

Birdhouse Supporting Web Processing Service for Climate Data

S. Kindermann¹, C. Ehbrecht¹, N. Hempelmann²

¹German Climate Computing Center (DKRZ)

²Le Laboratoire des Sciences du Climat et de l'Environnement, France (LSCE)

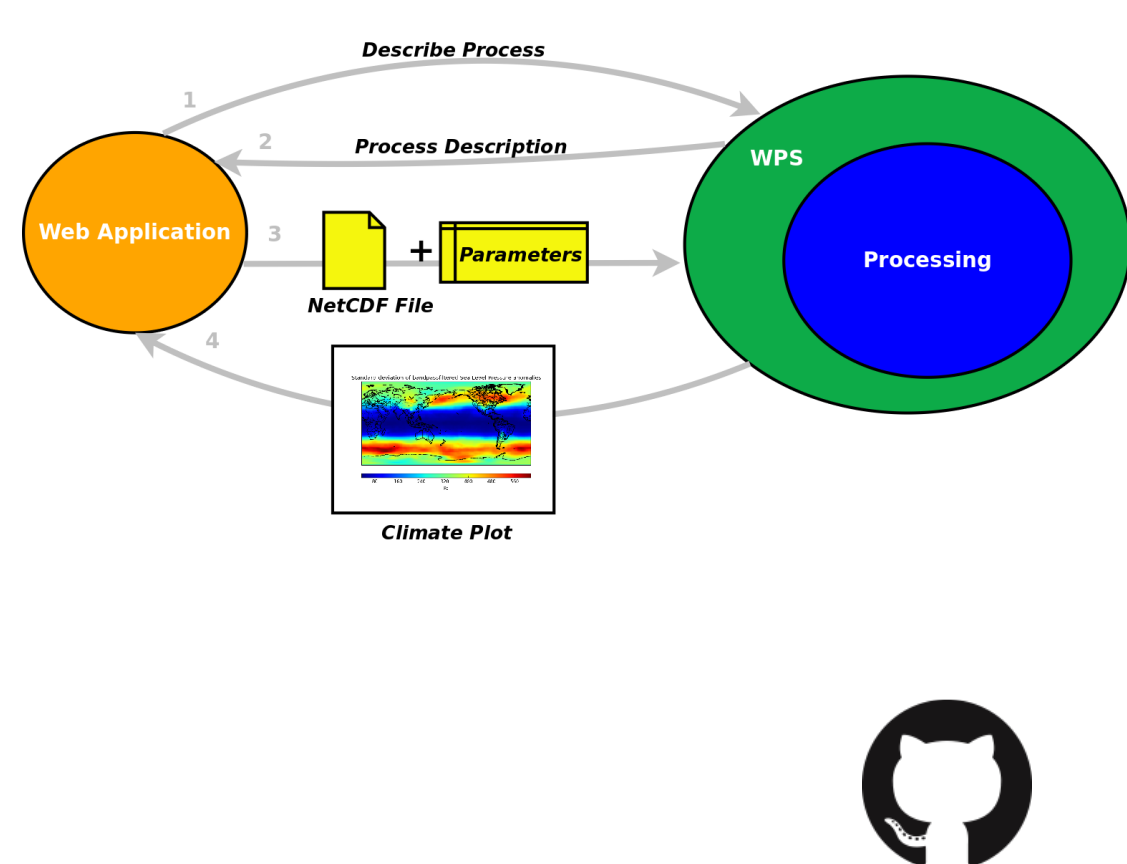
INTRODUCTION

Motivation



- The volume of climate data grows with the upcoming climate model projects (e.a. CMIP6).
- Even larger climate institutes and computing centers will not be able to keep all relevant data on one storage system.
- Climate processing services located at the data archives can be a valuable contribution to cope with the growing data challenge.
- A set of climate analyses processing tools are provided as services which can be accessed over the web.
- These web services are accessible to a larger range of scientific users.

