EUROPEAN MIDDLEWARE INITIATIVE

EMI-ES implementation in UNICORE

Bernd Schuller
Forschungszentrum Jülich GmbH

Outline

- UNICORE/X “Compute” Overview
- EMI-ES
- Implementation approach
- Status
- Plans

UNICORE 6

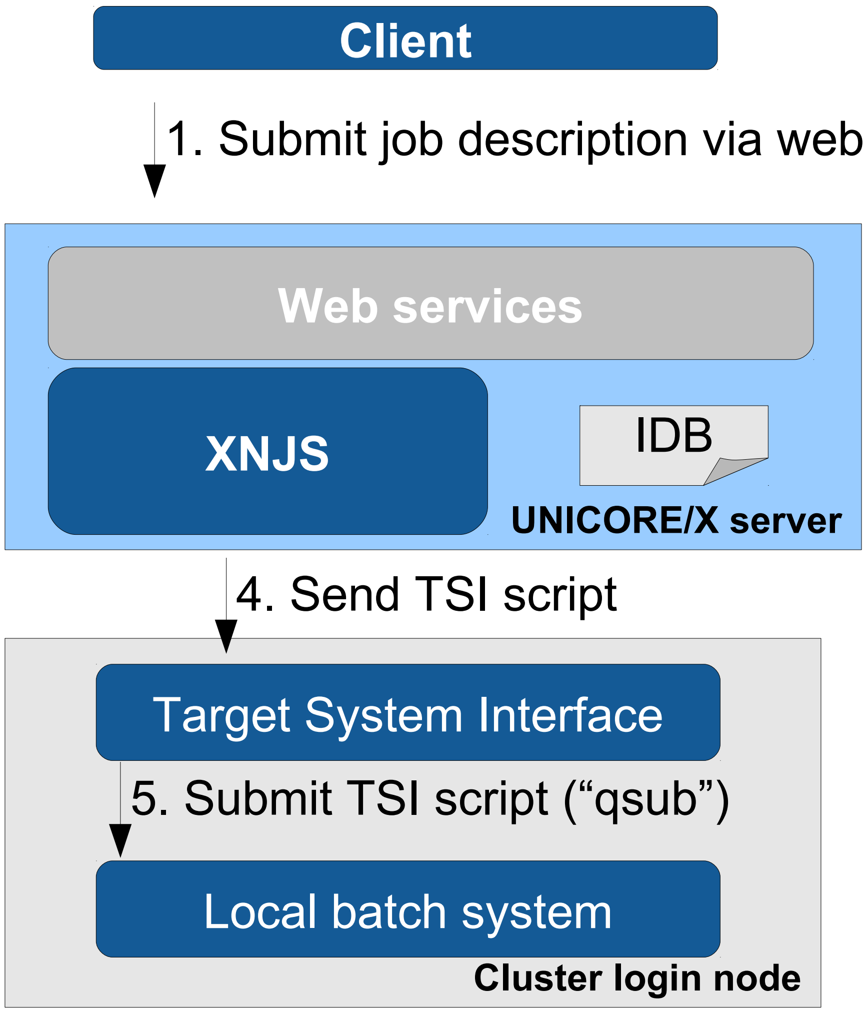
Overview

- Integrated, complete Grid middleware stack including graphical & commandline clients
- Focus on ease of use (both end users and admins)
- Lightweight and platform independent, coded in Java and Perl
- Supports many resource management systems and operating systems (Linux/Unix, Mac OS X, Windows)
- Strong support for applications and workflows

UNICORE 6

EUROPEAN MIDDLEWARE INITIATIVE

Components involved in job execution



2. Web service interfaces:
- UNICORE proprietary ("UAS")
- OGF standard OGSA-BES

3. The XNJS generates a TSI script from the job description and local configuration stored in IDB file

UNICORE 6

Component responsibilities

- Client (mostly thinking of UCC here)
  - Parse user-friendly(!) job description and interact with the services
- Web services stack
  - Security (authentication, XACML callouts for authorisation)
- XNJS
  - Processing engine
- TSI
  - Batch system and file system access

