Lessons Learned from UNICORE EMI-ES Adoption towards Improved Open Standards

Shahbaz Memon et al.

Jülich Supercomputing Center

29-03-2012

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

Outline

- Motivation
- EMI-ES Introduction
- UNICORE Overview
- UNICORE EMI-ES Implementation
- OGF Standards
- EMI-ES enhancements
- Conclusion

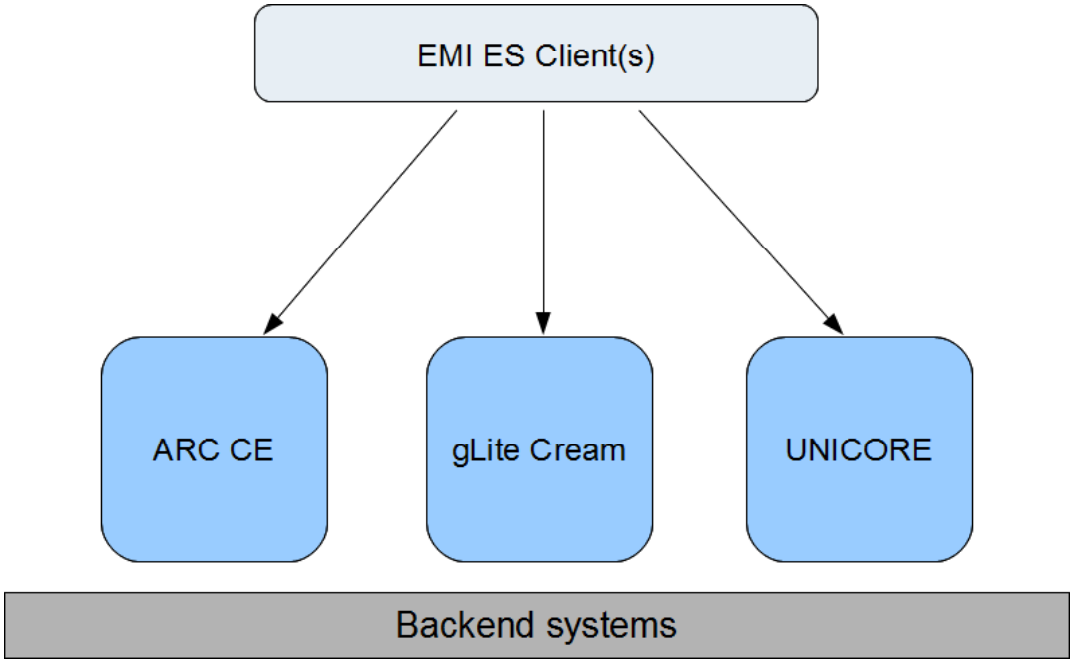
EMI INFSO-RI-261611

Motivation

- EMI mandate to support diversified scientific communities
- Proprietary interfaces was not useful for e-Infrastructures
- Job management standards permeated the Grid middleware stacks to attract myriad set of scientific user communities
- Evolving application requirements influence middleware stacks – EMI user requirements
- They can be improved with advanced execution service concepts
- EMI-ES: a step towards the next generation of Job management standards.

