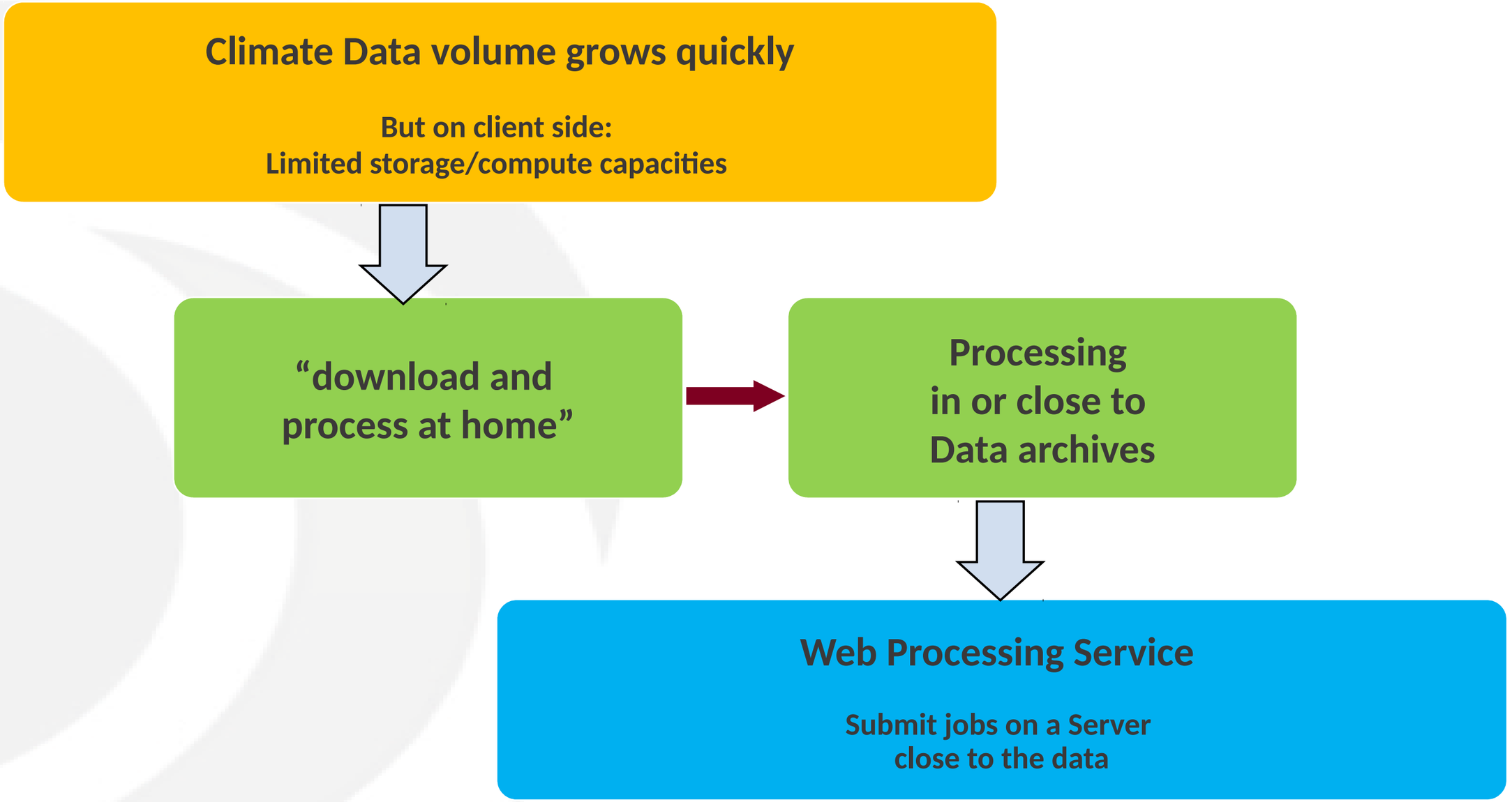
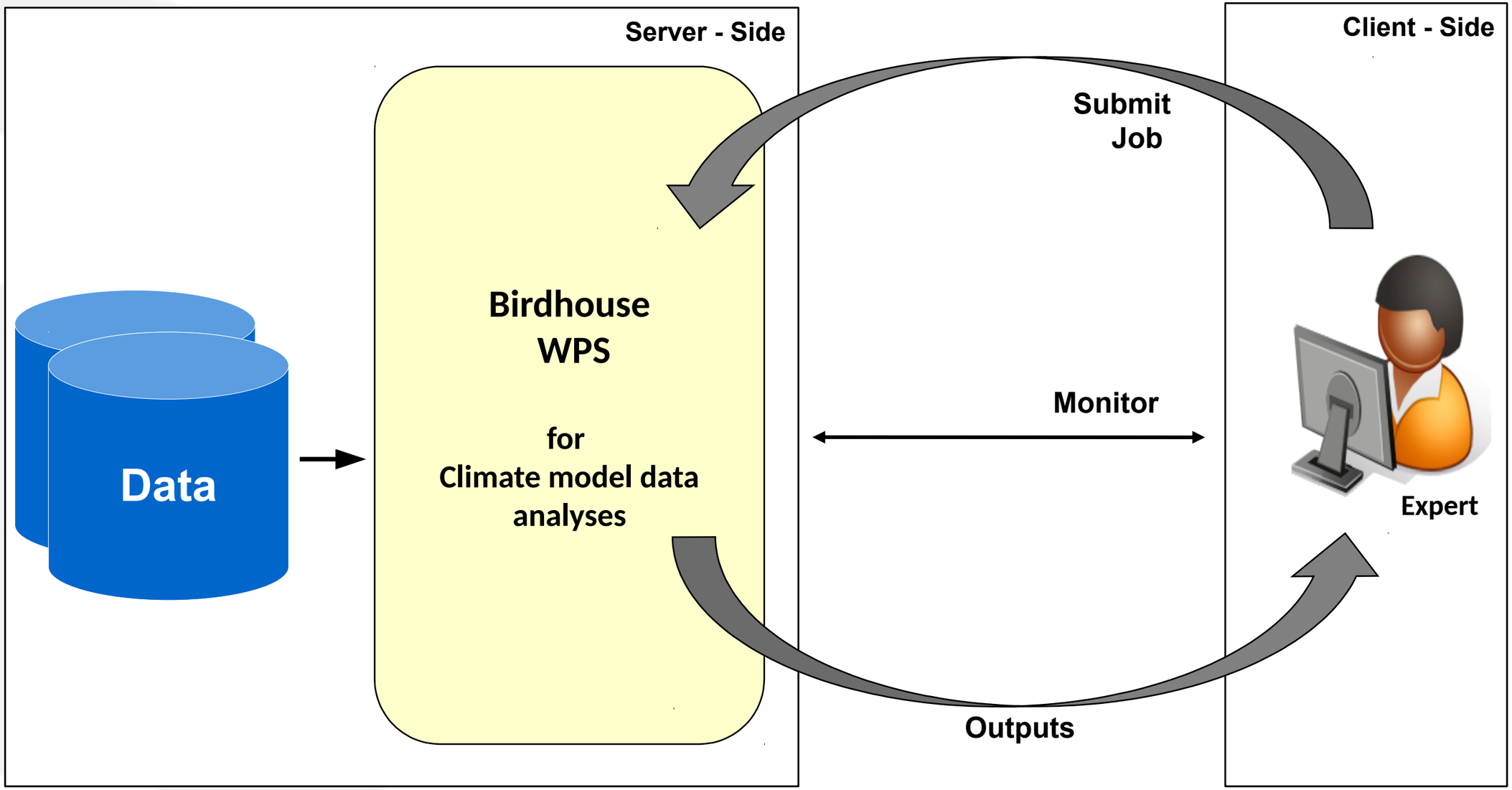
birdhouse:
supporting web processing services for climate data

