

# Interoperable Grid Scheduling

Ramin Yahyapour  
University of Dortmund, OGF Compute Area  
[ramin.yahyapour@udo.edu](mailto:ramin.yahyapour@udo.edu)

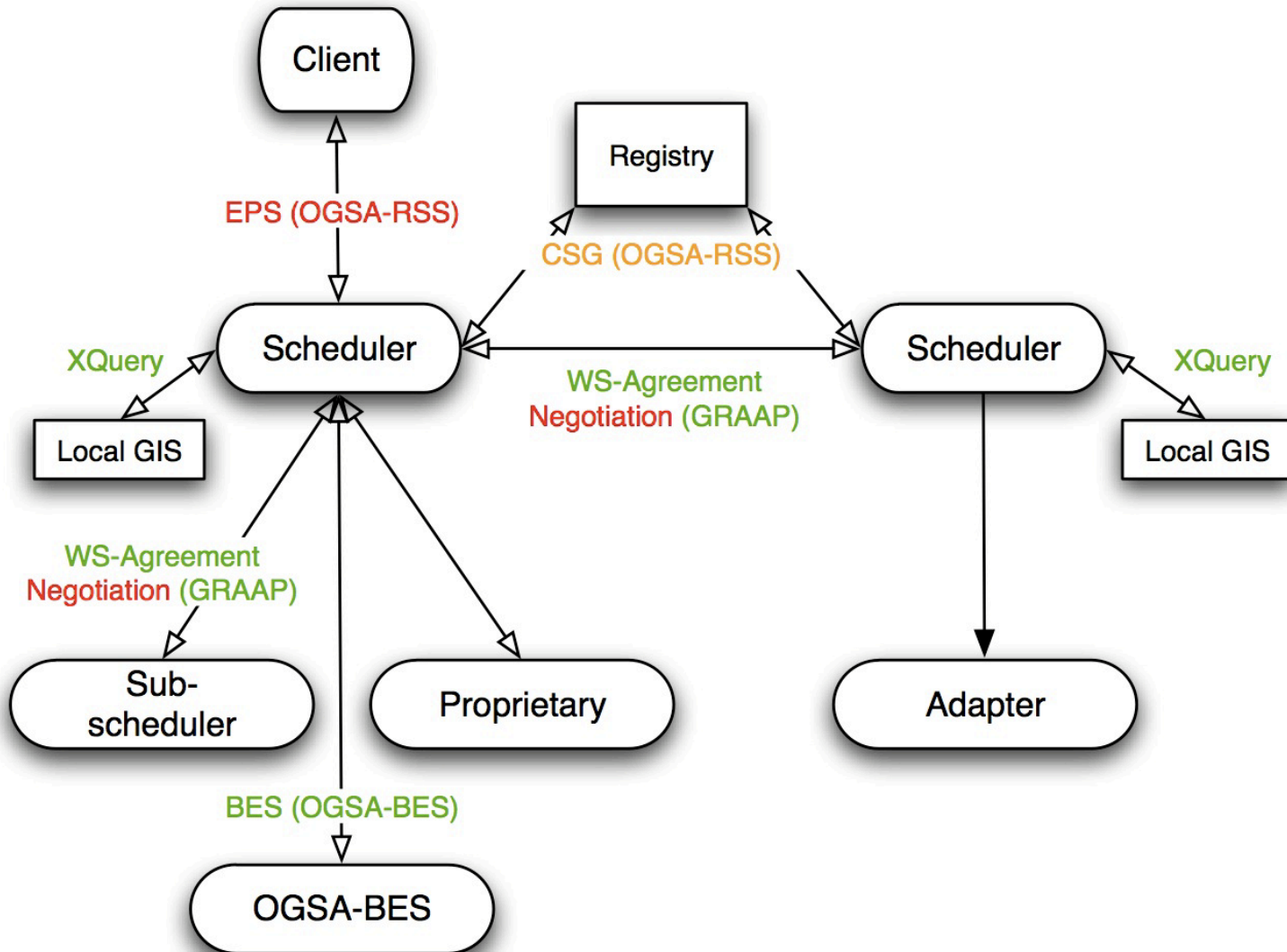
# Service vs Desktop Grids

- Service Grids
  - Guaranteed service quality
  - Limited number of management systems
  - Infrastructure is invisible
- Desktop Grids
  - Mostly best-effort execution
  - Limited QoS support
- Scheduling is needed in both cases
  - Different in scope, structure, algorithms
  - Interoperability provides higher flexibility
  - Standards support interoperability without the need for maintaining individual bridges