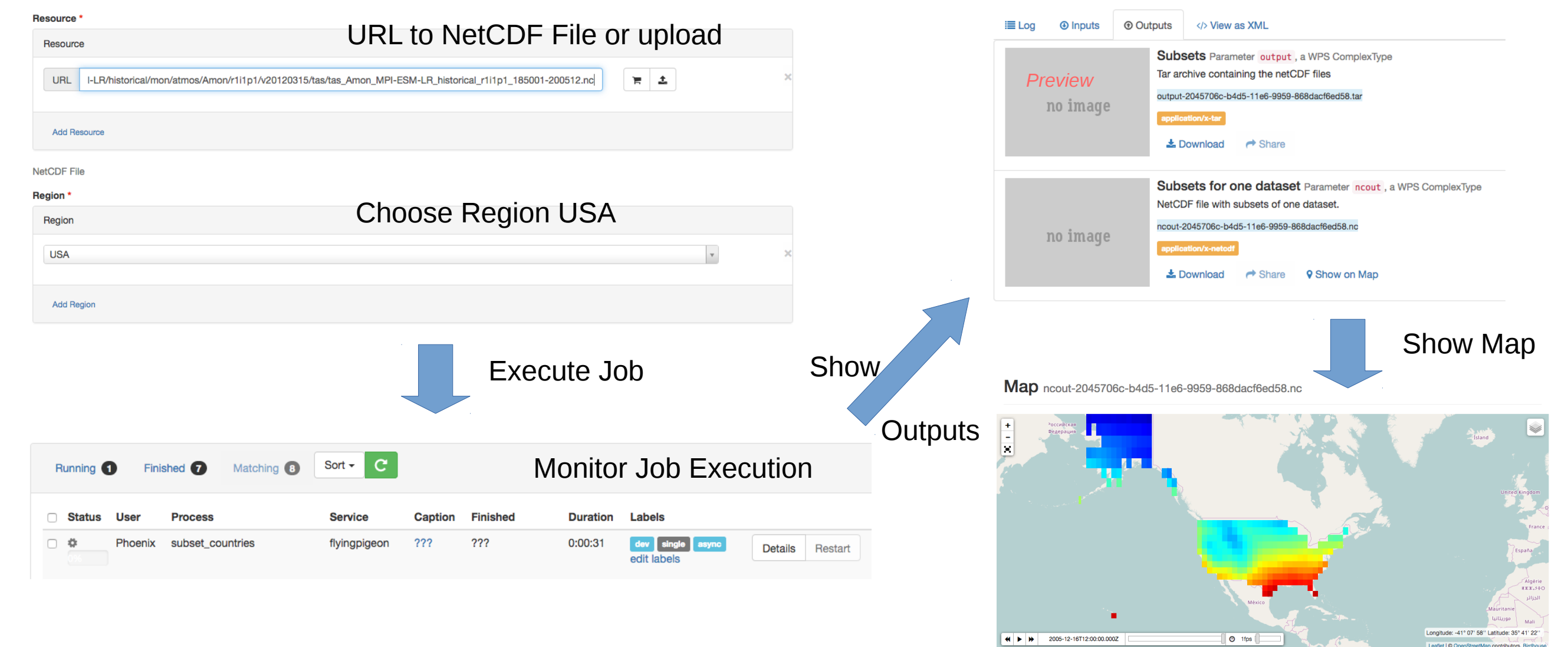
Birdhouse



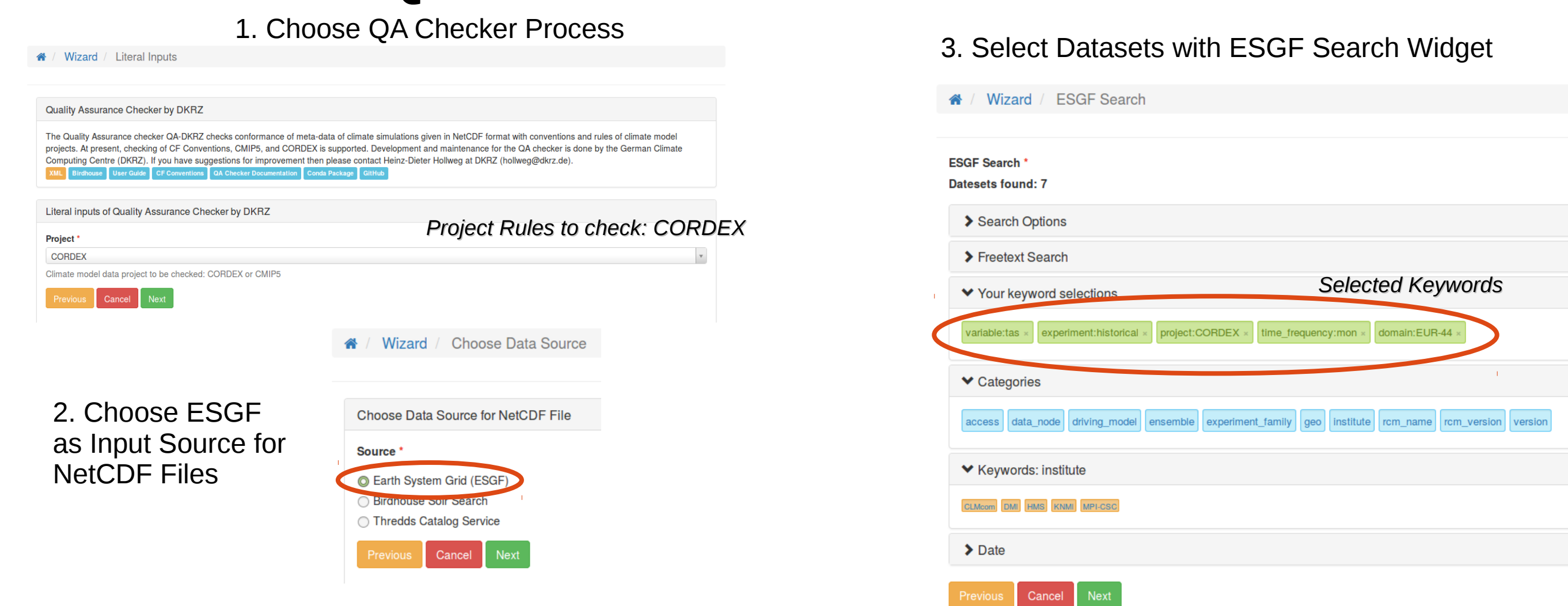
- Birdhouse uses the Web Processing Service (WPS) standard to realize services for climate processing tools.
- WPS is an open standard defined by the Open Geospatial Consortium (OGC) with several open source implementations.
- Birdhouse supports PyWPS but is not restricted to it (others: Zoo, GeoServer, COWS, 52North).
- Birdhouse provides components to setup a WPS for your own uses cases.
- The Birdhouse development uses GitHub and is open for contribution.

