Interaction between CDDLM & JSDL
EMS Architecture Document

1 Definitions

For the purposes of this document we define:

- Life Cycle: Both provisioning and deployment actions have a life cycle – they are created, used, and eventually destroyed. A mechanism is needed for activities to 'register' their use and interest in a particular activity. Before an activity is destroyed there have to be no outstanding references. [Look at EGA provisioning state model.]
- Provisioning: The instantiation of an environment on a resource (i.e. a deployment activity) that may be used by more than one activity. Generally these deployments are 'heavyweight' in that they take time to instantiate. Provisioning actions may be triggered by a specific activity (e.g. following a manual job submission) or as a result of general response to state of current resources (i.e. defined by a policy). E.g. my available free RHEL4 resources has dropped to 5%, dynamically provision new nodes until the total free nodes has increased to 15%.
- Deployment: The instantiation of an environment on a resource to meet the needs of a specific activity such as a submitted job. This is generally a 'lightweight' action in that provides a binary, dataset, etc. onto the resource. Once used by the specified activity that environment may be removed immediately, or left as part of a 'cache' to be cleanup up or reused at a later date. In such a case a 'keep alive' model is needed to show which *** E.g. an activity wants to run Gaussian, download and install the required version for this platform.
- Activity: The smallest part of possibly a larger sequence of activities generated from a single submitted job.
- Job Manager (JM): The entity that accepts a 'job' (defined by a JSDL document) from the user agent. The JM coordinates further invocations and interactions with other elements of the EMS architecture.
- Deployment Service (DS): The entity that accepts a 'deployment' action (defined by a CDL document) to instantiate an environment on the resource to support a BES job. The DS provides the bootstrapping for all deployment/provisioning actions. BES does not do provisioning/deployment.

Deleted:

2 Scenarios

2.1 Deploying an Application (3.1)

In this scenario the BLAST application needs to be deployed to support an activity that is to be initiated within a BES container. In this scenario the BES instance has already been determined but no BLAST application resides on that platform.

[NB: This (and other steps) should be converted to a UML diagram once agreed.]

1. The User Agent passes the JSDL document (which requires the presence of the BLAST application) to the JM requesting that the job (the document) be sent to a specific BES instance.
2. The JM queries the Information Service (IS), or the platform directly, to discover if the specified BES instance resides on a resource which already has the BLAST application.
3. In this scenario an empty set is returned as the specified BES instance does not have the BLAST application located on it and this needs to be provided for the JSDL document to be successfully executed.
4. A BLAST binary suitable for the platform on which the BES instance resides is discovered from an IS (or from the ACS?).
5. The CDL document template (i.e. a CDL document that still contains unresolved type references – lazy elements) MAY have some of these 'lazy elements' filled in by the JM using information retrieved from the IS. The CDL document (which may still contain lazy elements) is then passed to the DS located on the platform hosting the BES instance.
    a. The DS MUST complete the remaining lazy elements in the CDL document. For instance, the hostname in the CDL document cannot be specified until deployment if there is no specific requirement as to where the CDL document is being processed. [CDDLM example.]
    b. The DS MAY act as a proxy for deployment on other nodes OTHER than the one it is running on.
    c. The DS MUST advertise (somewhere in a currently unspecified manner) which nodes it is capable of deploying stuff onto, e.g. front end node for a cluster.
6. The DS returns an EPR to the JM providing a reference to the installed software.
7. The IS entry related to the resource is updated with the location of the BLAST application by the JM. This prevents the removal of the resource as 'something' the BLAST application is 'using' it.
8. The JM can use the EPR from the DS to find the defined environment variables to be pulled back into the JSDL document. This is effectively an operation to retrieve relevant chunks of the XML deployment tree (i.e. the instantiated CDL document) back to the JM for insertion into JSDL document.
9. The complete JSDL document is then sent to the BES instance on that platform.
10. The IS entry related to the BLAST application is updated with the information that the BES Activity now has an 'interest' in the application. If another BES Activity uses this BLAST application and performs this registration it will prevent un-deployment of the BLAST application even if the initial activity has completed.
11. The BES Activity completes successfully and its interest in the BLAST application is removed from the IS.
12. If no other activities are registering an interest in the BLAST application on this resource then the undeploy operation is invoked on the Deployment Service to remove this BLAST application from the resource.
13. The BLAST application entry in the IS is removed.

