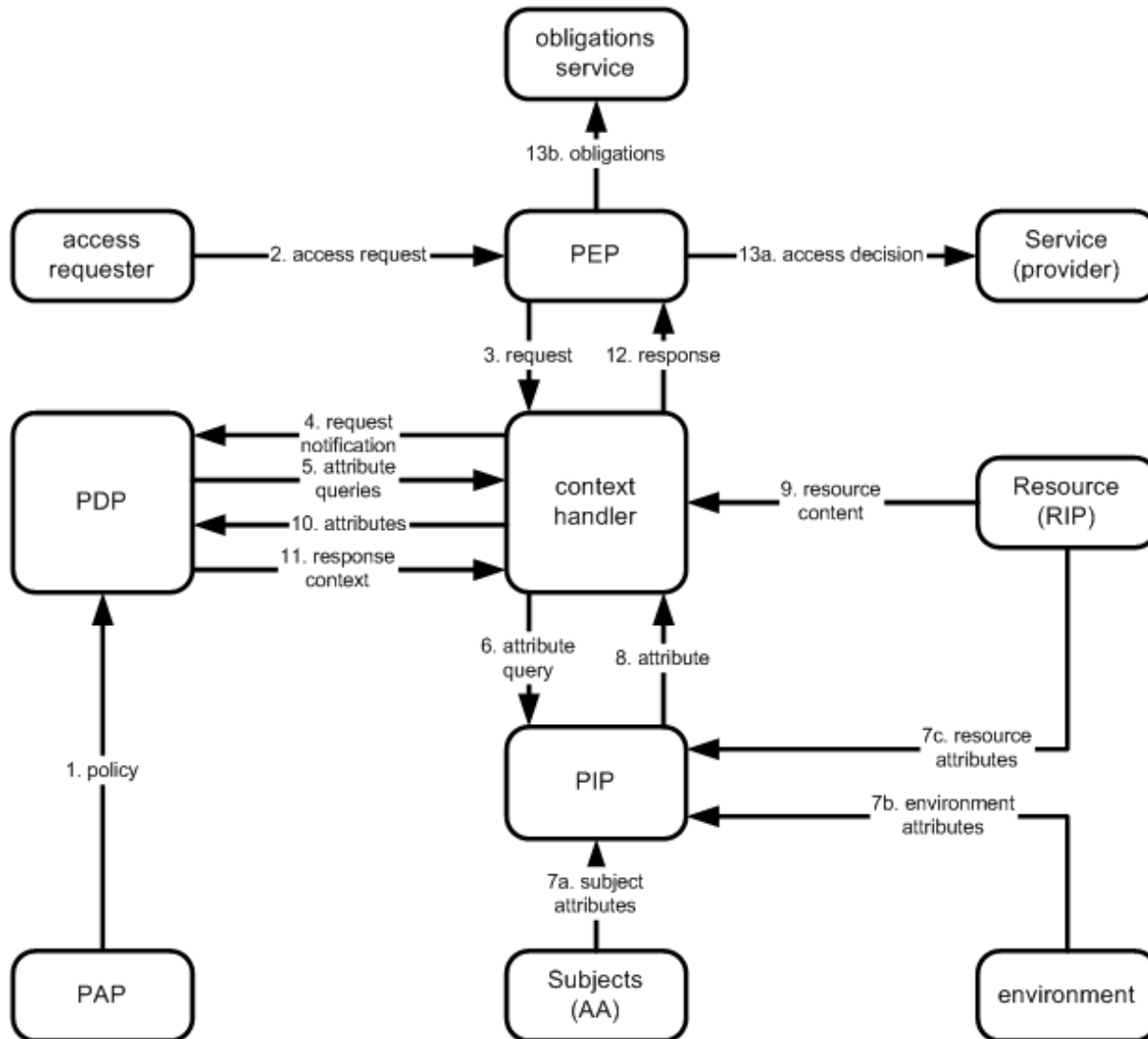
- Req {AuthNtoken, Attr/Roles, PolicyTypeId, ConditionExt}
- RBE (Req + Policy) => => Decision {ResponseAAA, ActionExt}
- ActionExt = {ReqAAAExt, ASMcontrol}
- ResponseAAA = {AckAAA/RejectAAA, ReqAttr, ReqAuthN, BindAAA (Resource, Id/Attr)}

•Defined by
Resource owner

•Translate logDecision => Action
•Translate State => LogCondition



(2) RBAC: main components and dataflow – XACML model



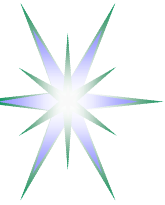
PEP/AEF - Policy Enforcement Point (authorisation enforcement function)

PDP/ADF - Policy Decision Point (authorisation decision function)

