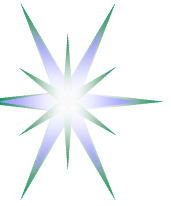


# **Security Context Management with XML based security tickets and tokens**

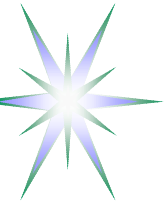
Yuri Demchenko <demch@science.uva.nl>  
AIRG, University of Amsterdam



# Outline

---

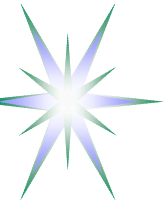
- Fine grained access control with Generic AAA Authorisation framework and RBAC
  - ◆ Combined push-pull and agent-push models using AuthZ tickets and tokens
- GAAAPI implementation detail and ticket/token examples
  - ◆ Collaboratory.nl Authorisation service



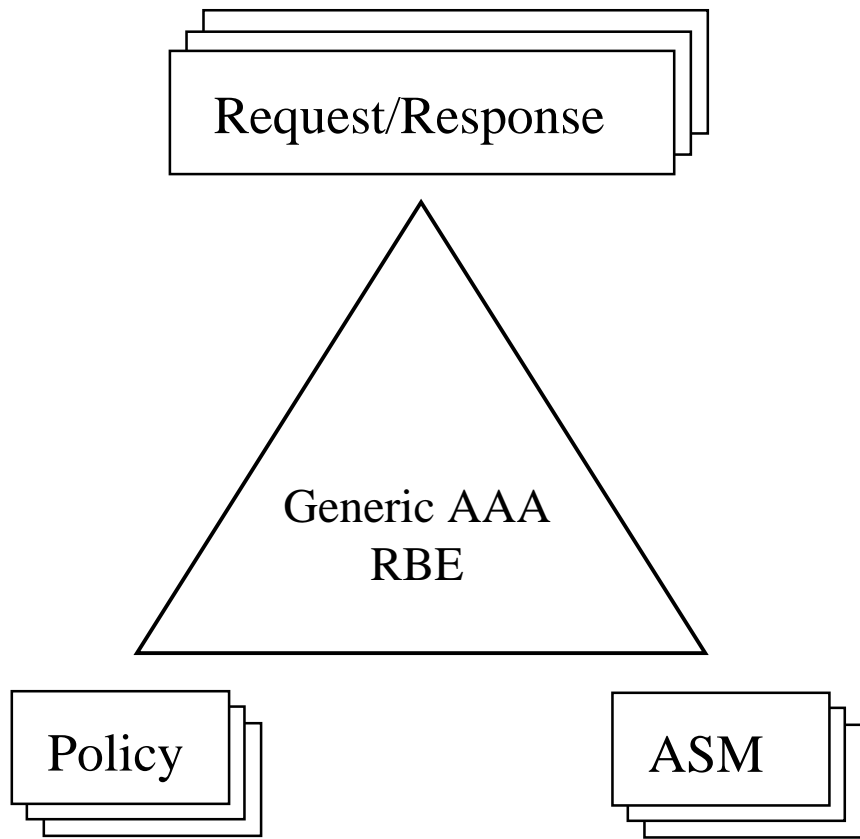
# Requirements to Policy based Access Control

---

- Multidomain and inter-institutional
- Multiple policy formats
- Policy combination
- Multiple policy authority
- Separate policy management
- Dynamic policy association



