GT SAML2.0-XACML2.0 Java Library:
Testing and gJAF/G-PBox integration

Håkon Sagehaug and Yuri Demchenko
University of Bergen and University of Amsterdam

Middleware Security Coordination Group Meeting
6-7 December 2007, Berkley, USA

Outline

- **Goal of our tests**
- **GT SAML2.0-XACML2.0 Library and suggested use**
- **gJAF Overview**
- **Test scenarios and setups: GT, gJAF, G-PBox**
- **Results and problems**
- **Way forward and further developments**
  - OpenSAML SAML2.0-XACML2.0 Extensions
  - gJAF extensions to support SAML-XACML protocol and Obligations

- **Additional materials (SAML2.0-XACML2.0 Specification overview)**

Goal of our tests

- **Use the GT SAML-XACML library to make callout to the external Site Central AuthZ Service (SCAS) from gJAF**
  - LCAS/LCMAPS based SCAS (C based)
  - Specific EGEE need - callout to G-PBox (as a potential Grid oriented native XACML-based SCAS)
  - Common requirement – exchanging, generating and parsing arbitrary SAML-XACML messages
- **Suggested testing stages**
  - First, run provided tests in the Globus environment
  - Testing with both gJAF and G-PBox
  - Additionally, testing with the proposed OpenSAML SAML-XACML Extension Library
- **Provide feedback to the GT developers**

GT SAML-XACML Library and Suggested Use

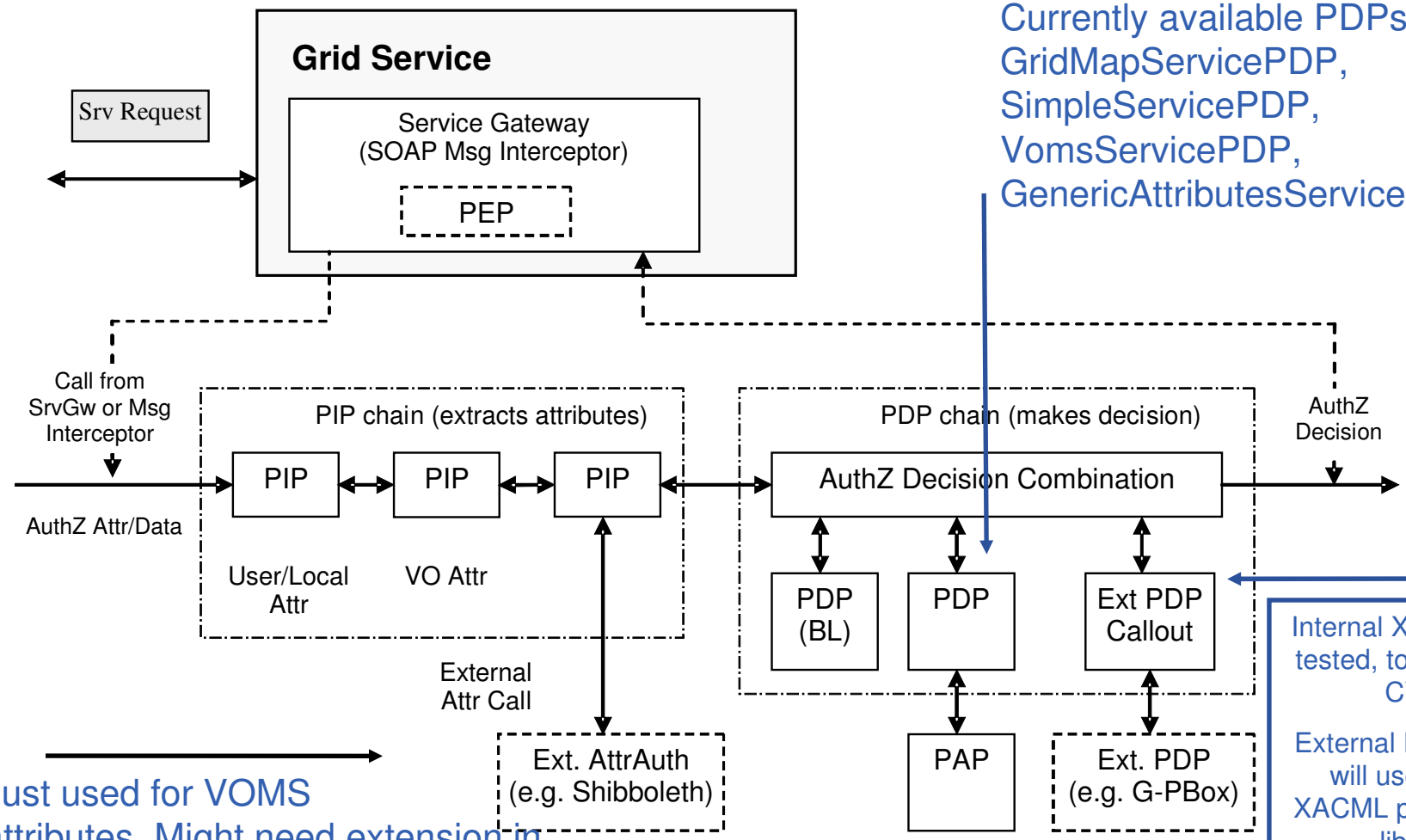
- **Uses Globus WSRF platform/environment**
- **All SAML-XACML classes created from the schema using AXIS2 tool**
  - org.oasis.xacml2.saml.assertion
  - org.oasis.xacml2.saml.protocol
- **Helper classes provided to handle Subject, Resource, Action, Environment information**
  - org.globus.wsrf.impl.security.authorization.xacmlUtil.saml2/xacml2
- **Uses command line client to call remote AuthZ service via Web service interface**