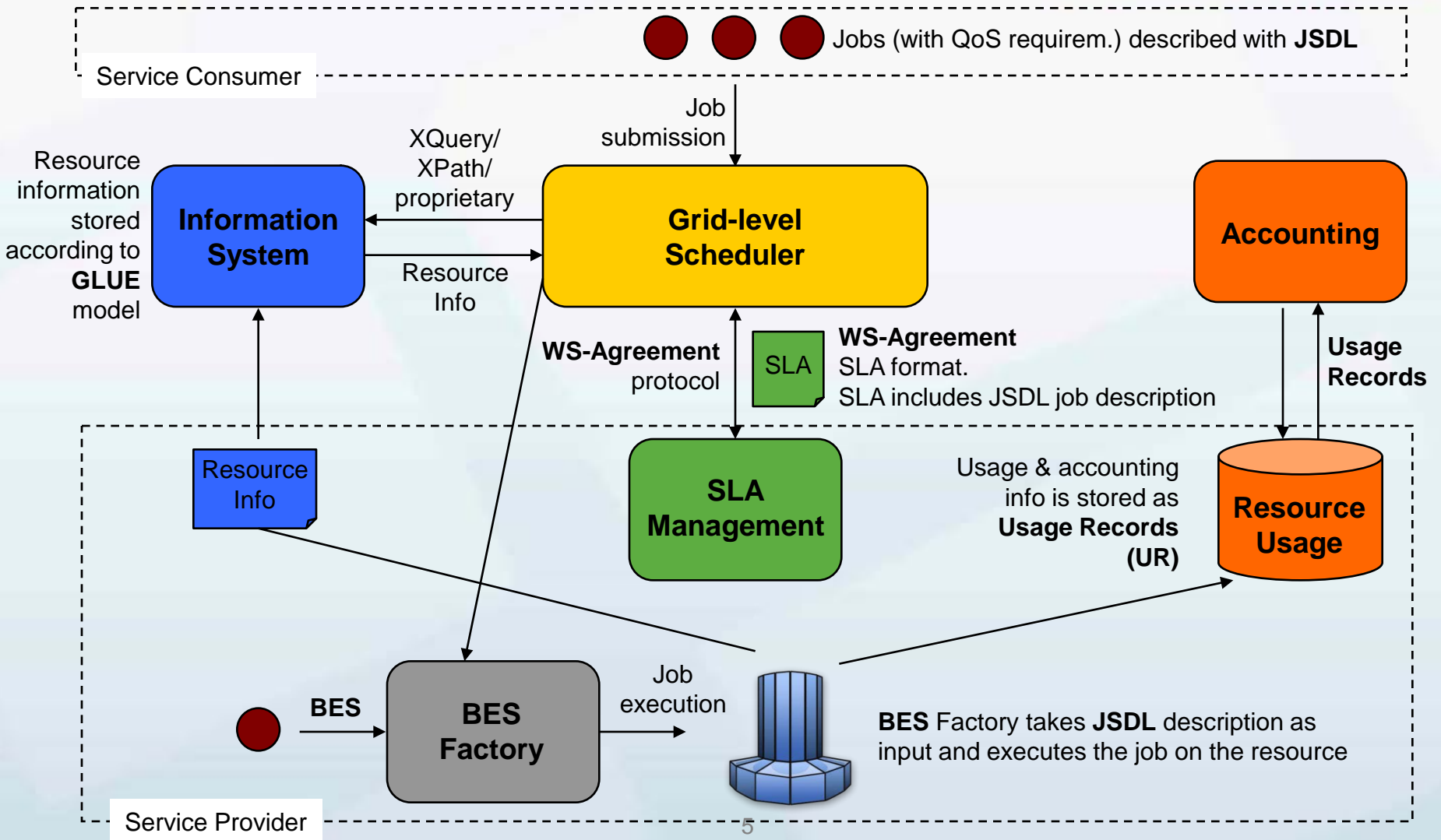
# Use-Case Overview

- Client submits a job to a grid-aware scheduling system
- The scheduling-system is responsible for detecting and selecting suitable resources, forwarding job to a Grid site
- SLA/QoS constraints are supported for job allocation
- Local resource management systems include local scheduling entites
- Available standards:
  - OGSA-BES, GLUE, JSDL, WS-Agreement, UR, RUS etc