# (1) Generic AAA Architecture by AIRG (UvA)



## Policy based Authorization decision

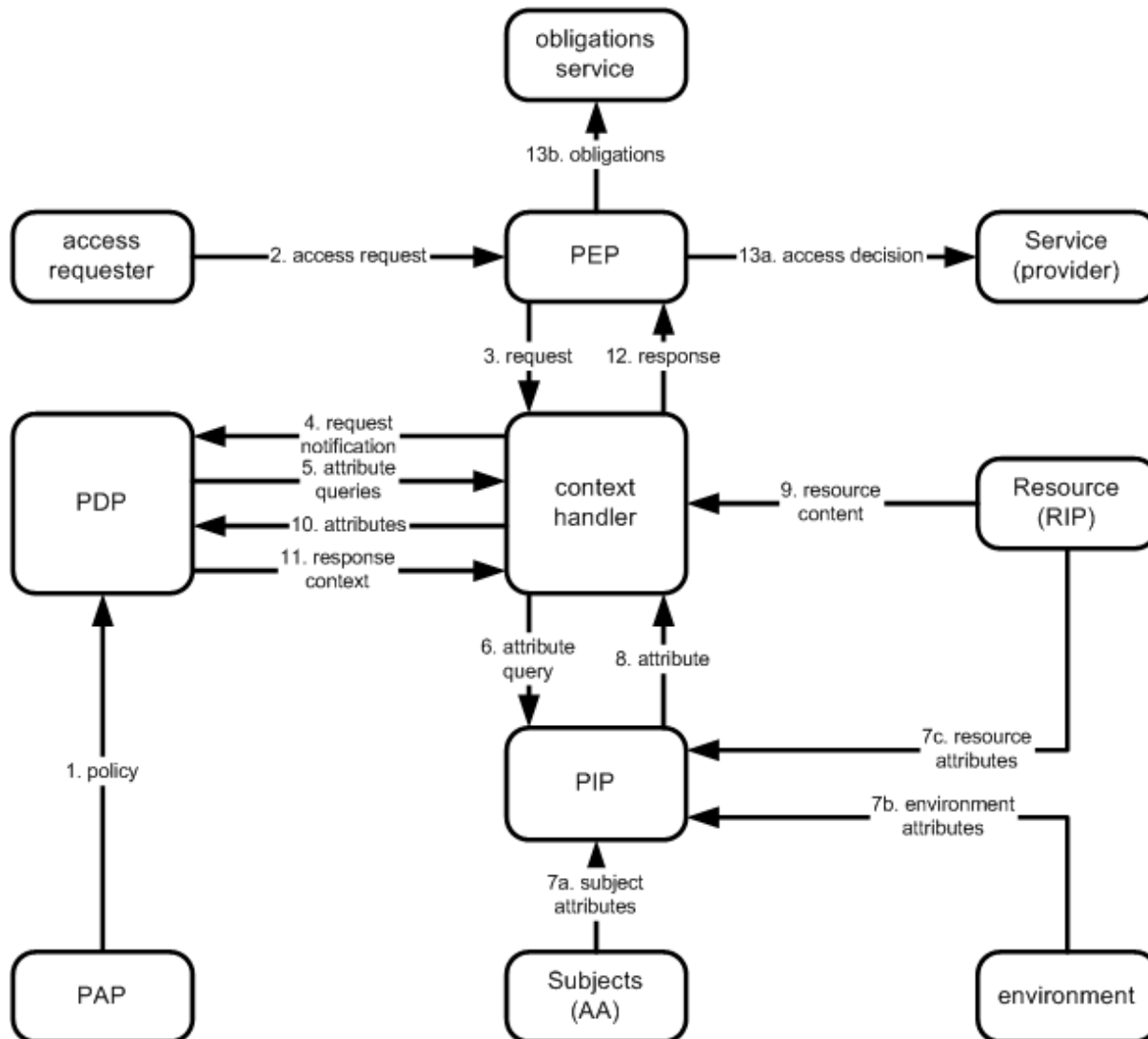
- Req {AuthNtoken, Attr/Roles, PolicyTypeId, ConditionExt}
- RBE (Req + Policy) => => Decision {ResponseAAA, ActionExt}
- ActionExt = {ReqAAExt, ASMcontrol}
- ResponseAAA = {AckAAA/RejectAAA, ReqAttr, ReqAuthN, BindAAA (Resource, Id/Attr)}

•Defined by Resource owner

- Translate logDecision => Action
- Translate State => LogCondition



# RBAC: main components and dataflow – XACML model



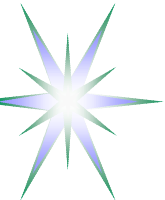
PEP/AEF - Policy Enforcement Point (authorisation enforcement function)

