GUOCCI – The Entryway to Federated Cloud for Small-scale Users

Radim Janča[*], **Dušan Baran, Boris Parák, Zdeněk Šustr**
CESNET z. s. p. o.
Zikova 4, 160 00, Praha 6, Czech Republic
E-mail: janca@ics.muni.cz, 445397@mail.muni.cz, boris.parak@cesnet.cz, zdenek.sustr@cesnet.cz

The Open Cloud Computing Interface (OCCI) – a standard released by the Open Grid Forum – has found wide adoption. First came, quite naturally, server-side components. Later, with a variety of attractive computing resources made available over the standardized protocol, user-side submission tools or science gateways followed. These are usually tailored with a specific use case in mind, and – as such – they are generally capable of setting up heterogeneous data processing platforms and orchestrating complex workflows specific to their given area of science. They put all this functionality at user's disposal, often "at a single click". On the other hand, small-scale users, backed by no software development teams, are usually relegated to the use of the simplest OCCI client in the command line. That is freely available and amply documented, but still, it by no means lowers the threshold of entry for potential federated cloud users, who often come from non-technical areas in the long tail of science. Therefore, the GUOCCI (GUI for OCCI) has been conceived as a rudimentary graphical user interface to OCCI-compliant cloud services. It is by design kept as simple as possible to address the needs of small-scale or one-off users who typically require little dynamism in their virtual resources, and who are perfectly happy to set up their virtual resources by hand, even one-by one, especially if they are offered a comprehensive graphical interface to do it. GUOCCI integrates not only with OCCI-compliant cloud sites, but also with the EGI Application Database and with authentication technologies used in academic federated clouds, namely with Virtual Organization Membership Services (VOMS). With that, the considerable resources available for instance in the EGI Federated Cloud are open up to all such small-scale or beginning users. This article introduces in greater depth the reasoning behind developing GUOCCI, and details the architecture of the product, making it into an example OCCI client implementation.

International Symposium on Grids and Clouds (ISGC) 2017
5 – 10 March, 2017
Taipei, Taiwan

[*]Speaker.

1. Introduction

This paper introduces a solution developed to ease cloud adoption and lower the threshold of entry into an OCCI-based federated cloud for small-scale users. To convey the message, Section 2 starts by explaining the target federation and its user-facing interfaces. Section 3 outlines the typical use cases expected from small-scale or one-off users. Section 4 explains the actual implementation of the GUOCCI component, which is to be understood not only as a specific product, but also as a model implementation of a federated cloud client, to be followed when large user groups implement their own field-specific workflow engines. Section 5 explains how the work presented herein fits with development effort done by other teams, Section 6 outlines future plans, and finally Section 7 sums up the information presented.

2. Cloud Federation Status

The EGI Federated Cloud [1] makes available resources provided by several organizations in Europe. It is heterogeneous, and individual connected sites currently employ products coming from one of three different cloud management framework families (Open Nebula, Open Stack or Synnefo). The federation relies on the use of uniform interfaces to perform essential operations in:

- identity federation
- virtual appliance distribution and per-site service availability
- accounting and monitoring
- virtual resource management

This paper relates closely to two of those interfaces, accessed directly by all users but especially those with small-scale use scenarios:

- Virtual resource management, i.e., instantiation and control of resources such as virtual machines, virtual disks or virtual networks, is performed through the primary user-facing interface, compliant with the Open Cloud Computing Interface (OCCI) Standard (Section 2.1).
- An umbrella service for the cloud federation, aware of resources available across individual connected sites, falls within the portfolio of services offered by the EGI Application Database (AppDB – see Section 2.2).

2.1 OCCI

OCCI is a standard developed by the Open Grid Forum [2], currently available in version 1.2. It primarily defines the standard way to manage virtual resources (typically Compute, Storage and Network) within a cloud site. There are extensions that standardize other aspects of user-cloud interaction (SLAs, monitoring) and others are in development (Security Groups, Availability Zones, etc.).

From the perspective of small-scale cloud use, the most important principle is that OCCI is intended for communication with a single cloud site. For a user this means that OCCI only comes into play once they know what site they need to use, i.e., what site has the resources to satisfy their need. (This is not unlike HTTP, which is only useful once the user knows what site they want to access, but once they do, they can rely on the standard protocol regardless of the underlying server-side implementation.) They may have this information already, obtained from a community source. If not, they must start their workflow a level higher – by selecting a suitable site through the AppDB.

