A New FedCloud Java Client Library
Michal Kimle, Boris Parák, Zdeněk Šustr, Jiří Sitera
CESNET, a. l. e., Prague, Czech Republic

OCCI
• The Open Cloud Computing Interface (OCCI) is an OGF standard for cloud resource management.
• The EGI Federated Cloud uses OCCI as a common interface to access its computing services.

For OCCI compliance, the client and server side must have at least these capabilities:

<table>
<thead>
<tr>
<th>Client must be able to...</th>
<th>Server must be able to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>receive, parse and understand model</td>
<td>render model: local capabilities, mixins, resource categories</td>
</tr>
<tr>
<td>render request, validate against model and send</td>
<td>parse request and act on it</td>
</tr>
<tr>
<td>receive response: location, rendered object, HTTP return code</td>
<td>return response</td>
</tr>
</tbody>
</table>

jOCCI
• Only the Ruby-based rOCCI framework with its command line interface available until very recently.
• Demand to support OCCI in Java. Wrappers around CLI or jRuby are not a suitable solution.
• jOCCI – a new native Java library implementing the OCCI class structure, rendering and transport, allowing developers to work with OCCI concepts natively in Java.
• Not specific to EGI FedCloud. Independently general-purpose OCCI client, helping to establish the OCCI standard.
• Provided by CESNET – developers of the existing rOCCI framework. Funded by EGI-InSPIRE.
• Already being tried out by early adopters.
• Works with any cloud platform implementing OCCI: rOCCI (OpenNebula, AWS, MS Azure\textsuperscript{in progress}), openstack, synnefo, ...

Examples

Creating a client instance (with X509 authentication):

```java
HTTPAuthentication auth = new X509Authentication("/path/to/certificate.pem", "password");
auth.setCAPath("/etc/grid-security/certificates/" + path to CA dir);
Client client = new HTTPClient(URI.create("https://remote.server.net"), auth);
```

Listing available resources:

```java
List&lt;URI&gt; list = client.list(URI.create("http://schemas.ogf.org/occi/infrastructure#compute"));
```

Creating a compute resource:

```java
Model model = client.getModel();
EntityBuilder entityBuilder = new EntityBuilder(model);
Resource resource = entityBuilder.getResource("compute");
resource.addMixin(model.findMixin("debian7", "os_tpl"));
resource.addMixin(model.findMixin("small", "resource_tpl"));
resource.addAttribute(Compute.MEMORY_ATTRIBUTE_NAME, "2048");
URI location = client.create(resource);
```

Future plans
• Supporting the jOCCI library, already made available to the community.
• Working with early adopters integrating jOCCI into their Java-based projects.
• Next step: Releasing a jclouds OCCI provider based on jOCCI.

Getting jOCCI
• Available from maven central repository. Search search.maven.org for “jOCCI”.
• Source code with documentation: github.com/EGI-FCTF

Contacting us for support
• E-mail: cloud@metacentrum.cz