Stephan Kindermann[1], Carsten Ehbrecht[1], Nils Hempelmann[2] et. al.

1. German Climate Computing Center, Germany
2. Le Laboratoire des Sciences du Climat et de l'Environnement, France


Climate Data volume grows quickly
But on client side:
Limited storage/compute capacities
“download and process at home”
Processing in or close to Data archives
Web Processing Service
Submit jobs on a Server close to the data

Server-Client Side

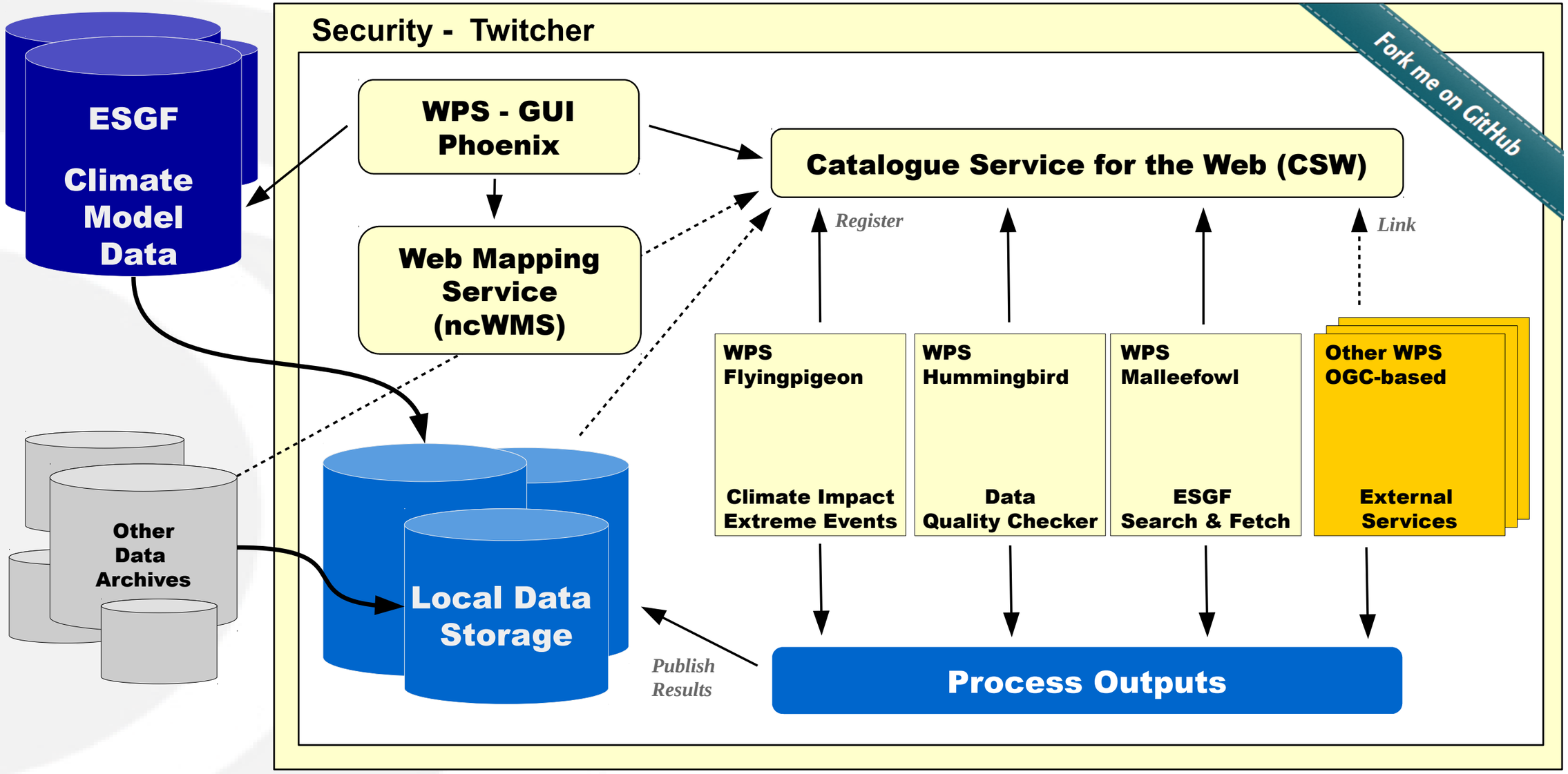


What does Birdhouse provide ?

- **Customizable installation of Web Processing Services using conda, buildout and ansible.**
- **Provides WPS as Docker Container.**
- **Web-based WPS client Phoenix.**
- **Terminal WPS client Birdy.**
- **Security Proxy Twitcher.**
- **Data Access : ESGF, Thredds, OpenStack Cloud, …**
- **WPS for quality checks, climate impact and extreme events.**
- **Supports PyWPS 3.x and 4.x … but not restricted to it (Zoo, GeoServer WPS, COWS, 52North, ...)**

