How the openEO project unifies access to big Earth Observation data processing platforms

Christoph Friedrich, Institute for Geoinformatics, University of Münster, Germany
c.friedrich@uni-muenster.de

Edzer Pebesma, Institute for Geoinformatics, University of Münster, Germany
edzer.pebesma@uni-muenster.de

Matthias Mohr, Institute for Geoinformatics, University of Münster, Germany
m.mohr@uni-muenster.de

Abstract

Due to the increasing difficulties in handling today's satellite imagery, more and more Earth observation (EO) processing calculations no longer happen on local machines with locally stored data, but happen in the cloud. Currently however, each provider defines their own interface and clients use a multitude of programming languages to access them. This makes cross-system usage bothersome and hinders reproducibility.

To solve this problem, the openEO project develops "an open API to connect R, Python, JavaScript and other clients to big Earth observation cloud back-ends in a simple and unified way" (www.openeo.org). The aim is to develop working software packages – in the form of client libraries and server drivers – that enable EO data users to uniformly access processing back-ends such as Sentinel Hub, GeoPySpark, GRASS GIS, WCPS/rasdaman, or Google Earth Engine. In a way, we strive to become for the domain of EO data processing what GDAL is for the domain of GIS file formats: a powerful mediator in the growing jungle of inoperable systems.

The core of the project is the development of an API through which clients and servers communicate. While this work package is coordinated by WWU, the specification is agreed upon by all nine project partners stemming from seven different countries and including universities, research institutes and industry partners. On top of this API, the software packages are developed and recurrently integrated into one big interoperable system of systems.

Among others, challenges are how to:

(1) incorporate all domain aspects into a simple yet universal interface – the API has to cover as many use cases as possible (including those of commercial providers), but should nevertheless not become overly complex,

(2) enable user-defined functions (UDFs) – we do not want to constrain users to a limited set of available processing functions, but instead allow them to use the versatility of custom scripts (e.g. written in Python or R) like in locally executed workflows.

(3) realise efficient access to large chunks of data on the server-side – moving big data is difficult, so concepts like the data cube are applied to both raster and vector data,

(4) validate back-ends against each other – all openEO-compatible back-ends should return identical output when given the same query,

(5) assure user adoption of this “two-sided market” – the new API will only experience widespread adoption if both clients and providers implement it *and* can discover each other.

During last year’s Geospatial Sensor Web Conference, we presented the first outcomes of a rather young project that had just presented its proof-of-concept. This year, we would like to provide an update on how we have progressed over the last year, including insights into conceptual and technical considerations, lessons-learned during the project, and a live demonstration.