

# FAIR-Cells: An interactive Jupyter extension for containerizing scientific code as services for eScience work

*Monday, 2 November 2020 15:00 (15 minutes)*

**Abstract**—Researchers nowadays often rapidly prototype and share their experiments using notebook environments, such as Jupyter. To scale experiments to large data volumes, or high-resolution models, researchers often employ Cloud infrastructures to enhance notebooks (e.g., Jupyter Hub) or execute their experiments as a distributed workflow. In many cases, a researcher needs to encapsulate subsets of their code (namely, cells in Jupyter) from the notebook to the workflow. However, it is usually time-consuming and burdensome for the researcher to encapsulate those code subsets and integrate them with a workflow. This process limits the Findability, Accessibility, Interoperability, and Reusability (FAIR) of those components are often limited.

To address this issue, we propose and develop a tool called FAIR-Cells, that can be integrated into the Jupyter notebook as a Jupyter extension to help scientists and researchers improve the FAIRness of their code. FAIR-Cells can encapsulate user-selected cells of code as standardized RESTful API services, and allow users to containerize such Jupyter code cells and 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 Light Detection and Ranging (LiDAR) to extract metrics that capture vegetation's vertical and horizontal structure. A novel opensource software called 'Laserchicken' allows the processing of country-wide LiDAR datasets in a local environment (e.g., the Dutch national ITC infrastructure called SURF). However, users have to employ the Laserchicken application as a whole to process the LiDAR data. Moreover, the volume of data that Laserchicken can process is limited by the capacity of the given infrastructure. In this work, we demonstrate how a user can use the FAIR-Cells extension. Namely, 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 catalog (e.g., LifeWatch) via the API (based on GeoNetwork). We also demonstrate how those containers can be assembled as a workflow (using Common Workflow Language) and deployed on a cloud environment (offered by the EOSC early adopter program for ENVRI-FAIR) to process a much bigger data sets than in a local environment. The demonstration results suggest that the technical roadmap of our approach can achieve FAIRness and behave good parallelism in large distributed volumes of data when executing the Jupiter-environment-based codes.

**Primary authors:** Dr KOULOUZIS, Spiros (University of Amsterdam); Mrs WANG, Yuandou (Multiscale Networked Systems, University of Amsterdam); Dr KISSLING, W. Daniel (LifeWatch ERIC, vLab&Innovation Center, Institute for Biodiversity and Ecosystem Dynamics (IBED)); Dr ZHAO, Zhiming (Multiscale Networked Systems, University of Amsterdam)

**Presenters:** Dr KOULOUZIS, Spiros (University of Amsterdam); Mrs WANG, Yuandou (Multiscale Networked Systems, University of Amsterdam); Dr KISSLING, W. Daniel (LifeWatch ERIC, vLab&Innovation Center, Institute for Biodiversity and Ecosystem Dynamics (IBED)); Dr ZHAO, Zhiming (Multiscale Networked Systems, University of Amsterdam)

**Session Classification:** Demos 2