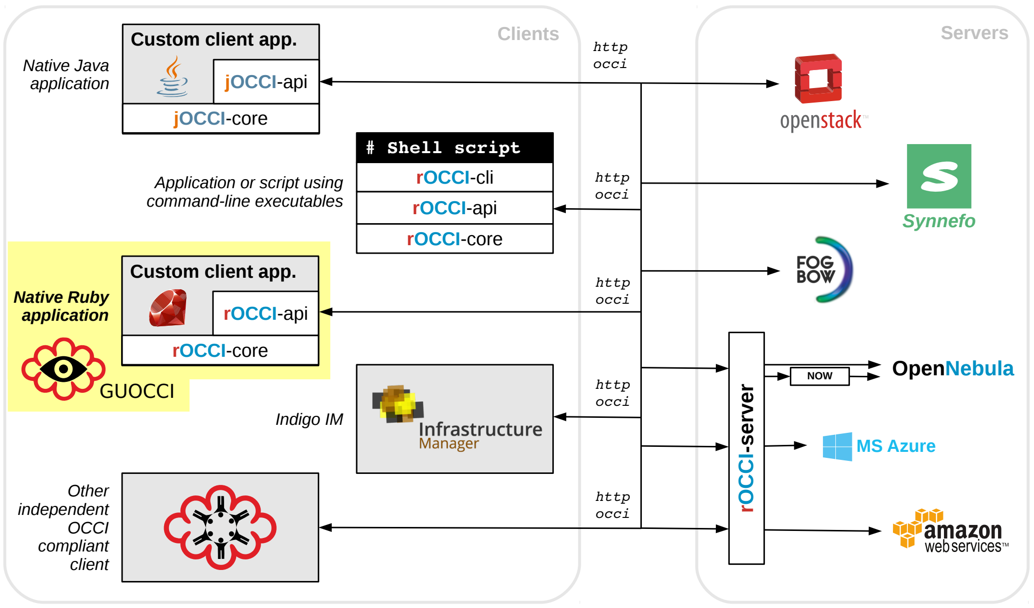

Figure 1: GUOCCI in the ecosystem of OCCI-enabled components, shown here as an example (highlighted) of a native Ruby application relying on rOCCI-core and rOCCI-api libraries.

2.2 AppDB

The EGI Application Database [3] provides a range of services to EGI grid and cloud users. From the point of view of a small-scale user in the federated cloud the most important service is the Cloud marketplace. It is a catalogue of cloud appliances organized by virtual organization. Not all appliances are available in all sites since each site only supports a subset of virtual organizations.

Hence the steps to be taken by a user are:

1. look up an appliance they wish to use in AppDB,
2. pick a site (from a list offered by the AppDB) that offers that appliance, and choose virtual machine size template,
3. instantiate virtual machines over OCCI using the site endpoint and appropriate template IDs obtained in step 2.

3. Small-scale Usage Scenarios in Federated Clouds

Small-scale usage typically entails occasional, short term or one-off use of cloud resources, typically by a small user group or even a single user. They are usually inexperienced, often encountering the cloud for the first time, at least as a computing resource. They lack the technical skill or personnel to have a custom client application developed for their specific purpose. Typical scenarios for small-scale use are briefly discussed below.

3.1 Alien or Disposable System

This use case is the least cloud-dependent since it requires no cloud-specific features beyond the ability to obtain virtual machines. The user will instantiate a single instance for experimental use, such as trying out potentially dangerous procedures, or evaluating software products requiring specific operating system family, distribution or version.

3.2 Prototyping

The user uses a small number of instances to prototype computing platforms or workflows that are intended for subsequent use at a larger scale. They require limited resources but may be already using more advanced cloud features such as linking storage, contextualization, security group settings, etc. They may also require the ability to make snapshots of their virtual machines.

3.3 Scaling out Local Workflows

The user has a workflow set up on local resources, a small departmental cluster or often even on their own office PC, and needs to scale out to speed up the processing. They will instantiate multiple virtual machines of equal configuration, choosing a manageable number of instances, and start processing their workload. Even that will be done manually or semi-automatically. There will be very little dynamism in the allocation of resources due to the fact that all virtual resource management is done manually. The user will keep the resources allocated until the work is processed.

Among the usage patterns discussed here this one most closely resembles the actual intended use of an HPC cloud. A user following this pattern may only be lacking better technical and field-specific support to utilize large sets of resources. But as long as they are managing their own resources in a manual fashion they may be considered small-scale and benefit from simple UIs.

4. Implementation

GUOCCI is designed to support operation in a Federated Cloud environment where information about sites such as endpoints, appliances and flavors is provided by the AppDB. This mode of operation is illustrated in figure 2.