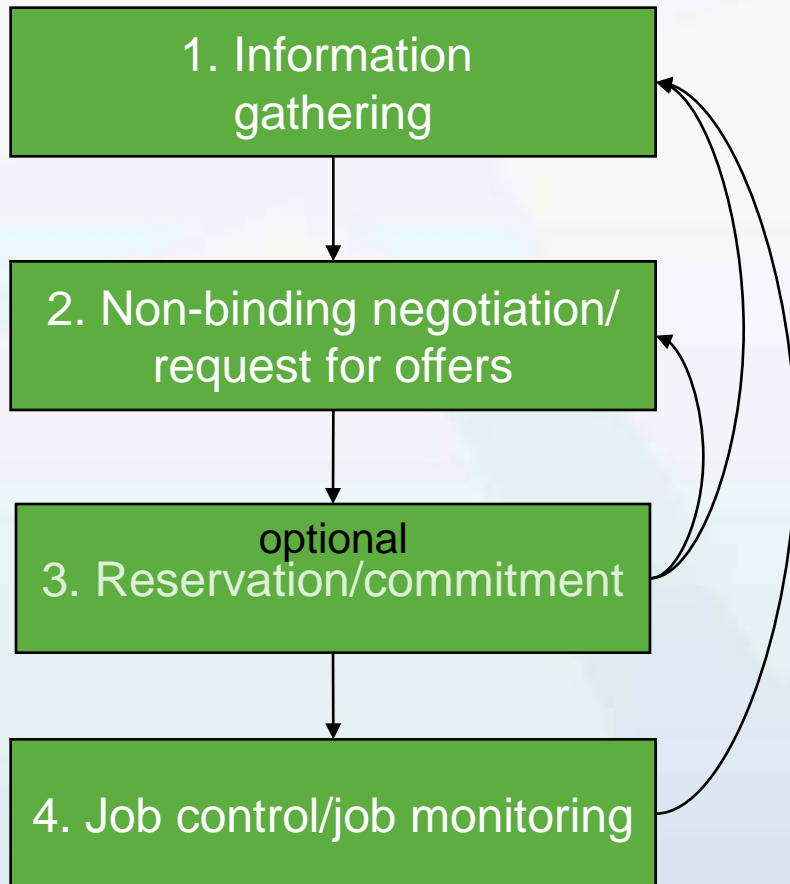
# Scheduler interoperation



# Use case overview

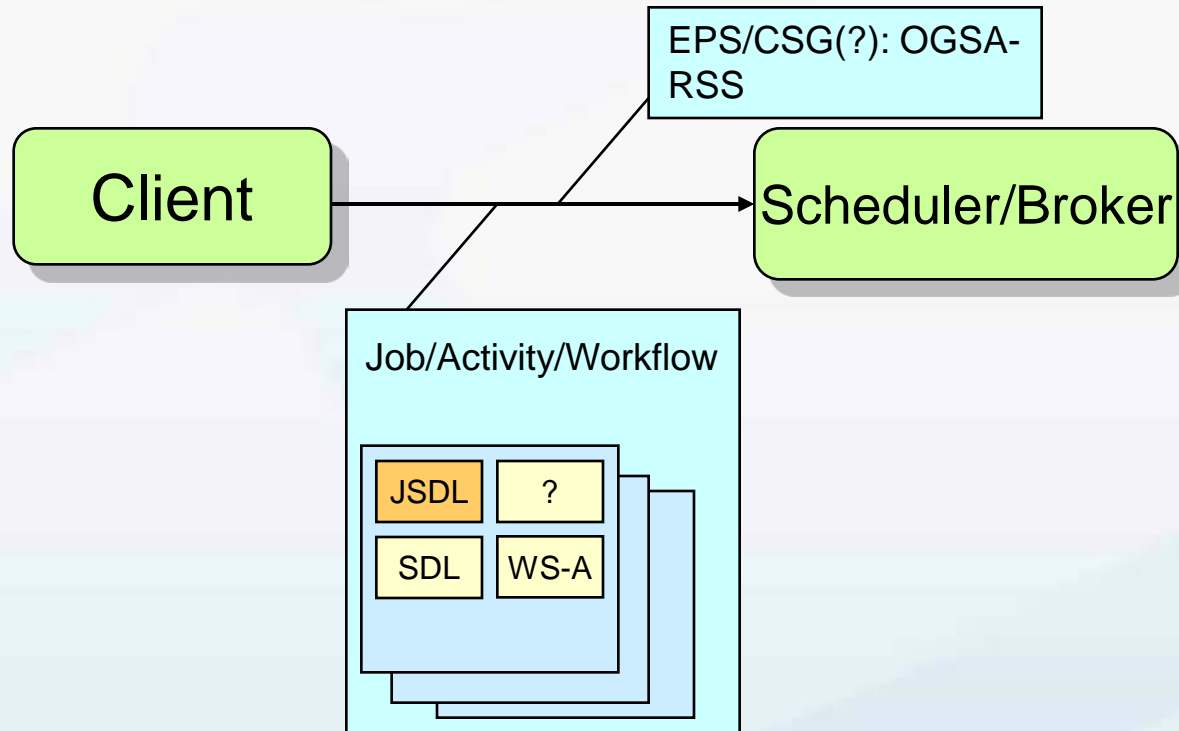


# Communication Stages



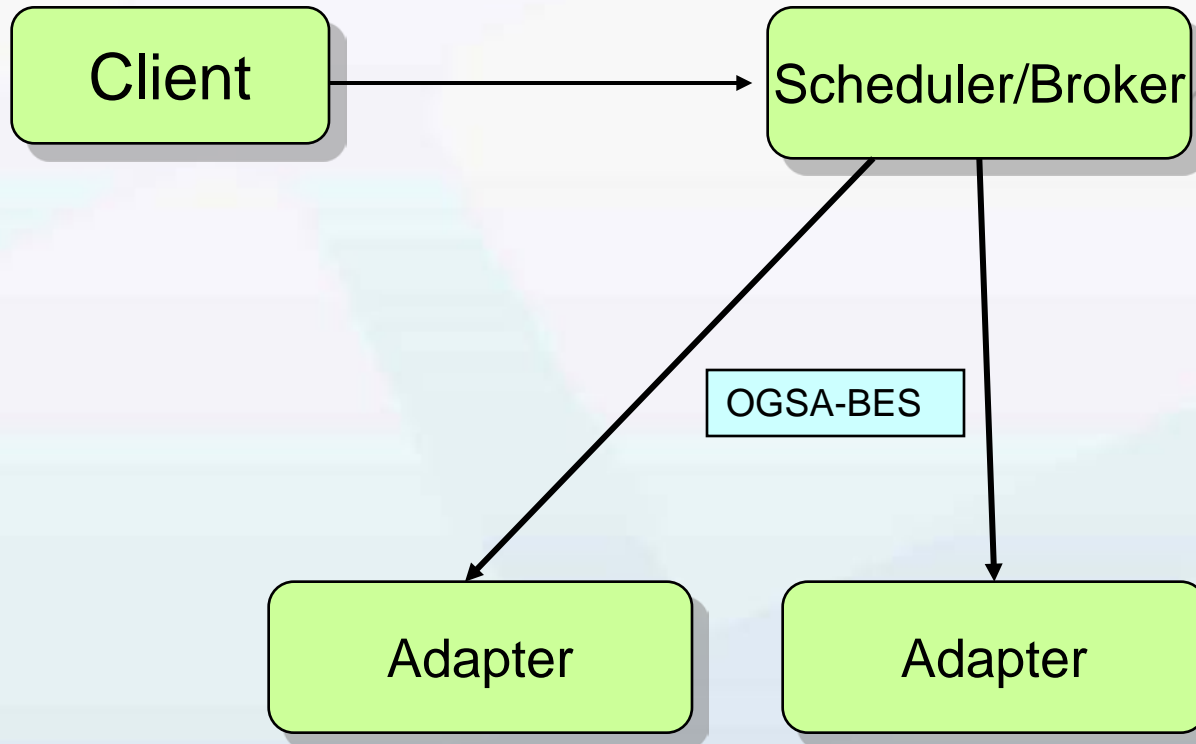
1. Information gathering about available remote Grid schedulers
2. Non-binding negotiation may end up with several possible agreement alternatives (possibly in parallel)
3. Agreement creation and commitment; may fail and require return to previous stages
4. Handing over job control to remote Grid scheduler (responsibility remains at initiator)