RUN PROCESSES

Run Subsetting Process with Region USA

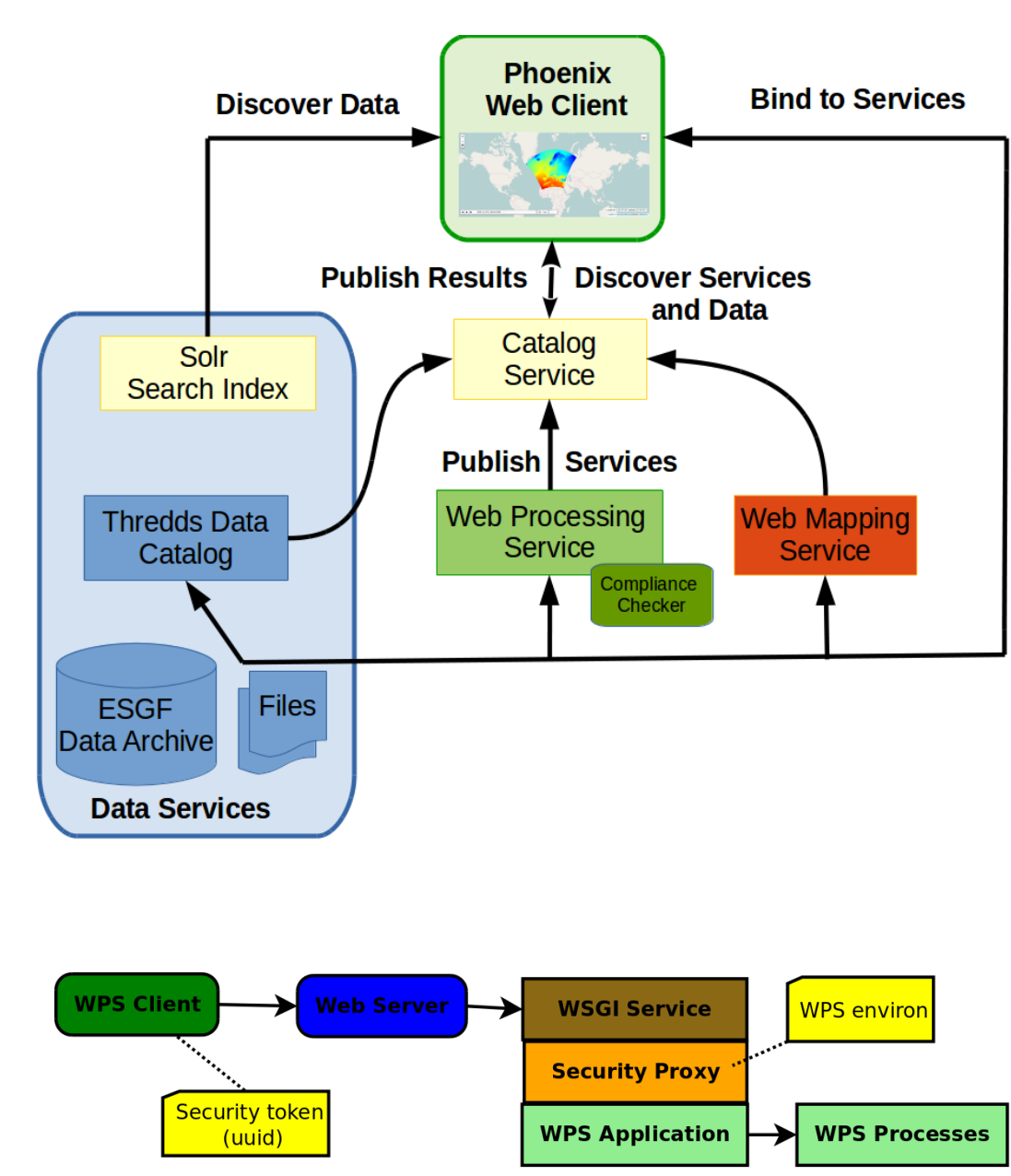


Wizard: run QA Checker on Data from ESGF



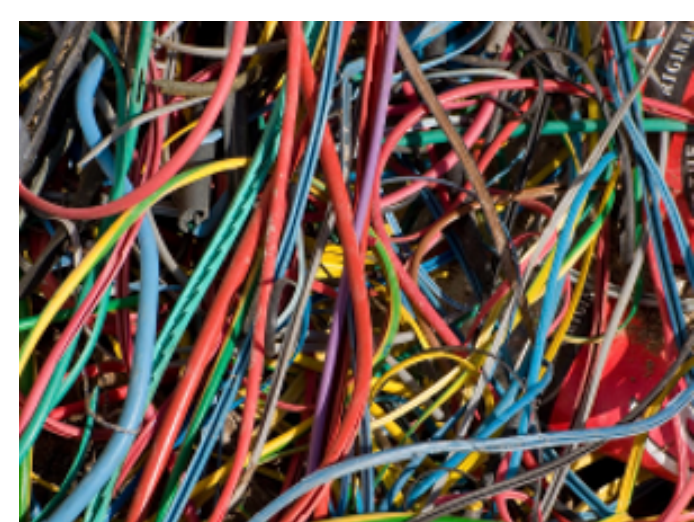
ARCHITECTURE

Overview of Birdhouse Components



- Birdhouse can interact with several OGC Web Processing Services.
- WPS services are registered at an OGC Catalog Service.
- The Phoenix web-client can display the processes of the registered WPS services. A process can be executed, monitored and the outputs are visualized.
- Phoenix uses a Web Map Service to show input and output NetCDF files on a Map.
- Data sources can be the ESGF Data Archive and external Thredds Data Servers.