Birdhouse - Ecosystem

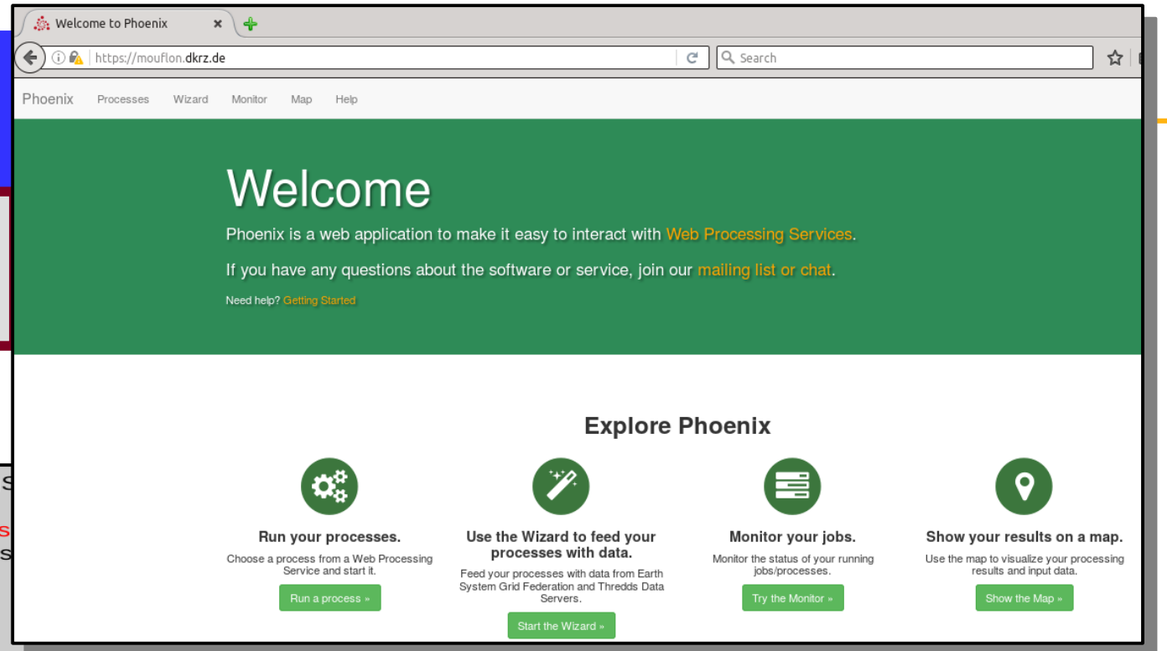
https://github.com/bird-house/



Client Side

Web Browser GUI

Authentication with OAuth or OpenID



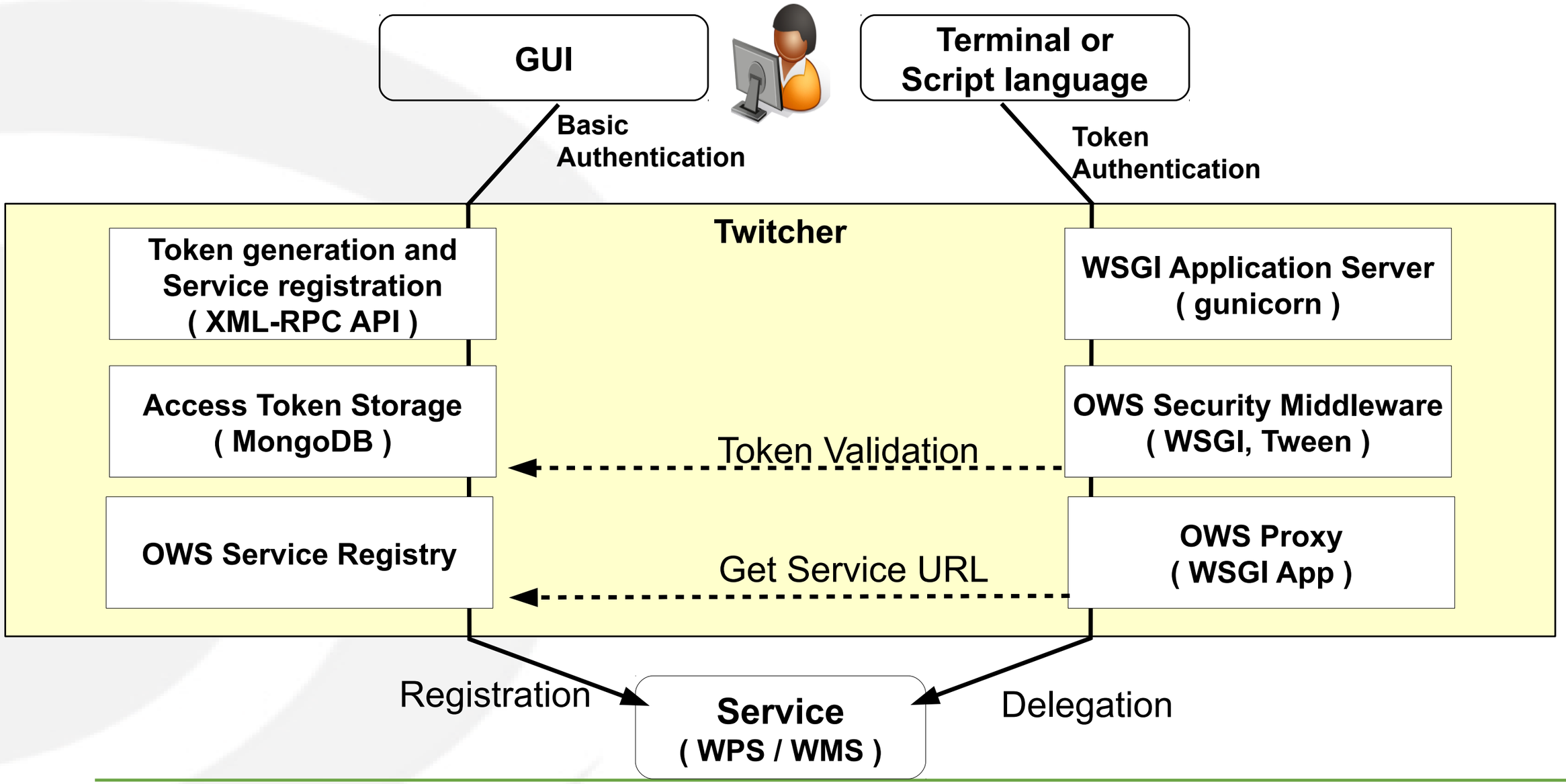
Script language Terminal Call

Token authentication

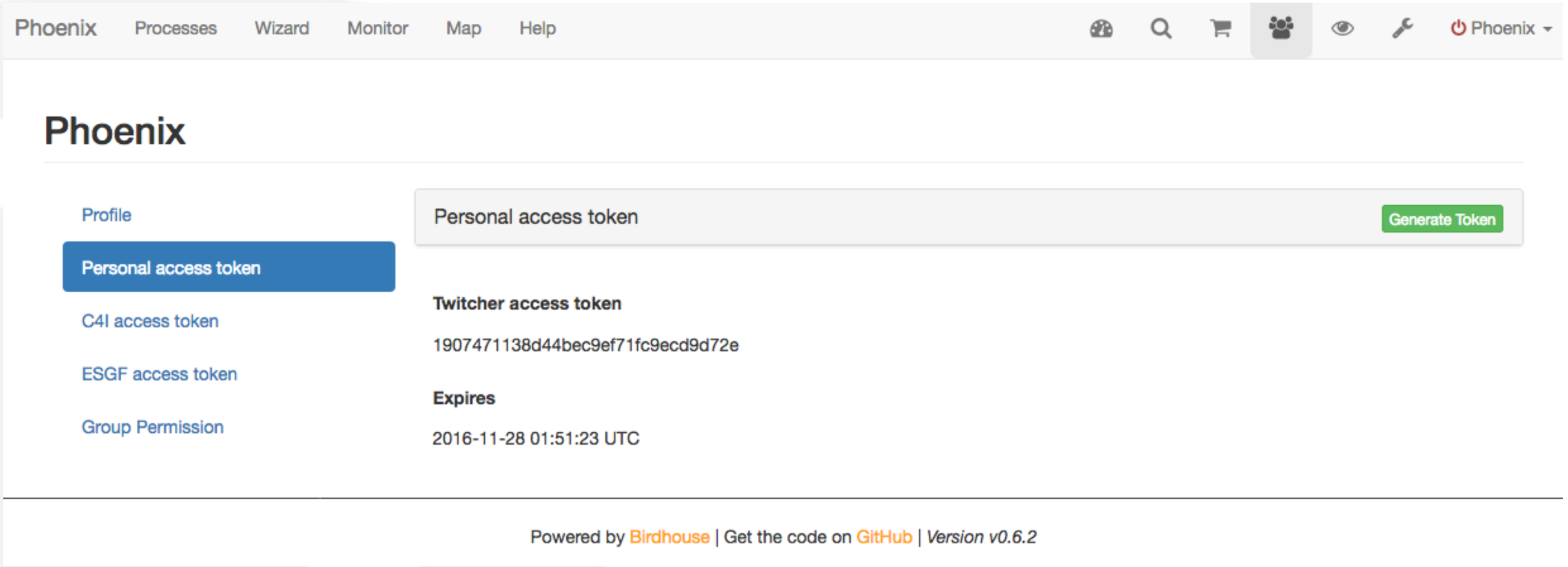
```
[nhempel@lsce3199 ~]$ export WPS_SERVICE=htt

[nhempel@lsce3199 ~]$ birdy -h

usage: birdy [<options>] <command> [<args>]

Flyingpigeon: Processes for climate data, indices and extrem events

optional arguments:
  -h, --help            show this help message and exit
  --debug               enable debug mode

command:
  List of available commands (wps processes)

  {visualisation,sdm,segetalflora,indices_single,subset_countries,eobs_t
                 Run "birdy <command> -h" to get additional help.

    visualisation       Visualisation of netcdf files:                  Just testing a nice script to visualise some variables
    sdm                 Species distribution model:                     Species distribution model
    segetalflora        Segetal Flora:                                  Species biodiversity of segetal flora.  Imput files: variable:tas , domain: EUR-
                                                                        11 or EUR-44
    indices_single      Calculation of climate indice (single variable):  This  process calculates climate indices based on one single variable.
    subset_countries    Subset netCDF files:                            This process returns only the given polygon from input netCDF files.
    eobs_to_cordex      EOBS to CORDEX:                                 downloads EOBS data in adaped CORDE format
    ensembleRobustness  Calculation of the robustness of an ensemle:    Calculates the robustness as the ratio of noise to
