Abstract: FutureGateway a new multi-infrastructure framework for customisable Science Gateways

Marco Fargetta and Riccardo Bruno
Italian National Institute of Nuclear Physics
Division of Catania
Catania, Italy
{marco.fargetta, riccardo.bruno}@ct.infn.it

Roberto Barbera
Department of Physics and Astronomy of the University
of Catania and INFN
Catania, Italy
roberto.barbera@ct.infn.it

Marcin Plociennik, Michal Owsiak, Tomasz Zok
Poznanskie Centrum
Superkomputerowo-Sieciowe
Poznan, Poland
{marcinp, michalo, tzok}@man.poznan.pl

Giovanni Aloisio and Sandro Fiore
Fondazione Centro Euro-Mediterraneo sui
Cambiamenti Climatici
Lecce, Italy
{giovanni.aloisio, sandro.fiore}cmcc.it

Keywords—Science Gateway; Framework; INDIGO-DataCloud, Multi-infrastructure; REST

ABSTRACT

INDIGO-DataCloud [1] (INtegrating Distributed data Infrastructures for Global ExplOitation) is an EU-funded project, which aims at developing a data & computing platform targeted at scientific communities, deployable on multiple hardware, and provisioned over hybrid e-Infrastructures. It is based on open source solutions addressing scientific challenges in the grid, cloud and HPC/local infrastructures and, in the case of cloud platforms (IaaS, PaaS and SaaS). SaaS solutions are exposed to end-users through Science Gateways (SG) and a new standard-based framework to create SG has been developed in the context of INDIGO: the FutureGateway (FG) [2]. The FG is a Science Gateway framework that provides both a presentation layer and a back-end API service.

The FG framework is an evolution of the CSGF [3] which aims at enabling the creation of SGs based on different technologies, such as web, mobile applications and others. The new framework integrates some existing tools to support specific activity (e.g. authentication) of the framework and implements a new service to support the concept of application lifecycle inside a Science Gateway. In this context an application can be a grid job, a cloud service or any activity performed on a remote infrastructure. The service responsible to manage the applications on behalf of the users executes behind the SG and communicates with all the interfaces implemented for the user. A set of components, developed for the Liferay enterprise portal framework, are provided to simplify the development of User Interface (UI) and portals. Libraries for Android, iOS and other platform are under development and development releases already available.

The backend service managing the applications is the main component of the new framework architecture and it is responsible to interact with many distributed computing infrastructures using different protocols. This consists of two separate daemons: the APIServer front-end and the APIServerDaemon. The former is responsible for processing incoming REST API calls. It essentially manages a queue table where API requests are stored. The queue table holds the interactions among the SG interfaces and DCI resources and it also allows to perform maintenance and accounting operations. Moreover the use of a queue table allows to decouple the front-end from the back-end operation increasing the responsiveness for the user and the system scalability, both vertically by increasing the consumer and horizontally by splitting the queue.

The APIServerDaemon is a process continuously polling the queue table. As soon as new entries are available, they will be delivered to the correct executor interface (EI) for the final processing. Currently, there are two EIs available: the Grid & Cloud Engine and the ToscaIDC. The former uses JSAGA [4] to target destination infrastructures supported by the available JSAGA adaptors, while the latter directly accesses TOSCA Orchestrator REST API developed by INDIGO.

The back-end service exposes a small set of REST APIs [5], which allow to manage three different resources: the '*application*' describing the activity to perform on top of the DCI, the '*infrastructure*' describing how the application accesses DCI resources, and finally the 'task' which is an application instance. Portal components, mobile applications and even workflow engines can access the APIs to operate with tasks. The lifecycle of the task is entirely managed by the API service so that the user-level code has only to check the status of the task and gets its output when available.

Authentication and authorisation are based on tokens and support several standards.

The FG has been tested in several use cases belonging to various disciplines, e.g. climate change, molecular dynamics of proteins and bioinformatics analysis. The initial tests demonstrate the easy integration with already established scientific applications and related communities. They were able to extend their workflows to the cloud with relatively few changes on existing front-ends.

ACKNOWLEDGMENTS

This work is supported by the INDIGO-DataCloud project funded by the European Union within its Horizon 2020 Programme (Grant Agreement no. 653549).

REFERENCES

[1] The INDIGO-DataCloud project, 2017. https://www.indigo-datacloud.eu

[2] The FutureGatewway framework, 2017. https://www.indigo-datacloud.eu/future-gateways-programmable-scientific-portal.

[3] Valeria Ardizzone, Roberto Barbera, Antonio Calanducci, Marco Fargetta, E. Ingrà, Ivan Porro, Giuseppe La Rocca, Salvatore Monforte, R. Ricceri, Riccardo Rotondo, et al. The DECIDE science gateway. *Journal of. Grid Computing*, 10 (4) (2012), pp. 689–707

[4] JSAGA, http://software.in2p3.fr/jsaga/dev/

[5] FutereGateway REST APIs, 2017. http://docs.fgapis.apiary.io.