NAREGI Middleware
(National Research Grid Initiative)

March 2005
NAREGI Middleware Objectives

Constructs Server Grid, realizing Meta-computing.

- Allow users to execute complex jobs with various interactions on resources across multiple sites on the Grid.
- Stable set of middleware to allow scalable and sustainable operations of centers as resource and VO hosting providers.
Coupled Simulation & NAREGI M/W Flow

- **User Registration**: a
- **Deployment**: b
- **Edit**: c

1. **Submission**
2. **Resource discovery**
3. **Negotiation**
4. **Reservation**
5. **IMPI starts**
6. **MPI job starts**
7. **MPI init.**
8. **Visualization**
9. **Accounting**

**GridMPI**

- **Site A** (SMP machine)
- **Site B** (PC cluster)
- **Site C** (PC cluster)

**Different (sub) Jobs**

- **Solvent sim.**
- **Solute sim.**

**Application requirement definition**

**... Multi-Physics simulation (MPI).**
NAREGI Middleware and OGSA-EMS

WP1: UNICORE NJS (as a WF engine)

- SuperScheduler
  - Execution Planning Services
  - Candidate Set Generator
  - Discover & Select
  - Reserve

WP2: WP1
- Deployment
- Register / Query

WP3: PSE
- Application Contents Service (ACS)
- Candidate Set Generator
- Resource Usage Services
- Information Services

Distributed Information Services
- GridVM
- Service Container

WP3: PSE
- GridVM
- Service Container
- Execution Planning Services
- Reservation

UNICORE NJS (as a security framework)
- Deployment
- Resource Usage Services
- Information Services

WP3: PSE
- Register
- Update
- Query
NAREGI WP1 Software stack and standards

Super Scheduler, GridVM, and Distributed Information Service consist the core of NAREGI middleware. We developed them as a set of interacting web services following the OGSA architectural document V1, and used many standards for interoperability, in particular WS-Agreement and JSDL.

1. WS-Agreement
   • To make resource provisioning be scalable, we believe that bilateral resource matching is the best way. We are using WS-Agreement for this reason.
   • Agreement based brokering is more secure than non-agreement based one, because the jobs, which do not match to local policy of site, will be rejected. This keeps the local resources safe, and autonomy of the site.

2. JSDL
   • For resource matching, a meta descriptive framework, which is portable, extensible, and enables unified description of job submission and resources, and it's standardized schema are needed. Currently, we employ JSDL because it is the only one available.

We also use WS-Notification, OGSA-DAI, and other standards.