# Example Process including Scheduling and Execution



Step 1: activity/job/workflow needs to be submitted

# Example Process including Scheduling and Execution

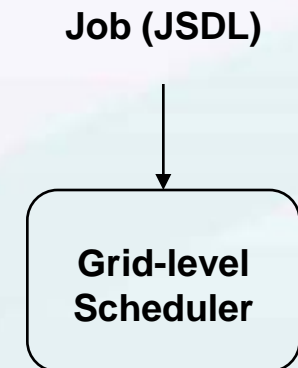


3a) Direct access to resources through BES or some other protocol

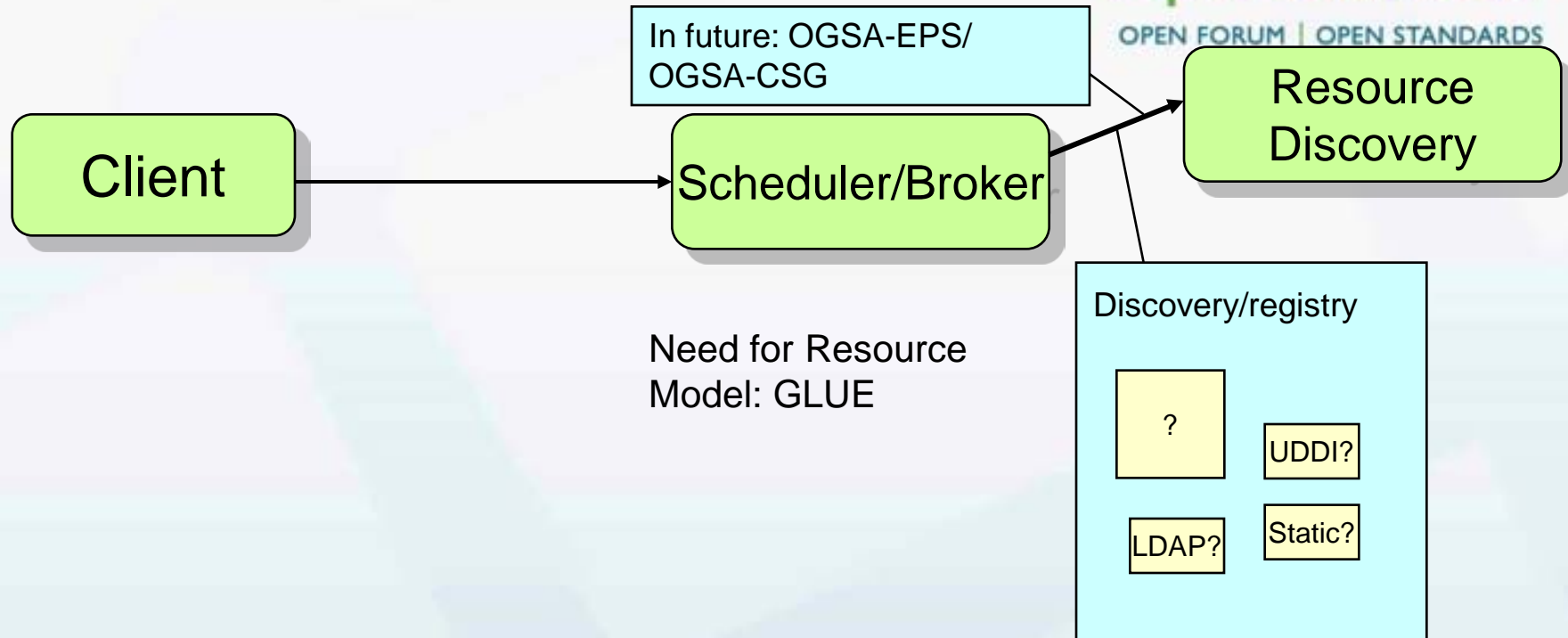


# Job Submission

- Job format: Job Submission Description Language (V 1.0; GFD.56/136;REC)