14. The User Agent is informed that the job is complete.

Missing bits:

- May be multiple things that need to be deployed for the activity
- And deployed things may be cached for later re-use or cleanup.

2.2 Provisioning a BES instance (3.2)

Provisioning is an activity initiated by policy and not directly in response to a specific job or some other activity. In the following scenario a provisioning service (PS) monitors ‘the system’ (in this particular example a compute centre) and undertakes provisioning actions in response to established policy. In effect the PS provides an autonomic capability.

A prerequisite is that there are available free platforms with a DS running on them.

1. The PS undertakes its monitoring of the BES instances within the compute centre by determining the current capacity of the BES instances registered in the IS.
2. The policy indicates that more BES instances need to be provisioned. The IS provides a list of platform (bare OS’s) that can be used.
3. A CDL document describing the instantiation of a BES instance is sent to the CDL server running on the selected platform.
4. The endpoint returned on success by the CDL server is used to register the presence of the BES instance in the IS.

At stage 2 the policy could also determine that there are too many BES instances, in which case stage 3 would generate a undeploy action through the CDL server to remove the BES instance which would then be un-registered from the IS in stage 4.

Deleted: document to

3 Worked Example

3.1 Running a BLAST (Basic Local Alignment Search Tool) job

3.1.1 JSDL Document Submitted to the Job Manager

```
<?xml version="1.0" encoding="UTF-8"?>
<jsdl:JobDefinition
xmlns:jsdl="http://schemas.ggf.org/jsdl/2005/11/jsdl"
xmlns:jsdl-posix="http://schemas.ggf.org/jsdl/2005/11/jsdl-posix"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://schemas.ggf.org/jsdl/2005/11/jsdl jsdl.xsd
http://schemas.ggf.org/jsdl/2005/11/jsdl-posix jsdl-posix.xsd">
  <jsdl:JobDescription>
    <jsdl:JobIdentification>
      <jsdl:JobName>Blast1</jsdl:JobName>
      <jsdl:Description>Blast query number 1</jsdl:Description>
      <jsdl:JobAnnotation>Blast1Annotation</jsdl:JobAnnotation>
      <jsdl:JobProject>BlastProject</jsdl:JobProject>
    </jsdl:JobIdentification>
    <jsdl:Application>
      <jsdl:ApplicationName>
BlastN
      </jsdl:ApplicationName>
      <jsdl:ApplicationVersion>
NCBI BLAST 2.2.13
      </jsdl:ApplicationVersion>
      <jsdl:Description>BlastN performs nucleotide similarity
searching</jsdl:Description>
      <jsdl-posix:POSIXApplication>
****
EXECUTABLE ELEMENT DELETED – TO BE INSERTED BY JM POST DEPLOYMENT
****
        <jsdl-posix:Argument>-p</jsdl-posix:Argument>
        <jsdl-posix:Argument>blastn</jsdl-posix:Argument>
        <jsdl-posix:Argument>-d</jsdl-posix:Argument>
        <jsdl-posix:Argument>est</jsdl-posix:Argument>
        <jsdl-posix:Argument>-T</jsdl-posix:Argument>
        <jsdl-posix:Argument>T</jsdl-posix:Argument>
        <jsdl-posix:Input
filesystemName="HOME">sequences1.txt</jsdl-posix:Input>
        <jsdl-posix:Output
filesystemName="HOME">sequences1.html</jsdl-posix:Output>
        <jsdl-posix:Error
filesystemName="HOME">sequences1.err</jsdl-posix:Error>
        <jsdl-posix:WorkingDirectory
filesystemName="HOME">blastqueries</jsdl-posix:WorkingDirectory>
****
ENVIRONMENT ELEMENT DELETED – TO BE INSTERTED BY JM POST DEPLOYMENT
****
        <jsdl-posix:UserName>csmith</jsdl-posix:UserName>
        <jsdl-posix:GroupName>bio</jsdl-posix:GroupName>
      </jsdl-posix:POSIXApplication>
    </jsdl:Application>
    <jsdl:Resources>
      <jsdl:FileSystem name="TMP">
        <jsdl:FileSystemType>temporary</jsdl:FileSystemType>
        <jsdl:Description>
          Temporary space that does not necessarily persist after the
job
          terminates and which might not be shared between resources,
but
          which will be fast.
        </jsdl:Description>
      </jsdl:FileSystem>
      <jsdl:FileSystem name="HOME">
        <jsdl:FileSystemType>normal</jsdl:FileSystemType>
        <jsdl:Description>Chris's home directory</jsdl:Description>
      </jsdl:FileSystem>
      <jsdl:ExclusiveExecution>true</jsdl:ExclusiveExecution>
      <jsdl:TotalCPUCount>
        <jsdl:Exact>1.0</jsdl:Exact>
      </jsdl:TotalCPUCount>
    </jsdl:Resources>
    <jsdl:DataStaging>
      <jsdl:FileName>blastqueries/sequences1.txt</jsdl:FileName>
```

