

OGSA WG F2F, Fujitsu Offices, Sunnyvale, 1/16 – 1/20 2006

January 18, 2006, Wed a.m.

Based on “EMS Architecture Composition Roadmap v0.6, 12/16/2005”

Present: Andrew Grimshaw, Hiro Kishimoto, Darren Pulsipher, Fred Maciel, Jem Treadwell, Jay Unger, Dave Snelling, Chuck Spitz, Fred Brisard, Alan Luniewsky (last part), Dave Berry (last part), Andreas Savva, Steven Newhouse, Steve McGough, Takuya Mori, Mark Morgan

Notes: Jem Treadwell

(See also Andrew's handwritten notes (ppt):

https://forge.gridforum.org/projects/ogsa-wg/document/EMS_Roadmap_notes/en/1)

Summary of Resolved items/actions:

- Separation of EPS and CSG is not clearly required – suggest to RSS to remove CSG and let them make the call. And clearly define the added value of CSG.
- EPS returns an ordered (by policy) list of (Activity Execution Candidates: <JSDL doc; EPR or path of BES container; rank (optional, numeric, extensible), ...>.
- Follow up on workflow discussion and the possibility of doing a Job Manager BoF. Focusing on the API and not the language aspect is one choice but we need to figure out the model and define the scope of the activity first.

Detailed Notes:

Andrew led review of document:

Section 2.2 (Indirect Job Execution) Steven N: No direct Job Manager interface specification (DRMAA is not appropriate)

2.3 User agent and client are the same thing – to be consistent, change to user agent throughout.

[Side question: What's going on with authorization from IBM for OGSA-WG to use Rational Rose? Jay: It's in progress, still waiting for authorization.]

Discussion about portals... separate user agent from job manager.

Jay: If an execution planning service is going to make intelligent decisions about placement it needs to see a sequence of jobs... Steven Newhouse (SN): that scenario is later – out of scope for this scenario. Jay: how do I know what's in scope? Why is this EPS? Not necessary... Jay: are the various scenarios distinguished by what's in scope? Is there more detail about what the EPS does? In terms of which components in EPS are hit, this scenario is not distinguished... SN: No, because the EPS might use different components... Jay: We need the EPS to do a best fit... Andrew: Global optimization... Jay: Concern is that if there's a big flow that does queue-wise optimization, whatever the step-wise interaction of the components is, if doing this requires a different traversal, that may be a problem for the architecture.

Andrew: Need to come back and address this problem.

Ian Foster's comment in the document: This was previously discussed and resolved by e-mail discussion; the comment can be removed.

The Todo item in section 2.3 can be deleted. (The text after it is the result.)

Section 2.4:

Termination cases for this scenario should be:

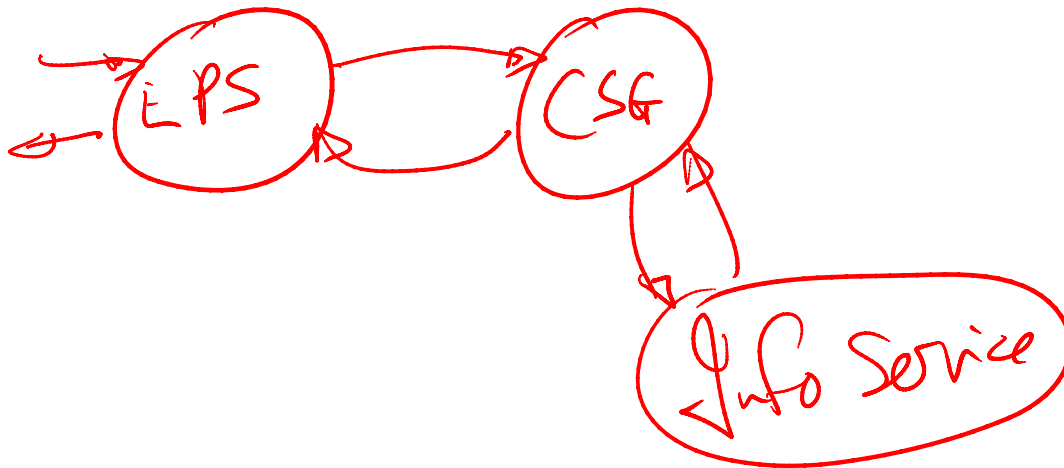
- 1: Job terminates normally
2. Job killed by user/agent
3. Activity dies unexpectedly
 - 3a) JM does not restart the job
 - 3b) JM restarts the job