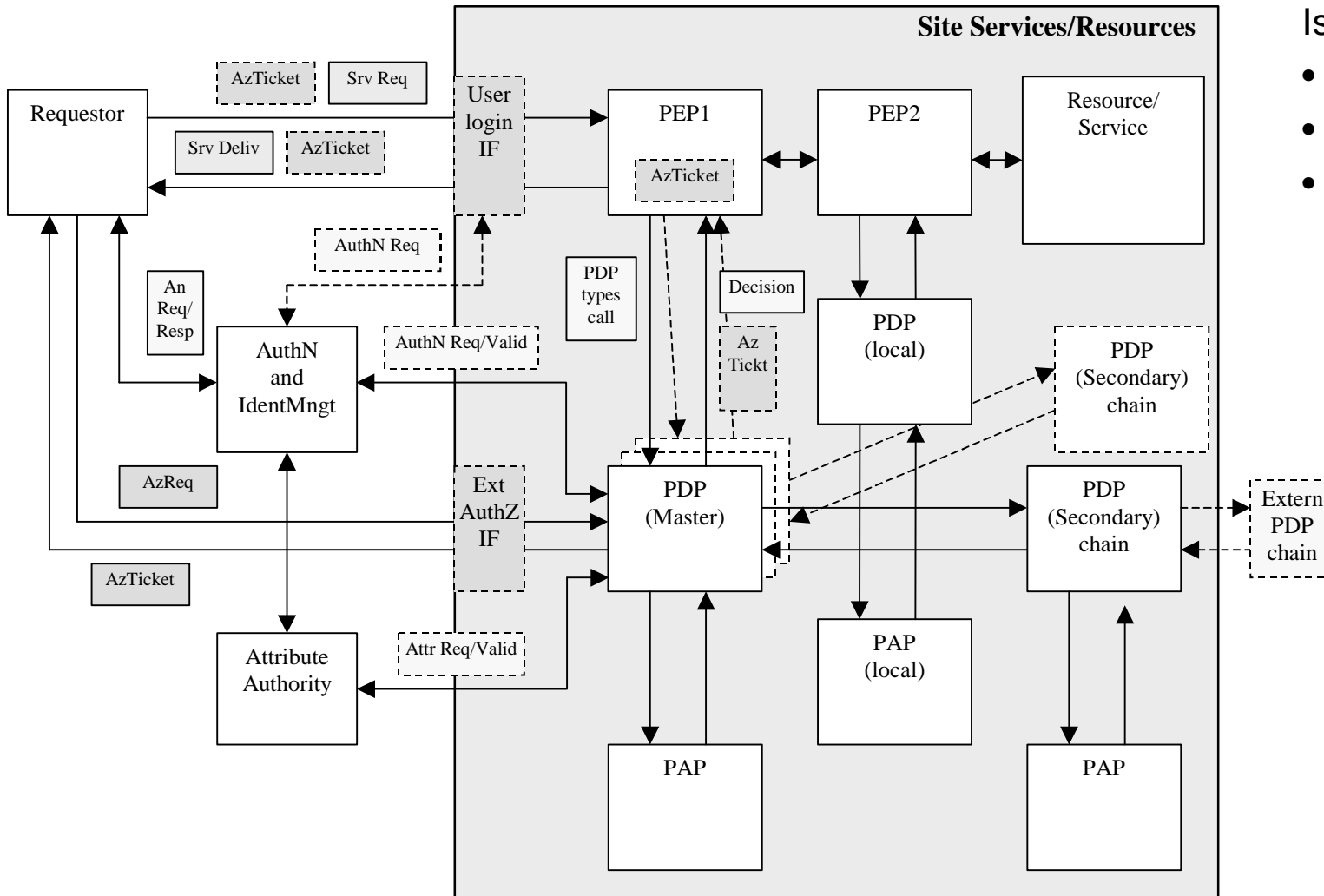
PIP - Policy Information Point

AA - Attribute Authority

PAP - Policy Authority Point

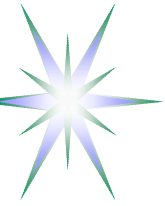


Site AuthZ service implementing RBAC and combined pull-push model

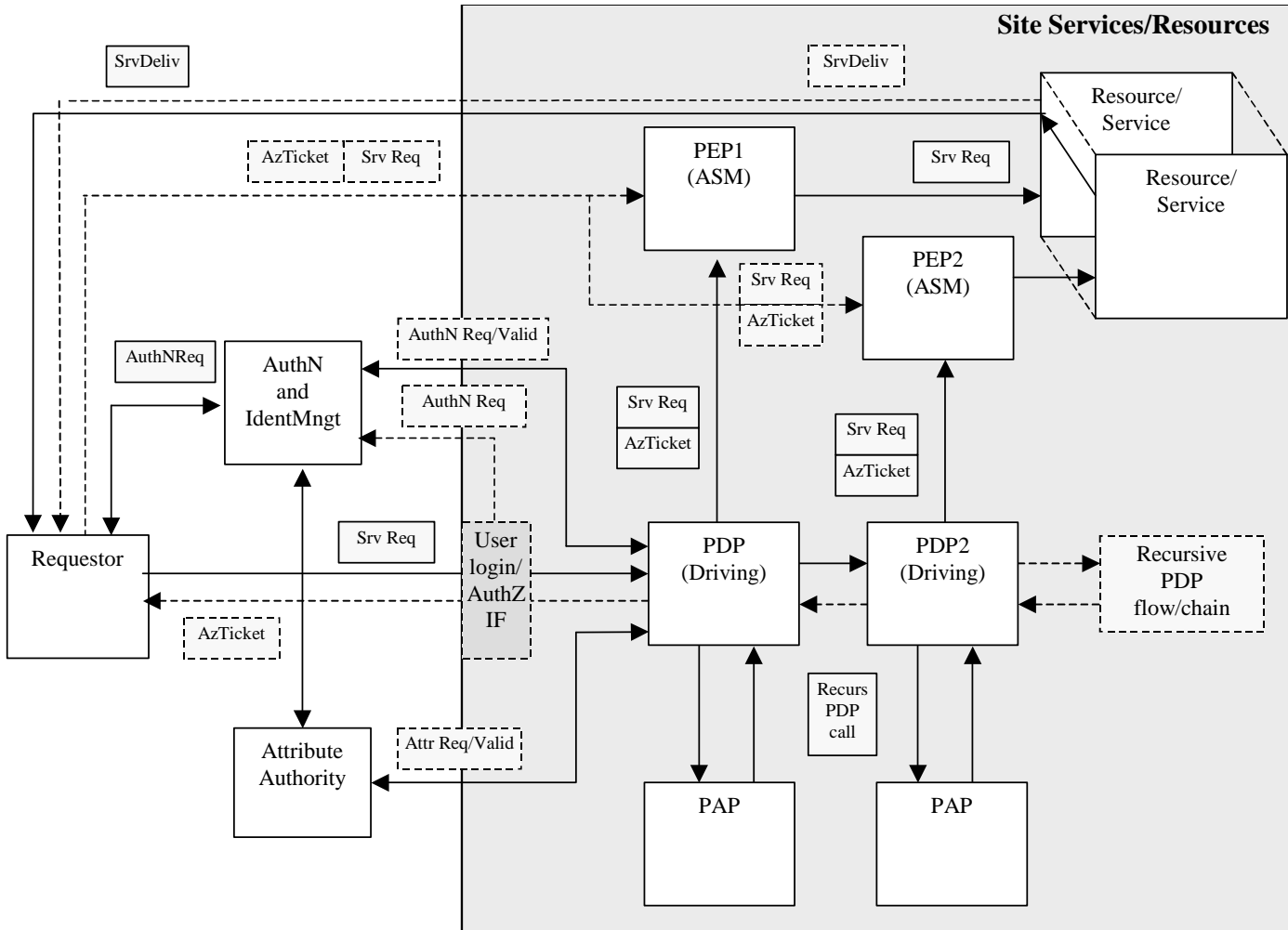


Issues to be addressed:

- PEP and PDP chaining
- Policy combining
- Multiple domains

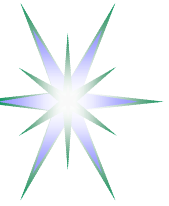


Site AuthZ service implementing combined agent and push model for complex resource



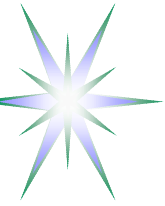
Issues to be addressed:

- Multi-component and multidomain resources
- Policy push and/or token based access control



Implementation suggestions for OCE

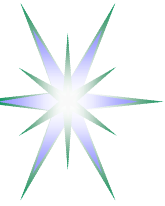
- PDP and PAP must share common namespace
- Policy and respectively PAP should be referenced in the request message explicitly or known to PEP and PDP a priory
- Every PEP in the chain of policy enforcement should take care of the whole request evaluation/enforcement by calling to a single (master) PDP. PEP should not do multiple decision combination.
- Only one PDP should provide a final decision on the whole request
- However, PEP may have a possibility to request different PDP types based on request semantics/namespace and referred policy
- When using ticket/token based access control model, the PEP should understand and have a possibility to validate the AuthZ ticket issued by trusted PDP
 - ◆ The AuthZ ticket should have validity and usage restriction and contain information about the decision and the resource.
- For the further validation of the AuthZ tickets/token, the PEP may cache the ticket locally to speed-up the validation procedure.