Deleted: <jsdl-posix:Executable>/usr/local/bin/blastall</jsdl-posix:Executable>

```
      <jsdl:FilesystemName>HOME</jsdl:FilesystemName>
      <jsdl:CreationFlag>overwrite</jsdl:CreationFlag>
      <jsdl:Source>

<jsdl:URI>file:/Users/csmith/blastqueries/sequences1.txt</jsdl:URI>
      </jsdl:Source>
    </jsdl:DataStaging>
    <jsdl:DataStaging>
      <jsdl:FileName>blastqueries/sequences1.html</jsdl:FileName>
      <jsdl:FilesystemName>HOME</jsdl:FilesystemName>
      <jsdl:CreationFlag>overwrite</jsdl:CreationFlag>
      <jsdl:Target>

<jsdl:URI>file:/Users/csmith/blastqueries/sequences1.html</jsdl:URI>
      </jsdl:Target>
    </jsdl:DataStaging>
    <jsdl:DataStaging>
      <jsdl:FileName>blastqueries/sequences1.err</jsdl:FileName>
      <jsdl:FilesystemName>HOME</jsdl:FilesystemName>
      <jsdl:CreationFlag>append</jsdl:CreationFlag>
      <jsdl:Target>

<jsdl:URI>file:/Users/csmith/blastqueries/sequences1.err</jsdl:URI>
      </jsdl:Target>
    </jsdl:DataStaging>
  </jsdl:JobDescription>
</jsdl:JobDefinition>
```

3.1.2 CDL Document

Define the CDL document submitted to the CDL Deployment Service that will deploy the BLAST application onto the system.

```
<BLASTSystem>
  <!—Component wakes up in the locally specified installation space →
 <!--- cd dir=”$INSTALL_ROOT” />
  <ant:untar archive=”http;//host:port/blast.tar” />
  <exportAtttributeValuePair atrribute=“BLASTALL_EXE”
value=”$INSTALL_ROOT/blast-x.y.z/bin/blastall</exportAttributePair>
<!—ASSUME SOMETHING WILL EXPAND & INSTER THE INSTALL_ROOT →
<!— Advertise the populated JSDL:FileSystem (elements) so they can be pulled back
from the CDL Deployment Service and inserted into thje JSDL document.
</BLASTSystem>
```