The backend is implemented in Ruby on Rails and it is responsible for communication with cloud site endpoints and with the AppDB. Communication with site endpoints is realized over the OCCI protocol. Libraries used for this are *rOCCI-core* [4] and *rOCCI-api* [5]. These libraries provide the necessary functionality to manage resources in each target cloud site. GUOCCI relies on the rOCCI libraries to perform operations ranging from simple appliance and flavor listing to resource creation and manipulation. The backend exposes a REST API designed in Swagger [6], which provides necessary functions to the frontend. The frontend is implemented with AngularJS [7], a JavaScript-based open-source frontend web application framework.

GUOCCI aims to provide a simple, usable authentication mechanism, but also capitalize on strong security tools currently used in Federated Cloud. This is done by an external component,

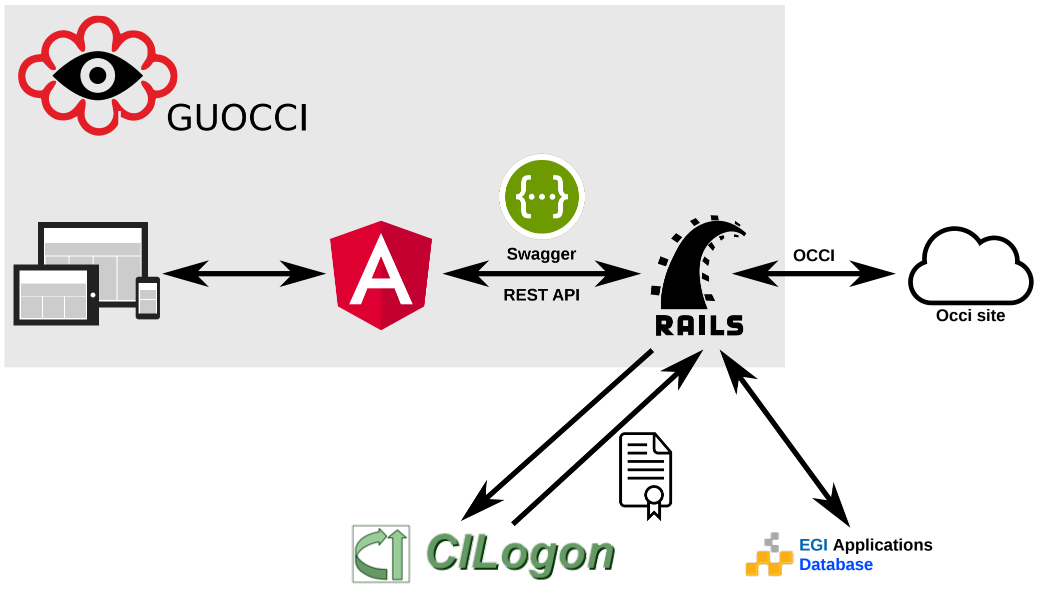

Figure 2: Internal components of GUOCCI in relation to external services provided by the federated cloud environment.

CILogon [8], which implements authentication to the application. CILogon allows each user to authenticate by the means they are used to, such as institutional login, Google account, etc. Once the authentication process is successfully finished, CILogon provides the back-end is provided with a VOMS-proxy certificate, which is used for authorization at the cloud sites.

5. Related Work

Other general-purpose OCCI clients for federated clouds, specifically for the EGI Federated Cloud Platform, exist but the approach they take is somewhat different. Together with GUOCCI, though, these products sufficiently cover the needs of the target audience (i.e., small-scale cloud users).

5.1 rOCCI-cli

rOCCI-cli is the first general-purpose OCCI client in existence. It is extensively used across the EGI Federated Cloud. As suggested by its title, it is a command line-based client, which exposes the full range of functionality over OCCI, and even offers additional convenience functions (such as blocking execution until a freshly created resource becomes ready). Workflow engine developers often choose to wrap around the command-line client rather than depend on underlying OCCI libraries.

On the other hand, it is somewhat awkward for human use, especially for beginner users who often complain that it exposes the underlying logic of OCCI too openly. In that respect GUOCCI can be viewed as a graphical alternative to *rOCCI-cli*.

5.2 EGI AppDB VMOps Dashboard / Indigo IM

The VMOps Dashboard [9] is presented as an extension to the AppDB itself. That, however, makes it exclusively compute-centric, i.e., one cannot independently manage other types of resources (storage, network).

In fact, the VMOps Dashboard – based on the Infrastructure Manager [10] developed by the INDIGO-DataCloud project – is being deployed as Platform as a Service (PaaS) manager that one can use to set up and manage larger topologies consisting of multiple computing resources, and the per-VM instance control it gives is the lowest level of management it provides.

Compared to it, GUOCCI has the ambition to be a full-fledged OCCI client at the IaaS level, supporting the scenario where users may choose sites based on what appliances they provide, or where their instances are already running, but once that choice is made, GUOCCI is intended to be used as a standard interface to that single site, and also to provide extensibility and adaptability to specific use cases.