- No standard JSDL submission interface exists for Grid-level scheduler
- The scheduler could provide a BES interface



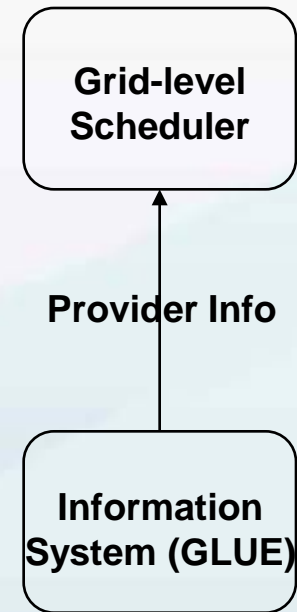
# Example Process including Scheduling and Execution



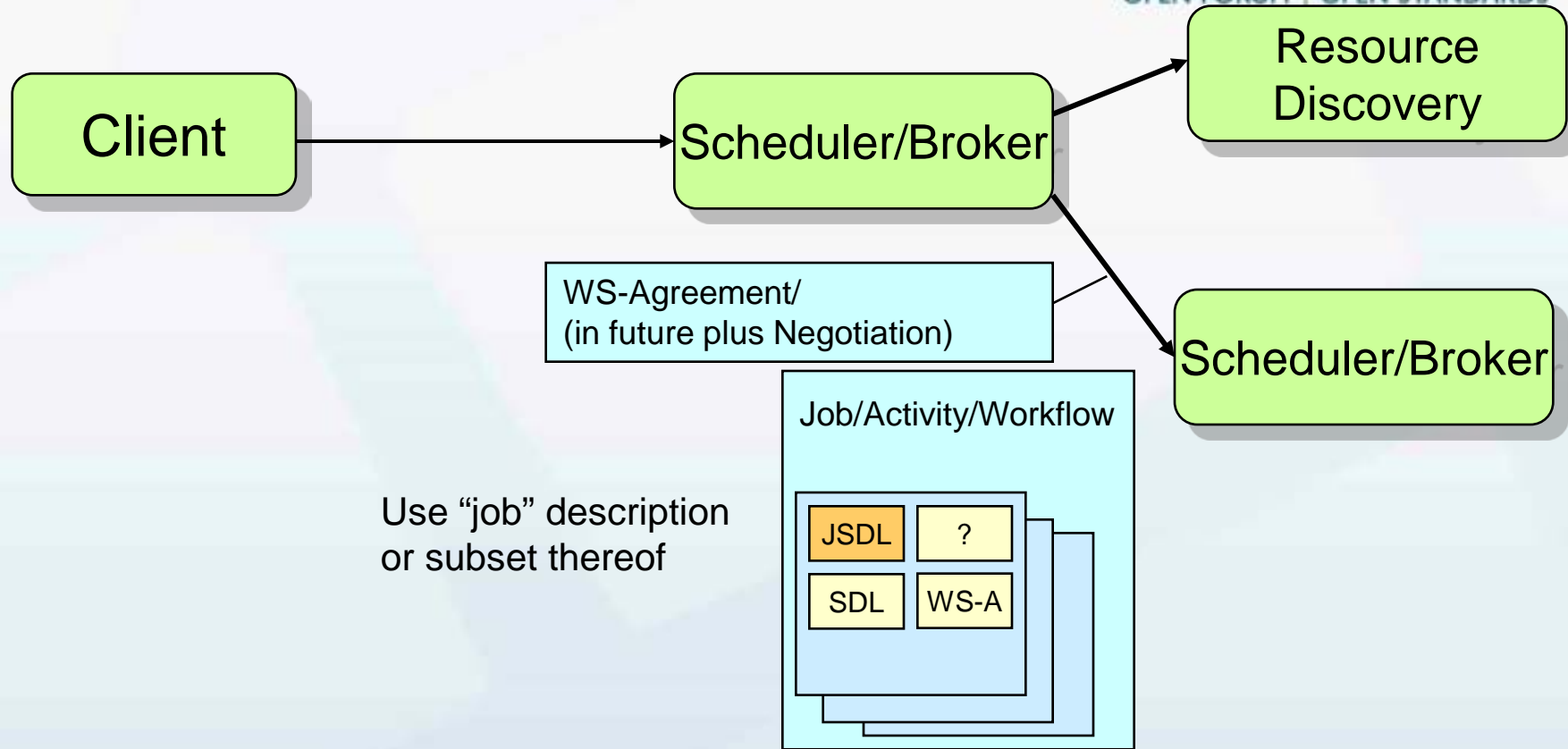
- 3b) Access resources through another broker/scheduler;
- 3b.1) identify available other scheduler/brokers through registry, statically, somehow...