3.1.3 JSDL Document submitted to the BES Instance

```
<?xml version="1.0" encoding="UTF-8"?>
<jsdl:JobDefinition
xmlns:jsdl="http://schemas.ggf.org/jsdl/2005/11/jsdl"
xmlns:jsdl-posix="http://schemas.ggf.org/jsdl/2005/11/jsdl-posix"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://schemas.ggf.org/jsdl/2005/11/jsdl jsdl.xsd
http://schemas.ggf.org/jsdl/2005/11/jsdl-posix jsdl-posix.xsd">
```

```
  <jsdl:JobDescription>
    <jsdl:JobIdentification>
      <jsdl:JobName>Blast1</jsdl:JobName>
      <jsdl:Description>Blast query number 1</jsdl:Description>
      <jsdl:JobAnnotation>Blast1Annotation</jsdl:JobAnnotation>
      <jsdl:JobProject>BlastProject</jsdl:JobProject>
    </jsdl:JobIdentification>
    <jsdl:Application>
      <jsdl:ApplicationName>BlastN</jsdl:ApplicationName>
      <jsdl:ApplicationVersion>NCBI BLAST
2.2.13</jsdl:ApplicationVersion>
      <jsdl:Description>BlastN performs nucleotide similarity
searching</jsdl:Description>
      <jsdl-posix:POSIXApplication>
        <jsdl-posix:Executable>/usr/local/bin/blastall</jsdl-
posix:Executable>
        <jsdl-posix:Argument>-p</jsdl-posix:Argument>
        <jsdl-posix:Argument>blastn</jsdl-posix:Argument>
        <jsdl-posix:Argument>-d</jsdl-posix:Argument>
        <jsdl-posix:Argument>est</jsdl-posix:Argument>
        <jsdl-posix:Argument>-T</jsdl-posix:Argument>
        <jsdl-posix:Argument>T</jsdl-posix:Argument>
        <jsdl-posix:Input
filesystemName="HOME">sequences1.txt</jsdl-posix:Input>
        <jsdl-posix:Output
filesystemName="HOME">sequences1.html</jsdl-posix:Output>
        <jsdl-posix:Error
filesystemName="HOME">sequences1.err</jsdl-posix:Error>
        <jsdl-posix:WorkingDirectory
filesystemName="HOME">blastqueries</jsdl-posix:WorkingDirectory>
        <jsdl-posix:Environment
name="PATH">/usr/bin:/usr/local/bin:/usr/local/bio/bin</jsdl-
posix:Environment>
        <jsdl-posix:Environment name="TMPDIR" filesystemName="TMP"/>
        <jsdl-posix:WallTimeLimit>60</jsdl-posix:WallTimeLimit>
        <jsdl-posix:FileSizeLimit>1073741824</jsdl-posix:FileSizeLimit>
        <jsdl-posix:CoreDumpLimit>0</jsdl-posix:CoreDumpLimit>
        <jsdl-posix:DataSegmentLimit>32768</jsdl-
posix:DataSegmentLimit>
        <jsdl-posix:LockedMemoryLimit>8388608</jsdl-
posix:LockedMemoryLimit>
        <jsdl-posix:MemoryLimit>67108864</jsdl-posix:MemoryLimit>
        <jsdl-posix:OpenDescriptorsLimit>16</jsdl-
posix:OpenDescriptorsLimit>
        <jsdl-posix:PipeSizeLimit>512</jsdl-posix:PipeSizeLimit>
        <jsdl-posix:StackSizeLimit>1048576</jsdl-posix:StackSizeLimit>
        <jsdl-posix:CPUTimeLimit>30</jsdl-posix:CPUTimeLimit>
        <jsdl-posix:ProcessCountLimit>8</jsdl-posix:ProcessCountLimit>
        <jsdl-posix:VirtualMemoryLimit>134217728</jsdl-
posix:VirtualMemoryLimit>
        <jsdl-posix:ThreadCountLimit>8</jsdl-posix:ThreadCountLimit>
        <jsdl-posix:UserName>csmith</jsdl-posix:UserName>
        <jsdl-posix:GroupName>bio</jsdl-posix:GroupName>
      </jsdl-posix:POSIXApplication>
    </jsdl:Application>
    <jsdl:Resources>
      <jsdl:CandidateHosts>
```

```
        <jsdl:HostName>cluster1</jsdl:HostName>
        <jsdl:HostName>cluster2</jsdl:HostName>
      </jsdl:CandidateHosts>
      <jsdl:FileSystem name="TMP">
        <jsdl:FileSystemType>temporary</jsdl:FileSystemType>
        <jsdl:Description>
          Temporary space that does not necessarily persist after the
job
          terminates and which might not be shared between resources,
but
          which will be fast.
        </jsdl:Description>
        <jsdl:DiskSpace>

