UNICORE - authorization

Krzysztof Benedyczak
ICM, University of Warsaw

Authorization overview

- Unicore/X base module (uas-core) provides an authorization stack.
- It is used by most of UNICORE servers:
  - Unicore/X,
  - Workflow engine,
  - Service orchestrator
  - optionally by Registry
  - others.
- The PDP engine is XACML based.

AuthZ attributes

- Authorization is attribute based.
- UNICORE provides mechanisms:
  - to configure several attribute sources and
  - to choose combining algorithm.
- Therefore users permissions can be flexibly configured by changing their attributes.
  - It's more convenient for administrators then modification of XACML policies.

XACML in UNICORE

- Version 1.1 used.
- Sun's XACML library.
- Only one implementation is used with a node-local policy.
- Other implementations may be configured.
- In most deployments a standard XACML policy is never modified.

Details

- UNICORE authorizes every web service call.
- XACML Request contains (in most cases):
  - Accessed WSRF resource id
  - Web Service name
  - WSRF Resource owner
  - Action (==web service operation)
  - Attributes of the caller (including identity).
- Two attributes are handled specially:
  - **role** (role 'user' is required for normal access)
  - **xlogin** (contains local username).

Default policy (simplified)

- Role 'admin' may do everything.
- Owner may do everything with the owned resource.
- Some basic read-only operations are allowed for everybody authenticated.
- Some fundamental operations (e.g. job submit) are allowed for users with 'user' role.

Current work

- There is ongoing work on integrating additional ACLs controlled by users.
  - Motivation: users want to share their resources.

Summary

- Implementation of support for a remote XACML authorization service in UNICORE should be straightforward.
- It seems that the functionality is not critical: up to now default policy was sufficient, when supported by a flexible attribute management solutions.