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

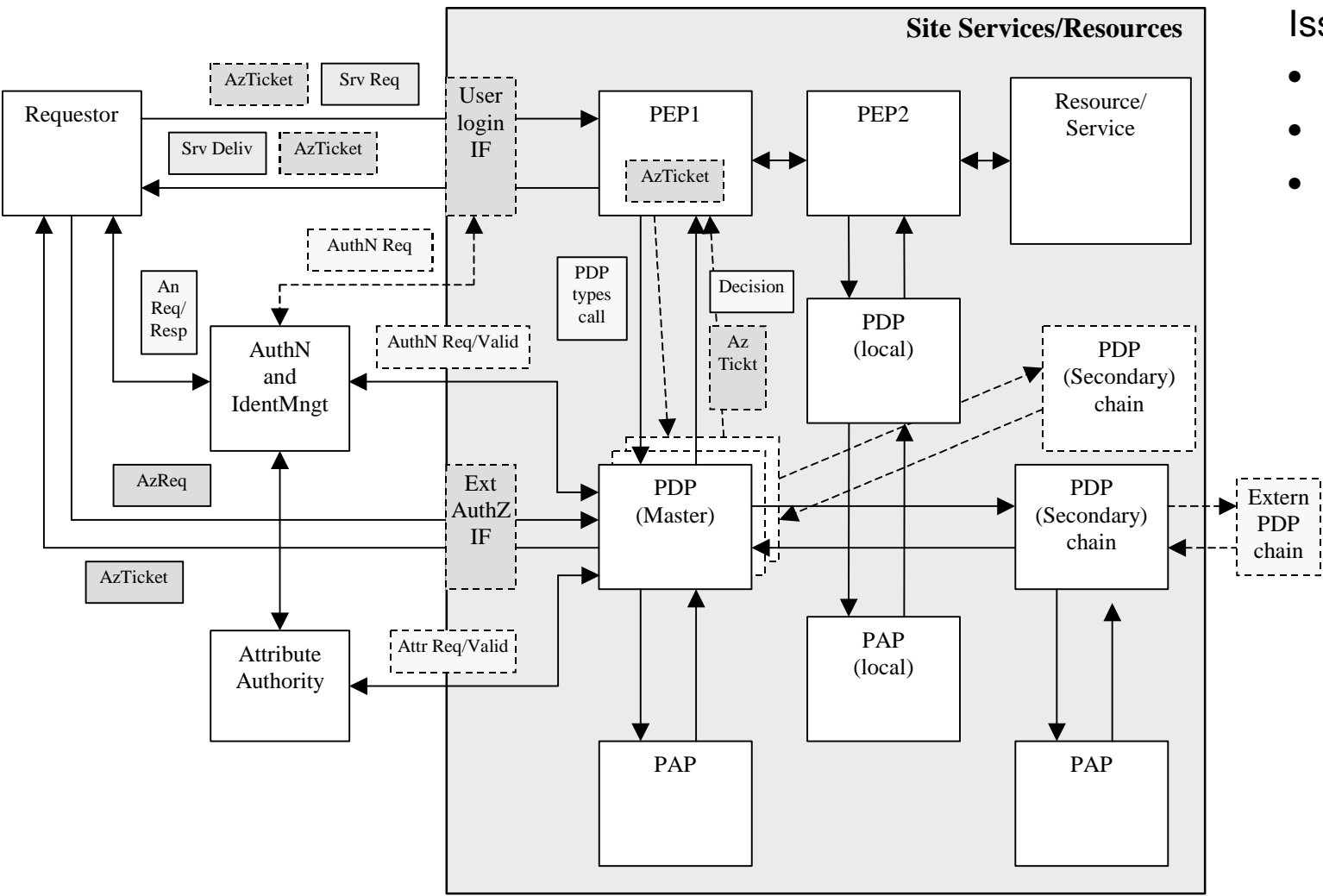
PIP - Policy Information Point

AA - Attribute Authority

PAP - Policy Authority Point

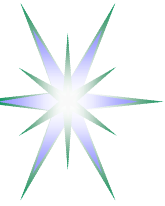


# Site AuthZ service implementing RBAC and combined pull-push model

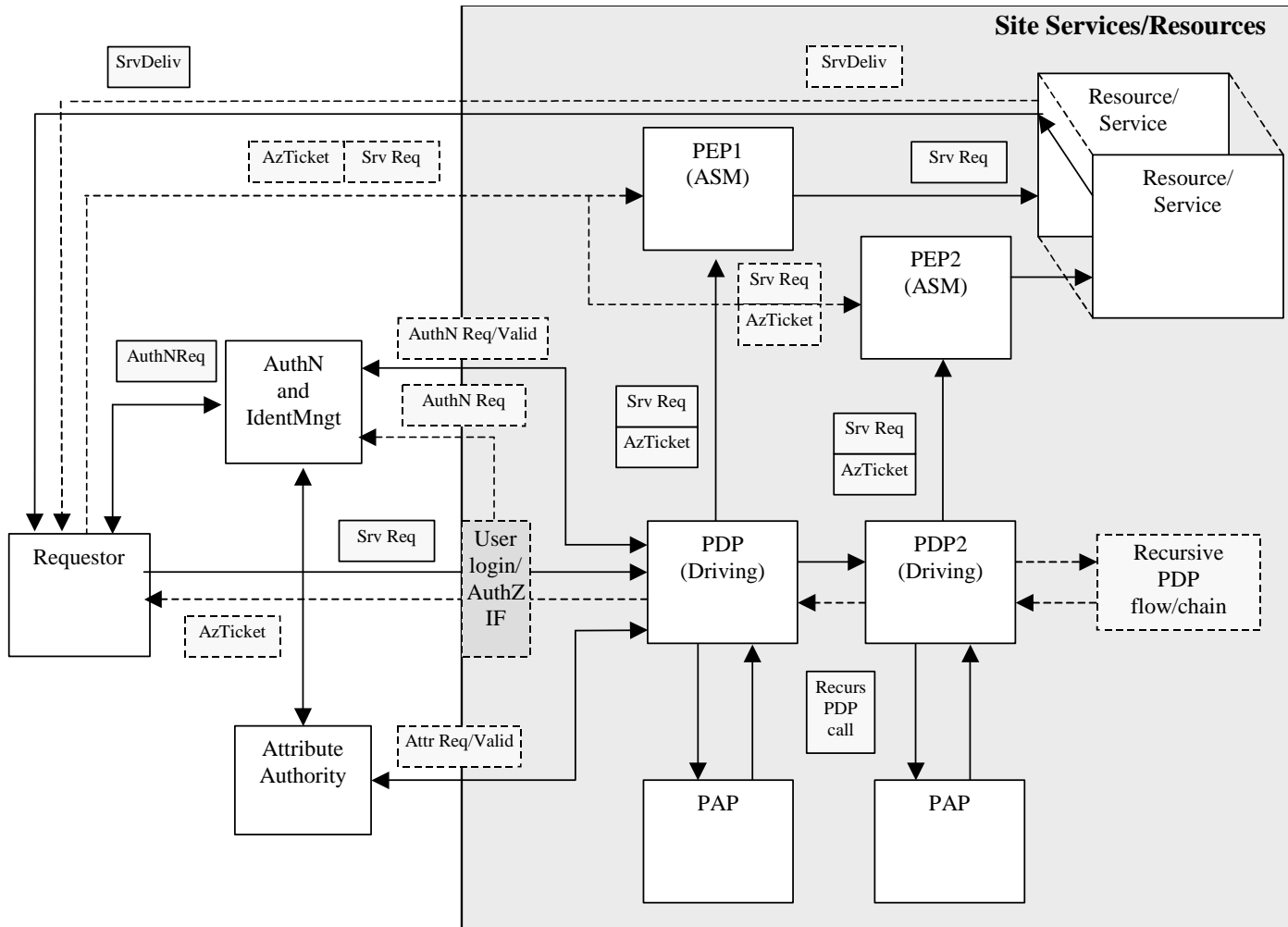


Issues to be addressed:

