Standards driven AAA for Job Management within the OMII-UK distribution

Steven Newhouse
Director, OMII-UK
s.newhouse@omii.ac.uk

OMII-UK

- A partnership between projects:
  - myGrid at Manchester (Carole Goble - Chair)
  - OGSA-DAI at Edinburgh (Malcolm Atkinson)
  - OMII at Southampton (Dave De Roure)
- Started January 2006

All funded for 3 years

  - Manchester – Expanded Engineering activity
  - Southampton – Expanded Community activity
  - Edinburgh – Continuation of OGSA-DAI team

National e-Science Centre

University of Southampton

MANCHESTER 1824

Objectives of OMII-UK

- To distribute a sustained, well-engineered, interoperable, documented and supported set of easily-used integrated composable services, components and tools
- To engage proactively with user communities in defining and developing this software
- To maintain a leading international role in advanced e-Infrastructure provision

Current Container Security

- Only consider a Web Service
- Primarily Authentication through WS-Security
  - Digital Signature on a signed message
- Signature **MUST** be signed by a certificate from a known CA
- Authentication data available to the service
- Outgoing message signed

OMII-UK Job Authorisation

- OMII 1.x: Application execution from GRIA
  - Defined model enforced by PBAC
  - PBAC: Process Based Application Control
  - User registration & account (quota) creation
  - Resource allocation for compute and data
  - Data in → Application execution → Data out.
  - Application needs to be installed on the machine

PBAC:
Process Based Access Control

- Specify server side workflow
  - Need to have performed Action Z on Service A before Action Y on Service B
  - Check authorisation policy rather than interaction state
- State interaction captured within a ‘conversation’
  - Authorisation action is related to a particular conversation
- Client interaction is planned & context dependent

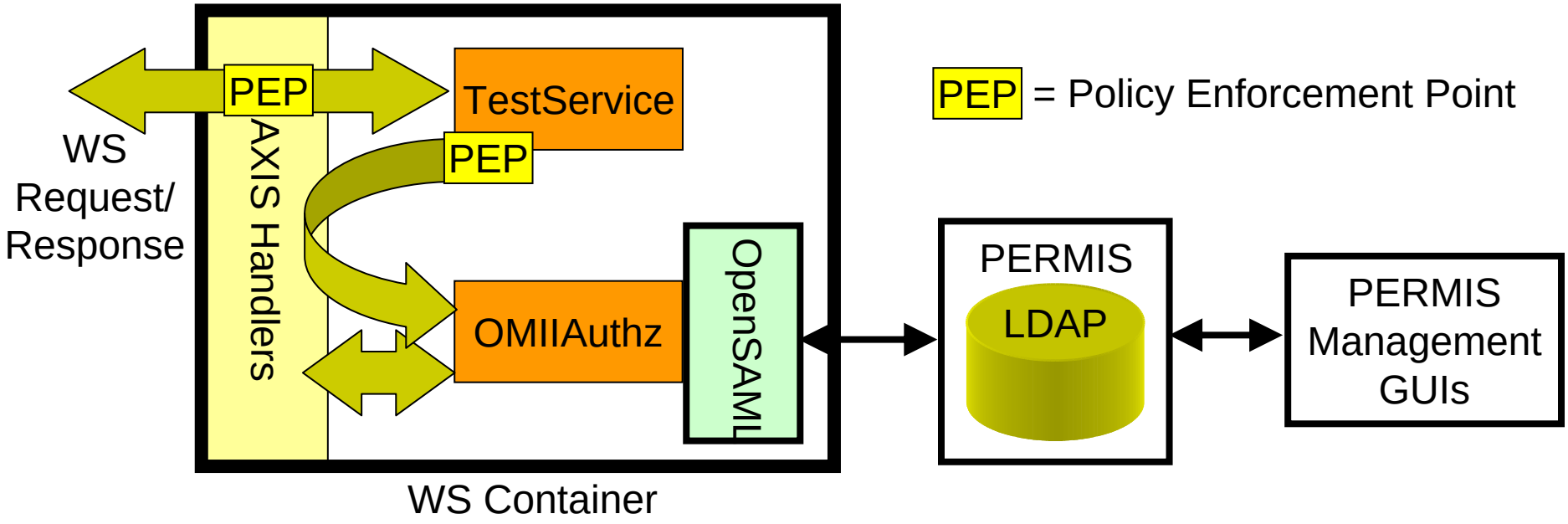
OMII-UK Job Authorisation

- OMII 2.x: GridSAM
  - SAM: Job **S**ubmission **a**nd Job **M**onitoring
  - Uses JSDL to define the ‘job’
  - Various back end environments ‘DRMConnector’
  - Service specific Authorisation
    - gridmap like
  - Connector specific Authorisation
  - WS Management Interface:
    - submit, status, terminate, …

Need to do better…

- An Authorisation policy that can be applied across consistently across all services
  - Within a hosting environment
  - A network of hosting environments (e.g. VO)
- A solution that can be reused:
  - Portlets
  - Service specific policies:
    - Data tables within a database
    - Queues or processor/memory limits within a job
- Standards driven

Current Prototype



- PERMIS: Generate Attribute Certs & Policy
- Authz Service: SAML 1.1 Assertion port type

Authorising Service Access

- Handler in the request chain
- Authorisation decision based on:
  - Requesting entity (user)
  - Target (service)
  - Action (operation)
- Great for job creation… but it is static
  - Same policy for job creation as termination, etc.

Dynamic Service Authorisation

- On job creation create a job specific policy
  - Steven's job – he can manipulate & delete it
  - But, the administrator can also delete it.
- But Steven may also want to allow Erwin to be able to manipulate the job
  - Provide an interface to manipulate policies
- Reuse the same Authorisation Service
  - Requesting entity (user)
  - Target (service)
  - Action (operation)

Other gaps…

- The third ‘A’ – Accounting
  - Looking at RUS & UR options
  - Account (quota) solution from GRIA
  - Applying for an account (e.g. GAMA, PURSe)
- The silent ‘A’ – Audit
- Attribute Management
  - VOMS
  - Standards?

Summary

- Mange authorisation policies across services
- Accounting (use against quota) is important
- Pick up on existing standards & tools
  - Authorisation infrastructure
  - User registration & account generation
- Currently a non-GSI world
  - But out-of bound use through MyProxy
- Emerging need for dynamic policies