

GGF9 OGSA-WG Program Execution (PE) Session

Oct. 07 2003, 12:00-1:30 Sheraton 4

Attendees: 66

Minutes: Ravi Subramaniam, Andreas Savva

PE subgroup update (Ming Xu)

- Setup recently (after Sep F2F)
- Goal: Allow transparent access to multiple resource infrastructures (managed by different RMS)
- WS-Agreement based
 - o OGSi not enough: need common way to submit/negotiate
- CSF
- Hierarchical VO
 - o upper layer does not know hierarchy
 - o negotiation between layers is via WS-Agreement
- Workload management generalization - Ravi.

Policy & agreement (Asit Dan)

- Support business decisions
 - o who is important and who is not; if we oversold which agreement to violate.
 - o provider's view
- Policy
 - o PEP to resource: push policy down to resource, this is specific to resource so the PEP is expected to know how to do this.
- Policy deployment
 - o Policy service agent
- When policy gets down to a policy enforcement then the interface between the last entity that knows about policy and the resource that is driven by the policy.
- Need to factor those entities that are common like agreement and negotiations and the policy framework.

GRAAP-WG (Jim Pruyne)

Requirements:

- WS-Agreement is a service management framework (what are the properties and behaviors and the description on how the services will interact)
- Generic extensible negotiation model

- Need to monitor and check agreements to monitor the agreement to make sure that the agreements are being delivered.
- Domain independence (try to apply outside of program execution)

Agreement

- Initiator and providers (accepts the request). Two parties come into agreement and the execution of the agreement may be done by other parties. Can create symmetry in the relationship if there is a two way commitment to the agreement.
- All agreements are based on domain specific terms based. Terms have state information with two dimensions (where are we in the negotiation: have we understood the terms and have reached an agreement and second dim. is where are we in the process and is there a chance of modifying the negotiations)
 - o Negotiation is the process of modifying term states.
- The former is maintained by state and is a key part of the standard here.
 - o Agreement service state represents agreement state (contains terms of agreement)

WS-Agreement architecture

- Agreements agents are added to the current roles and services that are the consumers and resources.
- The consumer talks to the agreement initiator which talks to a agreement factory. Can have further discussion with other agents to get agreement that need to tie into a higher level agreement.
- Have proposed services and portTypes. Factory model based on ogasi. Terms to be interpreted and have coarse agreement on whether.
- Provides a foundation of service management. Expect other efforts to tie in and/or leverage (for example: GESA and JSDL)

Ravi's Questions:

- Is there a trust model here? How do we know that the agreement is not a lie?
 - o No trust model in here yet. This issue has been raised before and is under investigation.
- Is there a concept of a dictionary? Can there be a context created by the dictionary?
 - o Porttype right now in where one can query to find the terms supported. Working on expanding this to a richer schema of terms.
 - o One level of negotiation may be on what terms (language) to use.

JDSL (Ali Anjomshoaa)

Job submission description language

- Talk based on concepts to help the OGSA effort.
- Goals: To provide an architecture independent job description language; allow job management in a grid environment and allow interoperability with job entities.

- JSDL applicable to all grid frameworks; not just OGSA.
- Have an abstract concept of a job.

Job management in Grids

- Expects that the job management is different from resource management
- job submission, negotiation, queuing, execution, job dependency, fault and recovery, job data management, job checkpointing and migration, job monitoring, control and lifetime, job information archiving, accounting (uses service and resources to deliver this)

Job description document (JDD)

- captures abstract information on the job and other protocols
- describes job over its entire lifetime
- provides information on lifetime of job, provenance of the job
- JDD combines JSDL and other items into the JDD (e.g. negotiation, agreement, security etc)

Job attributes

- identity, resources required, environment attributes, data attributes, scheduling attributes, security attributes.

Context with OGSA

- JME can be implemented as OGSA services
- a service or set of services
- JDD can be represented as a grid service with a suitable set of SDEs and defined interfaces.
 - o JDD has associated portTypes to create/manage/manipulate the document.

Ravi questions:

- o Is the job a static object or an executing entity with a manageability environment?
 - o JDD is a Grid Service (has a GSH)
- o Why is a schema required (esp. normative schema)?
- o Why is job different from a resource? E.g., a job might provide a resource for some other entity.
 - o No difference if use term 'manageable entity' instead of resource.
- o Why are services and resource different?
- o Why cannot the JDD be similar to a term dictionary and submission be an agreement and dispatch?
 - o JDD covers more than agreement. Agreement covers lifetime of job; what happens postmortem.

DRMAA (John and Harbri)

- Provide a single interface to a set of DRM systems.
- Submit, control & monitor and query status of jobs.
- New specs are being worked on (C binding, java binding, (G)WSDL binding)
- Would like to work with OGSA for GWSDL.

Would like to see the DRMAA interface join seamlessly into driving the OGSA?

- Possible interactions
 - o DRMAA i/f to ogsa service (on soa)
 - o DRMAA ogsci client mapping document or drmaa for ogsa document.

Provide portability across distributed computing applications.

Support products like GridIron and provides an interface to take a run on single machine and migrate to a DRM solution.

Help companies to use their current DRM solutions to be leveraged across the DRM solutions.

GESA (John MacLaren)

Motivation:

- allow people to charge for grid services
- Accounting and usage information.
- Mechanism to provide pricing schemes.
- also working on payment systems.

Identified negotiation and security (beyond delegated credentials) as key requirements for models to work. Have set up a special GESA security session for GGF9 to explore issues. Security experts from companies(?) participating. Others are welcome.

Looking at Negotiation to negotiate price and accept pricing schemes. Plan to provide use cases to the GRAAP-WG. Not a done deal that they will follow WS-Agreement but are exploring the options.

Chargeable grid services: A wrapper around existing Grid services, defined extra SDEs. allow portTypes to allow changing.

See presentation for architecture model.

Focusing on GRAAP-WG (for negotiation model) and JSDL (for term definition).