gJAF Overview (Briefly)

- **Provided as org.glite.security.authz Java package**
  - Uses actively java-utils library for VOMS

  (alternatively voms java api directly for voms libraries)
- **Called from applications via an interceptor (PEP)**
  - {MessageContext, Subject, operation}
- **Contains a configured chain of PIP and PDP modules**
  - PIP collects/extracts information to be sent to PDP
  - Each PDP evaluates its relevant attributes against its own Policy
  - Chain is configured to apply PDP decisions combination

gJAF components and connection to the Grid Service (Present)



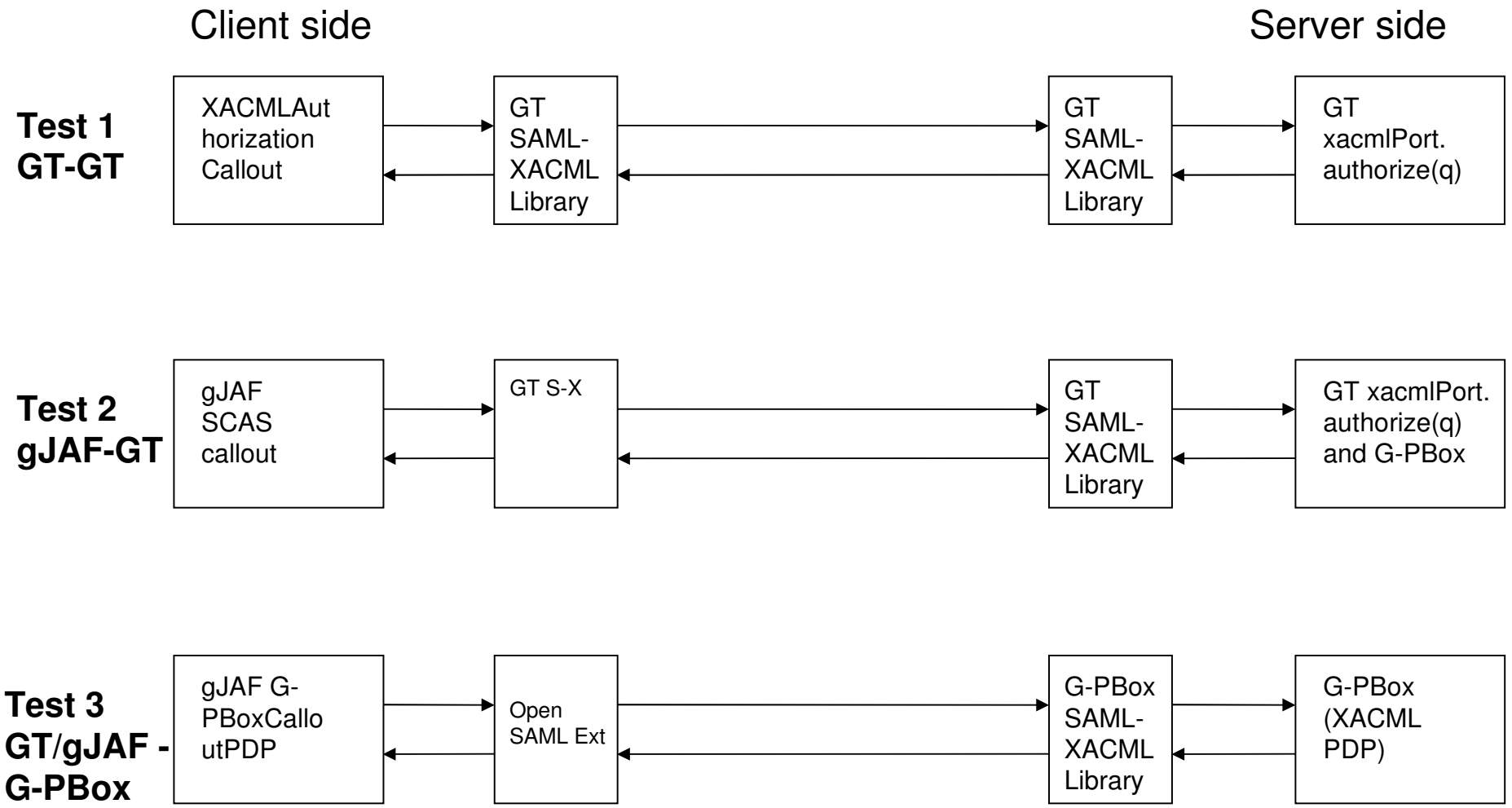
Currently available PDPs:
GridMapServicePDP,
SimpleServicePDP,
VomsServicePDP,
GenericAttributesServicePDP

Internal XACML PDP tested, to be put into CVS.