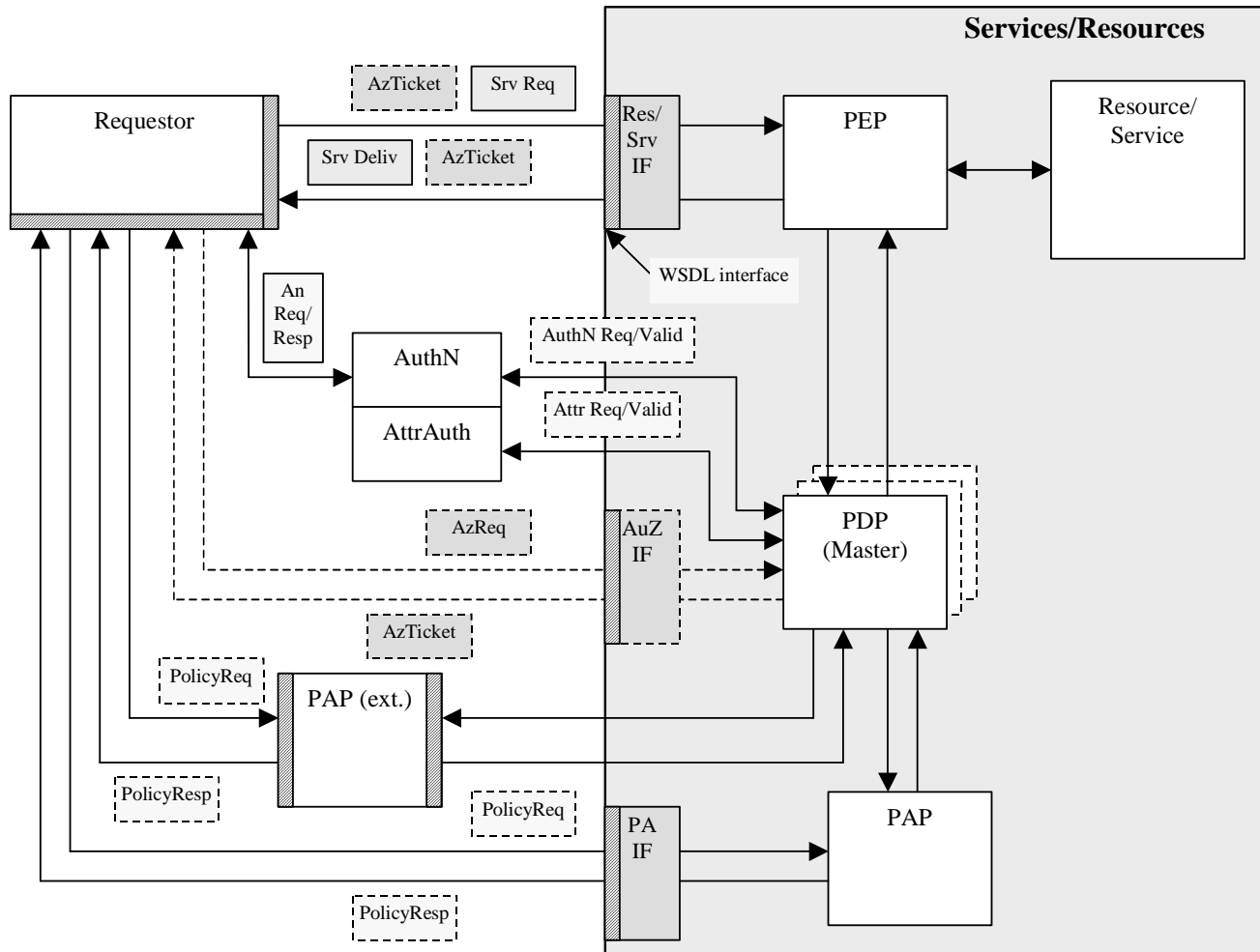
Traditional Access Control model – setting up trust and authority relations

- Policy, attributes semantics and namespaces are known a priori to all participating parties
 - A requestor knows what information to present to adhere to a specific policy and in what format (although PEP may act as ASM)
- PEP and PDP locations are known and interacting parties are known
- Trust relations between PDP, AA and resource are established
 - Resource trusts PDP's decision that can be delivered to a Resource in a form of AuthzTicket or based on default trust between PEP and Resource
 - Root of policy enforcement hierarchy, like in real life, belongs to the resource owner

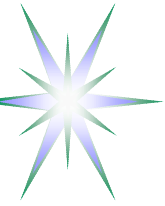
This approach is not sufficient for effective Service Oriented Architecture (SOA)



Open policy enforcement model in WSA/SOA using WS-Policy attachment mechanisms



- Linking dynamically all components of the access control system
- Policy is attached to any component of the service description in WSDL format
- Interacting services will fetch policy document and apply restrictions/rules to elements, which declared policy compliance requirements
- Provides a basis for mutual authorisation

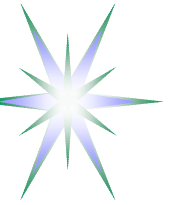


Attaching policy to WSDL - Example

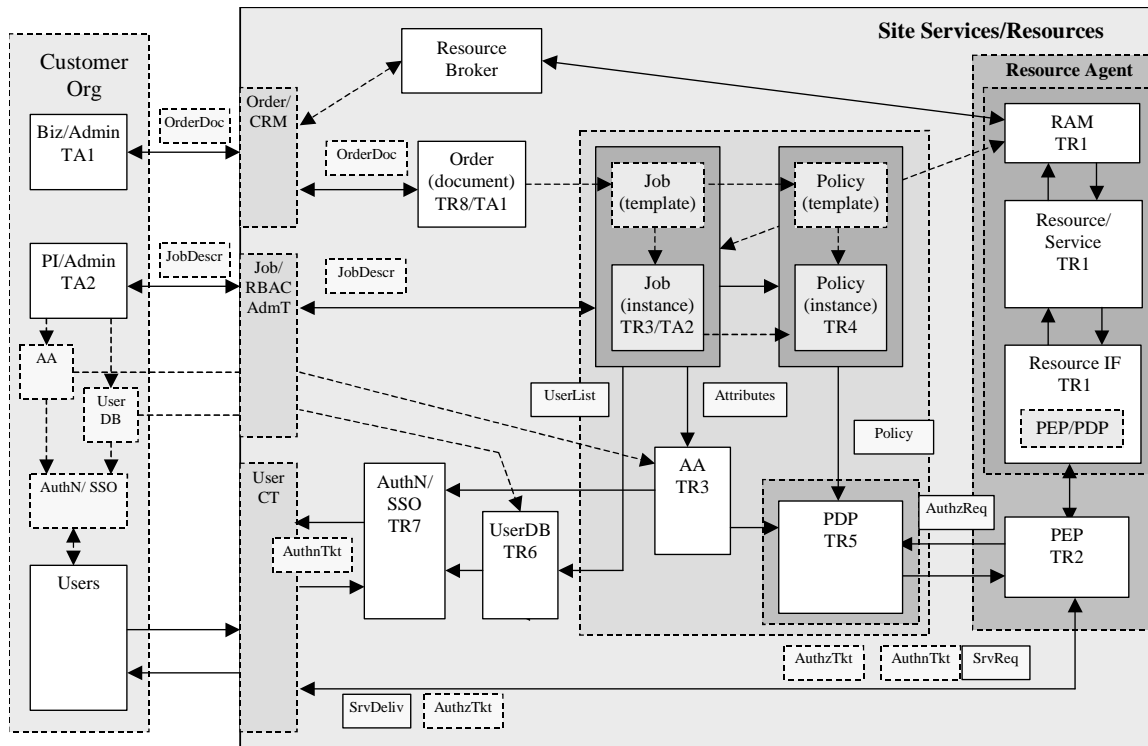
```
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/"
  < ... snip long namespace declaration ... >
  xmlns:wsp="http://schemas.xmlsoap.org/ws/2002/12/policy"
  xmlns:cnl="http://cnl.telin.nl/cnl" xmlns:policy="cnl-policy-schema.xsd"
  targetNamespace="http://cnl.telin.nl/cnl">
  <message name="ViewExperimentRequest" wsp:PolicyURIs="cnl-policy-02example.xml">
    <part name="coordinateX" type="xs:string"/>
    <part name="coordinateY" type="xs:string"/>
    <part name="zoom" type="xs:int"/>
  </message>

  <<< snip >>>>
  <wsp:PolicyAttachment ... >
    <wsp:AppliesTo>
      <x:DomainExpression/> +
    </wsp:AppliesTo>
    ( <wsp:Policy>...</wsp:Policy> |
      <wsp:PolicyReference>...</wsp:PolicyReference> ) +
      <wsse:Security>...</wsse:Security> ?
    ...
  </wsp:PolicyAttachment>

  <wsp:UsingPolicy wsdl:Required="true"/>
</definitions>
```