# Information System

- Information model:  
GLUE (V 2.0; GFD-147, P-REC)
- Rendering: most likely  
XML Schema
- Information query:  
XQuery, XPath,  
proprietary



# Example Process including Scheduling and Execution

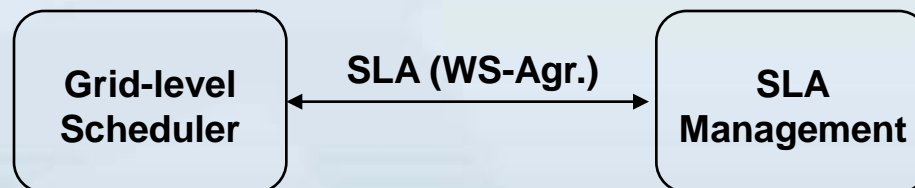


3b) Access resources through another broker/scheduler;

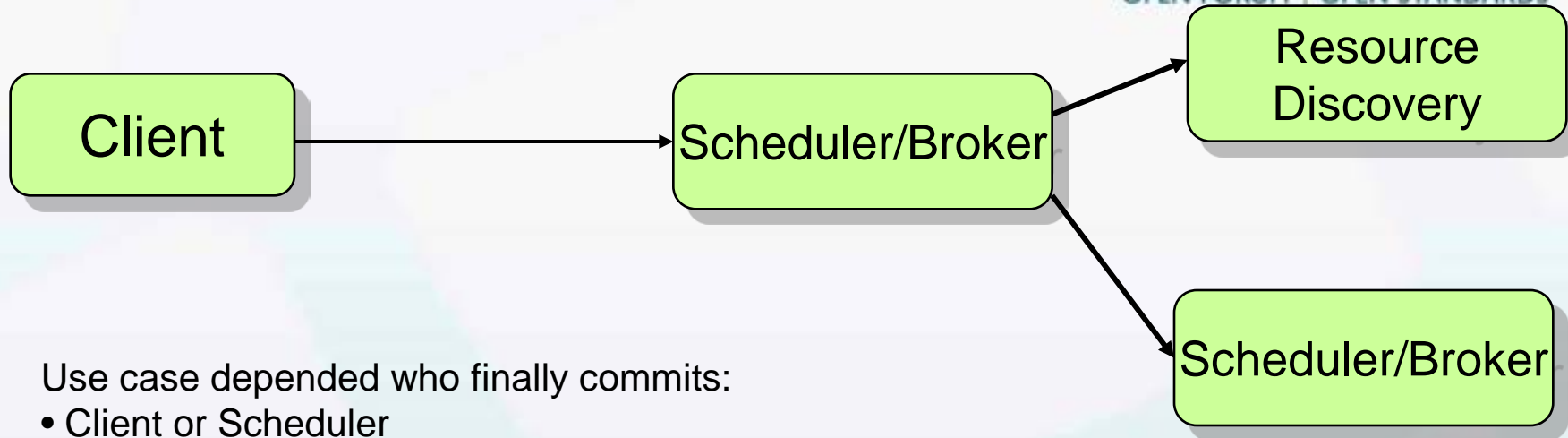
3b.2) communicate with remote scheduler about potential agreement

