GAAA Authorisation Framework
for Collaborative Applications

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

This work is funded by Collaboratory.nl, the EU project EGEE and GigaPort
Outline

- Security requirements to Open Collaborative Environment (OCE)
- Job-centric security model for OCE Security Architecture
- Using Generic AAA Authorisation framework and Role Based Access Control (RBAC) for fine grained access control
  - Optimised push-pull-agent model using AuthZ tickets and tokens
- GAAAPI and implementation details – 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
  ▪ 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) via PKI

---

**Signed Order Document**

- (BA/TA1)

**Job Description**

- JobID
- Job Attributes
- Job Priority
- Job Owner
- User List
- User Attributes
- RBAC Admin
- Policy Ref/Attach
- Trust Anchor (TA2)

**Job Manager** (Scheduler)

**Access Control System**

- UserDB
- Policy
- AuthN/Z context

---

VL-e Workshop  
5 April 2005, Amsterdam  
GAAA AuthZ Framework for OCE  
Slide_4
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

VL-e Workshop
5 April 2005, Amsterdam

GAAA AuthZ Framework for OCE
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/tickets format
    ▪ Standard vs proprietary
  ✓ Protocols and Messages format
    ▪ SOAP + XACML Request/Response
    ▪ SOAP + SAML + XACML
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
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”
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 AuthzToken, AuthzTicket must be cached; Resolution mechanism from token to ticket must be provided
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
Mapping between CNLAuthzTicket, XACML Request/Response and SAML Authorization Assertion
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/AzStatemt/Evidence/AttrAssert/Attr/AttrValue
- Ambiguous location for PolicyURIs and SessionID
- SAML1.1 ConfirmationData element is an extensible type – compatibility problems
- XACML Obligation element
  - Can be mapped to SAML Condition element or SAML Advice element
CNLAuthzTicket example – 1011 bytes

<cnl:CNLAuthzTicket 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>SeDFGTVHYTY83ZXxEdsweOP8Iok</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:CNLAuthzTicket>
CNLAuthzToken example – 293 bytes

```xml
<cnl:CNLAuthzToken TokenID="ed9d969e1262ba1d3a7f33dbd670dd94">
  <cnl:TokenValue>0IZt9WsJT6an+tIzhhTPtiztDp2+iy97K7X2Cxd2iBwCUTQ0n61Szv81DK1lwEq751sHfusnm56z3fhkULzEUo7p6oMLM7hb42+vjfvNeJu2roknhIzruMrr6hMDsIfaotURepu7QCT0sADm9IfX89Et55E9SE9o99qB8=
</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)
- Extended RBAC functionality based on GAAA Authorisation framework
  - AuthZ/Resource Session management
- GAAA_tk as RBE and AAA policy expression
  - XACML Request/Response messaging
  - Migration to XACML based policy exchange and combination
- Proprietary and SAML based AuthzTicket format including SAML2.0 extensions
- XML Signature and XML Encryption for JobDescription and AuthzTicket security
- Contribution to XACML, SAML and GT4 development
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
  - GT4 Authorization Framework
  - EGEE gLite Authorisation Framework
- 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
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

- Binding Policy to WSDL with WS-PolicyAttachment
- XACML AuthZ Request and Response messages format and example
- Detailed AuthZ and AuthN ticket and token examples
Traditional Access Control model – setting up trust and authority relations

✓ Policy, attributes semantics and namespaces are known a priori to all participating parties
  o 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
  o 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
  o 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-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>

  <policyLocation>
    <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
GAAAPI dataflow diagram (implements RBAC)
GAAAPI implementation – XACML Request message format

<?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>2SeDFGVHY828xEdsweOP8Iok)yGHxVhozm90</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>
GAAAPI implementation – XACML Response message format

```xml
<? 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">
  <Result ResourceId="http://resources.collaboratory.nl/Phillips_XPS1">
    <Decision>Permit</Decision>
    <Status>
      <StatusCode Value="OK"/>
      <StatusMessage>Request successful</StatusMessage>
    </Status>
  </Result>
</AAA:AAAAResponse>
```
CNLAuthzTicket example – 1011 bytes