Trust relations in distributed AAA infrastructure

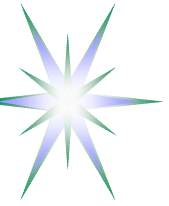


Trust/credentials chain and delegation between major modules:

User =>
=> HomeOrg.staff(TA2) =>
=> Job.members =>
=> Member.roles =>
=> Role.permissions

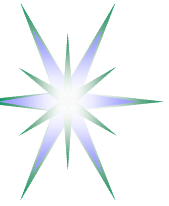
The process of obtaining required permissions to perform requested action by the user:

User => AuthN(HomeOrg.staff, Job.members) =>
=> AuthZ(Member.roles, Policy.permissions) =>
=> Resource.permissions



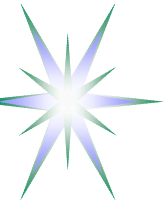
Implementation suggestions for OCE/CNL

- Root of trust and authority belong to the Resource
- Trust anchor TA2 embedded into the Job Description is the main trust anchor shared between the resource and the customer.
 - ◆ In more business integrated model the signed order may contain TA1
 - ◆ Both TA2 and TA1 may have the same trust path to the root/resource
- To become a shared trust anchor for the resource and the customer trust domains, the Order or JobDescription must contain mutually signed credentials/certificates
- Although the main PEP operation assumes authorisation decision request from the trusted PDP, in general PEP may accept an AuthzTicket from the trusted external PDP

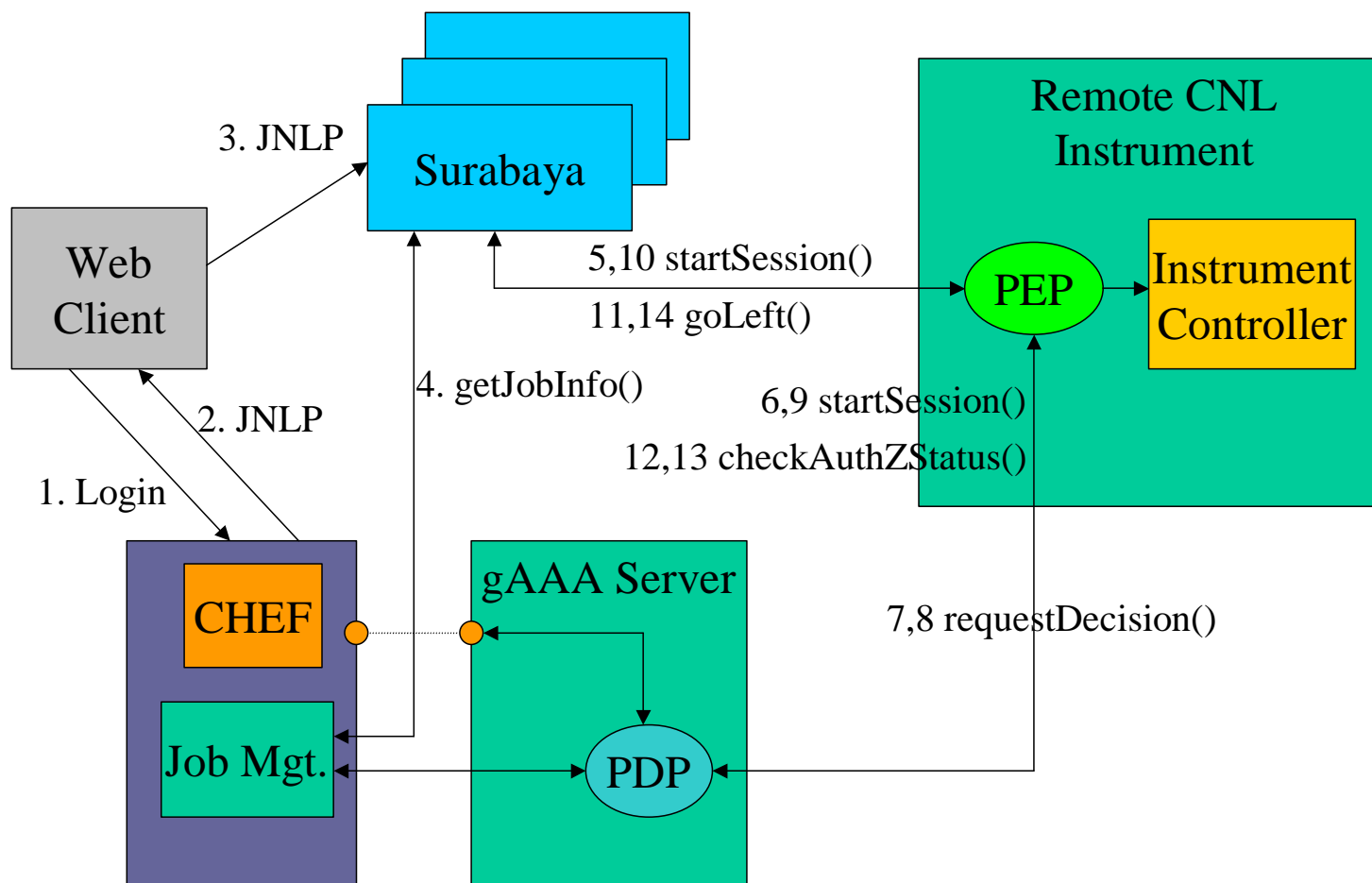


Before deploying security infrastructure

- Design conventions and agreements
 - Key distribution and trust establishing
 - ◆ In search of simple consistent model
 - Policy definition including subject, attributes, actions semantics and namespaces
 - ◆ Compatibility with existing, e.g. SAML, XACML
 - Security credentials format
 - ◆ Standard vs proprietary
 - Protocols and Messages format
 - ◆ SOAP + XACML Request/Response
 - ◆ SOAP + SAML + XACML