4) JM terminates job (all activities) for policy reasons

Darren: Activity Endpoint Descriptors (AED) are returned by BES: AEDs are not exposed outside the BES architecture – i.e. not passed back to the JM. Andrew: We don't know anything about the port types of the job. We need to define another operation where we take the endpoint from the JM and ask it to return the current activity endpoint(s) that we want from the workflow

What does the JM return? The EPR of the activity, or something else. Considered the need for a "job" portType, so that the user can ask for info about the job, as opposed to activities. Hiro: We need the portType to have the same interface, whether it comes from BES or the JM. Is it valid for a user agent to talk directly to BES? We could disallow this and get rid of this scenario (would simplify). After discussion it was determined that when a job is submitted the operation is called submit, so has a different return type. Agreed that this is not an issue.

2.5: Dynamic Job Selection. A better (more accurate) name would be "Dynamic Activity Placement."



Steve McGough: the architecture could be impossible to implement efficiently. We may need to collapse the functionality of CSG and EPS into one. Andreas: NAREGI published a paper on the performance. They seem to have good results. Also these are just interfaces. You do not have to expose both if you don't want to. We need to look at workflow in the future.

The RSS-WG does job placement – EPS shouldn't tell them how to do it. Dave Snelling: There's a split between EPS and CSG: would we ever talk to a CSG from anywhere other than an EPS? Steven: what is the value add that a CSG is providing? Need to generate candidates – is this a separate service? Leave to the RSS group, but we need to talk to them, and get their participation in a f2f.

Dave S: Did we get feedback from implementers that they didn't like having the CSG/EPS split forced on them? Andreas: NAREGI seems to have had fairly good results. Dave: Relationship between EPS and CSG is that CSG sends back offers of places to execute, and they may not come back all at once – potential here for complexity. Globus decomposition may be different – they don't have a separate CSG. Darren: what is the EPS returning to the JM? Andrew: The idea was that the EPS returns a set of mappings in space and time – not a reservation. Considered also passing a set of deltas to that mapping, but we're not focused on that right now. Was just going to return a set of mappings and let JM make decisions, but it may make more sense for the CSG to make decisions.