                                                                         signal in an ensemle of timeseries
    analogs             Days with analog pressure pattern:              Search for day with analog pressure pattern
    fetch               Download Resources:                             This process downloads resources (limited to 50GB) to the local file system
                                                                        and returns a textfile with appropriate pathe
```

```
from owslib.wps import WebProcessingS

wps = WebProcessingService(url="https
                           verbose=False, s

execute = wps.execute(
    identifier="niceprocess",
    inputs=[
    ("parameter_1", "argument"),
    ("parameter_2", "42"),
#     ("parameter_3", "0.987"), # use the default value
    ("file_identifier", "https://thredds/fileServer1/test/file1.nc"),
    ("file_identifier", "https://thredds/fileServer1/test/file2.nc"),
    ("file_identifier", "https://thredds/fileServer2/test/file3.nc")],
    output=[("output", True)])

# time for a coffee

for o in execute.processOutputs:
    print o.reference

https://mouflon.dkrz.de:8090/wpsoutputs/flyingpigeon/output_graphic-697dee76-d722-93ae-9789bf75cf44.png
https://mouflon.dkrz.de:8090/wpsoutputs/flyingpigeon/output_netCDF-697dee76-d722-93ae-9789bf75cf44.nc
https://mouflon.dkrz.de:8090/wpsoutputs/flyingpigeon/output_text-697dee76-d722-93ae-9789bf75cf44.txt
```

Security


GUI
Terminal or
Script language
Basic
Authentication
Token
Authentication
Twitcher
Token generation and
Service registration
(XML-RPC API)
Access Token Storage
(MongoDB)
OWS Service Registry
WSGI Application Server
(gunicorn)
OWS Security Middleware
(WSGI, Tween)
OWS Proxy
(WSGI App)
Token Validation
Get Service URL
Registration
Service
(WPS / WMS)
Delegation