- PEP and PDP chaining
- Policy combining
- Multiple domains

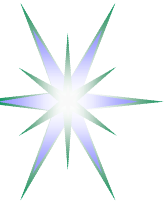


# Site AuthZ service implementing combined agent and push model for complex resource



Issues to be addressed:

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

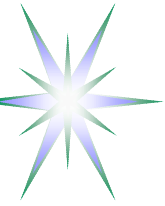


# Policy based AC: Implementation suggestions

---

- 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.

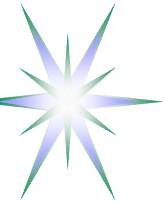




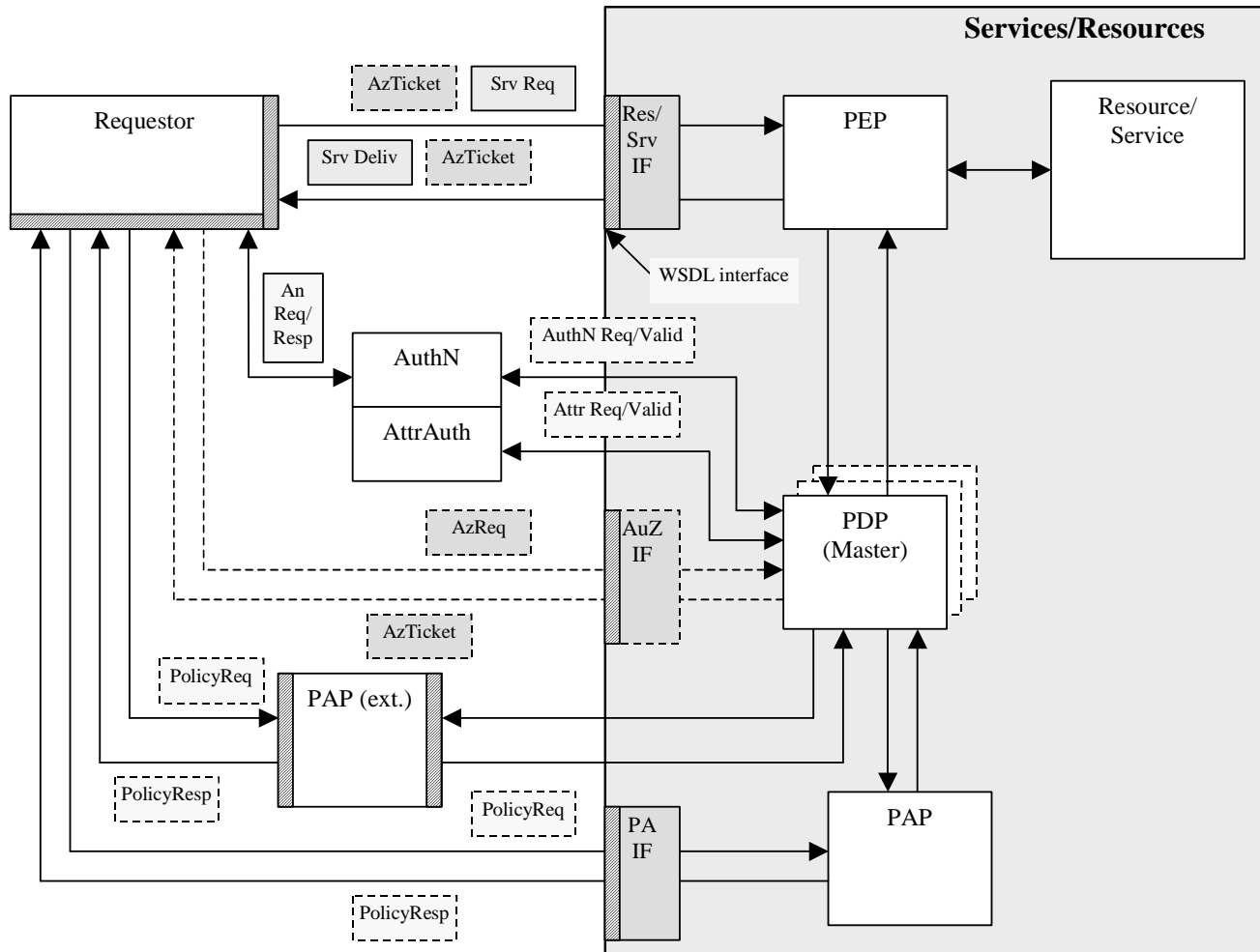
## Traditional Access Control model – setting up trust and authority relations

