XACML and SAML support in GAAAPI for CNL2 Authorisation Service

Yuri Demchenko <demch@science.uva.nl>
Advanced Internet Research Group
University of Amsterdam
Outline

- Security requirements to Open Collaborative Environment (OCE)
- Job-centric security model for OCE Security Architecture
- Using Generic AAA (GAAA) Authorisation framework and Role Based Access Control (RBAC) for fine grained access control
  - Optimised push-pull-agent model using AuthZ tickets and tokens
- Trust relations in distributed access control infrastructure
- Implementation details – GAAAPI and Collaboratory.nl project
- Summary - Used technologies and new developments
- Additional materials (technical)
OCE specific security requirements and common problems

- Open Collaborative Environment specific security requirements
  - Dynamic and multidomain
  - Customer driven
  - Human controlled and interactive
  - Data protection: personal, experimental data and metadata

- Common problems addressed
  - Authorisation service performance
    - Using XML based ticket/token – integrity and secure context management
  - Session management in RBAC Authorisation
  - Key management and trust relations in distributed access control infrastructure
  - Compatibility and integration with existing access control tools
    - Policy formats mapping for flexible policy exchange and combination
OCE/ CNL 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) through 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
Site Authorisation service implementing RBAC and combined pull-push model
Implementation suggestions for OCE/CNL

- 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 priori
- 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.
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
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)
Trust relations in distributed access control infrastructure

Obtaining required permissions to perform requested action by the user:

\[
\text{User} \rightarrow \text{AuthN(HomeOrg.staff(TA2), Job.members)} \\
\rightarrow \text{AuthZ(Member.roles, Policy.permissions)} \\
\rightarrow \text{Resource.permissions}
\]
Issues in using XACML and SAML for Authorization

XACML issues/problems
- No mechanisms for authenticity and integrity
- No communication protocol specified
- No AuthZ session management
- Policy doesn’t have Subject/Attribute (cryptographic) validation function

SAML issues/problems
- No direct mapping from XACML Authz decision to SAML AuthzStatement
- Full AuthZ Assertion is not elegant

Common SAML and XACML issues
- Complex in implementation
- Require separate key/trust management support
- Require application/community specific attribute namespace definition
Implementation: Authorisation Service operation in a CNL2 Demo system

1. Login
2. JNLP
3. JNLP
4. getJobInfo()
5,10 startSession()
6,9 startSession()
7,8 requestDecision()
11,14 goLeft()
12,13 checkAuthZStatus()

Note: we assume SSL TCP connections all over.

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

Locations/trust domains
GAAAPI Trust Domains Configuration

Options for trust domains configuration depend on possible PEP and PDP location:

- PEP is protecting Resource, and therefore should be located in the Resource trust domain
- PDP may be remote, in this case communication between PEP and PDP must be protected cryptographically

Trust domain identifiers:

- TRUSTDOMAIN_PEP = "urn:cnl:trust:pep";
- TRUSTDOMAIN_PDP = "urn:cnl:trust:pdp";
- TRUSTDOMAIN_PEP_PDP = "urn:cnl:trust:pep-pdp";

Authorities identifiers:

- TICKETAUTHORITY_PEP = "urn:cnl:trust:tickauth:pep";
- TICKETAUTHORITY_PDP = "urn:cnl:trust:tickauth:pdp";

Note: Current implementation is in class ConfigTrustDomains for debugging/demo purposes
Security Configuration Parameters

Key store location and access

- keystoreType = "JKS"
- keystoreFile = LOCAL_DIR_KEYSTORE + "keystore5cnlsec.jks"
- keystorePass = "********"
- trustedstoreFile = LOCAL_DIR_KEYSTORE_TRUSTED + "keystore5cnltrusted.jks"
- trustedstorePass = "******"

Trusted and local keys/credentials for PEP trust domain (Certs are selfsigned)

- pepprivKalias = "cnl_pep"
- peppubKalias = "cnl_pep"
- pepprivKpass = "Trust:pep"
- pdppubKalias = "cnl_aaapdp"