- The Twitcher Security Proxy is an access token (string uuid) based security layer in front of OGC/WPS (or WMS etc) services.

Deployment with Conda and Buildout



- Birdhouse uses the conda package manager to setup an environment with all used software components.
- It uses Buildout to setup a WPS (PyWPS) with all services (supervisor, gunicorn, nginx) and configuration files.
- "Managing the Chaos"

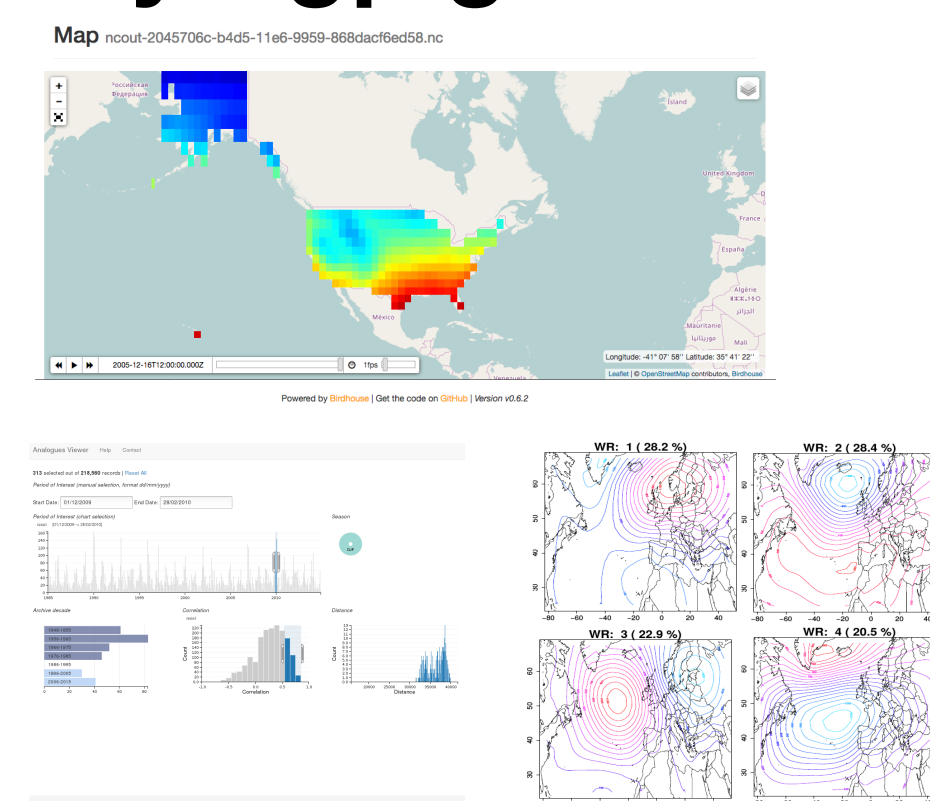
<http://birdhouse.readthedocs.io/en/latest/installation.html>



A Dockerfile is generated using the Buildout setup for each WPS service. Docker images are build automatically on DockerHub.