EMI-ES summary

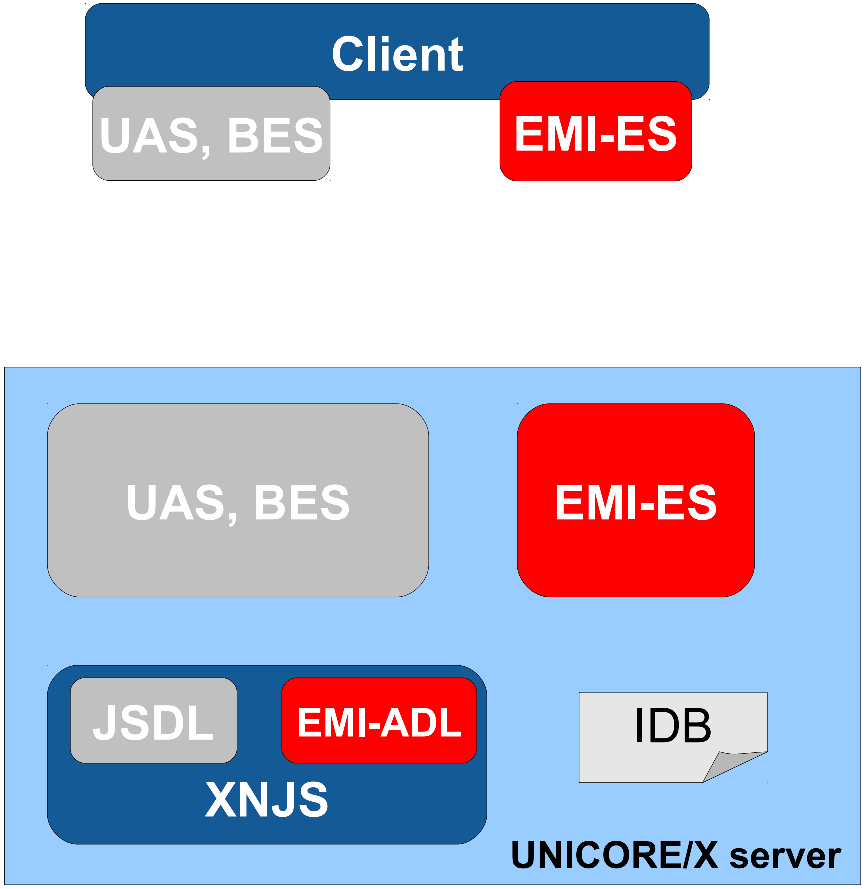
- Web service interface definitions
  - Delegation (issuing a proxy cert for (GridFTP) data staging)
  - Create and manage activities (i.e. jobs)
  - Get GLUE2 info about the compute service and the activities
- Job description
  - XML schema to describe a single job to be executed by EMI-ES
  - Covers serial and parallel jobs
  - Runtime environments

EMI-ES specification status

- v1.03
- Some small changes proposed (MPI-TF) to job description
- Home page

  https://twiki.cern.ch/twiki/bin/view/EMI/EmiExecutionService

EMI-ES in UNICORE: implementation tasks



1. create XMLBeans from XML schemas

3. create web service interface classes and implementation classes

3. extend/refactor XNJS to be able to handle the EMI-ES job description

4. create client classes wrapping the raw web service interfaces

5. (?) create client (UCC) module allowing to use EMI-ES from UCC

Status

- XNJS refactoring
  - done
- Core EMI-ES service implementation
  - Already functional, but still many things to do – estimated at 30% complete
- Client classes
  - Development has started
- UCC module
  - Not started

https://twiki.cern.ch/twiki/bin/view/EMI/EMI_ES_Status-Update

Some remaining issues

- GetActivityInfo
  - Allows to query ES for activities using GLUE2. Will be rather hard to implement for us (GLUE2 is not a first-class citizen in UNICORE)
- Activity management and access control
  - Can't re-use UNICORE security layer
    - activities are not individually addressable (like UNICORE jobs are), only through management service

Client issues – UCC

- UCC was not designed to target multiple "service types"

- New command group will be necessary for EMI-ES
  - e.g. "ucc run" → "ucc emi-es-run"
  - Added complexity for users :-(

- Helper classes might need refactoring
  - e.g. parser for UCC job description

People involved

- CINECA will put in the major effort
  - Michele Carpené
  - New person who is just starting work
- JUELICH
  - Shiraz Memon (GLUE2)
  - Shahbaz Memon (WS layer)
  - Björn Hagemeier (Client API)
  - Bernd Schuller (XNJS related tasks, job description parsing, features)

Plans

- EMI-ES implementation to be included in EMI-2
  - Probably not same level of “production ready” as existing services
- TODOs
  - Finish code as far as necessary and possible
  - Integration? Probably will release it together with other services in UNICORE/X package
  - Documentation?

EUROPEAN MIDDLEWARE INITIATIVE

Thank you!

EMI is partially funded by the European Commission under Grant Agreement RI-261611