Security Token



Script language

```
from owslib.wps import WebProcessingService, monitorExecution

#  using wps url with access token db6c...
wps = WebProcessingService(
                        url="https://mouflon.dkrz.de/ows/proxy/flyingpigeon/db6c1293d0444d919dcc3ce48fa610f7 ", \
                        verify=False,
                        verbose=False, skip_caps=False,
                        )

execute = wps.execute(
    identifier="niceprocess",
    inputs=[
    ("parameter_1", "argument"),
    ("parameter_2", "42"),
#    ("parameter_3", "0.987"), # use the default value
    ("file_identifier", "https://thredds/fileServer1/test/file1.nc"),
    ("file_identifier", "https://thredds/fileServer1/test/file2.nc"),
    ("file_identifier", "https://thredds/fileServer2/test/file3.nc")],
    output=[("output", True)])

for o in execute.processOutputs:
    print o.reference

https://mouflon.dkrz.de:8090/wpsoutputs/flyingpigeon/output_graphic-697dee76-d722-93ae-9789bf75cf44.png
https://mouflon.dkrz.de:8090/wpsoutputs/flyingpigeon/output_netCDF-697dee76-d722-93ae-9789bf75cf44.nc
```

Terminal Call

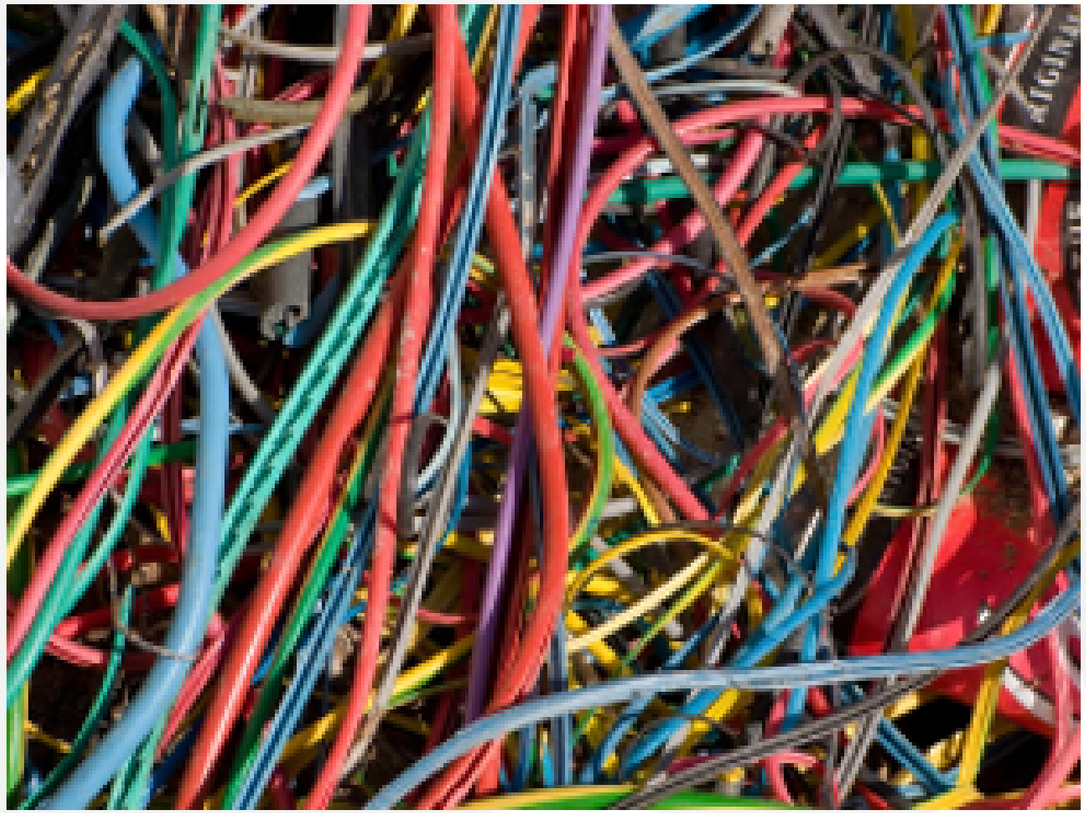
```
[nhempel@lsce3199 ~]$ conda install -c birdhouse birdhouse-birdy

[nhempel@lsce3199 ~]$ birdy -h
usage: birdy [<options>] <command> [<args>]

Flyingpigeon: Processes for climate data, indices and extreme events

optional arguments:
  -h, --help            show this help message and exit
  --debug               enable debug mode
  --token TOKEN, -t TOKEN
                        Token to access the WPS service.

command:
  List of available commands (wps processes)
```

Terminal Call

```
[nhempel@lsce3199 ~]$ export WPS_SERVICE=https://mouflon.dkrz.de/ows/proxy/flyingpigeon

[nhempel@lsce3199 ~]$ birdy –token 0c6d305b0f42452cbdcf31c7ac74f1e1 \
                                  analogs_detection --experiment 'NCEP_slp'

INFO:Execution status: ProcessAccepted
INFO:Execution status: ProcessStarted
INFO:Execution status: ProcessSucceeded

INFO:Output:

INFO:analogs=http://mouflon.dkrz.de/wpsoutputs/flyingpigeon/analogs-08bce60c-6a41-11e6-be7a-8fdf4b12fcf5.txt (text/plain)
INFO:config=http://mouflon.dkrz.de/wpsoutputs/flyingpigeon/config-08bce60c-6a41-11e6-be7a-8fdf4b12fcf5.txt (text/plain)

[nhempel@lsce3199 ~]$
```

http://twitcher.readthedocs.io/en/latest/tutorial.html

Deployment with conda and buildout



Using conda package manager to setup an environment with all used software components (python, R, matplotlib, PyWPS, ...)

