

## Providing OCCI support for arbitrary Cloud Management Frameworks

Boris Parák, Zdeněk Šustr, Florian Feldhaus



- Introduction
- Midterm Goals
- Framework Changes
- *rOCCI-server* Design
- Plans for the Future



What is rOCCI?

- a framework implementing OCCI in Ruby
- a client providing shell-based user interface
- an open source project hosted on GitHub

What is *rOCCI-server*?

- a server-side implementation leveraging rOCCI
- a bridge between a CMF and the world of OCCI

What is the goal of this project?

- extend the rOCCI framework
- design and implement a new *rOCCI-server*



What did we set out to do?

- analyze existing OCCl implementations
- find a way to provide OCCl support for CMFs
- work with other developers to utilize their solutions
- provide a design for a new implementation

How did we accomplish this?

- by rewriting the *rOCCl* framework
- by re-designing *rOCCl-server* from scratch

## Internal:

- completely new parser with relaxed parsing rules
- new attribute handling, including value validation
- growing code base split into *rOCCI-core*, *rOCCI-api* and *rOCCI-cli*

## Client:

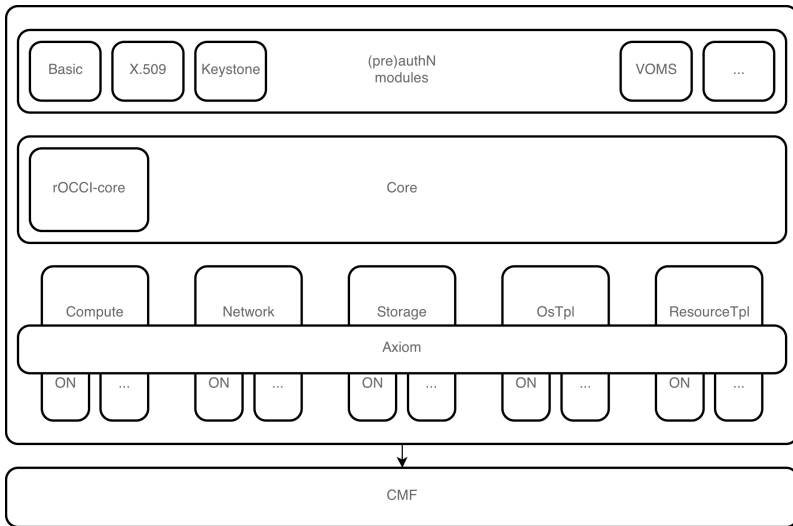
- support for third-party OCCI servers (OCCI-OS for OpenStack and snf-occi for Synnefo)
- support for Keystone-based authentication
- authentication fallbacks
- much better VOMS handling and output rendering

Provided you have **Ruby 1.9+** and **Rubygems**:

```
$ gem install occi-cli
```

```
$ occi --help
```

```
$ occi --action create --resource compute \  
  --mixin os#egi_demo --mixin resource#small \  
  --attributes title='rOCCI_VM' \  
  --context public_key='file:///tmp/id_rsa.pub'
```



## Key design elements:

- modular (pre)authentication
- real authentication left to the backend CMF
- backend interface implemented with Axiom
  - relational algebra on structured data
  - possibility to “merge” multiple data sources
- core functionality provided by *rOCCI-core*

## Why not use Fog as the backend interface?

- was part of the original design
- turned out not to be very usable
- did not provide enough of unified functionality



- keep supporting EGI FedCloud
- extend *rOCCI-core* with the latest developments in OCCI
- extend *rOCCI-cli* with dynamic linking capabilities for attaching block storage or network interfaces to running instances
  
- implement the new *rOCCI-server*
- provide detailed documentation
- deploy the new server in EGI FedCloud

What to read if you want to know more?

- <http://occi-wg.org>
- <https://www.egi.eu/infrastructure/cloud>
- <https://github.com/gwdg/rOCCI>
- <https://github.com/gwdg/rOCCI-server>
- <https://github.com/dkubb/axiom>

Do you have any questions?

- ask us directly at [parak@cesnet.cz](mailto:parak@cesnet.cz) or [sustr4@cesnet.cz](mailto:sustr4@cesnet.cz)
- ask in our mailing lists [rocci@gwdg.de](mailto:rocci@gwdg.de) or [inspire-mp-rocci@mailman.egi.eu](mailto:inspire-mp-rocci@mailman.egi.eu)