OGSA-WG interim meeting #14 — April 6th 2006 — Day 3 morning

Participants:

Hiro Kishimoto (Fujitsu)
Andreas Savva (Fujitsu)
Michel Drescher (Fujitsu)
Allen Luniewski (IBM)
Jun Tatemura (NEC)
Dejan Milojicic (HP)
Duane Merrill (UVA) (first hour)
Mark Morgan (UVA) (first hour)
Steven Newhouse (OMII)

Minutes: Michel Drescher (and Andreas Savva)

Joint session with CDDLM

Jun presents "CDL for BLAST Deployment"

Question Hiro: For BLAST there are two kinds of data – the input data and the parameters.
Answer Jun: It is assumed that the parameters are given as part of the application program

Question Hiro: With the JSDL doc for BLAST, the resource (server) is chosen with the resource specification in the JSDL doc. How's the file server chosen?
Answer Steven: This is the wrong moment to ask as this is more pointing towards RSS etc.
Answer Andreas: With JSDL you can select to mount a directory and not necessarily specify a file server (a file serve may also be specified)
Answer Hiro: So we will now concentrate on that we have a specific exec server and a specific file server chosen already, and how to deploy BLAST and let it run as requested by the user

Question Andreas: The <Server> and <FileServer> XML elements are considered as templates?
Answer Jun: Yes they are.

The group discussed and reached consensus on the high level interaction between ACS, JM and CDL Deployment service. (This was a confirmation of the current (general) EMS interaction picture.) While CDDLM leaves the exact interaction flexible to the implementation, EMS will profile this to a particular interaction pattern. In particular it was clarified that the CDDLM deployment service may manage a number of machines and that the method by which content is deployed on one of those machines is implementation dependent (e.g., a deployment agent).

Hiro asks for a concrete example for this interaction pattern

The group discussed how a CDL document with a concrete cdl:CodeBase element can be picked up and executed on a concrete system. Components are **not** elements that can be deployed but elements that **help** to deploy job requests (e.g. one component for .NET, one for SmartFrog, etc.)

In order to be able to deploy a BLAST job, we must also know which components are configured at the CDL deployment server

A platform independent CDL document is later-on mapped to a platform-specific CDL script that takes the platform-specific BLAST archive to install on the system. The same with the file server configuration.

In case a JSDL job description specifies that the executable is to reside in “/usr/bin/blast”, the current example scenario presented in the slides would fail to deploy BLAST, as in the scenario, the component model implementation defines the root directory of where the BLAST application is installed. In a more generic CDDLM deployment scenario, this would be possible, though.

ACS may be used to resolve how cdl:CodeBase is resolved on the target platform. (But in that case the contents of the CodeBase would have to be an EPR plus other information.)

From EMS point of view, we need to define a set of common component models, as they do exist in current implementations, but not in a

published definition document. In other words there is no standard definition of such common components.

Action: Hiro and Jun will look at what needs to be provided and make a proposal to the group.

Steven Newhouse presents an interaction document for OGSA EMS scenarios

The group discusses the domain specific terms and semantics of “deployment” and “provisioning”. Deployment could not only been policy driven, but also user driven (by job submission?), workload driven, etc.?

Dejan: Deployment happens almost instantaneously, provisioning is a more heavyweight process that will take longer. But otherwise, provisioning and deployment are the same.

Dejan: Both deployment and provisioning may be triggered by automated or manual (i.e. job submission) activities. (This statement is left open for consideration, as no consensus could be reached)

Andreas proposed to look at the specific scenario rather than try to work through these terminology again.

Steven added Dejan’s comments to the definitions and also pointed out that these are the definitions used in the scenarios below and that the group should keep this in mind while reviewing.

The group discusses the consequences for a CDL enabled system when the user submits a JSDL document with either a specific host name or no host name at all, in an environment where the component model in the CDL system provides CDL templates with cdl:lazy host binding.

Jun explained that this issue was not discussed before and the current specification would not allow for a CDL definition that has the host as cdl:lazy and for that value to be supplied by the caller of the Deployment Service.

Action: Jun and Dejan will take this issue back to the CDDLM-WG for discussion.

The group discusses the roles and responsibilities of where to define how to deploy a particular application and which information is needed to do so, and where or which instance is to provide this concrete information to enable a successful deployment of a requested application

Action: Steven will update the scenarios with the comments from this review and merge into the draft Scenarios document

06/04/2006 08:30:00