<jsdl:LowerBoundedRange>10737418240.0</jsdl:LowerBoundedRange>
        </jsdl:DiskSpace>
      </jsdl:FileSystem>
      <jsdl:FileSystem name="HOME">
        <jsdl:FileSystemType>normal</jsdl:FileSystemType>
        <jsdl:Description>Chris's home directory</jsdl:Description>
        <jsdl:MountPoint>/home/csmith</jsdl:MountPoint>
      </jsdl:FileSystem>
      <jsdl:ExclusiveExecution>true</jsdl:ExclusiveExecution>
      <jsdl:OperatingSystem>
        <jsdl:OperatingSystemType>
          <jsdl:OperatingSystemName>MACOS</jsdl:OperatingSystemName>
        </jsdl:OperatingSystemType>

<jsdl:OperatingSystemVersion>10.4.4</jsdl:OperatingSystemVersion>
        <jsdl:Description>Mac OS version 10.4.x</jsdl:Description>
      </jsdl:OperatingSystem>
      <jsdl:CPUArchitecture>
        <jsdl:CPUArchitectureName>powerpc</jsdl:CPUArchitectureName>
      </jsdl:CPUArchitecture>
      <jsdl:IndividualCPUSpeed>
        <jsdl:LowerBoundedRange>2147483648.0</jsdl:LowerBoundedRange>
      </jsdl:IndividualCPUSpeed>
      <jsdl:IndividualCPUTime>
        <jsdl:UpperBoundedRange>60.0</jsdl:UpperBoundedRange>
      </jsdl:IndividualCPUTime>
      <jsdl:IndividualCPUCount>
        <jsdl:Exact>2.0</jsdl:Exact>
      </jsdl:IndividualCPUCount>
      <jsdl:IndividualNetworkBandwidth>
        <jsdl:LowerBoundedRange>104857600.0</jsdl:LowerBoundedRange>
      </jsdl:IndividualNetworkBandwidth>
      <jsdl:IndividualPhysicalMemory>
        <jsdl:LowerBoundedRange>1073741824.0</jsdl:LowerBoundedRange>
      </jsdl:IndividualPhysicalMemory>
      <jsdl:IndividualVirtualMemory>
        <jsdl:LowerBoundedRange>1073741824.0</jsdl:LowerBoundedRange>
      </jsdl:IndividualVirtualMemory>
      <jsdl:IndividualDiskSpace>
        <jsdl:LowerBoundedRange>1073741824.0</jsdl:LowerBoundedRange>
      </jsdl:IndividualDiskSpace>
      <jsdl:TotalCPUTime>
        <jsdl:UpperBoundedRange>600.0</jsdl:UpperBoundedRange>
      </jsdl:TotalCPUTime>
      <jsdl:TotalCPUCount>
        <jsdl:Exact>10.0</jsdl:Exact>
      </jsdl:TotalCPUCount>
      <jsdl:TotalPhysicalMemory>
        <jsdl:LowerBoundedRange>10737418240.0</jsdl:LowerBoundedRange>
      </jsdl:TotalPhysicalMemory>
      <jsdl:TotalVirtualMemory>
        <jsdl:LowerBoundedRange>10737418240.0</jsdl:LowerBoundedRange>
      </jsdl:TotalVirtualMemory>
      <jsdl:TotalDiskSpace>
        <jsdl:LowerBoundedRange>10737418240.0</jsdl:LowerBoundedRange>
      </jsdl:TotalDiskSpace>
      <jsdl:TotalResourceCount>
        <jsdl:Exact>5.0</jsdl:Exact>
      </jsdl:TotalResourceCount>
    </jsdl:Resources>
    <jsdl:DataStaging>
      <jsdl:FileName>blastqueries/sequences1.txt</jsdl:FileName>
      <jsdl:FilesystemName>HOME</jsdl:FilesystemName>
      <jsdl:CreationFlag>overwrite</jsdl:CreationFlag>
      <jsdl:Source>

<jsdl:URI>file:/Users/csmith/blastqueries/sequences1.txt</jsdl:URI>
      </jsdl:Source>
    </jsdl:DataStaging>
    <jsdl:DataStaging>
      <jsdl:FileName>blastqueries/sequences1.html</jsdl:FileName>
      <jsdl:FilesystemName>HOME</jsdl:FilesystemName>
      <jsdl:CreationFlag>overwrite</jsdl:CreationFlag>
      <jsdl:Target>

<jsdl:URI>file:/Users/csmith/blastqueries/sequences1.html</jsdl:URI>
      </jsdl:Target>
    </jsdl:DataStaging>
    <jsdl:DataStaging>
      <jsdl:FileName>blastqueries/sequences1.err</jsdl:FileName>
      <jsdl:FilesystemName>HOME</jsdl:FilesystemName>
      <jsdl:CreationFlag>append</jsdl:CreationFlag>
      <jsdl:Target>

<jsdl:URI>file:/Users/csmith/blastqueries/sequences1.err</jsdl:URI>
      </jsdl:Target>
    </jsdl:DataStaging>
  </jsdl:JobDescription>
</jsdl:JobDefinition>
```

