



GlobalGridForum

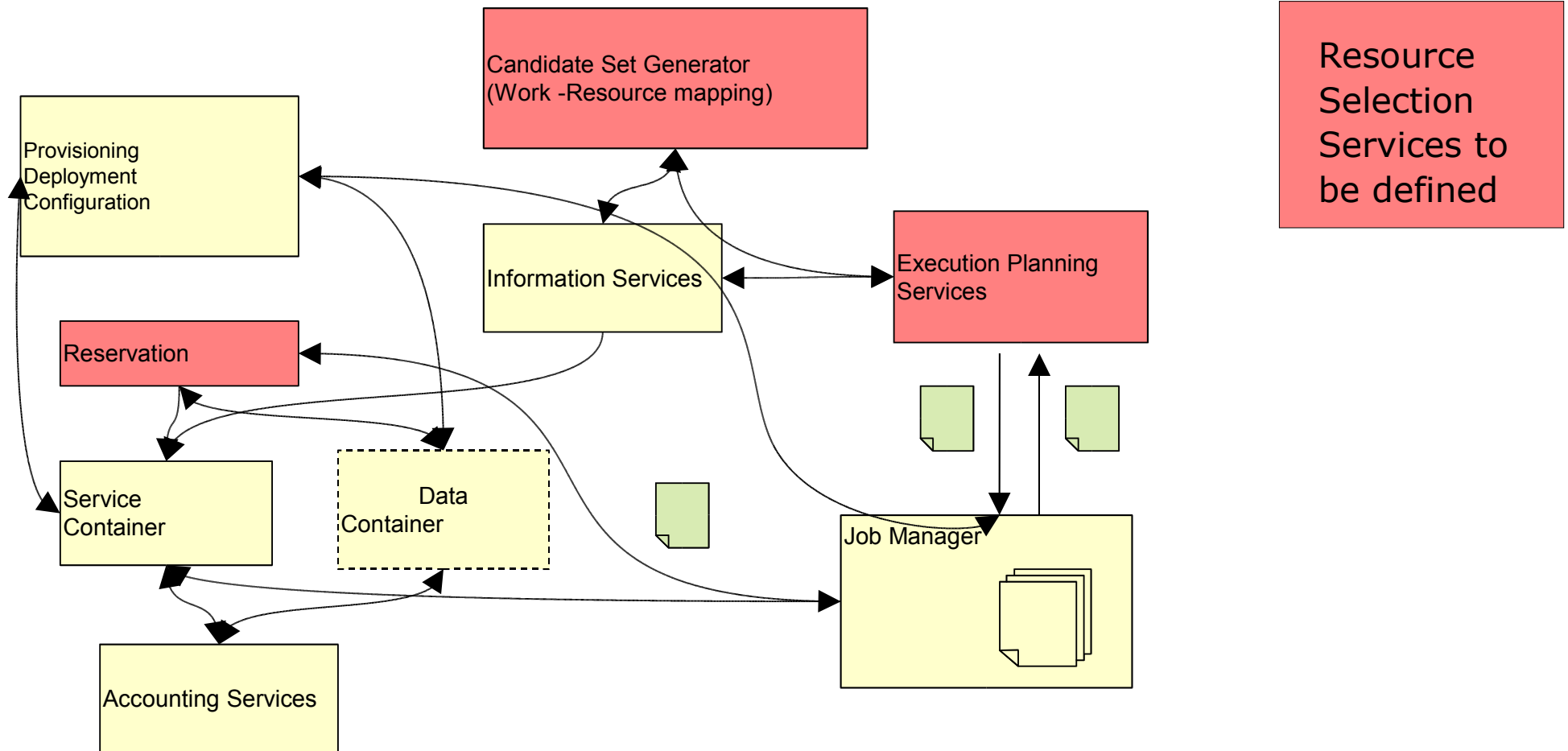
Leading the pervasive adoption of grid computing for research and industry

OGSA Resource Selection Services Charter Discussion

OGSA F2F, May 22 – 26, London

Mathias Dalheimer, dalheimer@itwm.fhg.de

What are the services OGSA-RSS wants to define?