# Service Level Agreements

- Grid-level scheduler & Service Provider agree upon the service to be provided
- SLA is negotiated (contains JSDL)
- SLA format & protocol: WS-Agreement (V1.0; GFD.107; P-REC)
- Current negotiation is “take-it-or-leave-it”, but there is work on enhancements



# Example Process including Scheduling and Execution



Use case depended who finally commits:

- Client or Scheduler

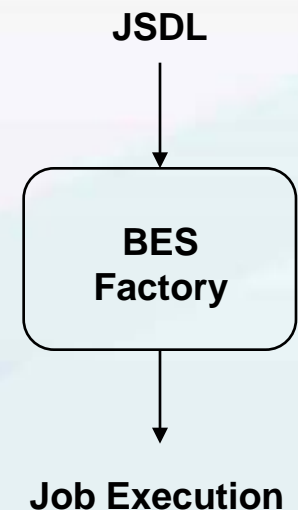
responsibility of execution management:

- Client or scheduler

4) Transition to execution management: OGSA-BES

# Job execution

- Via Basic Execution Service (V1.0; GFD.108; P-REC)
- BES-Factory receives JSDL job description
- BES defines all interfaces to create & manage “activities” (i.e. jobs)



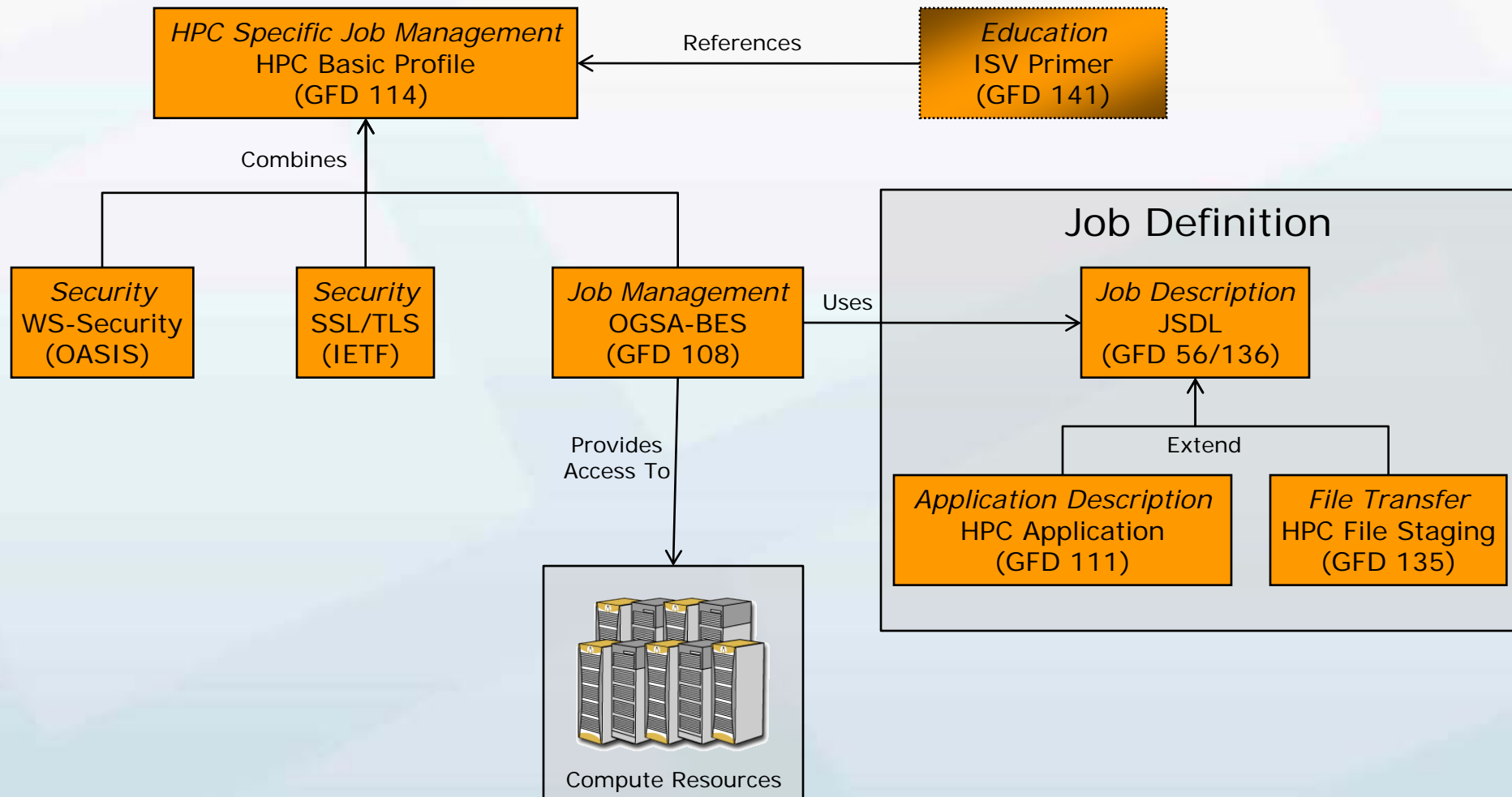
# Accounting

- Service provider stores accounting and usage data using Usage Records (V1.0; GFD.98; P-REC)
- Interface on how to retrieve URs is not yet defined (candidate by RUS-WG?)