External PDP callout will use SAML-XACML protocol and library

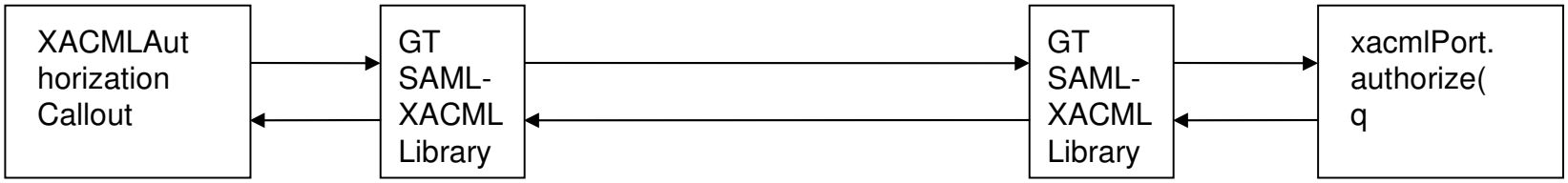
Just used for VOMS attributes. Might need extension in Shibboleth phase III

Test scenarios and involved components



- **These tests has a pragmatic approach, to test full required functionality with different libraries**

Testing in the GT Environment



- **Testing in the Globus environment**
  - Launches two web service containers that hosts the needed web services
  - Uses the command line client to call the web service
  - This sample authorization service checks a file to see if the user is allowed to perform the operation
- **Result: SUCCESS**
  - After some initial problems the test passed successfully

Experience testing in the Globus environment

- The tester should be familiar with Globus before starting on this.
- One could maybe wanted a more light weight test set-up.
- The documentation, can as always in a software development, be better. This of course will be better in time.
- Maybe wanted a more XACML profile of the authorization test service. Where the the actual XACML request was used more together with XACML policy.

Test Scenario 2 - Using GT library with gJAF

- **Modules used in the test**
  - Used the stubs from Globus library
  - Callout service to the SimpleAuthz service provided in Globus web service container and G-PBox
  - Also used the library for native XACML PDP internally
- **Problems encountered**
  - Parsing/extracting S,R,A,E information from the XACML Request
  - No possibility to put multiple Subjects, Resource, Environments and arbitrary AttributeId

Other test experiences - Helper classes

- Setting the subject is done by extracting the DN from the users X509 certificate/proxy.
- Resource and action is set by giving values as string to a helper class for construction the respective element
- Environment element is just setting a date and time value in the element
- The attribute Id is set in a very statical way. With the use of predefined constants

Experience using library for a callout to G-PBox

- The use was easy, after I had some tips from Valerio.
- Since the AuthzService in Globus and G-PBox has the same wsdl, the only thing that had to be changed was the location of the web service
- When the query arrived at G-PBox there was a problem of validating it, but this was solved.
- Evaluate the response with the use of the helper classes was easy, and then returned the result to the calling PEP.

Helper classes - Suggestions

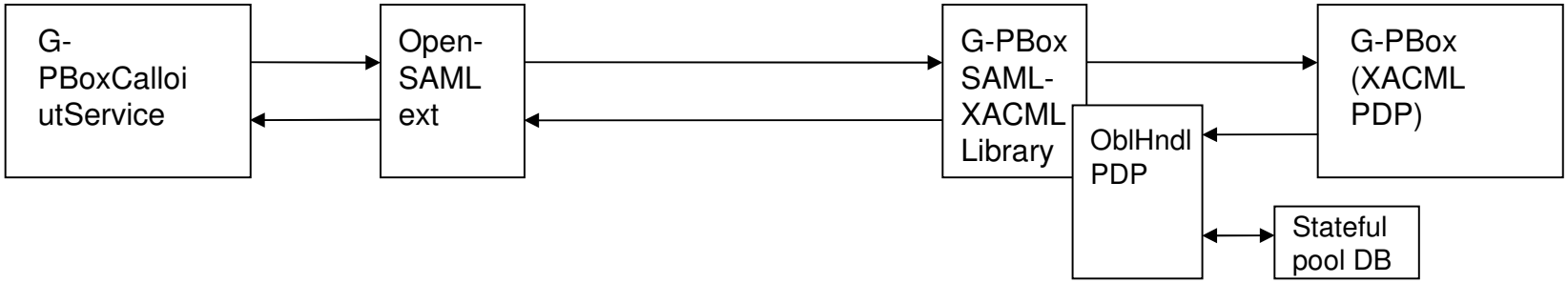
- **Helper classes must support arbitrary XACML Request message generation**
  - Multiple Subject, Resources, Environment elements but single Action element
- **Setting the Subject**
  - More flexible way of setting Subject parameters is needed
  - Should be possible to add a users VO etc.
- **Setting the Environment**
  - This should allow placing arbitrary information into Environment attributes
  - XACML specification defines Environment as containing information relevant to the decision, but independent of Subject, Action and Resource
- **Setting the attributes Id**
  - This should be also more flexible and allow arbitrary Id value (and namespace)
- **Must ensure metadata capabilities**
  - Namespaces
  - Constants