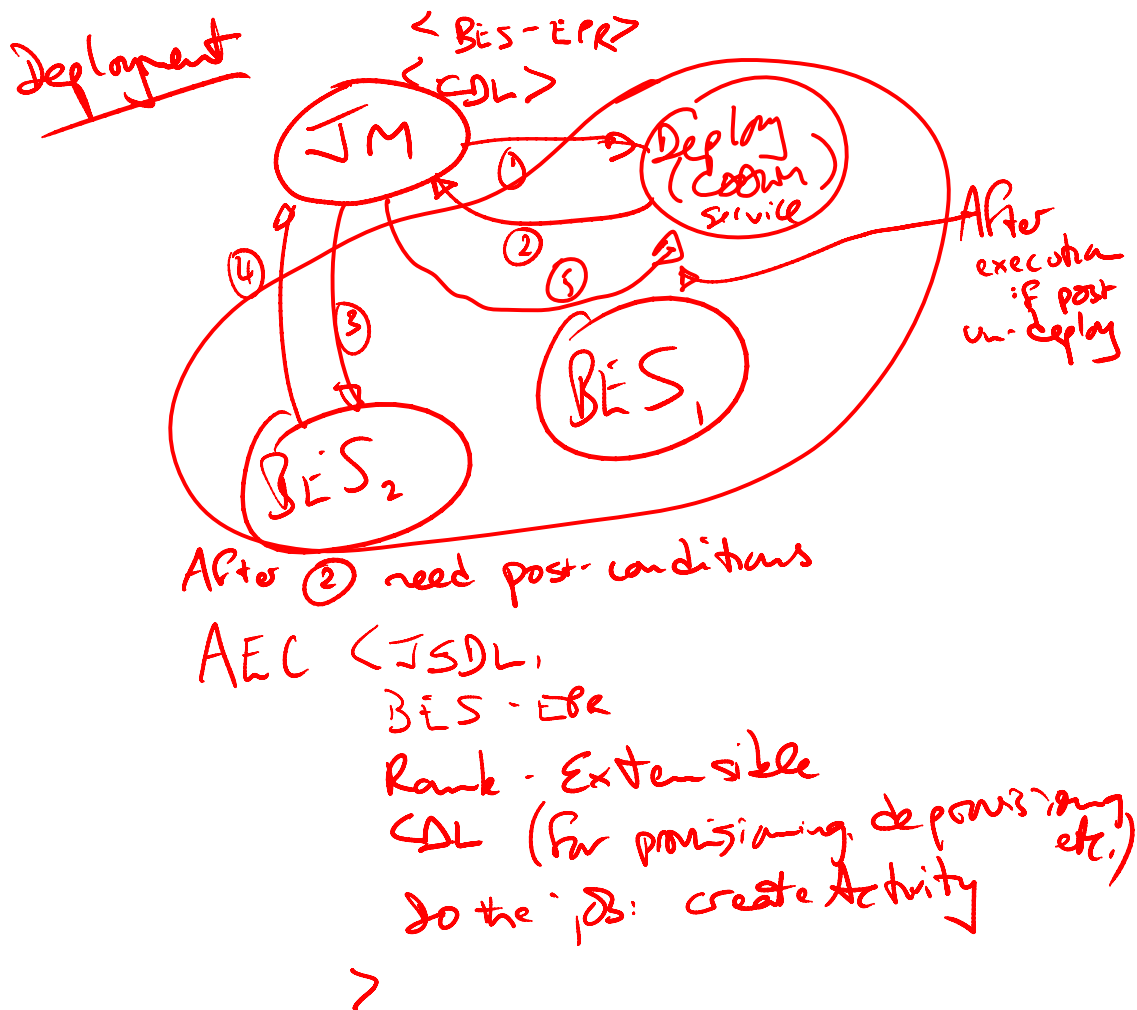
Hiro: In an earlier scenario EPS only returns one JSDL document; Andreas proposes that it returns more. It should return a list, ordered in by recommendation. The information contained in a concrete JSDL documents may be different depending on the chosen BES. So a list is necessary in the general case.

Resolved:

- Separation of EPS and CSG is not clearly required – suggest to RSS to remove CSG and let them make the call. And clearly define the added value of CSG.

- EPS returns an ordered (by policy) list of (Activity Execution Candidates: <JSDL doc; EPR or path of BES container; rank (optional, numeric, extensible), ...>.

Section 3:



An AEC could contain:

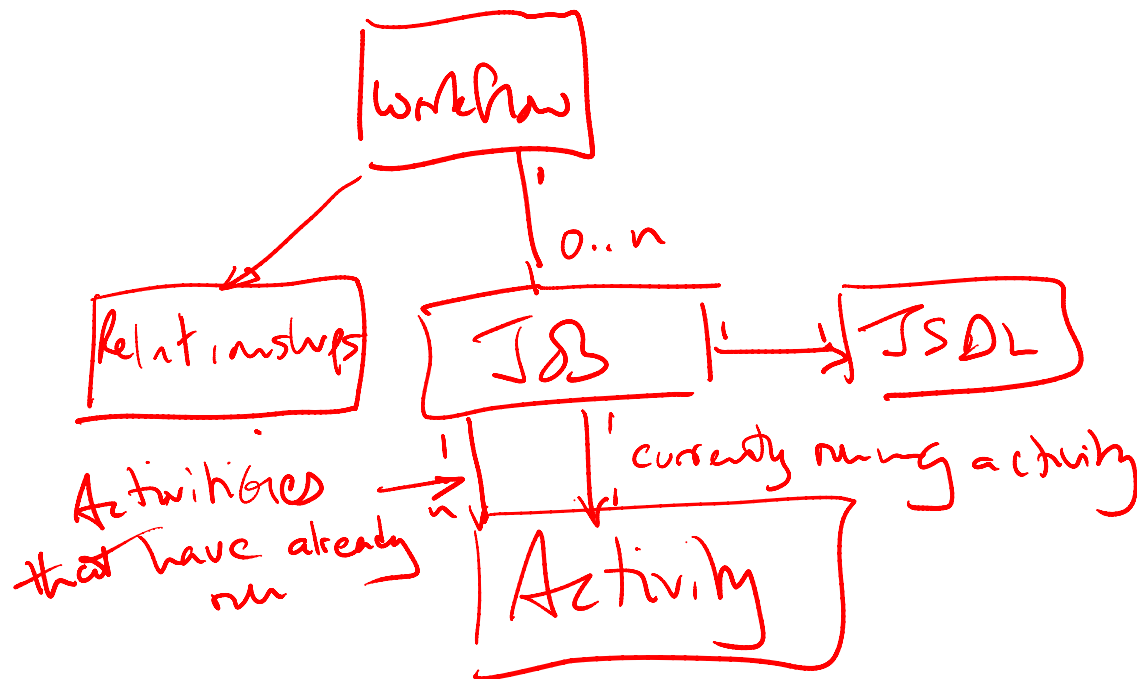
- JSDL
- The EPR for a suitable BES
- A rank (numeric, optional, extensible) – indicates EPS's recommended order of suitability of returned AECs
- CDL (CDDL configuration description language: declarative, not procedural: specifies how to deploy, undeploy, etc.)
- And the EPR of a suitable Deployment service (where to submit the CDL)

An AEC is a plan. Also discussed whether the AEC should include or be structures as pre/exec/post stages. E.g., the CDL and the EPR to the deployment service, if present define the pre-step. (And also possibly the post-steps if undeployment is necessary.) The JSDL and BES EPR define the exec step. In any case a detailed definition is work of the RSS-WG.

Discussion about Complex Jobs:

Jay: I see two general use cases: (i) dependency graph, and (ii) requirement for co-scheduling, parallel jobs, explicitly stated that they have to be run at the same time because they communicate. Darren: parallel jobs are categorized as MPI jobs (even if not MPI). Andrew: can model as constraints, e.g. must be started at the same time, or must be disjoint etc. Jay: We

have a relationship between jobs and activities – jobs may be embarrassingly parallel, simple relationship. A job can be made up of one or more activities, and pushed onto a container independently – no particular order, etc. – simplest case. If nothing else is specified then it's OK to have the notion of a basic complex job that's just a bag – it may have a constraint such as time to completion, rather than execution order). Then consider mandatory parallelism: still just a bag, but it has to be run at the same time. There's a big dichotomy in how those things are represented: all on completion, or all on dependences. Darren: The Workflow Management RG has done quite a bit of work. Steven: We don't need another workflow language... Jay: what alternatives are there? (Steven and others enumerate various workflow languages). Jay: need to agree the model. Darren: A job is made up of a set of activities, e.g. an activity might fail, but the job description might say retry it, so there may be multiple activities in the course of completion of a given job. Jay: Is only one activity associated with a job at one time? Yes. Is the relationship between a job and an activity that there is a one-to-one relationship between a job and an activity? Darren draws diagram:



Jay: We need to figure out the model – I and others were under the impression that a workflow is a collection of activities. Each component has its own JSDL – the relationship between them has to be defined. We need some basic commonality between workflow languages. If I'm trying to write a scheduler I need to be able to interrogate all parts to understand the dependences – I believe workflow is well understood, but we need to specify a container around the relationship stuff, and to have the capability to ask a workflow if there's a temporal relationship between them. Workflow languages need to be compliant [with what we define] to be able to answer these questions. We should do a use-case based exploration to find out what we need to be able to schedule... Steven: There are many workflow use cases and many workflow languages. We should be careful not to do one more.

Andrew wraps up discussion: Interesting topic – we need to have calls on this. We should schedule a BOF, start a WG to interact with OGSA, RSS etc. – call it the Job Manager WG? We may want to start it as a design team. Fred Brisard is volunteered/agrees to lead this group. Need to establish baseline with phone calls.