Using buildout to setup PyWPS with all services (supervisor, gunicorn, nginx) and configuration files.

To install a *Bird* just run :

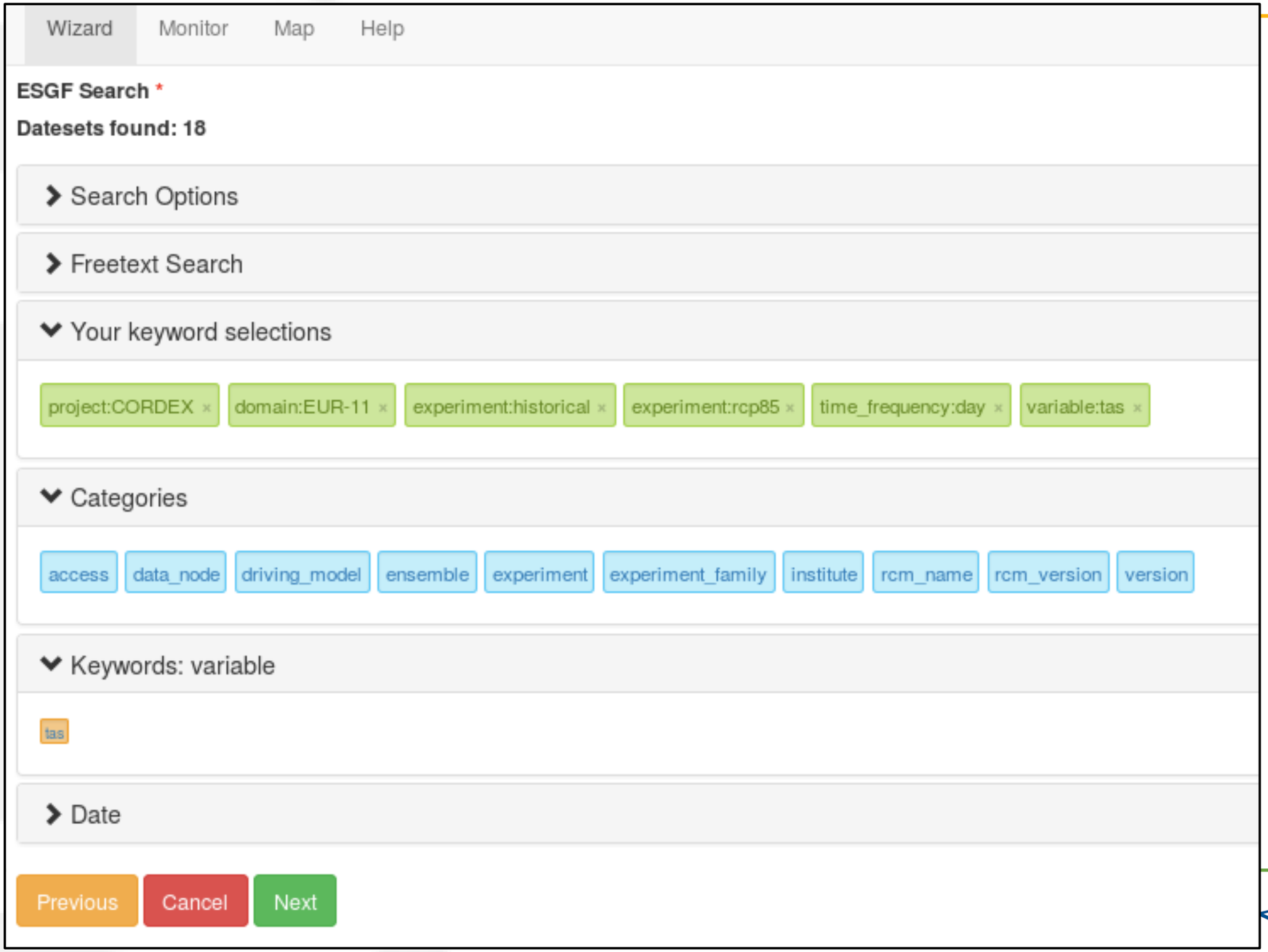
```
$ git clone https://github.com/bird-house/emu.git
$ cd emu
$ make clean install
$ make start
```