---

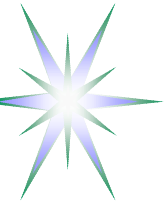
- Policy, attributes semantics and namespaces are known a priori to all participating parties
  - A requestor knows what information to present to adhere to a specific policy and in what format
- PEP and PDP locations are known and interacting parties are known
- Trust relations between PDP, AA and resource 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 emerging 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

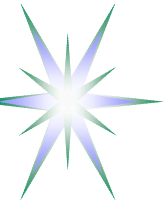


# Attaching policy to WSDL - Example

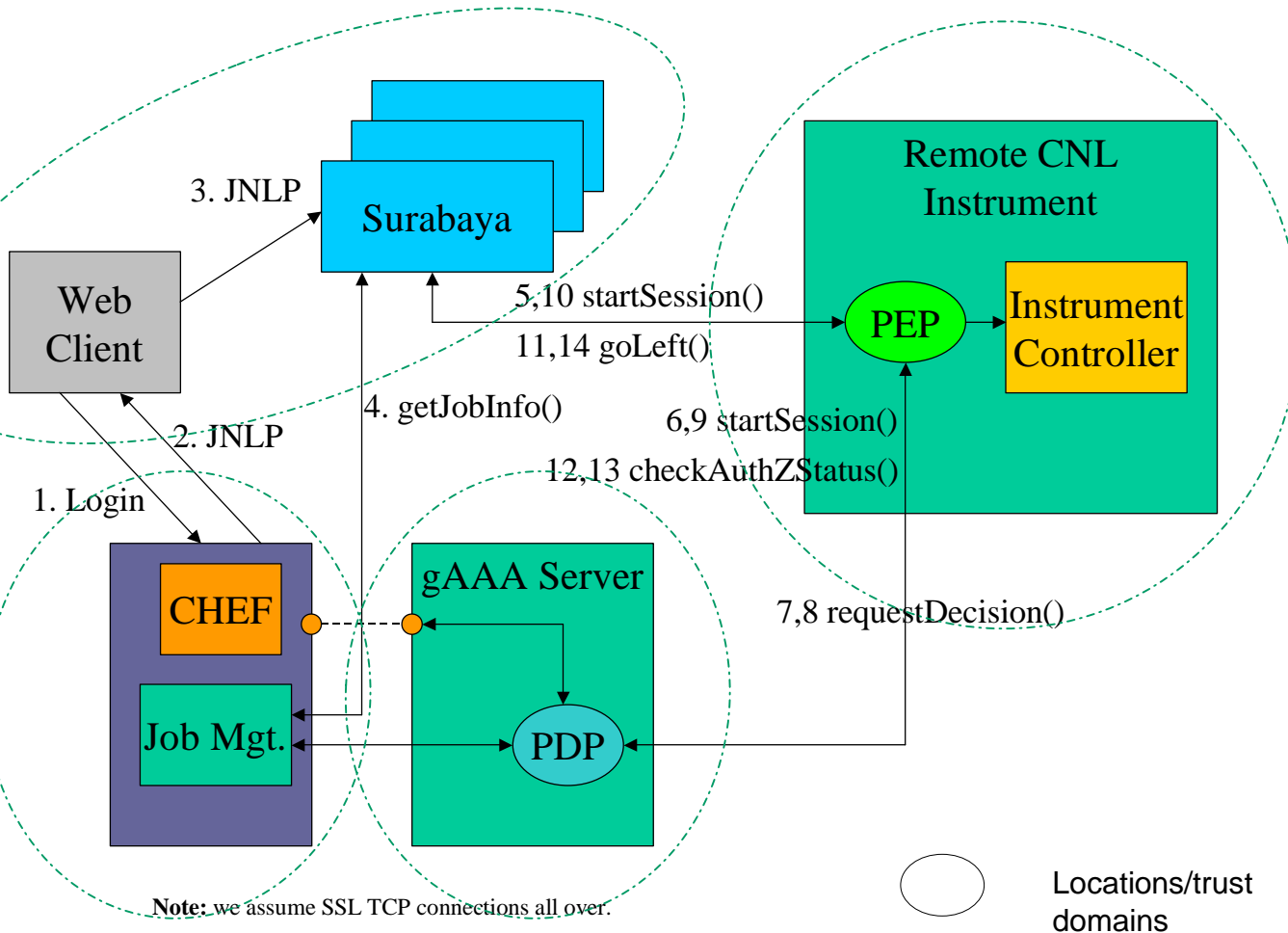
```
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/"
  < ... snip long namespace declaration ... >
  xmlns:wsp="http://schemas.xmlsoap.org/ws/2002/12/policy"
  xmlns:cnl="http://cnl.telin.nl/cnl" xmlns:policy="cnl-policy-schema.xsd"
  targetNamespace="http://cnl.telin.nl/cnl">
  <message name="ViewExperimentRequest" 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>

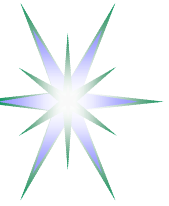
  <wsp:UsingPolicy wsdl:Required="true"/>
</definitions>
```



