

EGU2020-1612

<https://doi.org/10.5194/egusphere-egu2020-1612>

EGU General Assembly 2020

© Author(s) 2022. This work is distributed under the Creative Commons Attribution 4.0 License.



Building Web Processing Services with Birdhouse

Carsten Ehbrecht¹, Stephan Kindermann¹, Ag Stephens², and David Huard³

¹DKRZ, Hamburg, Germany (ehbrecht@dkrz.de)

²CEDA, Didcot, UK (ag.stephens@stfc.ac.uk)

³Ouranos, Montreal, Canada (huard.david@ouranos.ca)

The Web Processing Service (WPS) is an OGC interface standard to provide processing tools as Web Service.

The WPS interface standardizes the way processes and their inputs/outputs are described, how a client can request the execution of a process, and how the output from a process is handled.

Birdhouse tools enable you to build your own customised WPS compute service in support of remote climate data analysis.

Birdhouse offers you:

- A Cookiecutter template to create your own WPS compute service.
- An Ansible script to deploy a full-stack WPS service.
- A Python library, Birdy, suitable for Jupyter notebooks to interact with WPS compute services.
- An OWS security proxy, Twitcher, to provide access control to WPS compute services.

Birdhouse uses the PyWPS Python implementation of the Web Processing Service standard. PyWPS is part of the OSGeo project.

The Birdhouse tools are used by several partners and projects.

A Web Processing Service will be used in the Copernicus Climate Change Service (C3S) to provide subsetting

operations on climate model data (CMIP5, CORDEX) as a service to the Climate Data Store (CDS).

The Canadian non profit organization Ouranos is using a Web Processing Service to provide climate indices

calculation to be used remotely from Jupyter notebooks.

In this session we want to show how a Web Processing Service can be used with the Freva evaluation system.

Freva plugins can be made available as processes in a Web Processing Service. These plugins can be run

using a standard WPS client from a terminal and Jupyter notebooks with remote access to the

Freva system.

We want to emphasise the integrational aspects of the Birdhouse tools: supporting existing processing frameworks
to add a standardized web service for remote computation.

Links:

- <http://bird-house.github.io>
- <http://pywps.org>
- <https://www.osgeo.org/>
- <http://climate.copernicus.eu>
- <https://www.ouranos.ca/en>
- <https://freva.met.fu-berlin.de/>