Trusted sites or authorities

- trustedAuth = "cnl-trust.xml" // similar to and to be compatible with Shibboleth

AuthzTicket authority

- tickauth = (tickauthPDP | tickauthPEP)
GAAAPI security related directories configuration

Configuration directories

LOCAL_DIR_ROOT = ""
LOCAL_DIR_KEYSTORE_CNLSEC = LOCAL_DIR_ROOT + "data/keystore/cnlsec/"
LOCAL_DIR_KEYSTORE_TRUSTED = LOCAL_DIR_ROOT + "data/keystore/trusted/"
LOCAL_DIR_SYMKEYSTORE = LOCAL_DIR_ROOT + "data/keystore/cnlsec/symkeystore/"
LOCAL_DIR_SCHEMAS = LOCAL_DIR_ROOT + "data/schemas/"

Temporal directory and cache

LOCAL_DIR_AAADATA_CACHE_AZTICKETS = LOCAL_DIR_ROOT + "_aaadata/cache/aztickets/"
LOCAL_DIR_AAADATA_TMP = LOCAL_DIR_ROOT + "_aaadata/tmp/"

Note. Currently configured in ConfigCNLSecurity class
CNL2 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/Phillips_XPS1”
- Subject – “WHO740@users.collaboratory.nl”
- Roles - “role“ or “role@JobID”
AAA Policy and XACML Policy formats

- CNL AAA Policy
  - Subject
  - Resource/Environment
  - Rules

- RBAC/XACML Policy
  - Target
    - {S, R, A, (E)}
  - PolicySet
    - Policy
      - {Rules}
    - ...

- XACML Policy
  - Rule Combination Algorithm
  - Policy Target
    - {S, R, A, (E)}
  - Rule ID#1
    - Rule Target
      - {S, R, A}
    - Condition
      - AttrDesignat
      - Match List
  - Rule ID#n
Policy generation conventions

- Subject validation
- Resource and Environment checking
- Access rules evaluation
  - Rules are expressed as permissions to perform an action against Subject role
CNL2 AuthZ policy: RBAC using XACML format

Policy generation conventions

- Policy Target is defined for the Resource and may include Environment checking
- Policy combination algorithm is “ordered-deny-override” or “deny-override”
- Rule Target is defined for the Action
  - 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
  - Rules effect is “Permit”
- Subject validation – is not supported by current XACML functionality
  - TODO: add Function or do validation at/by PEP or Context Handler
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 SAML Authorization Assertion

**XACMLRequest**
- **(Subject)**
  - SubjectID
  - AuthnToken
  - JobID
  - [Roles]
- **{Resource}**
  - ResContent
  - [ResAttrs]
- **Action**
  - [ActionAttrs]
- **Environment**
  - [EnvirAttrs]

**XACMLResponse**
- **Result**
  - ResourceID
- **Decision**
  - Status
    - Status/Msg
    - StatusDetail
  - Obligations
    - [Obligation]

**CNLAuthzTicket**
- SessionID
- Issuer
- PolicyURIs

**Decision**
- ResourceID

**Validity**
- ValidityTime
- [communRest]

**Subject**
- SubjectID
- AuthnToken
- JobID
- [Roles]

**Resource**
- ResourceID

**Action**
- [Actions]

**Obligations**
- [Obligation]

**SAML20AuthzAssertion**
- Issuer
- Subject
  - SubjNameID
  - SubjConfirm
- Conditions
  - ValidityTime
  - [AudienceRstr]
  - Proxy/ITime
  - [Condition]
- Advice
  - [Assertions]
  - OtherInfo
  - [Assertions]

**SAML11AuthzStat**
- Action
  - [ActionID]

**SAMLAuthzStat**
- Action
  - ResourceID

**SAMLAuthzAssertion**
- Evidence
  - AttrAssertion
  - [Assertions]
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
<cnl:CNLAuthzTicket xmlns:AAA="http://www.AAAarch.org/ns/AAA_BoD"
xmlns:cnl="http://www.aaauthrearch.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#" ...
</cnl:CNLAuthzTicket>
CNLAuthzToken example – 293 bytes