Related groups in GGF

- OGSA of course, especially OGSA-EMS
Andrew Grimshaw will act as liaison
- Grid Scheduling Architecture-RG
See following transparencies (Ramin Yahyapour/Philipp Wieder)
- Grid Resource Allocation Agreement Protocol-WG
“... addresses the protocol between a Super-Scheduler (Grid Level Scheduler) and local Schedulers necessary to reserve and allocate resources in the Grid...” (from WG's website)
Philipp Wieder acts as liaison
- JSDL-WG
Donal Fellows will act as a consultant when needed.

GSA-RG: Scope and objectives

- Scope
 - Grid Scheduling Architecture
 - Service and usage pattern level, not interface and protocol level
 - Arbitrary resources: from compute and data to network and currently unknown resource types
 - Advance reservation, co-allocation, workflow management, policy integration, ...
- Objectives
 - Gather Grid scheduling use cases & usage patterns
 - Specify functional service requirements
 - Specify service interactions & high-level workflows

Contributions to RSS

- Documents
 - Grid Scheduling Use Cases (to be submitted before GGF14)
 - Grid Scheduling Architecture – Requirements (working draft)
- Expertise
 - GSA work driven by EU projects like CoreGRID, NextGRID, UniGrids
- Roles in RSS
 - R. Yahyapour: liaison, editor, secretary (if necessary)
 - Ph. Wieder: liaison, editor

More information at <https://forge.gridforum.org/projects/gsa-rg>

Related Projects

So far, four projects have committed themselves to adopt the work

- The NAREGI SuperScheduler project (Soonwook Hwang / Kazushige Saga)
- The Unicore/GS Broker (being developed in the UniGrids project) (Donal Fellows)
- The MyGrid Broker of the OurGrid project (Walfredo Cirne)
- Fraunhofer ITWM's Calana Scheduler (Mathias Dalheimer)

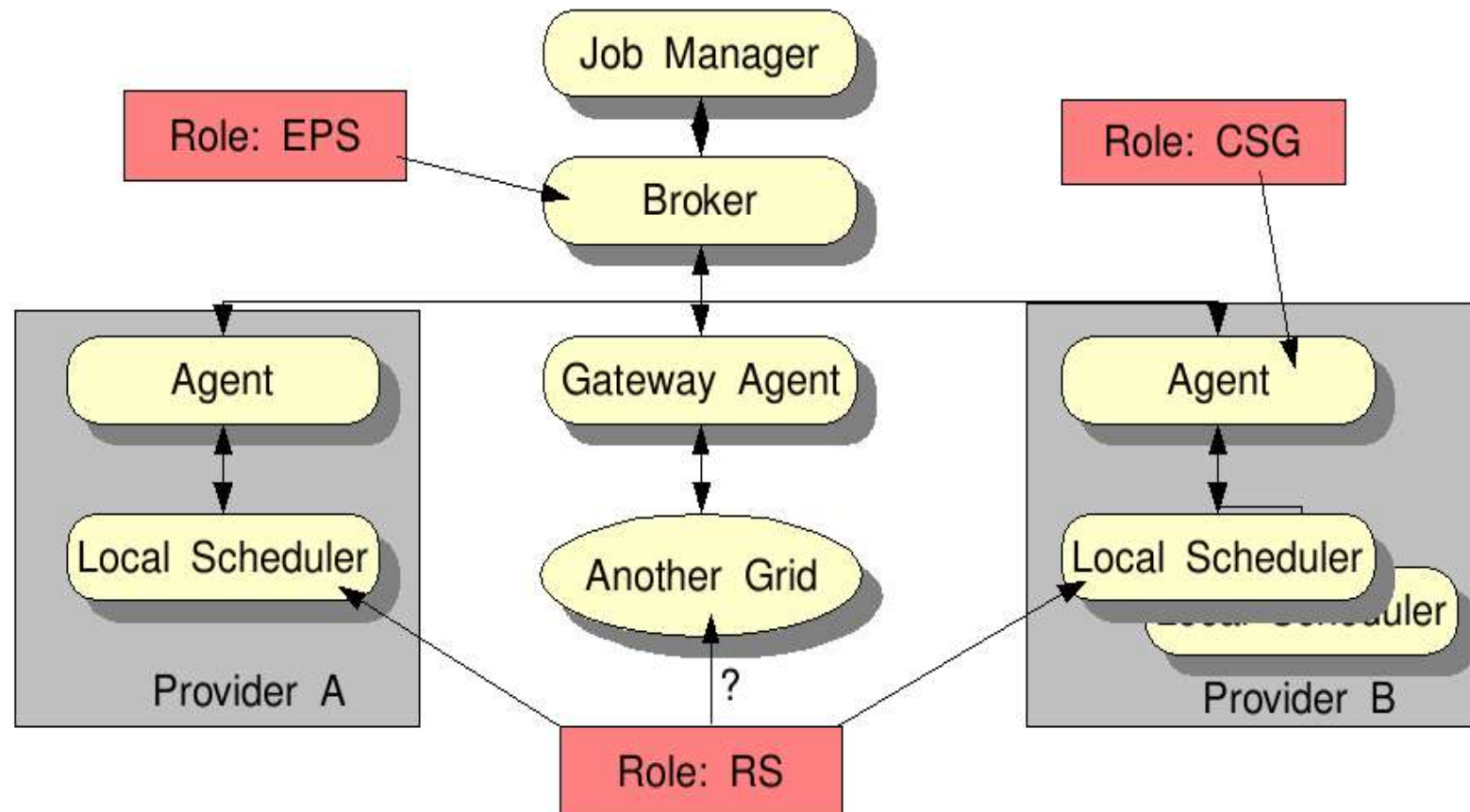
In addition, the GSA-RG has identified typical use-cases for grid schedulers which will be used to start the discussion.