EMI INFSO-RI-261611

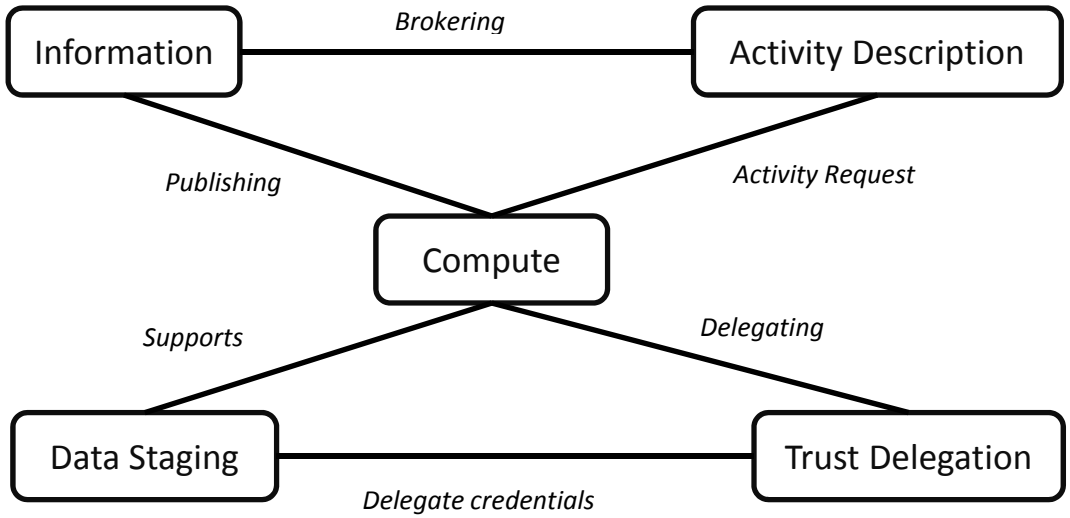
Approach



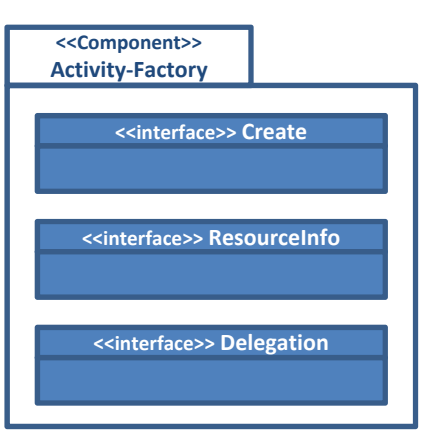
EMI-ES Overview

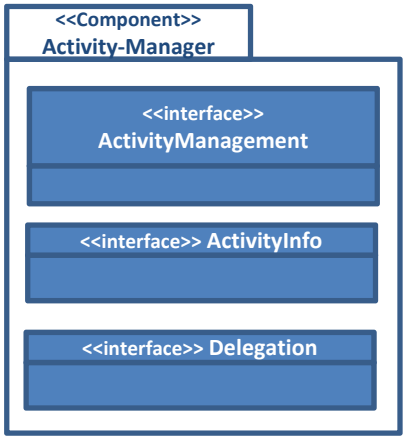
- Web Service interface definitions
  - Create and manage vector of activities
  - Expose capabilities as GLUE-2 instance
  - Delegation (issue a proxy cert for data staging)
- Job information Model - ADL
  - Describes a job to be executed by EMI-ES
  - Encompass execution environments
  - Supporting the concept of serial and parallel jobs