<cnl:CNLAuthzToken TokenID="ed9d969e1262ba1d3a7f33dbd670dd94">
  <cnl:TokenValue>
    OIZt9WsJT6an+tIxhhTPtiztDpZ+iynx7K7X2Cxd2iBwCUTQ0n61Szv81DKl1Wsq75IsHfusnm56zT3fhKUlzEUsob7p6oMLM7hb42+vjfvNeJu2roknhIDzruMrr6hMDsIfaotURepu7QCT0sADm9IfX89Et55EkSE9oE9qBD8=
  </cnl:TokenValue>
</cnl:CNLAuthzToken>

- CNLAuthzToken is constructed of the CNLAuthzTicket TicketID and SignatureValue
- CNLAuthzToken use suggests caching CNLAuthzTicket
Summary - Used technologies and new developments

- Job-centric security model that responds OCE dynamic distributed requirements
  - Job description format – to be compatible with WS-Agreement and GGF JSDL (Job Submission Description Language)
- Trust model for distributed access control system
- Extended RBAC functionality based on GAAA Authorisation framework
  - XACML Request/Response messaging
  - Current policy expression format is AAA and migration to XACML based policy exchange and combination
- GAAA Authorisation performance optimisation using tickets/tokens
  - Proprietary and SAML based AuthzTicket format
  - AuthZ/Resource Session management
- XML Signature and XML Encryption for JobDescription and AuthzTicket security
Summary - Future development

- Common policy expression and exchange format based on XACML
- GAAAPI/GAAA_tk profile for multidomain AuthZ and pushing policy
- Integrating with existing Access Control and other tools
  - EGEE gLite Authorisation Framework - http://hepunx.rl.ac.uk/egee/jra1-uk/glite-r1/
- Binding Policy to WSDL service description
  - Using WS-Security Framework and OGSA/WSRF
- Adding VO and VOMS functionality - for user and resource attributes management
- AuthN and Identity management
- More information
  - GAAAPI - http://staff.science.uva.nl/~demch/projects/aaauthreach/
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 is a part of ongoing research and development of the Generic AAA Authorisation framework by the Advanced Internet Research Group at the University of Amsterdam.
Additional information

- Open policy enforcement model
  - binding policy to WSDL with WS-PolicyAttachment
- Generic AAA Architecture and RBAC model
- XACML AuthZ Request and Response messages format and example
- Detailed AuthZ and AuthN ticket and token examples
Open policy enforcement model in WSA/SOA using WS-PolicyAttachment 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

```xml
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/">
  <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>
</definitions>

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

<wsp:UsingPolicy wsdl:Required="true"/>
</definitions>
```
(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
<?xml version="1.0" encoding="UTF-8"?>
<AAA:AAARequest
    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</ActionID>
    </Action>
</AAA:AAARequest>
<?xml version="1.0" encoding="UTF-8"?>
<AAA:AAAAResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="aaa-cnl-response-00.xsd" version="0.0">
  <Resource ResourceId="http://resources.collaboratory.nl/Phillips_XPS1">
    <Decision>Permit</Decision>
    <Status>
      <StatusCode Value="OK"/>
      <StatusMessage>Request successful</StatusMessage>
    </Status>
  </Result>
</AAA:AAAAResponse>
AAA Policy and RBAC/XACML Policy

<table>
<thead>
<tr>
<th>CNL AAA Policy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subject</td>
</tr>
<tr>
<td>Resource/Environment</td>
</tr>
<tr>
<td>Rules</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RBAC/XACML Policy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target (S, R, A, (E))</td>
</tr>
<tr>
<td>PolicySet</td>
</tr>
<tr>
<td>Policy {Rules}</td>
</tr>
<tr>
<td>...</td>
</tr>
<tr>
<td>Policy {Rules}</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>XACML Policy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target (S, R, A, (E))</td>
</tr>
<tr>
<td>Rules</td>
</tr>
</tbody>
</table>
CNLAuthzTicket example – 1011 bytes