Suggestions

- Multiple subject with the use of HashMap, with Attribute ID and a array of attributes values
- Add methods to set/input XML request into the XACMLAuthzDecisionQuery

All these issues have being discussed on the AUTHZ_INTEROP mailing list

Test Scenario 3 – GT/gJAF and G-PBox

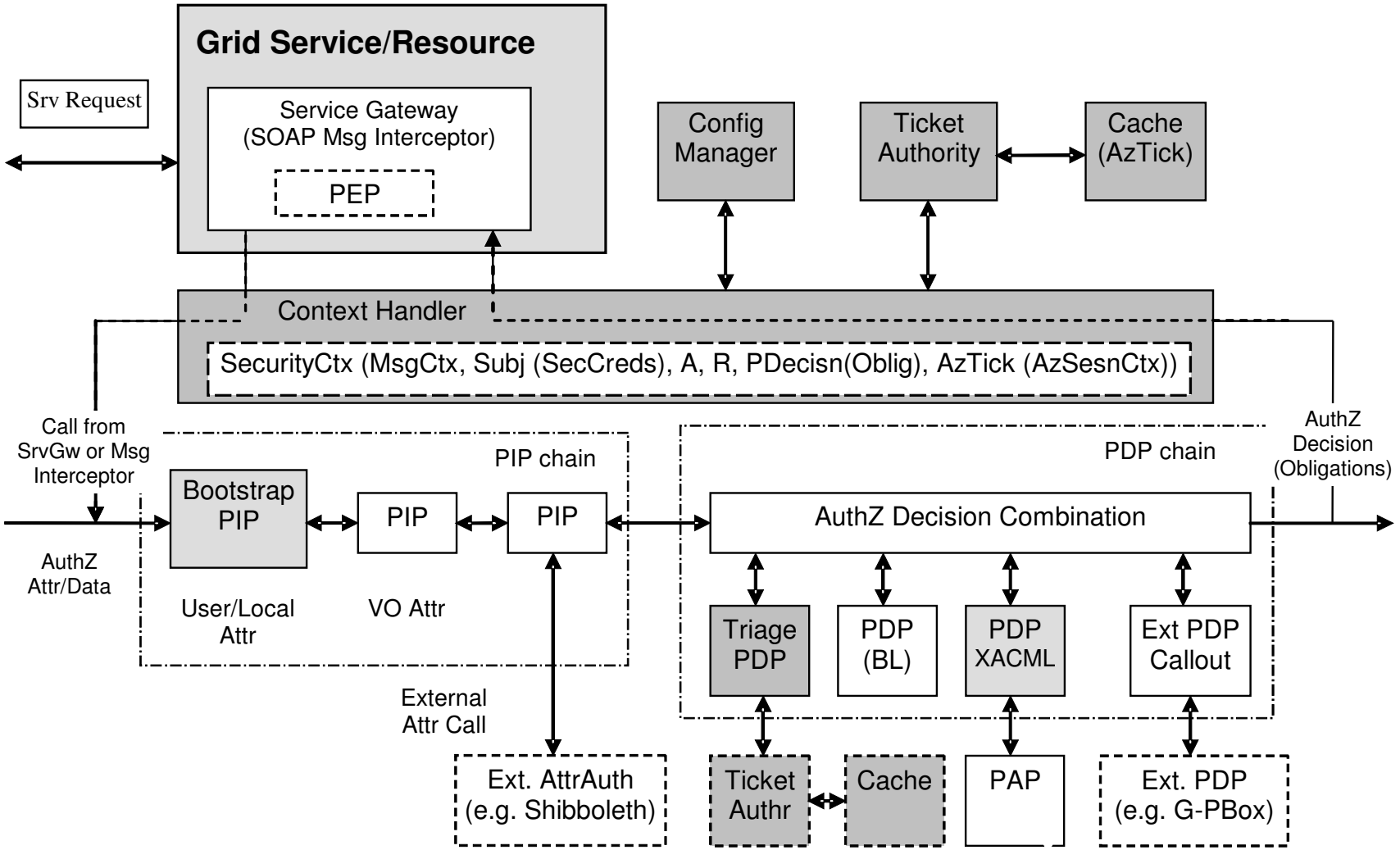


- **Components participating in the tests**
  - Used SAMLXACML stubs from the OpenSAML extension
  - Fixed policy so we knew what result to expect.
  - Input is a XACMLAuthzDecisionQuery
  - Output is a Response message, that contains the decision
  - The response contains the result from the XACMLPDP plus the obligation from G-PBox's Obligation Handler, UID and GID