Authorisation Service operation in a CNL2 Demo system

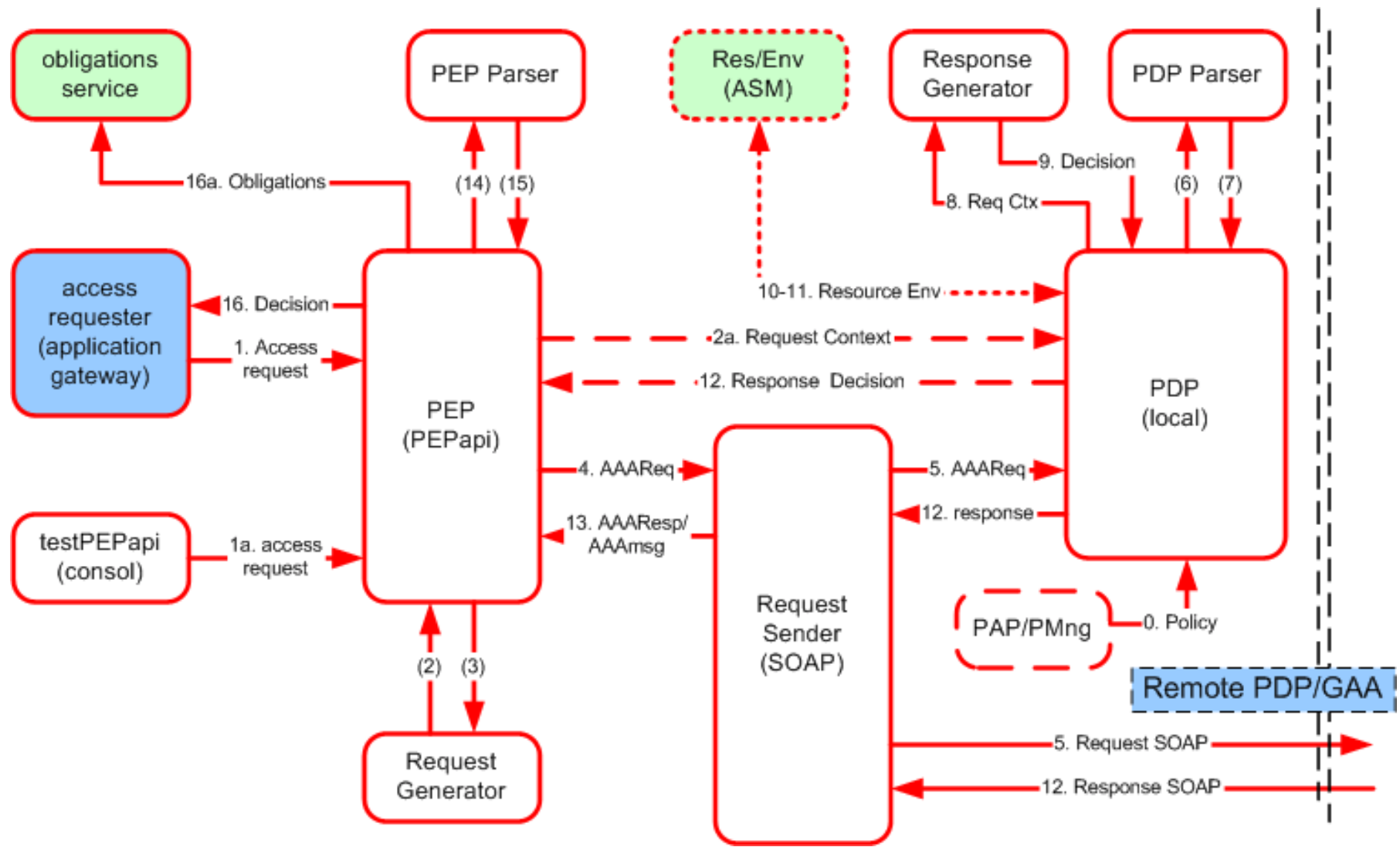


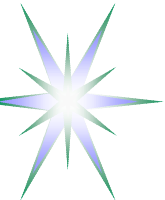
JNLP – Java Network Launch Protocol
CHEF – Collaborative tool
Surabaya – Collaborative Workspace environment

Note: we assume SSL TCP connections all over.

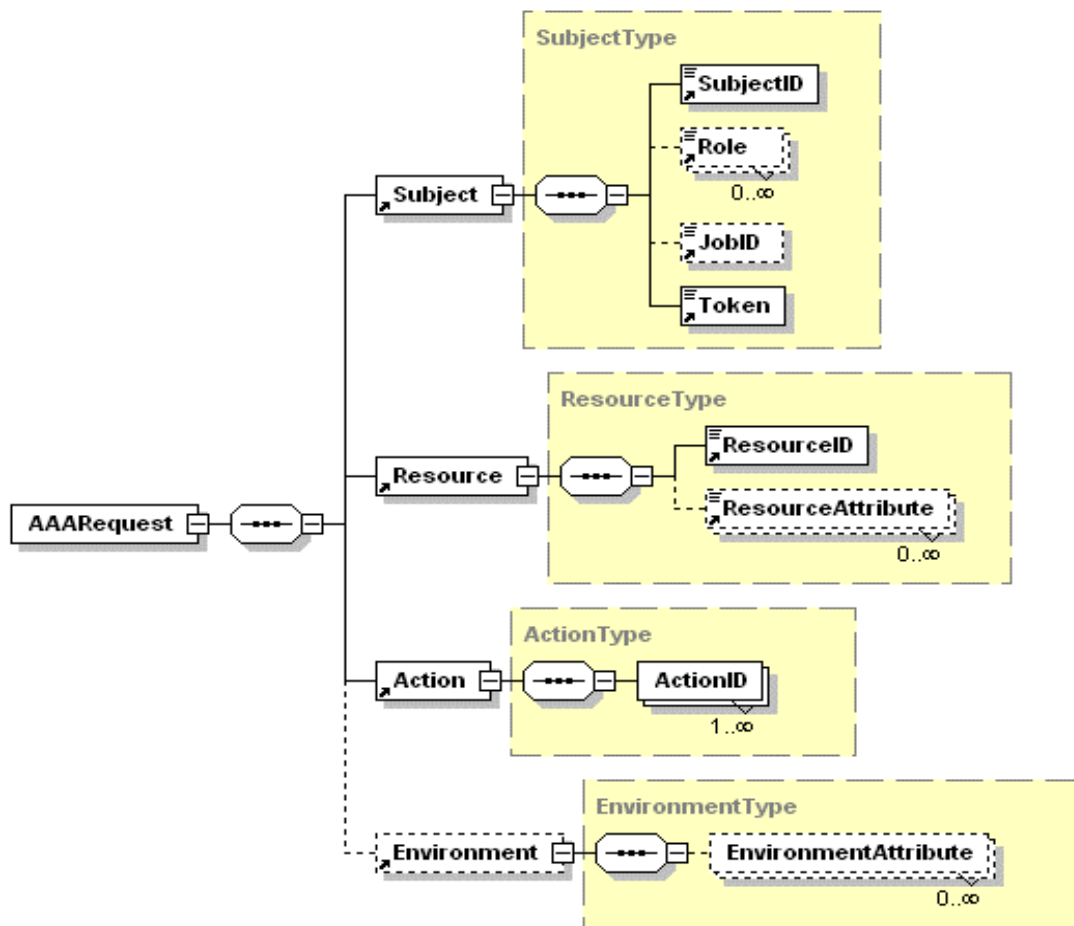


GAAAPI flow diagram (implements RBAC)

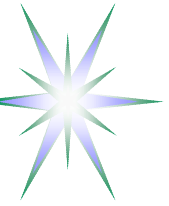




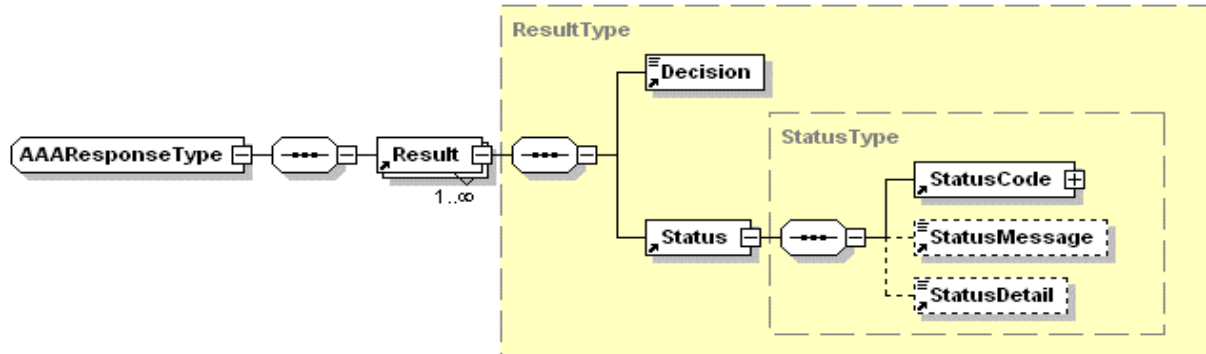
GAAAPI implementation – XACML Request message format (1)



