

# A New FedCloud Java Client Library

Michal Kímle, Boris Parák, Zdeněk Šustr, Jiří Sitera

CESNET, a. i. 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:

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

## jOCCI

- Only the Ruby-based **rOCCI** framework with its command line interface available up 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. Independent 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<sup>in progress</sup>), openstack, synnefo, ...

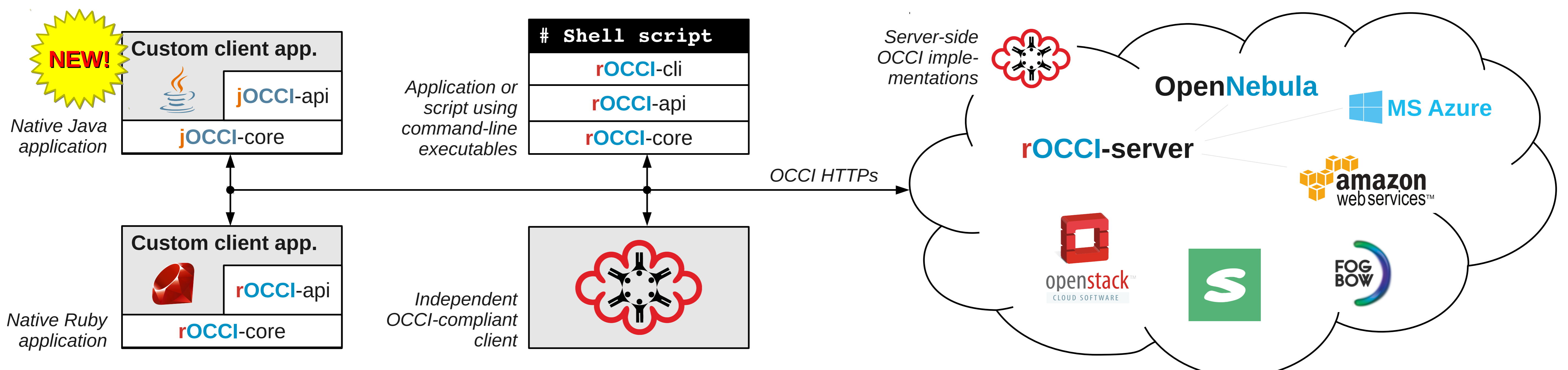


Fig. 1: The role of the new jOCCI library in OCCI client-server implementations.

## Examples

Creating a client instance (with X509 authentication):

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

Listing available resources:

```
List<URI> list = client.list(URL.create("http://schemas.ogf.org/occi/
infrastructure#compute"));
```

Creating a compute resource:

```
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](http://search.maven.org) for "jOCCI".
- Source code with documentation: [github.com/EGI-FCTF](https://github.com/EGI-FCTF)

### Contacting us for support

- E-mail: [cloud@metacentrum.cz](mailto:cloud@metacentrum.cz)