An Authorisation Interface for the GRID

David Chadwick
d.w.chadwick@salford.ac.uk
Outline

• What is an authorisation interface?
• Why is a standard GRID authorisation interface needed?
• Interface requirements
• A brief introduction to SAML
• GRID authorisation interface implementation in SAML
• Future research work
What is an Authorisation Interface

• An authorisation infrastructure should be able to say who is authorised to access which GRID resources in which ways and under what conditions, according to the policy set by the resource owner

• An authorisation interface should be able to answer the following sorts of questions
  – Is this user allowed to access this Grid resource in this way
    • Ans. Yes, No, Yes subject to, Don’t know
  – Is this user who possesses these attributes allowed to run this Grid job
    • Ans. Yes or No
  – Tell me all the rights that this user has to this resource
    • Ans. User S with role R is allowed to access target T in modes N and P
Why is a standard GRID authorisation interface needed?

- Several different authorisation infrastructures now exist e.g. VOMS, CAS, PERMIS, Akenti etc.
- Authorisation is complex. All infrastructures have different features, capabilities, strengths, weaknesses.
- GRID application developers should be able to pick the best one to suit their requirements.
- There is no one-size fits all at the moment – it's still too new and we are still learning.
- Thus we need a standard interface, so application developers can mix and match authorisation infrastructures with their applications.
Interface Requirements (1)

• First, split the authorisation function into an application dependent function and an application independent function, according to the ISO 10181-3 model

• This allows an application independent decision making engine to be built, that makes it decisions based on the authorisation policy. It does not need to be changed when the Grid application changes
X.812|ISO 10181-3 Access Control Framework

**User’s Site**
- Initiator
- Submit Access Request
- Internet
- Decision Request
- ADF
  - ADF= application independent
  - Access control Decision Function

**Target Site**
- AEF= application dependent
- Access control Enforcement Function
- Present Access Request
- Decision
- Target

All Hands, Nottingham, 3rd Sept 2003
Interface Requirements (2)

- Support standalone decision server and built-in integrated decision server

- Support different types of client
  - The AEF, the user himself, or an agent acting on behalf of the user
  - This places requirements on authenticating the client (out of scope of current work)
Interface Requirements (3)

• Support return of simple decisions
  – Boolean, granted/denied

• Support return of authorisations
  – The named user with this set of attributes is allowed to access this named resource in the following ways

• Support conditional responses for when insufficient input is provided
  – Access is granted subject to the following conditions being satisfied
Interface Requirements (4)

• What information should be passed in the request?
  – The subject’s name (their X.500 distinguished name)
  – The subject’s credentials (attributes, roles)
  – The requested action(s)
  – The target resource
  – Environmental parameters
    • Time of day, IP calling address, account balance etc.
Interface Requirements (5)

- Need to support default values for the input information
  - Subject default name means “public”
  - Subject default credentials are none (or those available to everyone)
  - Requested action is all known to the ADF
  - Target resource is all targets protected by this ADF
Interface Requirements (6)

• Support the Attribute Push and Attribute Pull models, and if Pull mode, the client should be able to say where the attributes are located.

 Retrieve User Attributes

PULL MODEL

PUSH MODEL

Authorisation Interface

AF

User attributes +
Decision Request

Decision Response

Decision Request
+ location of attrs

Authorisation Interface

AF

Decision Response

All Hands, Nottingham, 3rd Sept 2003
Interface Requirements (7)

• Need some mechanism for telling the ADF which policy to use, or where to get it from
  – Pre-configured into the ADF
  – Client could pass (part of) the policy along with the decision request
  – The policy could be distributed into several sub-policies
  – Could be a separate Policy Management Interface

• Decided to specify Policy Management at a later date
Interface Requirements (8)

• Allow optimisation of the interface through multi-step decision making

• Step 1. Here is a set of credentials for the user. Validate them against the policy
  – Check signatures, throw away invalid, expired, untrusted credentials.
  – Cache the result (in the client or the server)
  – Typically takes place during user log in

• Step 2. The user want to access this resource in this way. Return granted or denied based on cached attributes
  – Repeat this step as often as needed
Security Assertions Markup Language (SAML)

- Industry standard specified by OASIS
- Allows security assertions to be encoded in XML
- Authentication assertion states how the subject was authenticated
- Attribute assertion states which attributes have been assigned to the subject
- Authorisation decision assertion states which access to which targets has been granted or denied to the subject
SAML Assertion Contents

• The assertion
• name of the issuer of the assertion
• the date and time the assertion was issued
• an assertion ID (for ease of subsequent reference)
• the version of SAML the assertion conforms to
• some optional conditions (that must be obeyed by the assertion user)
• some optional advice (that can be ignored by the assertion user)
• Optionally, the digital signature of the issuer
SAML Request/Response Protocol

- SAML assertions can be Requested and returned in SAML Responses
- SAML Request/Response Protocol can be mapped onto any underlying protocol. Http and SOAP is standarised
- SAML authentication request asks “What assertions containing authentication statements are available for this subject?”
- SAML attribute query asks “Can you return the requested attributes for this subject?”
- SAML authorisation decision query asks “Should these actions on this resource be allowed for this subject, given this evidence?”
SAML Request Contains

• the SAML query
• a unique request ID
• the date and time the request was issued
• the type of assertion responses the requestor would like to be returned to the query (called Respond With parameters)
• the digital signature of the requestor (optional)
• the version of SAML this message conforms to
SAML Response Contains

- zero or more SAML assertions
- a status code (indicating if the request was successful or not)
- the unique ID of the original request
- a unique ID for this response (optional)
- the time the response was issued (optional)
- the intended recipient of the response (optional)
- the digital signature of the responder (optional)
- the version of SAML this message conforms to
Interface Implementation in SAML

- Use the SAML Authorisation Decision Query Request message to make an authorisation request, BUT
- It is limited, therefore extensions needed to be defined
- Use the SAML Authorisation Decision Response message to return the authorisation decision, BUT
- It is limited, therefore extensions needed to be defined
SAML Authorisation Decision Query
Request Message

• the name of the subject
  – X.500 DN from the PKC or null for public access

• the URI of the resource
  – Grid Service Handle (GSH) of the service as described in OGSI

• the action being requested
  – string describing the operation and a URI specifying the namespace of the action, or a special value for “all”

• evidence to support the request
  – holds the user’s credentials and the environment parameters

• Technical details are given in the paper
Extensions needed to SAML Request Message

• No way to signal attribute PULL model
  – Push model is signalled by including the credentials in the request
  – Pull model signalled with new Reference statement, holding the URI of the repository where the credentials may be found

• No way to signal multi-step decision making
  – RespondWith Attribute defined, which signals first step of multi-step decision making is being requested
SAML Authorisation Decision Response Message

• Response is Permit/Deny/Indeterminate
  – but this does not include Granted-subject-to, so we chose to use Indeterminate for this

• Authorisation decision assertion states which actions to which targets has been granted/denied to the subject, given the attached evidence and optional conditions
  – I.e. an authorisation, not a simple decision
Extensions needed to SAML Response Message

• A simple Decision Response message is needed, containing a Boolean only

• Respond With set to either Authorisation or Decision to signal which type of response is needed
Future Research

- Working with OASIS SAML and XACML groups to standardise the SAML extensions defined here in SAML v2
- Implement it in GT3 and PERMIS, and pilot with UK Grid projects
- Start to standardise the policy and the conditions, based on XACML
- Specify a policy management interface
- Distribute the policy, and allow subjects to present (sub)policies to the target