On the following transparencies: How do the projects above match the OGSA?

Calana: An agent-based grid scheduler

- We use multi-criteria auctions to allocate jobs. A bid consists of a price and a estimated finish time
- Agents bid on the execution of jobs issued by a central and trustworthy broker
- When bidding, an agent has to ensure that the job can run as stated in the bid.
 - We need a Reservation Service
- No information system is needed:
 - Only the broker must be announced. Agents subscribe to broker(s) and participate in the bidding, according to local policies.
 - Since the agents can directly interact with their resources, they do not rely on an information system for dynamic information.

Calana: Architecture



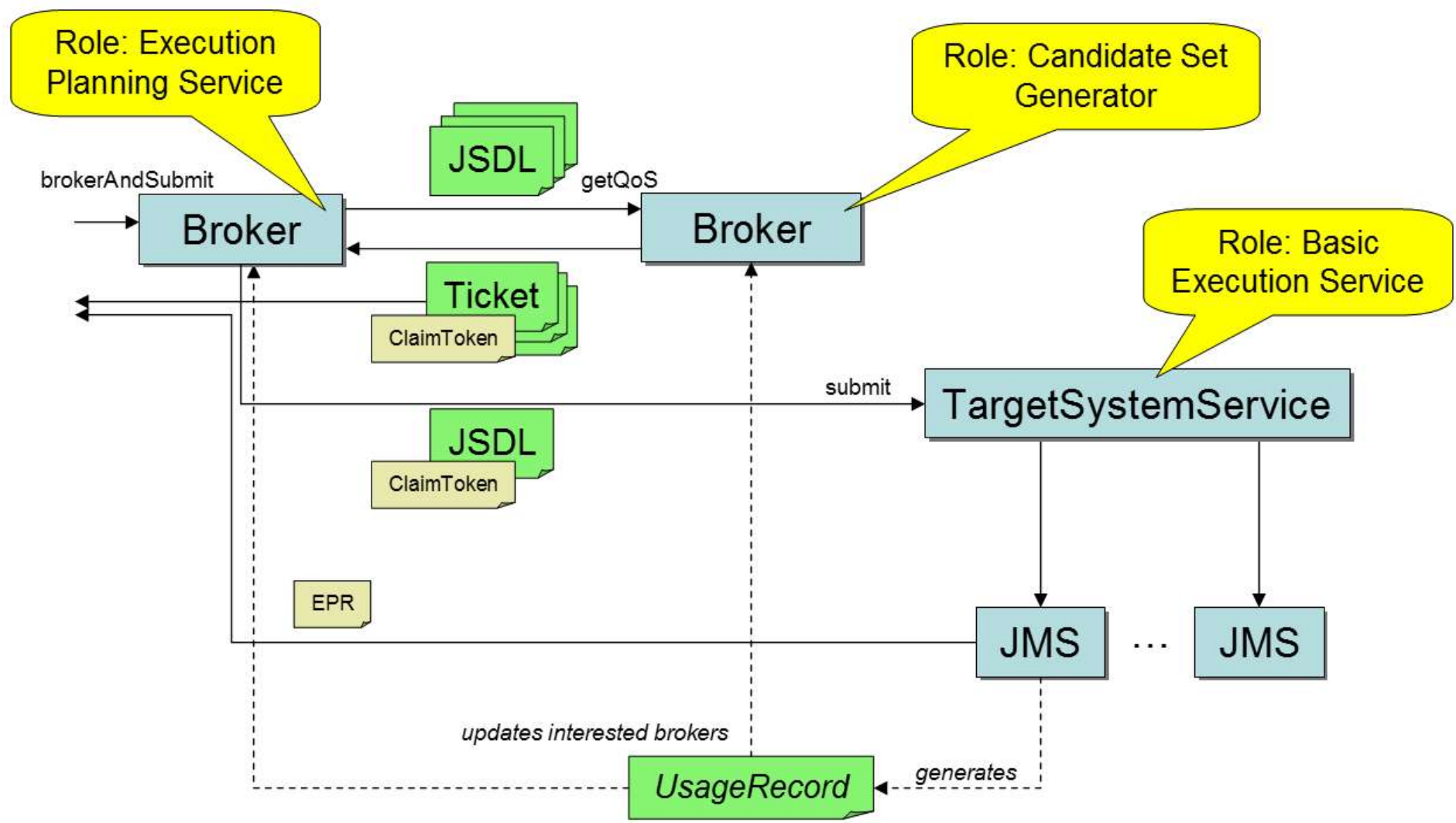
- OurGrid is a peer-to-peer grid
- Peers can freely join the system without any human intervention
- To keep it doable, we focus on Bag-of-Tasks application
 - Parameter Sweep, Data Mining,
- uses **MyGrid**: A broker that schedules BoT applications

- MyGrid could implement CSG and EPS
- We don't have use for a RS as now.
 - Replication-based scheduling approach to avoid needing guarantees from the underlying system.
- If there are RS available, we could certainly evolve MyGrid for it to provide stronger guarantees to its clients.

The Unicore/GS Broker