EMI-ES Conceptual Model



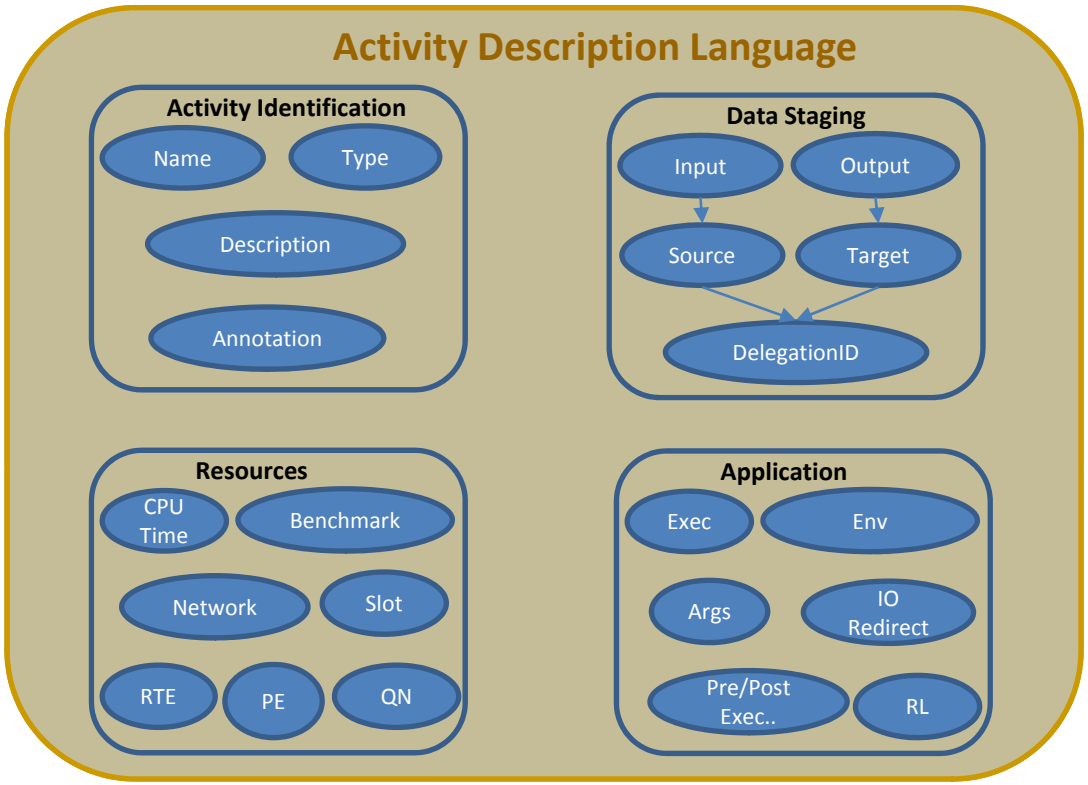
EMI-ES Component Architecture



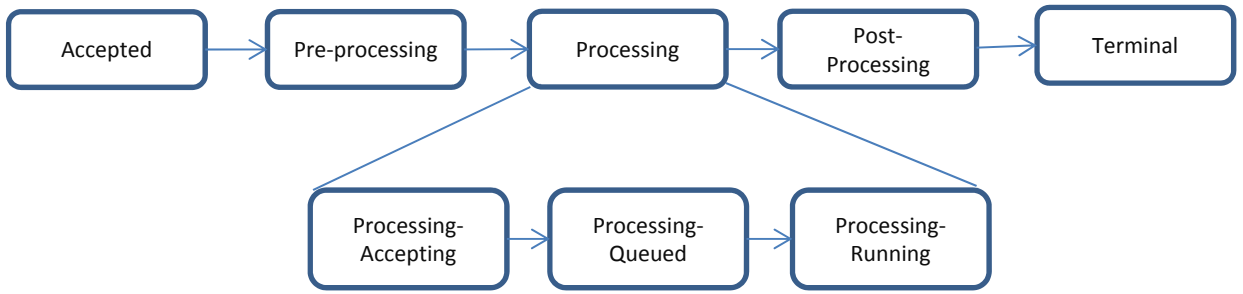


EMI INFSO-RI-261611

EMI-ES Job Description:ADL



Activity State Model – Optimal Flow

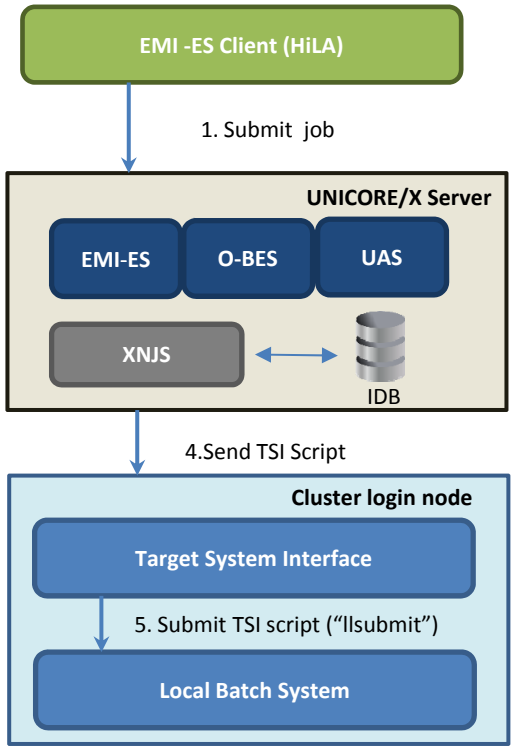


- State model is exposed to clients (not necessarily used by the implementation internally)
- Each state may be assigned multiple attributes
- Validating, Server-Paused, Provisioning, App-Failure,..

UNICORE Overview

- Integrated, complete Grid middleware stack including Graphical & command line 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 (PBS, Torque, LSF) and operating systems (Linux/Unix, Mac OS X, Windows)
- Strong support for applications and workflows

EMI-ES in UNICORE



2. Remote request received by EMI-ES (alternatively OGSA-BES, UAS)

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

OGSA-BES and JSDL

- OGSA-BES: a web service interface to manage and monitor Grid jobs
  - Create single activity
  - Get activities statuses
  - Terminate activities
- JSDL
  - Job request model
  - Job identification, Resources, Application, and Data staging elements

OGF Profiles

- Scope the use of multiple standards for a particular use case
- In OGSA-BES and JSDL
  - HPC-BP, HPC-FSP
  - JSDL-SPMD, JSDL-HPC

EMI INFSO-RI-261611

OGF JSDL Extensions

- Application
  - Pre & Post-Executable
  - RemoteLogging
  - WipeTime
  - Notification
- Resources
  - Runtime Environment
  - Parallel Environment

GLUE2 Extensions

- ComputingActivity
  - StageInDirectory
  - StageOutDirectory
  - SessionDirectory
  - ComputingActivityHistory
  - ComputingActivityProgress

EMI INFSO-RI-261611

EMI-ES and OGSA-BES

Approach	EMI-ES	OGSA-BES
Manage Activities	CreateActivities	CreateActivity (Single)
	Pause	
	Resume	
	Cancel	TerminateActivities (Vector)
	Notify	Use of WSN
	Wipe	
Monitor	ListActivities (return on ly Ids)	
	GetActivityStatus	GetActivityStatuses
	GetActivityInfo	GetActivityDocuments

EMI-ES and OGSA-BES

Approach	EMI-ES	OGSA-BES
Monitor Computing Service	GetResourceInfo	GetFactoryAttributes
	QueryResourceInfo	Not in the spec (only WSRF renderings)
	GetActivityInfo	GetActivityDocuments
Information Model	ResourceInfo (**Glue2**: ComputingManager, ComputingEndpoint, ExecutionEnvironment, ApplicationEnvironment)	FactoryAttributes (BES)
	ActivityInfo(**Glue2**:ComputeActivity)	ActivityDocument (JSDL)
Data Staging	Client Initiated Server Initiated	Server Initiated
	Stage-in /Session / Stage-out directories	Not distinguished

Conclusion

- Comes with suite of interfaces, activity state model, support of vector operations, integrated job description language
- Expose capabilities and state by associating GLUE2 entities
- EMI-ES supported by ARC, gLite, and UNICORE
- Job standards and profiles could be improved to support production infrastructure requirements via EMI-ES extensions