```xml
<cnl:CNLAuthzTicket 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>SeDFG6HY83ZxEdsweOP8Iok</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>
</cnl:CNLAuthzTicket>
```

CNLAuthzTicket 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: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>
      MIICADCCAwKCMgEXzrb9kZ1lvcNAQEEBQAwrzZ1YhW8AEBhMCTkwxGTAxBgNVBAoTEENv
bGxhYm9yYXRvcnkubmwxHTABBgNVBAMTFEFBQXV0aHJ1YWNObmF2Ny5jb20My2fEA/
NDYxMzAwMDAxMTA1NDEwNzQ4MDAw

      <ds:X509Data>
        <ds:X509Certificate>
          MIIwDQYJKoZIhvcNAQEFBQAwRzZ1YhW8AEBhMCTkwxGTAxBgNVBAoTEENv
bGxhYm9yYXRvcnkubmwxHTABBgNVBAMTFEFBQXV0aHJ1YWNObmF2Ny5jb20My2fEA/
NDYxMzAwMDAxMTA1NDEwNzQ4MDAw

          <ds:KeyValue>
            <ds:RSAKeyValue>
              <ds:Modulus>3Q6wYVZq9Zw/Xqou1O5uMoyEcX74wCr99xKbmo70y6S1IC423AQt4zSeBe2tBo10HFW7hf2g8f8
nSFCv0JLbxcStEpu11XknpdHC3NYWH1Eg1nrbX7e6wi/gszZ1Xa+kSxaUvWBRJb+abRmjrKm
8B41NNh5kxBi91vtAM=

              <ds:Exponent>AQAB</ds:Exponent>
            </ds:RSAKeyValue>
          </ds:KeyValue>
        </ds:X509Certificate>
      </ds:X509Data>
    </ds:X509Certificate>
  </ds:X509Data>
</ds:KeyInfo>
```
CNLAuthzToken example – 293 bytes

<cnl:CNLAuthzToken TokenID="ed9d969e1262bald3a7f33dbd670dd94">
  <cnl:TokenValue>
    0IZt9WsJT6an+tIxhhTPtiztDpZ+iynx7K7X2Cxd2iBwCUTQ0n61Szv81DKl1Wsq75IsHfusnm56
    zT3fhKUlzEUob7p6oMLM7hb42+vjfvNeJu2rokhIDzruMrr6hMDsIfaotUREpu7QCT0s
    ADm9If
    X89Et55EkSE9oE9qBD8=
  </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="f3a7e74e515ffe776b10a7eef0119d7" IssueInstant="2005-02-15T14:53:23.542Z" Issuer="cnl:subject:CNLAAAauthority">  
        <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>PBLIR0aZrtdZmg9791j855e0j5YT68xW6sJApC5BpJisfrHUCoOpWwOwXBo2TmO2dJGnxFWhMnzd0U78J3J7aJ17819Vb71s150uUa777r7871Hv9/DP1ksMMDZ5DnPmEc+2yqnKogJm8h5K+DRqPwFHF6U</ConfirmationData>  
            </SubjectConfirmation>  
          </Subject>  
          <Attribute xmlns="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 – 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:Subject Id="subject">
        <cnl:SubjectID>WHO740@users.collaboratory.nl</cnl:SubjectID>
        <cnl:SubjectConfirmationData>
            0+qQNAVuZW4t5Mi8DH6DFy7eLMGxSfKDJY6ZnY4UW5Dt0JFtat1EprUtgnjCkzrJUMvWk9qtUzna
            sDdUG+P4ZY7dgb+PHiU91C1usZbztu/ZIjNqCnw5su1BQLTumC8ZTtYKKJi4WWs+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 signed/encrypted – 401/269 bytes

```xml
<cnl:CNLAuthnToken xmlns:cnl="http://www.aaauthreach.org/ns/#CNL"
    TokenID="f35585dfb5edec48de0c7eadb11c17e">
    <cnl:SubjectID>WHO740@users.collaboratory.nl</cnl:SubjectID>
    <cnl:TokenValue>0+qQNAVuZW4txMi8DH6DFy7eLMGxSfKDJY6ZnY4UW5Dt0JFtat1EprUtgjCKzrJUMvWk9qtUzna
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

```xml
<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>
```