<cml:CNLAuthzTicket xmlns:AAA="http://www.AAAarch.org/ns/AAA_BoD"
    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>cml:actions:CtrlInstr</cnl:Action>
        <cnl:Action>cml:actions:CtrlExper</cnl:Action>
    </cnl:Actions>
    <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#" ...</ds:Signature>
</cnl:CNLAuthzTicket>
<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+iynx7K7X2Cxd2iBwCUTQ0n61Szv81DK11Wsq75IsHfusnm56zT3fhKULzEUosb7p6oMLM7hb42+vjfvNeJu2roknhIDzruMrr6hMDsIfaotURepu7QCT0sADm9IfX89Et55EkSE9oE9qBD8=
  </ds:SignatureValue>
  <ds:KeyInfo> << ... snip ... >> </ds:KeyInfo>
</ds:Signature>
RSA `<ds:KeyInfo>` element – 1010 bytes
(total signed ticket with KeyInfo - 3078 bytes)

```xml
<ds:KeyInfo>
  <ds:X509Data>
    MIICADCCAwkCBEgX/FywDQYJKoZIhvcNAQEEBQAwRzELMAkGA1UEBhMCTkwGTAXBgNVBAoTEENvbGxhYm9yX1NleABEBgNVBAsMMHgCAwECAwIBBQIBBQIBAggrBgEE directory>
    <ds:X509Certificate>
      MIICADC...[RSA public key]
    </ds:X509Certificate>
  </ds:X509Data>
</ds:KeyInfo>
```
CNLAuthzToken example – 293 bytes

```
<cnl:CNLAuthzToken TokenID="ed9d969e1262ba1d3a7f33dbd670dd94">
  <cnl:TokenValue>
    0IZt9WsJT6an+tIxhhTPtiztDpZ+iynx7K7X2Cxd2iBwCUTQ0n61Szv81DKl1Wsq75IsHfusnm56zT3fhKU1zEUsob7p6oMLM7hb42+vjfvNeJu2roknhIDzruMrr6hMDsIfaotUREpu7QCT0sADm9IfX89Et55EkSE9oE9qBD8=
  </cnl:TokenValue>
</cnl:CNLAuthzToken>
```

CNLAuthzToken is constructed of the CNLAuthzTicket TicketID and SignatureValue. CNLAuthzToken use suggests caching CNLAuthzTicket’s...
    <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">
    <Evidence>
      <Assertion
        AssertionID="f3a7ea74e515ffe776b10a7eeef0119d7"
        Issuer="cnl:subject:CNLAAAauthority" MajorVersion="1" MinorVersion="1">
        <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>0aZRtdZmq9791j8eDpJ5VT6BxxWbtsApC55PnIsfHRUcOOpMQowXb2TmO2dJGNzFWhMinzXU3/wSDLjv+siO2JGfY279eqM0GqT8VixM15uRu0Asr7AIHv9/DP1ksMND25DnGosMc+YynKogfMqhK+DKqPwFHF6U=/</ConfirmationData>
            </SubjectConfirmation>
          </Subject>
            <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 – 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="f35585dfb5edec48de0c7eadb11c17e">
    <!-- Mandatory elements -->
    <cnl:Subject Id="subject">
        <cnl:SubjectID>WHO740@users.collaboratory.nl</cnl:SubjectID>
        <cnl:SubjectConfirmationData>
            0+qQNAVuZW4txMi8DH6DFy7eLMGxSfKDJY6ZnY4UW5Dt0JFtat1EprUtnjCkzrJUMvWk9qtUzna
            sDdUG+P4Y7qgab+PHiU91ClusZbztu/ZIjNqCnw5su1BQLTumC8ZtYKKJi4WWs+bMMbP8mFNQm
            +M7F4bJIPBfLcxfobk4=
        </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 is constructed of the CNLAuthnTicket TicketID and SubjectConfirmationData which is encrypted SubjectID value

• CNLAuthnToken must be self-sufficient and doesn’t require caching CNLAuthnTicket’s