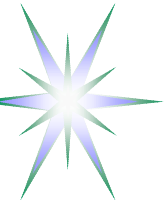
```
<?xml version="1.0" encoding="UTF-8"?>
<AAA:AARequest
  xmlns:AAA="http://www.AAA.org/ns/AAA_BoD"
  xsi:schemaLocation="http://www.AAA.org/ns/AAA_BoD http://146.50.22.64/CNLdemo1.xsd"
  version="0.1" type="CNLdemo1">
<Subject>
  <SubjectID>
    WHO740@users.collaboratory.nl</SubjectID>
  <Token>
    2SeDFGVHYTY83ZXxEdsweOP8Iok)yGHxVfHom90
  </Token>
  <JobID>JobID-XPS1-212</JobID>
  <Role>Analyst@JobID</Role>
</Subject>
<Resource>
  <ResourceID>
    http://resources.collaboratory.nl/Phillips_XPS1
  </ResourceID>
</Resource>
<Action>
  <ActionID>ControlInstrument</AttributeID>
</Action>
</AAA:AARequest>
```



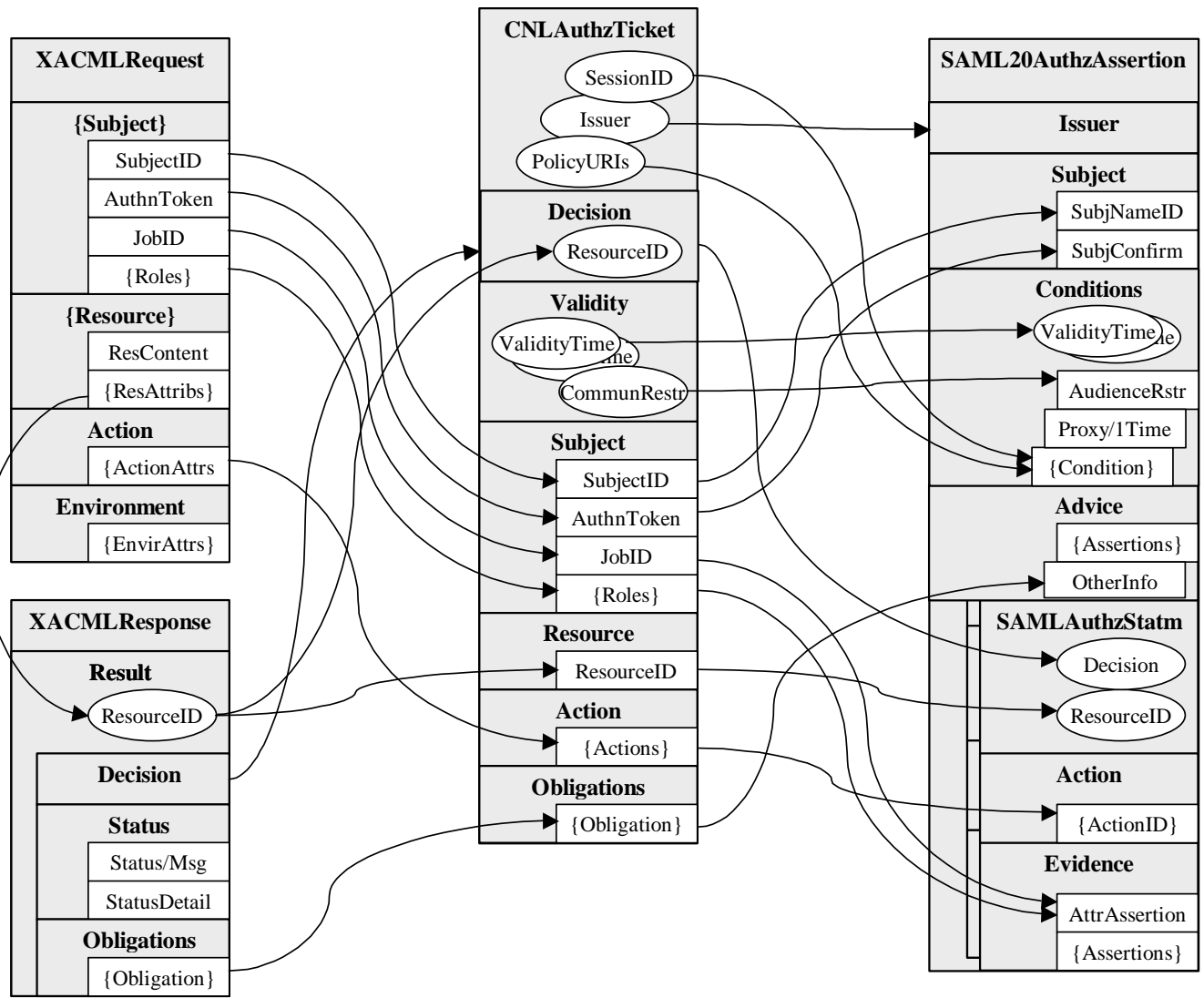
GAAAPI implementation – XACML Response message format



```
<?xml version="1.0" encoding="UTF-8"?>
<AAA:AAAResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="aaa-cn1-response-00.xsd" version="0.0">
  <Result ResourceId="String">
    <Decision>Permit</Decision>
    <Status>
      <StatusCode Value="OK"/>
      <StatusMessage>Request succes7ful</StatusMessage>
    </Status>
  </Result>
</AAA:AAAResponse>
```



Mapping between CNLAuthzTicket, XACML Request/Response and SAML2.0 Authorization Assertion

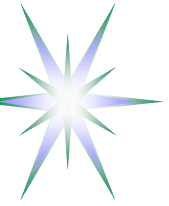


SAML 2.0 vs SAML 1.1

- Better security features
- Issuer and Subject are top level elements
- Encrypted elements for Subject, Attributes, Evidence
- Special profile for XACML

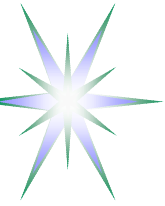
General problems for AuthZ

- Attributes can be placed only as deep as 5 level down: Assert/AzStm/Evid/AttrAsrt/Attr/AttrValue
- Ambiguous location for PolicyURIs and SessionID
- SAML1.1 ConfirmationData element is extensible type – compatibility problems