Way forward and further developments

- **OpenSAML SAML-XACML Extension Library**
  - First version is available from – http://www.bccs.uib.no/~hakont/SAMLXACMLExtension/
  - Currently in the process of contributing to the OpenSAML project
- **ObligationHandler API definition**
  - As a common/unresolved issue for all AuthZ frameworks
- **gJAF extension to support SAML-XACML protocol**
  - Add internal simple XACML PDP
  - Modify SecurityContext information
    - Can be done without modifying AuthZ chain configuration (very sensitive component)
  - Add handling extended AuthZ context in the form of SAML-XACML AuthZ Assertion or other AuthzTicket format

gJAF – Proposed Extensions



Issues for Discussion

- **What should we do with all the information inside a SAML Response**
  - There is assertion element with many possible values
  - What values would we need and not?
  - Important to not have to much inside it
- **How should we process the information**
  - Where/when and how should we handle obligation?
  - Develop a API and a common understanding of
    - Syntax
    - Semantics
    - Proper values for the obligation
- **Should we pass more information to the PEP for instance inside the Message context, instead of just return true or false?**

Additional Information

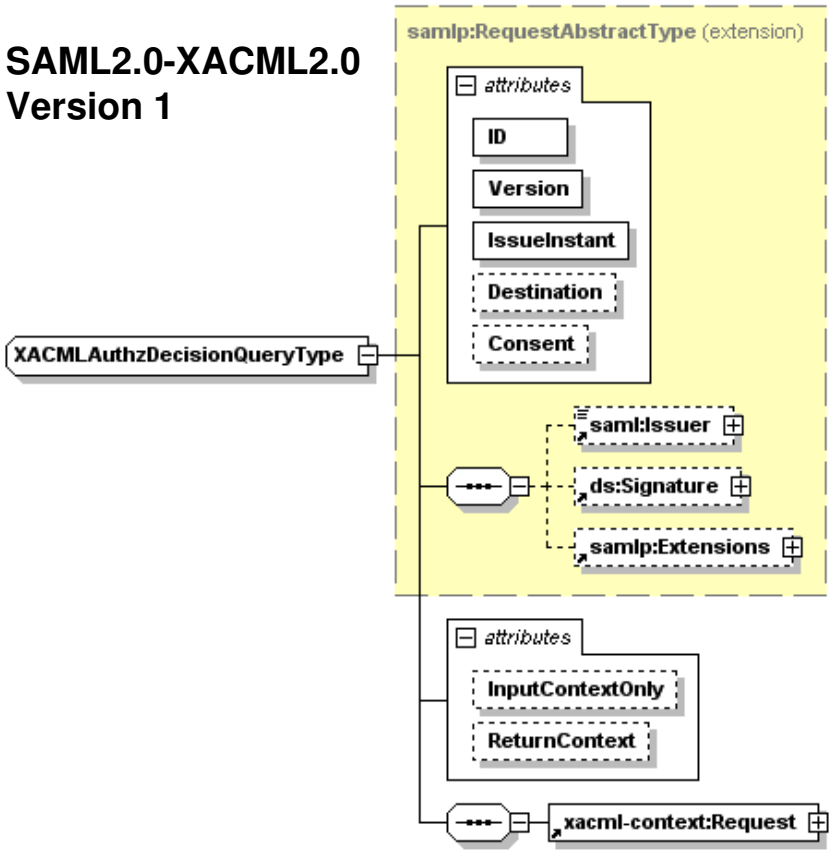
- SAML2.0-XACML2.0 profile Version 1 and Version 2 overview
- OpenSAML SAML-XACML Extension Library
- Examples XACML and SAML-XACML request/response messages