# HPC Basic Profile

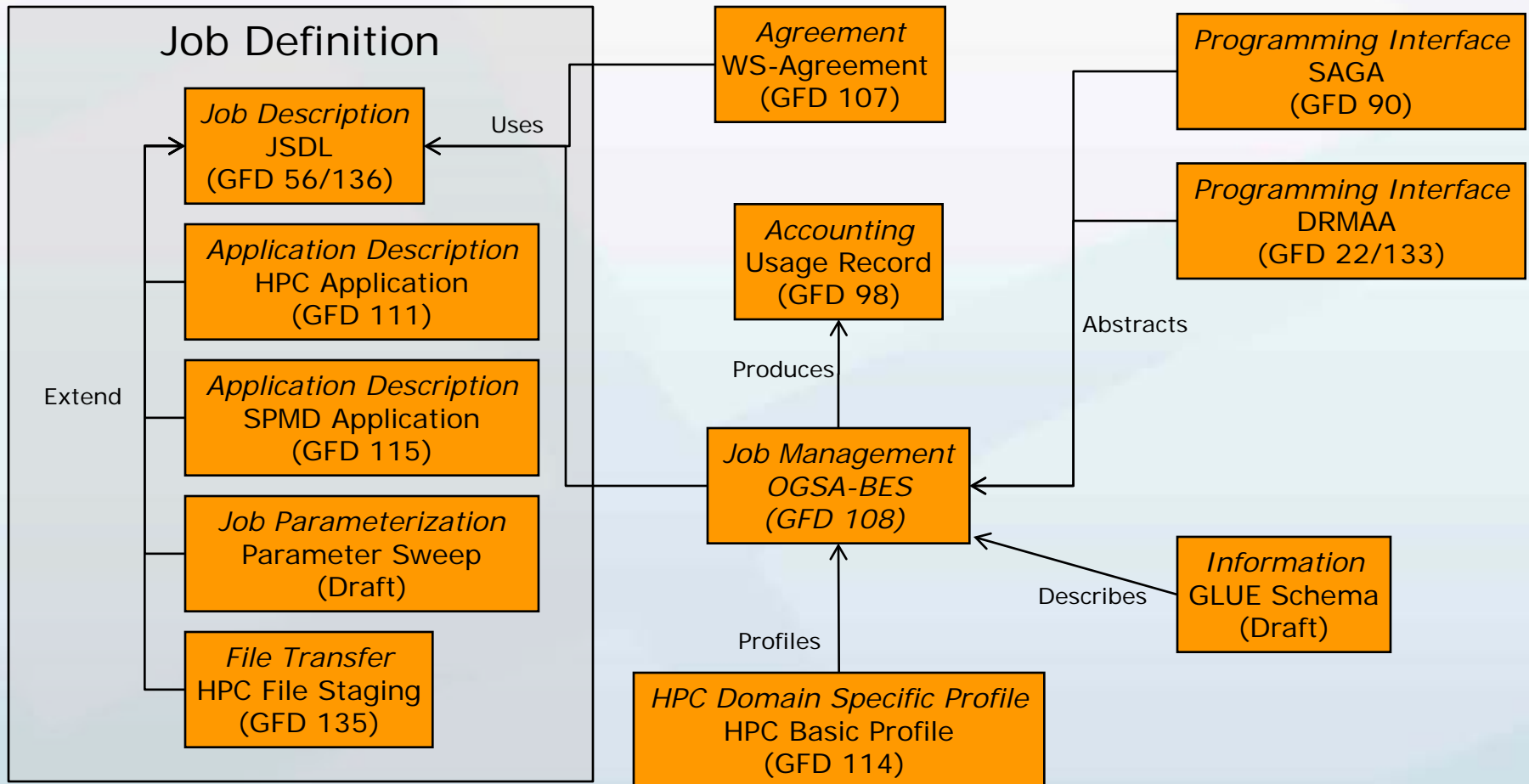


# Compute Area - Standards

*Architecture*  
OGSA EMS Scenarios  
(GFD 106)

*Use Cases*  
Grid Scheduling Use Cases  
(GFD 64)

*Education*  
ISV Primer  
(GFD 141)



# Summary



- There are several standards available:
  - facilitating Grid scheduling
  - Potential base for interoperability between Service and Desktop Grids
- The support of standards limits the dependence on individual interfaces
  - Less effort for maintaining different bridges
- JSDL/HPCB are good starting points
  - PGI-WG addresses some of the basic interoperability issues for production environments