# Implementation: Authorisation Service operation in a CNL2 Demo system



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

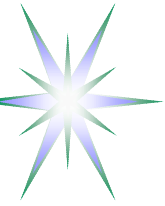


# Before deploying security infrastructure

---

## Design conventions and agreements

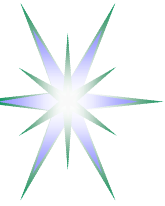
- Key distribution and trust establishing
  - ◆ *Currently, in search of simple consistent model*
- Policy definition and format including subject, attributes/roles, actions semantics and namespaces
  - ◆ Compatibility with existing formats, e.g. SAML, XACML
  - ◆ Policy format defines/defined by the PDP implementation
- Secure credentials/ticket format
  - ◆ Standard vs proprietary
- Protocols and Messages format
  - ◆ SOAP + XACML Request/Response
  - ◆ SOAP + SAML + XACML



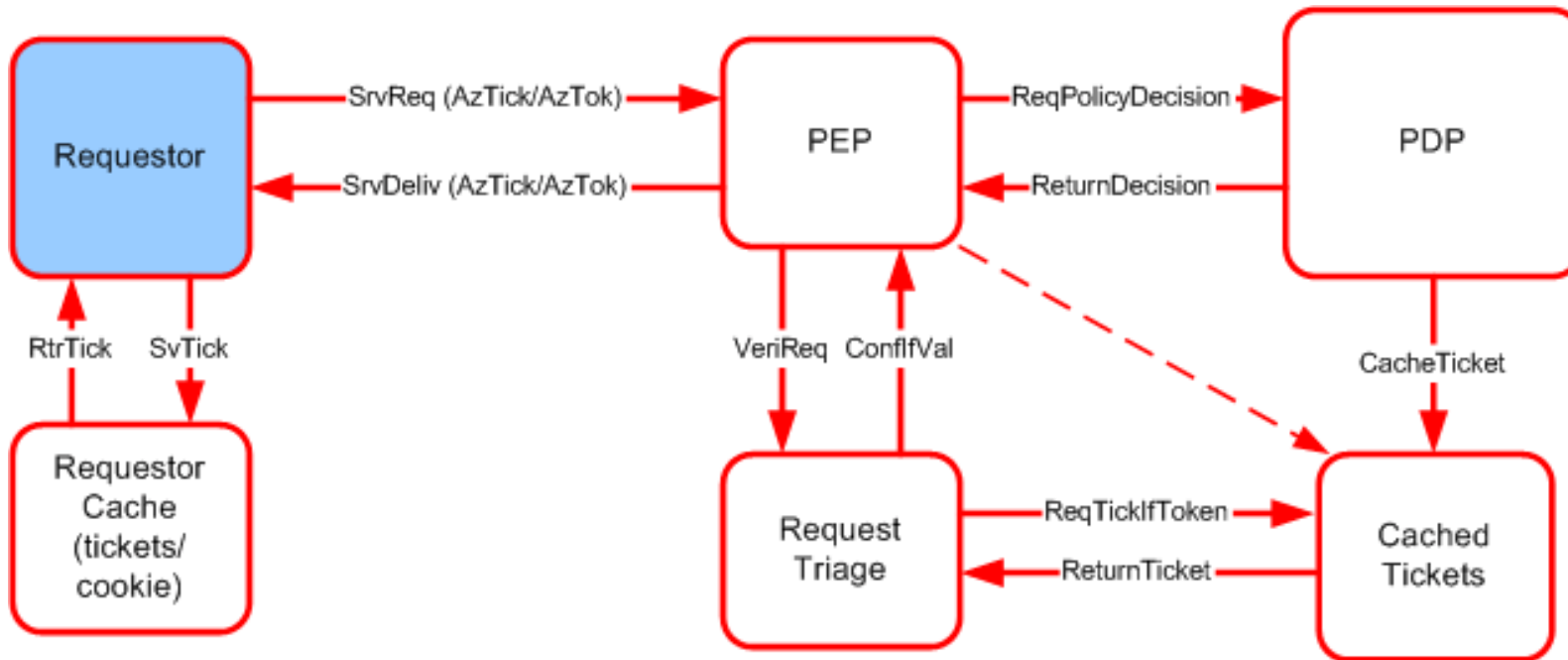
# Session management in CNL2 AuthZ system

---

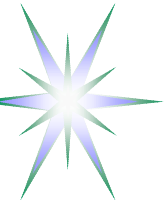
- Maintaining session is a part of generic RBAC functionality
- Session can be started only by authorised Subject/Role
  - ◆ Session can be joined by other less privileged users
- SessionID is included into AuthzTicket together with other decision attributes
  - ◆ Signed AuthzTicket is cached by PEP or PDP
- If session is terminated, cached AuthzTicket is deleted
  - ◆ Note: AuthzTicket revocation should be done globally for the AuthZ trust domain