SAML2.0-XACML2.0 Profile Version 1 Overview

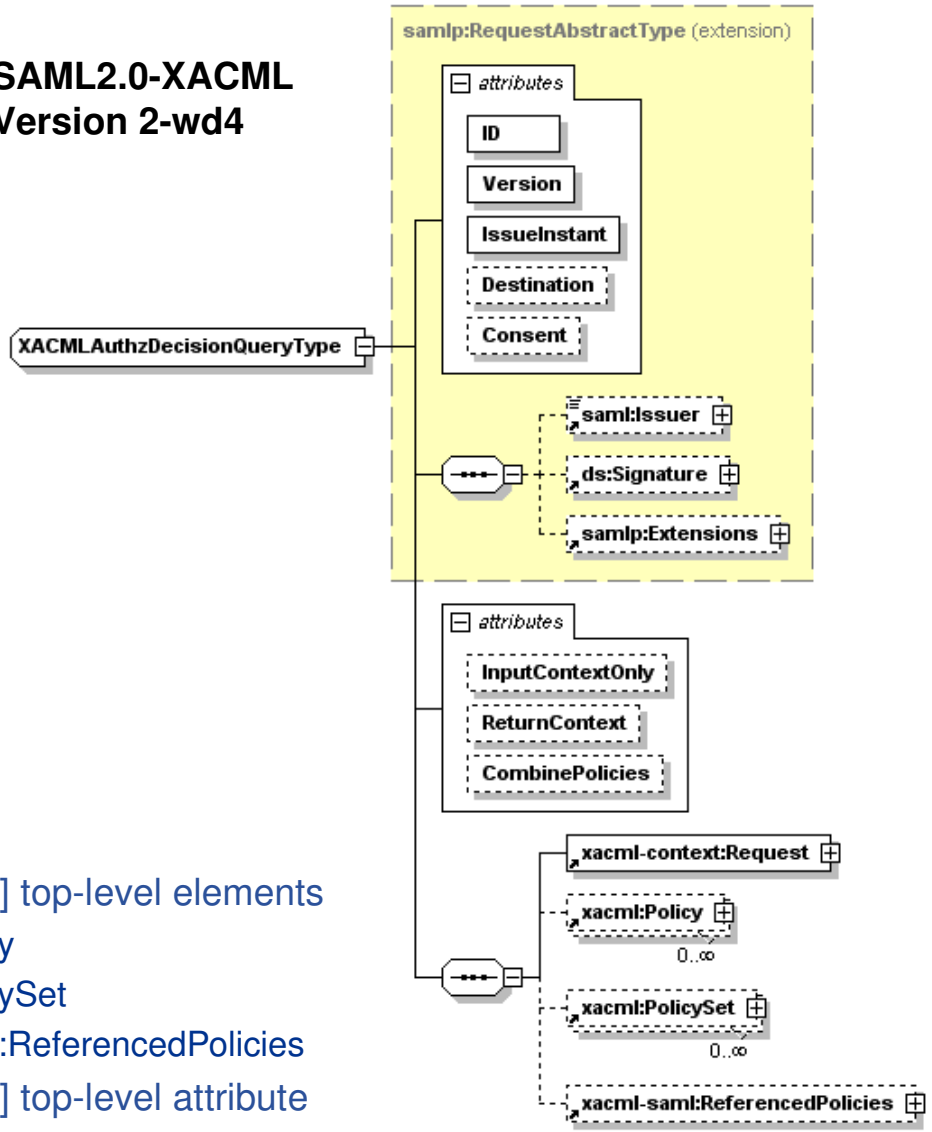
- **Uses SAML2.0 Protocol and Assertions format for wrapping XACML Request/Response messages**
- **Introduces the following new queries and statements**
  - XACMLPolicyQuery
  - XACMLPolicyStatement
  - XACMLAuthzDecisionQuery
  - XACMLAuthzDecisionStatement
- **The *Query is sent to the remote AuthZ service or Policy repository and *Statement is returned back *enclosed* into the SAML Assertion**
- **Recommended by OGF OGSA AUTHZ-WG as AuthZ service interface**

XACMLAuthzDecisionQuery (Version 1 vs Version 2-wd4)