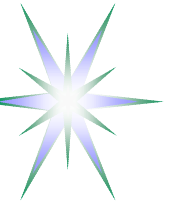
CNLAAuthzTicket example

```
<cnl:CNLAAuthzTicket xmlns:AAA="http://www.AAAarch.org/ns/AAA_BoD"
  xmlns:cnl="http://www.aaauthreach.org/ns/#CNL" Issuer="http://www.AAAarch.org/servers/AAA"
  PolicyURIs="CNLpolicy01" SessionIndex="JobXPS1-2005-001"
  TicketID="c24d2c7dba476041b7853e63689193ad">
  <!-- Mandatory elements -->
  <cnl:Decision
    ResourceID="http://resources.collaboratory.nl/Philips_XPS1">Permit</cnl:Decision>
  <cnl:Validity NotBefore="2005-02-13T01:26:42.699Z" NotOnOrAfter="2005-02-
    14T01:26:42.699Z"/>
  <!-- Additional elements -->
  <cnl:Subject Id="subject">
    <cnl:SubjectID>WHO740@users.collaboratory.nl</cnl:SubjectID>
    <cnl:SubjectConfirmationData>SeDFGVHYTY83ZXxEdsweOP8Iok</cnl:SubjectConfirmationData>
    <cnl:JobID>CNL2-XPS1-2005-02-02</cnl:JobID>
    <cnl:Role>analyst@JobID;expert@JobID</cnl:Role>
  </cnl:Subject>
  <cnl:Resource>http://resources.collaboratory.nl/Philips_XPS1</cnl:Resource>
  <cnl:Actions>
    <cnl:Action>cnl:actions:CtrlInstr</cnl:Action>
    <cnl:Action>cnl:actions:CtrlExper</cnl:Action>
  </cnl:Actions>
  <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#"> ... </ds:Signature>
</cnl:CNLAAuthzTicket>
```



CNLAAuthzTicket XML Signature element

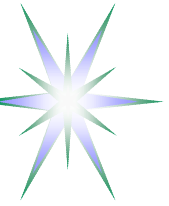
```
<ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
  <ds:SignedInfo>
    <ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
    <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
    <ds:Reference URI="">
      <ds:Transforms>
        <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
        <ds:Transform Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315#WithComments"/>
      </ds:Transforms>
      <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
      <ds:DigestValue>nrNrZZDiw/2aDnKXFEHSeoixnsc=</ds:DigestValue>
    </ds:Reference>
  </ds:SignedInfo>
  <ds:SignatureValue>
0IZt9WsJT6an+tIxhhTPtiztDpZ+iynx7K7X2Cxd2iBwCUTQ0n61Szv81DKl1Wsq75IsHfusnm56
zT3fhKU1zEUsob7p6oMLM7hb42+vjfvNeJu2roknhIDzruMrr6hMDSifaotURepu7QCT0sADm9If
X89Et55EkSE9oE9qBD8=
  </ds:SignatureValue>
</ds:Signature>
```



CNLAUTHZToken example

```
<cnl:CNLAUTHZToken TokenID="ed9d969e1262ba1d3a7f33dbd670dd94">  
<cnl:TokenValue>  
0IZt9WsJT6an+tIxhhTPtiztDpZ+iynx7K7X2Cxd2iBwCUTQ0n61Szv81DK1lWsq75IsHfusnm56  
zT3fhKU1zEUsob7p6oMLM7hb42+vjfvNeJu2roknhIDzruMrr6hMDSifaotURepu7QCT0sADm9If  
X89Et55EkSE9oE9qBD8=  
</cnl:TokenValue>  
</cnl:CNLAUTHZToken>
```

- CNLAUTHZToken is constructed of the CNLAUTHZTicket TicketID and SignatureValue
- CNLAUTHZToken use suggests caching CNLAUTHZTicket's



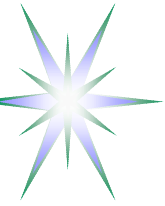
CNLSAMLAUTHzTicket example – uses SAML1.1/2.0

```
<Assertion xmlns="urn:oasis:names:tc:SAML:1.0:assertion" xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion"
  xmlns:samlp="urn:oasis:names:tc:SAML:1.0:protocol" AssertionID="c236b047d62db5cecec6b240996bcb90" IssueInstant="2005-
  02-15T14:53:23.542Z" Issuer="cnl:subject:CNLAAAauthority" Version="1.1">
<Conditions NotBefore="2005-02-15T14:53:11.745Z" NotOnOrAfter="2005-02-16T14:53:11.745Z"/>
<AuthorizationDecisionStatement Decision="Permit" Resource="http://resources.collaboratory.nl/Philips_XPS1">
  <Action Namespace="urn:oasis:names:tc:SAML:1.0:action:cnl:action">cnl:actions:CtrlInstr</Action>
  <Action Namespace="urn:oasis:names:tc:SAML:1.0:action:cnl:action">cnl:actions:CtrlExper</Action>
<Evidence>
  <Assertion AssertionID="f3a7ea74e515ffe776b10a7eef0119d7" IssueInstant="2005-02-15T14:53:23.542Z"
  Issuer="cnl:subject:CNLAAAauthority" MajorVersion="1" MinorVersion="1">
    <Conditions NotBefore="2005-02-15T14:53:11.745Z" NotOnOrAfter="2005-02-16T14:53:11.745Z"/>
    <AttributeStatement>
      <Subject>
        <NameIdentifier Format="urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress"
        NameQualifier="cnl:subject">WHO740@users.collaboratory.nl</NameIdentifier>
        <SubjectConfirmation>
          <ConfirmationData>SeDFGVHYTY83ZXxEdsweOP8Iok</ConfirmationData>
        </SubjectConfirmation>
      </Subject>
      <Attribute xmlns:typens="urn:cnl" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" AttributeName="AttributeSubject" AttributeNamespace="urn:cnl">
        <AttributeValue xsi:type="typens:cnl:job-id">CNL2-XPS1-2005-02-02</AttributeValue>
        <AttributeValue xsi:type="typens:cnl:role">analyst@JobID;expert@JobID</AttributeValue>
      </Attribute>
    </AttributeStatement>
  </Assertion>
</Evidence>
</AuthorizationDecisionStatement>
</Assertion>
```



CNLAuthnTicket example

```
<cnl:CNLAuthnTicket xmlns:AAA="http://www.AAAarch.org/ns/AAA_BoD"
  xmlns:cnl="http://www.aaauthreach.org/ns/#CNL" Issuer="http://www.AAAarch.org/servers/AAA"
  TicketID="f35585dfb51edec48de0c7eadb11c17e">
  <!-- Mandatory elements -->
  <cnl:Validity NotBefore="2005-02-15T14:33:10.548Z" NotOnOrAfter="2005-02-16T14:33:10.548Z"/>
  <cnl:Subject Id="subject">
    <cnl:SubjectID>WHO740@users.collaboratory.nl</cnl:SubjectID>
    <cnl:SubjectConfirmationData>
      0+qQNAVuZW4txMi8DH6DFy7eLMGxSfKDJY6ZnY4UW5Dt0JFtat1EprUtgnjCkzrJUMvWk9qtUzna
      sDdUG+P4ZY7dgab+PHiU91ClusZbztu/ZIjNqCnw5sulBQLTumC8ZTtYKKJi4Wws+bMMbP8mFNQm
      +M7F4bJIPBfLcxf0bk4=
    </cnl:SubjectConfirmationData>
    <!--Optional elements -->
    <cnl:SubjectAttribute attrname="urn:cnl:subject:attribute:job-id">
      CNL2-XPS1-2005-02-02
    </cnl:SubjectAttribute>
    <cnl:SubjectAttribute attrname="urn:cnl:subject:attribute:role">
      analyst@JobID;expert@JobID
    </cnl:SubjectAttribute>
  </cnl:Subject>
</cnl:CNLAuthnTicket>
```