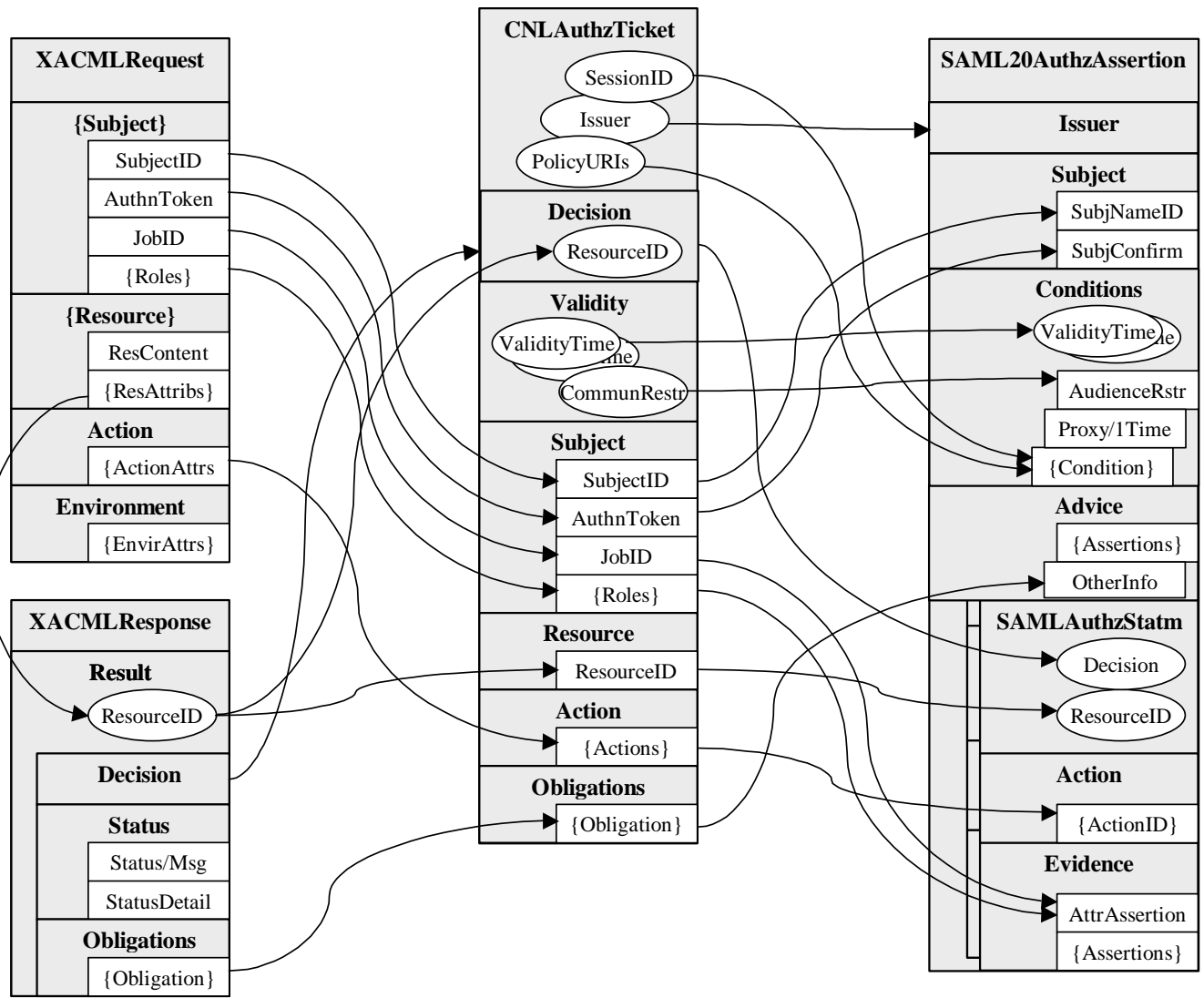
# Tickets/Tokens handling in AuthZ system



- 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



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



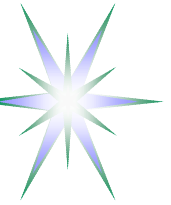
## SAML 2.0 vs SAML 1.1

- 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





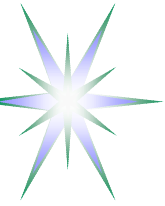
# Using SAML 1.1/2.0 for AuthzTicket expression

## **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 XACMLAuthzStatement

## **General problems for Authorisation assertion**

- Attributes can be placed only as deep as 5 level down:  
Assertion/AuthzStatement/Evidence/AttributeAssertion/Attribute/AttributeValue
- Ambiguous location for PolicyURIs and SessionID
- Ambiguous mapping for XACML/Obligation to SAML/(Condition or Advice)
- SAML1.1 ConfirmationData element is an extensible type – compatibility problems
- XACML Obligation element
  - ◆ Can be mapped to SAML Condition element or SAML Advice element



# CNLAAuthzTicket example – 1011 bytes

```
<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>
```

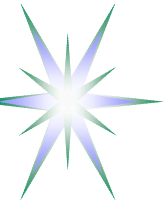


# CNLAAuthzTicket XML Signature element – 957 bytes (total signed ticket 1968 bytes)

```
<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+iynx7K7X2Cxd2iBwCUTQ0n61Szv81DKllWsq75IsHfusnm56
zT3fhKU1zEUsob7p6oMLM7hb42+vjfvNeJu2roknhIDzruMrr6hMDsIfaotURepu7QCT0sADm9If
X89Et55EkSE9oE9qBD8=
  </ds:SignatureValue>

  <ds:KeyInfo> << ... snip ... >> </ds:KeyInfo>

</ds:Signature>
```