SAML2.0-XACML2.0
Version 1



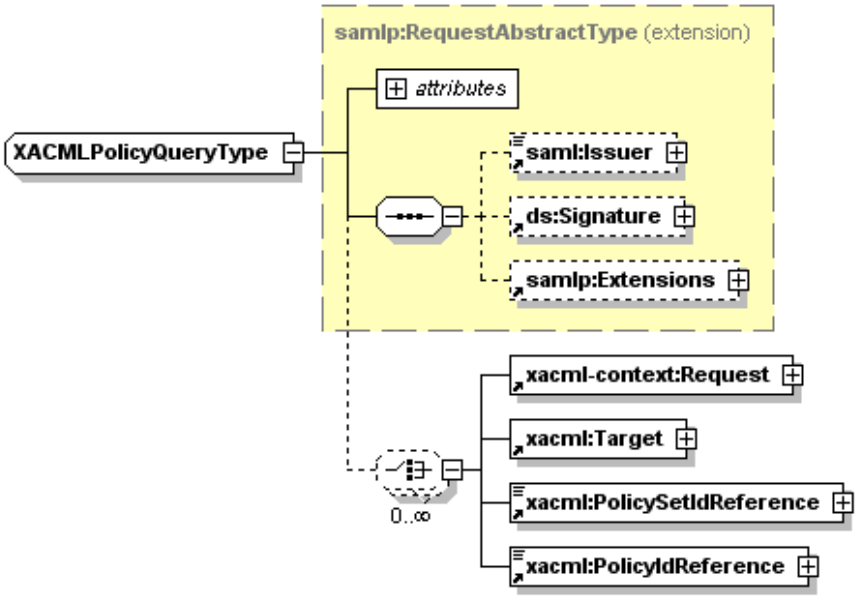
SAML2.0-XACML
Version 2-wd4



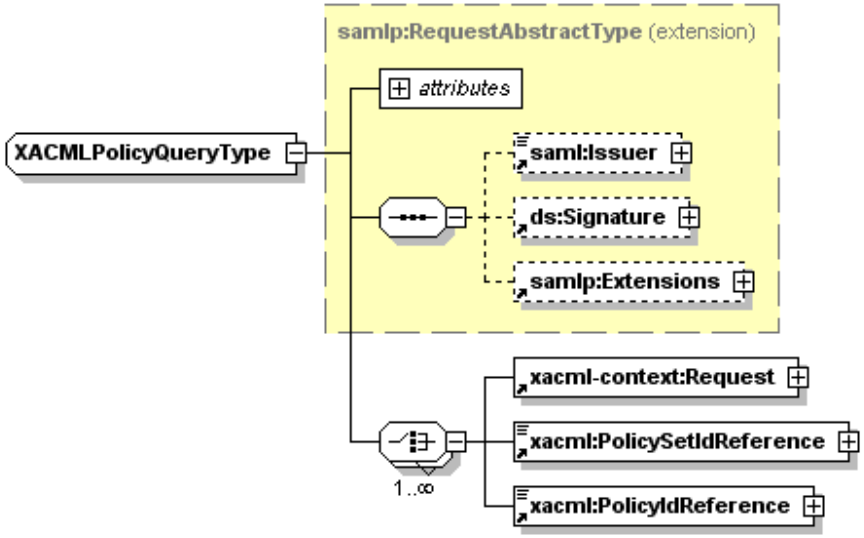
- Added [optional] top-level elements
  - xacml:Policy
  - xacml:PolicySet
  - xacml-saml:ReferencedPolicies
- Added [optional] top-level attribute
  - CombinePolicies

XACMLPolicyQuery (Version 1 vs Version 2 – wd4)

