



## **Lessons learned while creating Open Source Scientific Software in eWaterCycle and MAGIC**

Niels Drost (1), Wim Som de Cerff (2), Nick van de Giesen (3), Rolf Hut (3), and Maarten Plieger (2)

(1) Netherlands eScience Center, Amsterdam, Netherlands (n.drost@esciencecenter.nl), (2) Royal Netherlands Meteorological Institute, R&D Observations and Data technology, Netherlands, (3) Delft University of Technology, Faculty of Civil Engineering and Geoscience, Netherlands

At the Netherlands eScience Center we are involved in numerous projects with scientists, enabling them to go further and do more and better science through the use of digital technology. Perhaps surprisingly, problems faced are often not technical in nature. In this session we will showcase lessons learned in creating open science software platforms for a number of projects at the eScience Center. We will particularly focus on two projects: eWaterCycle and MAGIC.

eWaterCycle is a project aiming to create an Global Hydrological forecasting system, using only open source software and open data. The system has been up and running for a few years, and we are in the process of "FAIRifying" the results of eWaterCycle through a European Open Science Cloud (EOSC) demonstrator project. We will show lessons learned, and for some of the harder lessons show how we are trying to improve matters. We will also show plans for the eWaterCycle II follow-up project, where we are heavily focusing on creating a global modelling community, and open science is critical for the success of the project. We are actively seeking input for this project from the community.

As a second project we will showcase lessons learned in the Metrics and Access to Global Indices for Climate Projections(MAGIC) project, part of the Copernicus Climate Change Service (C3S). The MAGIC project is developing a system allowing users to visualise and analyse Petabytes of climate model data without having to download them to their own machine. MAGIC brings together software from a large number of scientists and organisations, giving some unique challenges for creating re-usable software, a prerequisite for open science. To face these challenges we heavily rely on standards and best practices for creating open source software.

By using these two projects as examples, we will highlight some lessons learned in creating software for Open Science. Many of these lessons apply to most of not all projects at the eScience Center, and we argue they can be applied throughout science.