Enabling “LiDAR data processing” as a service in a Jupyter environment

Spiros Koulouzis\textsuperscript{1,2}, Yifang Shi\textsuperscript{1,3}, Yuandou Wan\textsuperscript{2}, Riccardo Bianchi\textsuperscript{1,2}, Daniel Kissling\textsuperscript{1,3}, and Zhiming Zhao\textsuperscript{1,2}

\textsuperscript{1}LifeWatch ERIC, vLab&Innovation Center
\textsuperscript{2}Multiscale Networked Systems, University of Amsterdam
\textsuperscript{3}Institute for Biodiversity and Ecosystem Dynamics (IBED)

Airborne Laser Scanning (ALS) data derived from Light Detection And Ranging (LiDAR) technology allow the construction of Essential Biodiversity Variables (EBVs) of ecosystem structure with high resolution at landscape, national and regional scales. Researchers nowadays often process such data, and rapidly prototype using script languages like R or python, and share their experiments via scripts or more recently via notebook environments, such as Jupyter. To scale experiments to large data volumes, extra data sources, or new models, researchers often employ Cloud infrastructures to enhance notebooks (e.g. Jupyter Hub) or execute the experiments as a distributed workflow. In many cases, a researcher has to encapsulate subsets of the code (namely, cells in Jupyter) from the notebook as components to be included in the workflow. However, it is usually time-consuming and a burden for the researcher to encapsulate those components based on the workflow systems' specific interface, where the Findability, Accessibility, Interoperability and Reusability (FAIR) of those components are often limited. We aim to enable the public cloud processing of massive amounts of ALS data across countries and regions and make the retrieval and uptake of such EBV data products of ecosystem structure easily available to a wide scientific community and stakeholders.

We propose and develop a tool called FAIR-Cells, that can be integrated into the Jupyter Lab environment as an extension, to help scientists and researchers improve the FAIRness of their code. It can encapsulate user-selected cells of code as standardized RESTful API services, and allow users to containerize such Jupyter code cells and to publish them as reusable components via the community repositories.

We demonstrate the features of the FAIR-CELLS using an application from the ecology domain. Ecologists currently process various point cloud datasets derived from LiDAR to extract metrics that capture vegetation's vertical and horizontal structure. A new open-source software called ‘Laserchicken’ allows the processing of country-wide LiDAR datasets in a local environment (e.g.
the Dutch national ICT infrastructure called SURF). However, the users have to use the Laserchicken application as a whole to process the LiDAR data. The capacity of the given infrastructure also limits the volume of data. In this work, we will first demonstrate how a user can apply the FAIR-Cells extension to interactively create RESTful services for the components in the Laserchicken software in a Jupyter environment, to automate the encapsulation of those services as Docker containers, and to publish the services in a community catalogue (e.g. LifeWatch) via the API (based on GeoNetwork). We will then demonstrate how those containers can be assembled as a workflow (e.g. using Common Workflow Language) and deployed on the cloud environment (offered by the EOSC early adopter program for ENVRI-FAIR) to process a much bigger dataset than in a local environment. The demonstration results suggest that our approach’s technical roadmap can achieve FAIRness and behave good parallelism in large distributed volumes of data when executing the Jupyter-environment-based codes.