3.2 CDL Document

Deploy the BLAST application
CDL Document going in to the CDL Deployemnt Service

```
<BLASTSystem>
   <!—Component wakes up in the locally specified installation space →
  <!--- cd dir=”$INSTALL_ROOT” />
   <ant:untar archive=”http;//host:port/blast.tar” />
   <exportAtttributeValuePair atrribute=“BLASTALL_EXE”
value=”$INSTALL_ROOT/blast-x.y.z/bin/blastall</exportAttributePair>
<!—ASSUME SOMETHING WILL EXPAND & INSTER THE INSTALL_ROOT →
<!— Advertise the populated JSDL:FileSystem (elements) so they can be pulled back
from the CDL Deployment Service and inserted into thje JSDL document.
</BLASTSystem>

   <Server cdl:extends=“d:Server”/>
       <cmp:CodeBase>blast.tar</cmp:CodeBase>
       <d:FileSystem>
            <d:ID cdl:ref=“/File/d:ID”/>
            <d:Dir>data</d:Dir>
       </d:FileSystem>
       <d:Env name=“ROOT” cdl:ref=“/Server/RootDir”/>
       <d:Env name=“DATA”>$ROOT/data</d:Env>
   </Server>
   <File cdl:extends=“d:FileSystem”>
       <cmp:CodeBase>blastdb.tar</cmp:CodeBase>
   </File>
</BLASTSystem>
```

CDL Document representing the deployed BLAST application

Completed JSDL document

BLASTSystem element

Components help deployment for a specific component for deployment through a particular API.

Deployment API services do not declare what they are (.NET/Unix). They do not declare what components exist.

Need to deploy ‘Components’ into the CDL system to enable the deployment of applications.

ACS: Binary, CDL document, Components needed to support CDL deployment

Common case component class – assume
What can be returned

Need codebase definitions
Need to better define the message body more tightly.