5.3 GUOCCI Prototype

To make the picture complete the prototype for GUOCCI [11] also needs to be mentioned here. Being a prototype it offers only a subset of features but it has already been adopted by user groups in the ELIXIR project [12] to develop their workflows, which shows the viability of the idea of a simple graphical OCCI client, although the prototype needs to be – naturally – replaced by a regular release before it is put into production.

6. Future Work

Since the ambition of the development team is to maintain GUOCCI as a full-fledged OCCI client, most future challenges will stem from the continued development of the OCCI standard.

Firstly there are new specifications arriving in the near future. Several extensions are currently in preparation, namely OCCI Security Groups, OCCI IP Reservations and OCCI Availability Zones, to name just those the GUOCCI development team is aware of. They all introduce new resource types that must be correctly presented in the graphical interface at the right moment.

Secondly there are developments to the main standard itself. For example, the recently introduced JSON rendering, which is scheduled to replace the current text rendering. At first sight it does not affect the current implementation in any way, because the transport protocol is implemented by the underlying library. But the use of JSON rendering will allow bulk operations, which were previously impossible. And that will be a topic to investigate.

Its is important to state here that tasks outlined above are not as much of a challenge from a technical standpoint. The definitions and procedures are clear. What makes them a real challenge is to maintain the ergonomics of the interface, i.e., to present the user with a full-featured interface without overloading their perception. That is a future challenge all by itself.

But finally there will also be developments to enable GUOCCI's use outside the scope of a federated cloud. It will be possible to use GUOCCI as a simple, standardized interface to a non-federated site. Available appliances will be obtained over OCCI directly from the target site, rather than from the AppDB. The user's workflow will be simplified by removing the site choice step.

Since GUOCCI is a Web-based application, there may be several differently configured instances used against a single endpoint. Each may be configured to support a different user group, for instance by further filtering or augmenting the choice of options users will be given.

7. Summary

This paper has briefly presented the development of a graphical user interface to OCCI-enabled clouds. The product, GUOCCI, has been developed to provide small-scale users with a tool that is simple to use, and easy to customize for user groups with set tools and procedures.

The amount of effort put into the development proves that developing a dedicated client is beyond all but the most powerful user groups. Building on top of a generic, general-purpose client such as GUOCCI will make this much easier even for smaller communities, and single, small-scale users can successfully rely on the basic client as it is.

Acknowledgements

Access to computing and storage facilities owned by parties and projects contributing to the National Grid Infrastructure MetaCentrum, provided under the programme "Projects of Large Research, Development, and Innovations Infrastructures" (CESNET LM2015042), is greatly appreciated.

References

[1] European Grid Infrastructure, *Federated Cloud*, [Online] Available: https://www.egi.eu/infrastructure/cloud/ [Accessed: April 5, 2017].

[2] *Open Grid Forum*, [Online] Available: http://www.ogf.org [Accessed: April 5, 2017].

[3] European Grid Infrastructure, *Applications database*, [Online] Available: https://www.egi.eu/services/catalogue/appdb.html [Accessed: April 5, 2017].

[4] MetaCentrum, *rOCCI-core - A Ruby OCCI Framework*, [Online] Available: https://github.com/arax/rOCCI-core [Accessed: April 5, 2017].

[5] MetaCentrum, *rOCCI-api - A Ruby OCCI Framework*, [Online] Available: https://github.com/arax/rOCCI-api [Accessed: April 5, 2017].

[6] SmartBear Software, *Swagger*, [Online] Available: http://swagger.io/ [Accessed: April 5, 2017]

[7] Google, *AngularJS*, [Online] Available: https://angularjs.org/ [Accessed: April 5, 2017].

[8] National Center for Supercomputing Applications, *The CILogon project*, [Online] Available: http://www.cilogon.org/ [Accessed: April 5, 2017].

[9] EGI, *EGI AppDB VMOps Dashboard*, [Online] Available: https://dashboard.appdb.egi.eu/vmops [Accessed: April 5, 2017].

[10] INDIGO-DataCloud, *IM – Infrastructure Manager (With TOSCA Support)*, [Online] Available: https://www.gitbook.com/book/indigo-dc/im [Accessed: April 5, 2017].

[11] Boris Parák, *A proof-of-concept GUI for (r)OCCI*, [Online] Available: https://github.com/arax/guocci [Accessed: April 5, 2017].

[12] ELIXIR, *A distributed infrastructure for life-science information*, [Online] Available: https://www.elixir-europe.org [Accessed: April 5, 2017].