- **Being Developed by UniGrids project**
 - Based on EUROGRID Resource Broker
- **Computational Job Brokering**
 - Built over Basic Job Execution Service Layer
- **Designed to Leverage VO Structure**
- **Multiple Application Domains**
 - Bioinformatics
 - Geology
 - Meteorology
 - Engineering
- **Supports Multiple Use Styles**
 - Interactive Use
 - Distributed Brokering
 - Parameter Space Studies

Unicore/GS Broker Architecture



Unicore/GS Broker Operations

- GetQoS (*Get Execution Candidates from CSG*)
 - Input: JSDL Document Set
 - Handle multiple JSDL docs in one request for performance reasons
 - May have application-specific resources
 - Output: Ticket array
 - TargetSystem EPR
 - Issuer EPR
 - Cost Information
 - Offer Validity Period
 - Claim Token (allows actual claim of offer)

Unicore/GS Broker Operations #2

- **BrokerAndSubmit (*Create Job through EPS*)**
 - Input: JSDL Document & Offer Selector
 - Handed off to GetQoS initially
 - Offer Selector controls choice between tickets
 - Claim Token will be attached to JSDL as resource before submission
 - Output: Job EPR, Selected Ticket & Selector Decision Log
 - Caller uses Job EPR to manage job
 - Selected Ticket describes what offer was made
 - Decision Log provides info about why choice was made
- **AcceptUsageRecord**
 - To allow for feedback of usage information

