



## Put your models in the web - less painful

**Nils Brinckmann**<sup>1</sup>, Massimiliano Pittore<sup>1,2</sup>, Matthias Ruster<sup>1</sup>, Benjamin Proß<sup>3</sup>, and Juan Camilo Gomez-Zapata<sup>1,4</sup>

<sup>1</sup>Helmholtz Centre Potsdam GFZ German Research Centre for Geosciences, Potsdam, Germany

<sup>2</sup>Eurac Research, Bolzano, Italy

<sup>3</sup>° North, Münster, Germany

<sup>4</sup>University of Potsdam, Potsdam, Germany

Today's Earth-related scientific questions are more complex and more interdisciplinary than ever, so much that is extremely challenging for single-domain experts to master all different aspects of the problem at once. As a consequence, modular and distributed frameworks are increasingly gaining momentum, since they allow the collaborative development of complex, multidisciplinary processing solutions.

A technical implementation focus on the use of modern web technologies with their broad variety of standards, protocols and available development frameworks. RESTful services - one of the main drivers of the modern web - are often sub optimal for the implementation of complex scientific processing solutions. In fact, while they offer great flexibility, they also tend to be bound to very specific formats (and often poorly documented).

With the introduction of the Web Processing Service (WPS) specifications, the Open Geospatial Consortium (OGC) proposed a standard for the implementation of a new generation of computing modules overcoming most of the drawbacks of the RESTful approach. The WPS allow a flexible and reliable specification of input and output formats as well as the exploration of the services' capabilities with the GetCapabilities and DescribeProcess operations.

The main drawback of the WPS approach with respect to RESTful services is that the latter can be easily implemented for any programming language, while the efficient integration of WPS is currently mostly relying on Java, C and Python implementations. In the framework of Earth Science Research we are often confronted with a plethora of programming languages and coding environments. Converting already existing complex scientific programs into a language suitable for WPS integration can be a daunting effort and may even result in additional errors being introduced due to conflicts and misunderstandings between the original code authors and the developers working on the WPS integration. Also the maintenance of these hybrid processing components is often very difficult since most scientists are not familiar with web programming technologies and conversely the web developers cannot (or do not have the time to) get adequately acquainted with the underlying science.

Facing these problems in the context of the RIESGOS project we developed a framework for a Java-based WPS server able to run any kind of scientific code scripts or command line programs. The proposed approach is based on the use of Docker containers encapsulating the running processes, and Docker images to manage all necessary dependencies.

A simple set of ASCII configuration files provides all information needed for WPS integration: how to call the program, how to give input parameters - including command line arguments and input files - and how to interpret the output of the program - both from stdout and from serialized files. There are a bunch of predefined format converters and we also include mechanisms for extensions to allow maximum flexibility.

The result is a encapsulated, modular, safe and extendable architecture that allows scientists to expose their scientific programs on the web with little effort, and to collaboratively create complex, multidisciplinary processing pipelines.