AVAILABLE PROCESSES

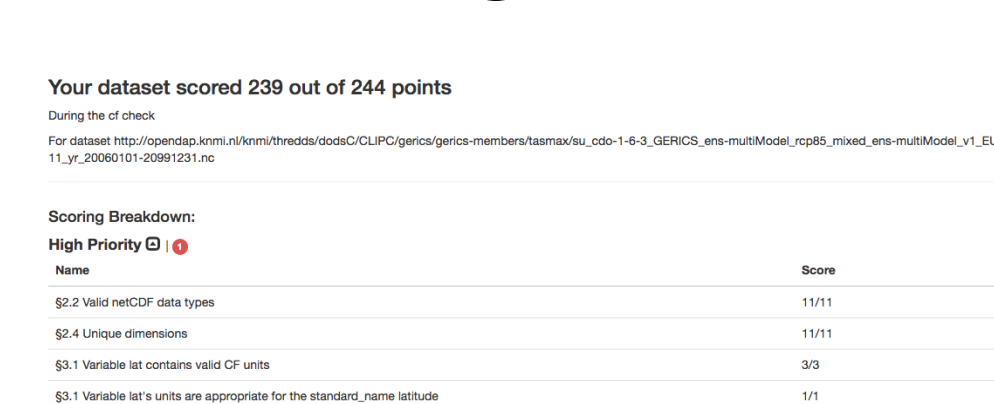
Flyingpigeon: Processes for Climate Impact



- Subsetting of pre-defined regions using OCGIS. <https://www.earthsystemcog.org/projects/openclimategis/>
- Calculation of Climate Indices using OCGIS/ICCLIM. <http://iclim.readthedocs.io/en/latest/>
- Statistical analyses of Extreme Events: Weather Regimes and Analogues detection of atmospheric circulations.

<http://flyingpigeon.readthedocs.io/en/latest/>

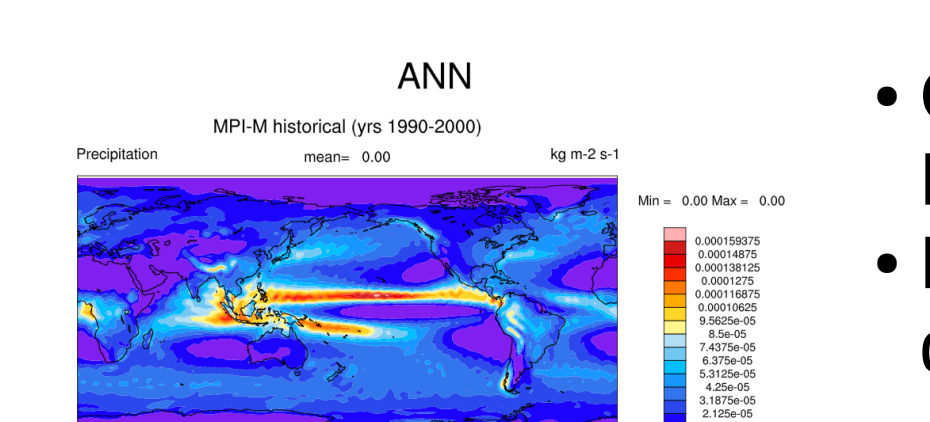
Hummingbird: Compliance Checker and CDO



- Compliance Checks for Climate and Forecast (CF) Conventions and project specific rules (CORDEX, CMIP5 and CMIP6)
- Processes for CDO operators like sinfo, monmax, sellatlonbox.

<http://birdhouse-hummingbird.readthedocs.io/en/latest/>

Processes for ESMValTool Diagnostics



- Currently only processes of tutorial diagnostics of the ESMValTool like a contour-plot are implemented.
- ESMValTool *namelists* are generated and the ESGF data coupling module is used.

<http://www.esmvaltool.org/>

OUTLOOK & REFERENCES

Using Birdhouse in Copernicus

- Using PyWPS 4.x: ready for WPS 2.0 (pause, resume, delete).
- Run processes by batch processing schedulers like SLURM and in Docker Containers.
- Delegation to SLURM and Docker is handled internally of PyWPS (new Feature in PyWPS-4).
- WPS Process definition and code is not changed when run as batch job or in a Docker container.
- Deployed at three sites: BADC, IPSL and DKRZ

References

- <http://bird-house.github.io/>
- Demo Installation: <https://mouflon.dkrz.de/>



ESGF F2F, Washington, 2016