SAML2.0-XACML2.0
Version 1

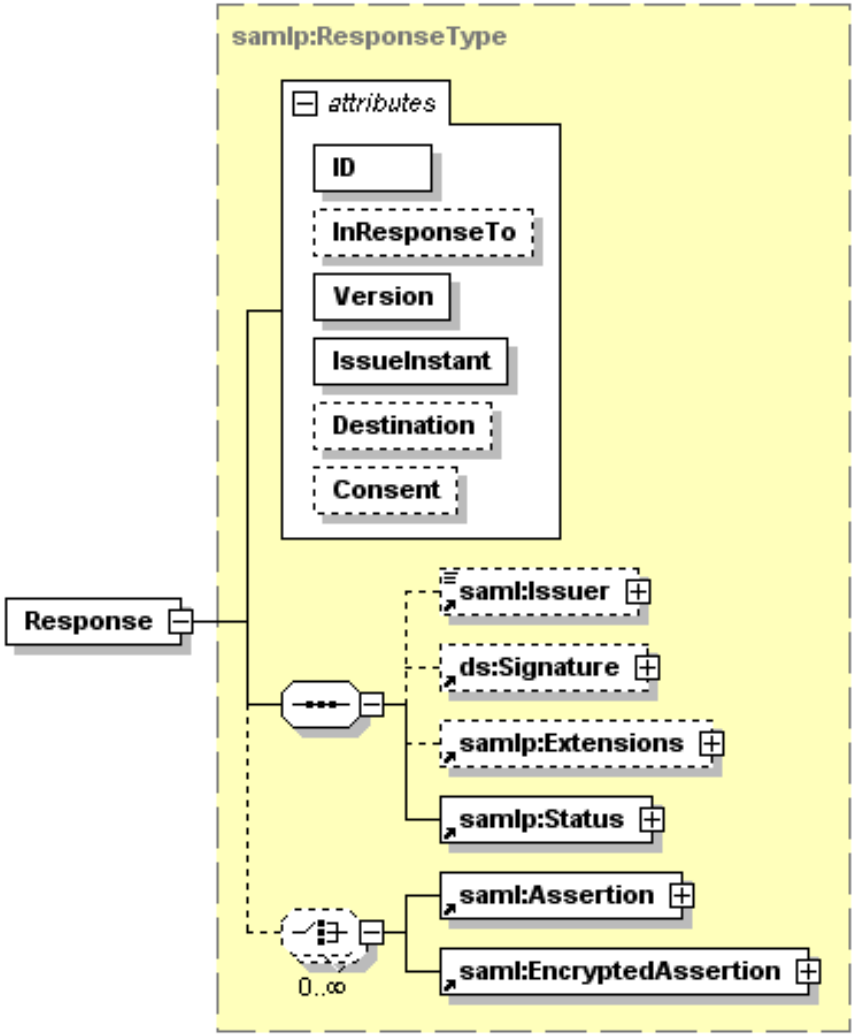


SAML2.0-XACML
Version 2-wd4

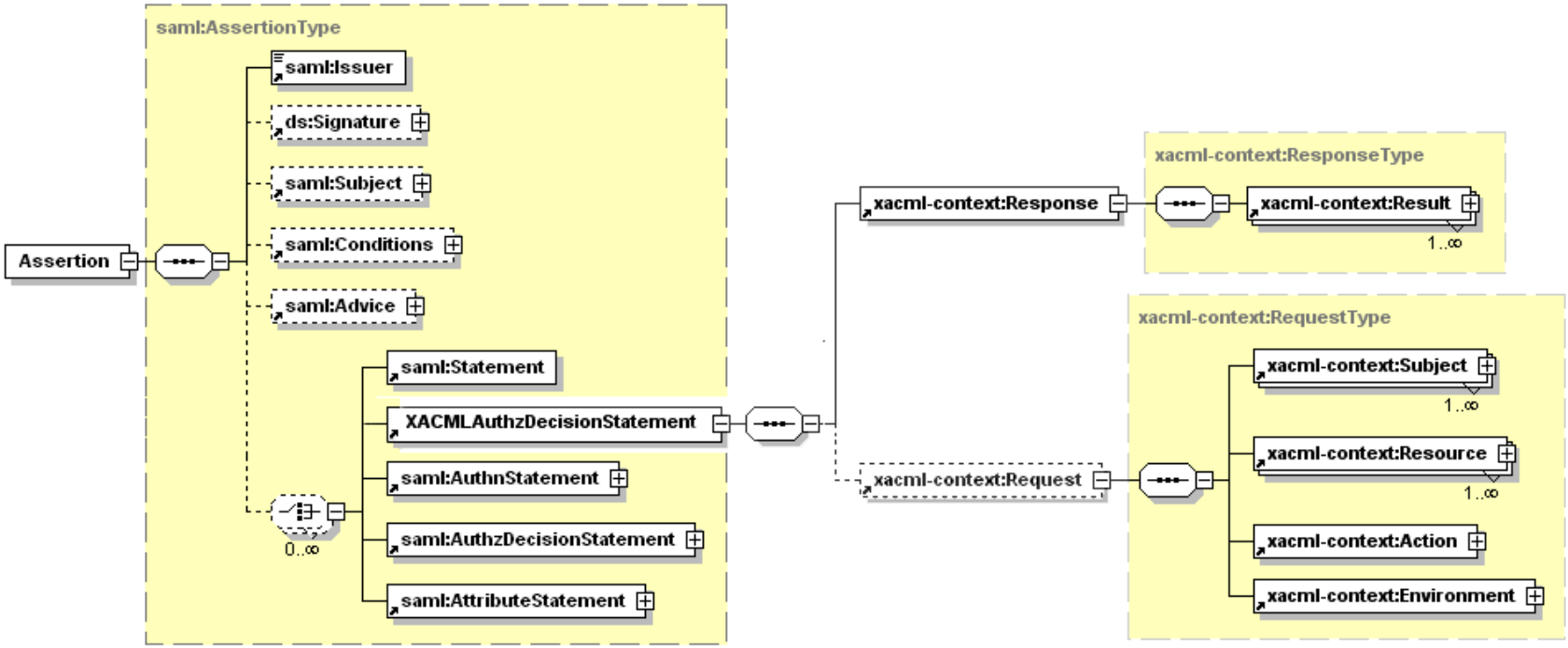


- Removed top-level element
  - xacml:Target

SAML2.0 Response



SAML Assertion containing XACMLAuthzDecisionStatement



OpenSAML SAML-XACML extension library

- Implements SAML2.0 profile of XACML2.0 Version 1 (with errata)
- Builds upon the source of OpenSAML
- Every XML-element/object in OpenSAML and the extension consists of
  - An interface
  - The implementation
  - Builder for creating it
  - Marshaller, Java->XML
  - Unmarshaller, XML->Java
- Remaining issues before ready for OpenSAML 2
  - Add object providers so we can construct XACML elements without the use of third party library, e.g. Sunxacml implementation.
  - Getting it up to date in regards to the last documents
  - Make a helper class for making a XACML Request context from a SAML Assertion
    - Described in the latest version of the SAML XACML profile document