# RSA <ds:KeyInfo> element – 1010 bytes (total signed ticket with KeyInfo - 3078 bytes)

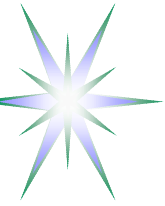
```
<ds:KeyInfo>
  <ds:X509Data>
    <ds:X509Certificate>
      MIIICADCCAWkCBEGX/FYwDQYJKoZIhvcNAQEEBQAwwRzELMAkGA1UEBhMCTkwGTAXBgNVBAoTEENv
      bGxhYm9yYXRvcnkubmwxHTAbBgNVBAMTFEFBQXV0aHJlYWNoIFNlY3VyaXR5MB4XDTA0MTEwNTAw
      NDYxNFoXDTA1MDIxMzAwNDYxNFowRzELMAkGA1UEBhMCTkwGTAXBgNVBAoTEENvbGxhYm9yYXRv
      cnkubmwxHTAbBgNVBAMTFEFBQXV0aHJlYWNoIFNlY3VyaXR5MIGfMA0GCSqGSIb3DQEBAQUAA4GN
      ADCBiQKBgQDdDrBhVmr1nD9eqi7U7m4yjiRxfvjAKv33EpuajvTKHpKUGLjbcBC3jNJ4F7a0GiXQ
      cVbuF/aDx/ydIUJXQktvFxK0Sm77WVeSel0cLclhYfUSAg4mudtfsB7rAj+CzNnVdr6RLFps9YFE
      lv5ptGaNGSbwHjU02HnArEGL2K+0AwIDAQABMA0GCSqGSIb3DQEBAUUA4GBADHKqkOW4mP9DvOi
      bMvf4oqXTth7yv8o3Zol7+nqlB9Tqf/bVNLMk8vNo5fWRHbpnHIFFGTk3lnrJf8kEZEofvwAeW9s
      lgQtYfsloxvsMPKHxFjJDiZlLkHRViJl/slz5a7pkLqIXLRsPFRziTksemRXB/ft8KDzMl4pzQZg
      HicO
    </ds:X509Certificate>
  </ds:X509Data>
  <ds:KeyValue>
    <ds:RSAKeyValue>
      <ds:Modulus>
        3Q6wYVZq9Zw/Xqoul05uMoyEcX74wCr99xKbmo70yh6S1IC423AQt4zSeBe2tBol0HFW7hf2g8f8
        nSFCV0JLbxcStEpu+1lXknpdHC3NYWH1EgIOJrnxB7Ae6wI/gszZ1Xa+kSxaUvWBRJb+abRmjRkm
        8B41NNh5wKxBi9ivtAM=
      </ds:Modulus>
      <ds:Exponent>AQAB</ds:Exponent>
    </ds:RSAKeyValue>
  </ds:KeyValue>
</ds:KeyInfo>
```



# CNLAUTHZToken example – 293 bytes

```
<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

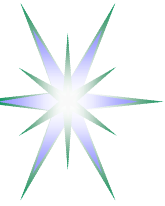


# CNLSAMLAAuthzTicket example – 2254 bytes

```
<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="2005-02-16T14:32:12.506Z" NotOnOrAfter="2005-02-17T14:32:12.506Z">
    <Condition xsi:type="typens:cnl:session-id">JobXPS1-2005-001</Condition>
    <Condition xsi:type="typens:cnl:policy-uri">CNLpolicy01</Condition>
  </Conditions>
  <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>
              <ConfirmationMethod>signed-subject-id</ConfirmationMethod>
              <ConfirmationData>
                PBLIR0aZRtdZmq9791j8eDpJ5VT6BxxWBtSApC5BPnIsfHRUcOOpWQowXBw2TmOzdJGNzFWhMinz
                XU3/wSdLjv+siO2JGfyZ7U9eqkM0GqY8VizM15uRuUAsrr7AIHv9/DPlksJMNDZ5DnGosMc+ZyqN
                KogfMqhK+DKqPwfHF6U=</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</AttributeValue>
            <AttributeValue xsi:type="typens:cnl:role">analyst@JobID;expert@JobID</AttributeValue>
          </Attribute>
        </AttributeStatement>
      </Evidence>
    </AuthorizationDecisionStatement>
  </Assertion>
```

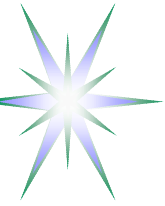
===> moved to attr in SAML 2.0

===> level 5 of element



# CNLAUTHNTicket example – 1752 bytes

```
<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+qQNAVuZW4txMi8DH6DFy7eLMGxSfKDJY6ZnY4UW5Dt0JFtatlEprUtgnjCkzrJUMvWk9qtUzna
      sDdUG+P4ZY7dgab+PHiU9lClusZbztu/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>
```



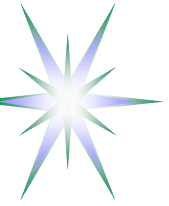
# CNLAUTHNTOKEN signed/encrypted – 401/269 bytes

```
<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>
```





# Conclusion and Discussion

---

- Tickets and tokens handling infrastructure can be used for general security context handling, including
  - ◆ Combining pull and push models
  - ◆ Simple mechanisms for push-policy operation in open environment
  - ◆ SSO and Identity management
- Further development (by AIRG) will include dynamic trust and virtual associations management based on Job/VO agreement