NAREGI Super Scheduler (SS)

- **Introduction**

- NAREGI is a grid middleware project aiming at developing computational infrastructure for scientific and engineering research
- SS aims at providing resource discovery and brokering service
 - e.g., Co-allocation of MPI jobs across multiple sites
- SS is designed based on the OGSA-EMS architecture
 - JM, EPS, CSG and RS

- **Support the RSS-WG**

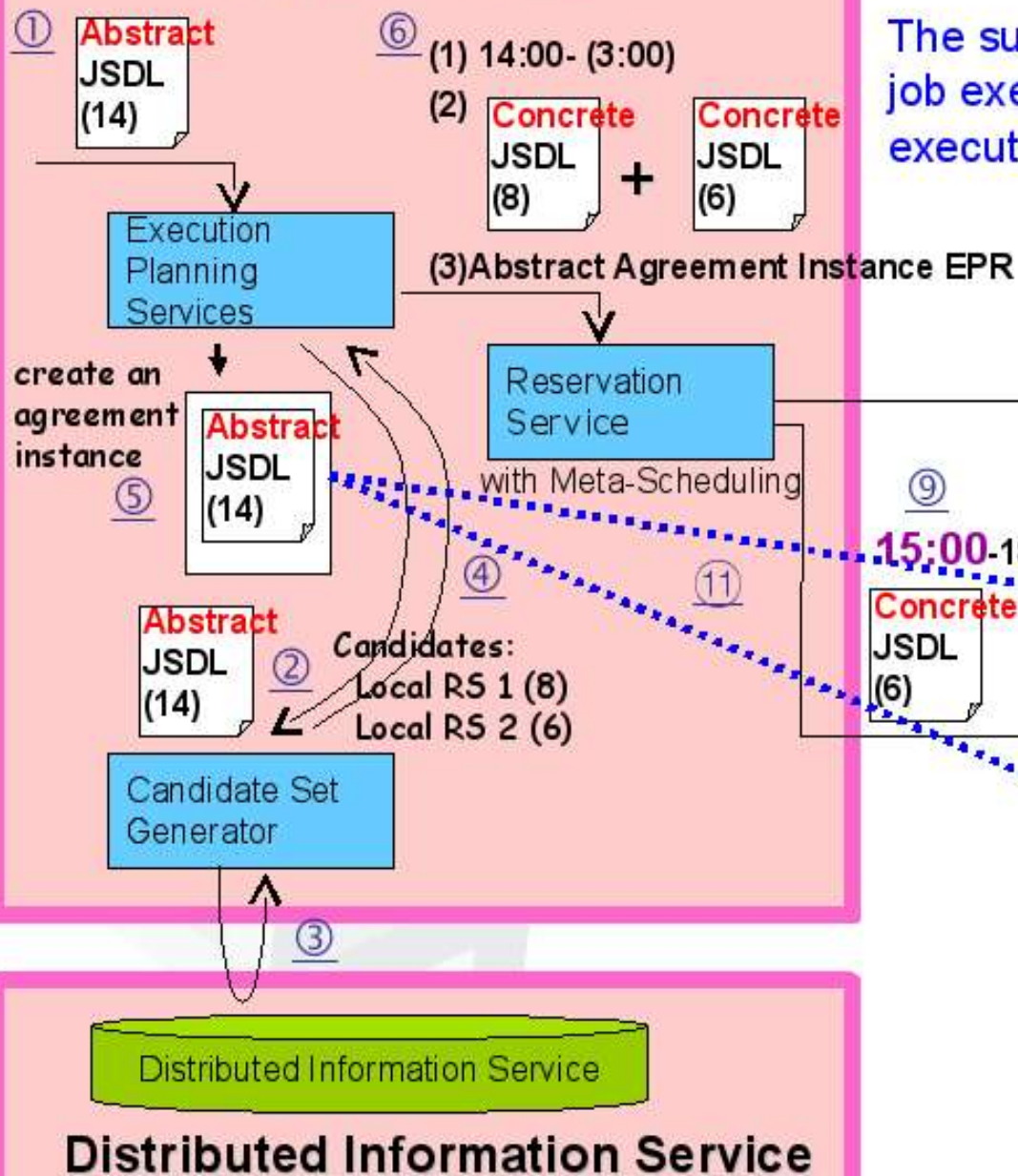
- Sharing Ideas with the RSS-WG
- NAREGI will implement the RSS specification, and distribute it as open software