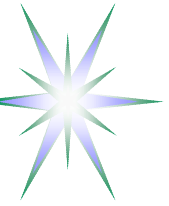


CNLAUTHNToken signed/encrypted – 401/269 bytes

```
<cnl:CNLAUTHNToken xmlns:cnl="http://www.aaauthreach.org/ns/#CNL"
  TokenID="f35585dfb51edec48de0c7eadb11c17e">
  <cnl:SubjectID>WHO740@users.collaboratory.nl</cnl:SubjectID>
  <cnl:TokenValue>
    0+qQNAVuZW4txMi8DH6DFy7eLMGxSfKDJY6ZnY4UW5Dt0JFtat1EprUtgnjCkzrJUMvWk9qtUzna
    sDdUG+P4ZY7dgab+PHiU91ClusZbztu/ZIjNqCnw5su1BQLTumC8ZTtYKKJi4Wws+bMMbP8mFNQm
    +M7F4bJIPBfLcxf0bk4=</cnl:TokenValue>
</cnl:CNLAUTHNToken>
```

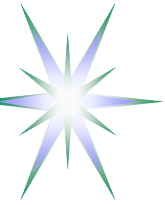
- CNLAUTHNToken is constructed of the CNLAUTHNTicket TicketID and SubjectConfirmationData which is encrypted SubjectID value
- CNLAUTHZToken must be self-sufficient and doesn't require caching CNLAUTHNTicket's

```
<cnl:CNLAUTHNToken xmlns:cnl="http://www.aaauthreach.org/ns/#CNL"
  TokenID="a392a20157698d201d77b2c6e8e444ef">
  <cnl:SubjectID>WHO740@users.collaboratory.nl</cnl:SubjectID>
  <cnl:TokenValue>qij9zJgKZp9RiJxYN1QJAN0vhjLJSMGVLD/doQtmCsk=</cnl:TokenValue>
</cnl:CNLAUTHNToken>
```

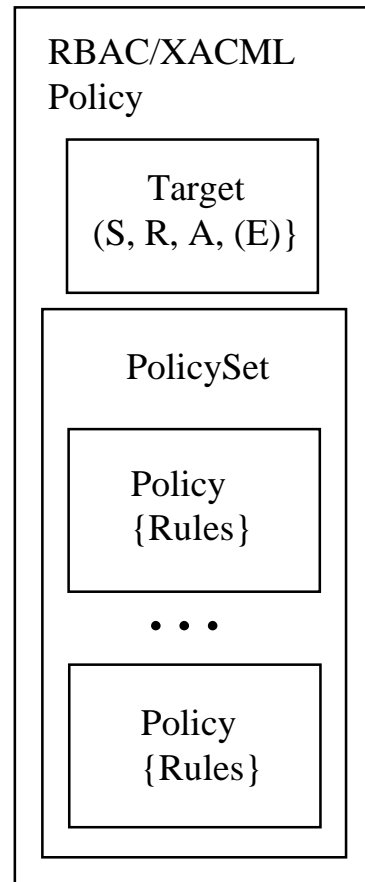
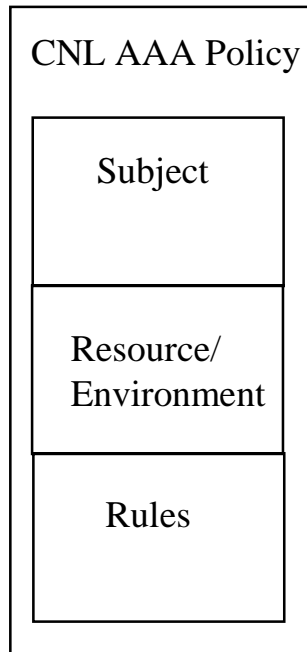


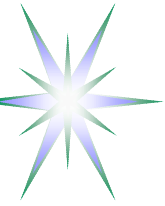
Integrating with existing Access Control and other tools

- Policy mapping between XACML, AAA Policy Language and other formats
- GT4 Authorization Framework
- EGEE gLite Authorisation Framework



AAA Policy and RBAC/XACML Policy



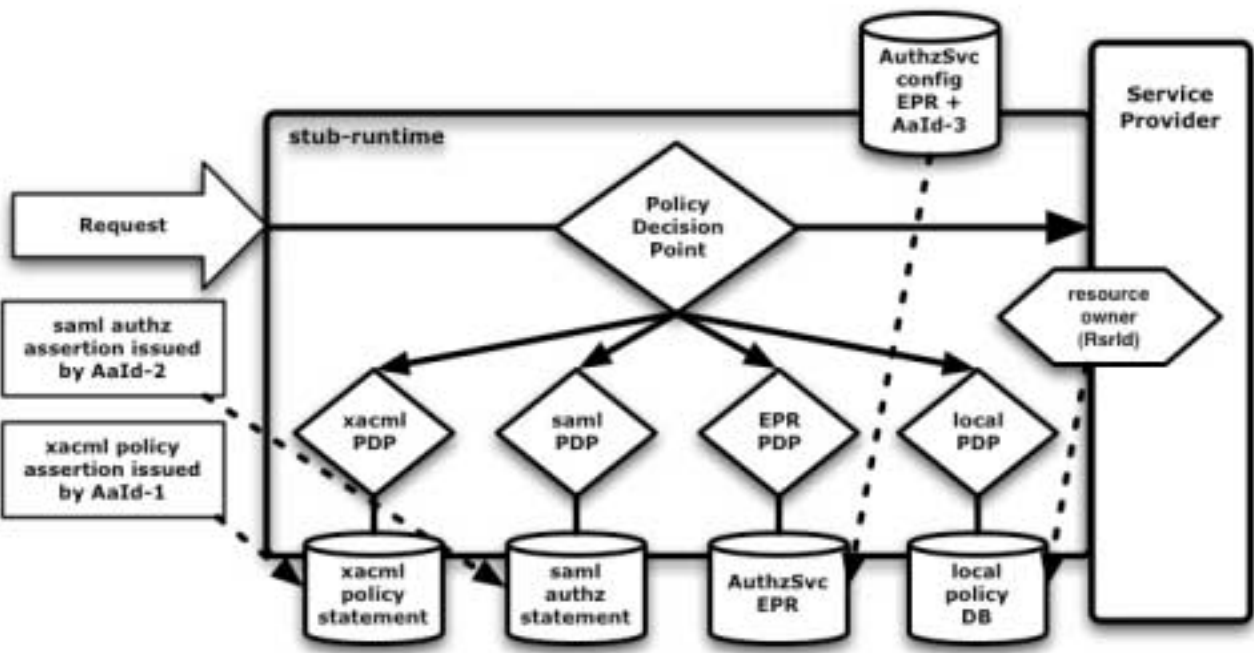


GT4 AuthZ framework: Implementation details

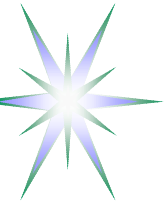
- Source code tree
 - org.globus.wsrp.impl.security.authorization
- Still using grid-map file as a major option
- Special interface for PDP and PIP to interact with Interceptor
- Very simple example provisioning for XACML
 - ◆ Simple policy format



GT4 AuthZ framework: Multiple configured PDPs



- GT4 implementation uses Interceptor concept
- Originated from POSIX AuthZ f/w
 - Supported by Axis Handlers
 - PEP function is (virtually) eliminated
 - “Deny-override” vs “Permit-override” combination
 - Configured by Interceptor PDP/PIP call-out list
 - PDP are called directly or via PIP



Acknowledgements

This work results from the Collaboratory.nl project, a research initiative that explores the possibilities of remote control and use of advanced lab facilities in a distributed and collaborative industrial research setting. The Collaboratory.nl consortium consists of DSM, Philips, Corus, FEI, Telematica Instituut and the University of Amsterdam.

This work also contributes to further development of the Generic AAA Authorisation framework by the Advanced Internet Research Group at the University of Amsterdam.