- **Special Issue**

- NAREGI needs the Reservation Service (RS) in the RSS to support for NAREGI's typical scenarios of running MPI jobs across multiple sites

NAREGI's RSS Interaction Scenario

Super Scheduler



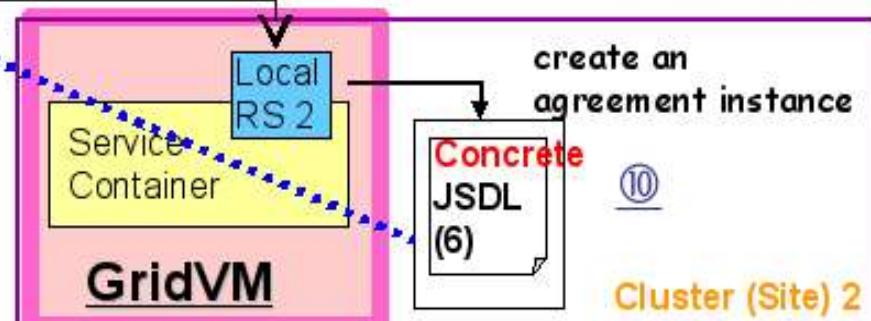
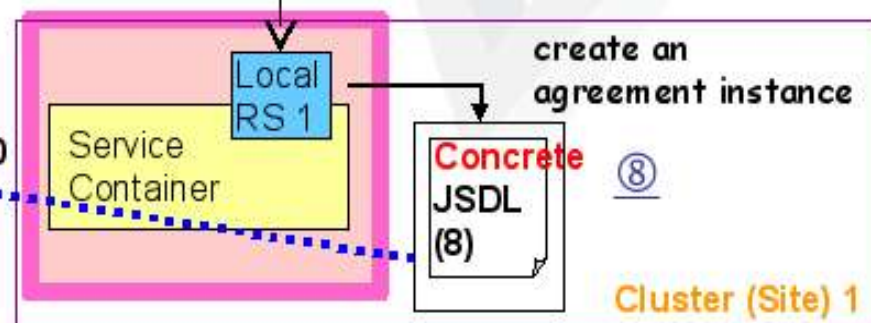
Meta computing scheduler is required to allocate and to execute jobs on multiple sites **simultaneously**.

The super scheduler negotiates with local RSs about job execution time and reserves resources which can execute the jobs simultaneously.

⑦ 15:00-18:00

Concrete JSDL (8)

⑨ 15:00-18:00
Concrete JSDL (6)



Local RS #:

Local Reservation Service #

OGSA-RSS charter summary

- The group will define interfaces and protocols for Resource Selection Services, namely
 - Execution Planing Service (EPS),
 - Candidate Set Generator (CSG), and
 - Reservation Services (RS)
- Provide feedback to the OGSA-WG
- Use general scheduling models from GSA-RG
- Deliverables:
 - D1: A spec doc describing the EPS and CSG protocol
 - D2: A spec doc describing the Reservation Service
here: focus on a basic reservation service

Reservation Service

- On one hand, a Reservation Service is needed by most schedulers
- On the other hand, a complete and extensive Reservation Service is out of scope
- Therefore:
 - Start with a “Basic Reservation Service” to satisfy the requirements
 - Placeholder for future Reservation Service
 - Provide input to a future GGF group

Timeline

- By now, we have defined our charter and contacted other working groups
- GGF14: BoF and start work on D1
- GGF15: Kickoff, Service description Milestone, discussion of D1
- GGF16: First draft of D1, discussion of D2
- GGF17: Revised draft of D1, first draft of D2
- GGF18: D1 in public comment, revised version of D2