http://conda.pydata.org/docs/

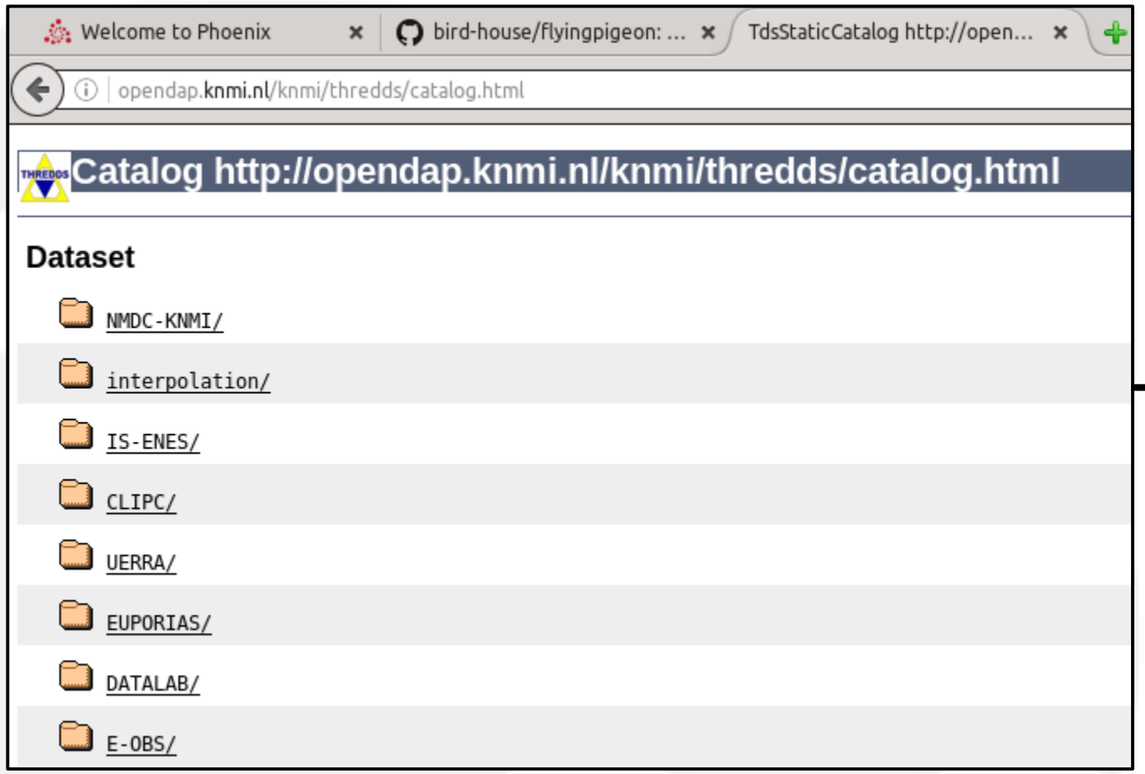
http://www.buildout.org/en/latest/

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

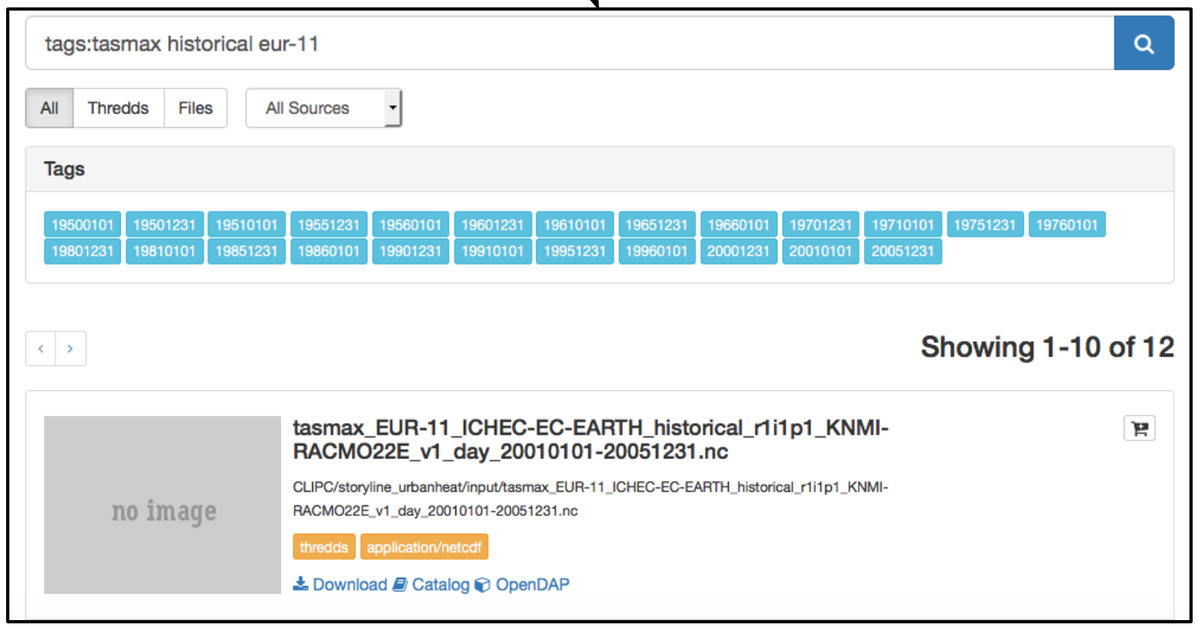
ESGF – search


Wizard
Monitor
Map
Help
ESGF Search *
Datesets found: 18
Search Options
Freetext Search
Your keyword selections
project:CORDEX ×
domain:EUR-11 ×
experiment:historical ×
experiment:rcp85 ×
time_frequency:day ×
variable:tas ×
Categories
access
data_node
driving_model
ensemble
experiment
experiment_family
institute
rcm_name
rcm_version
version
Keywords: variable
tas
Date
Previous
Cancel
Next

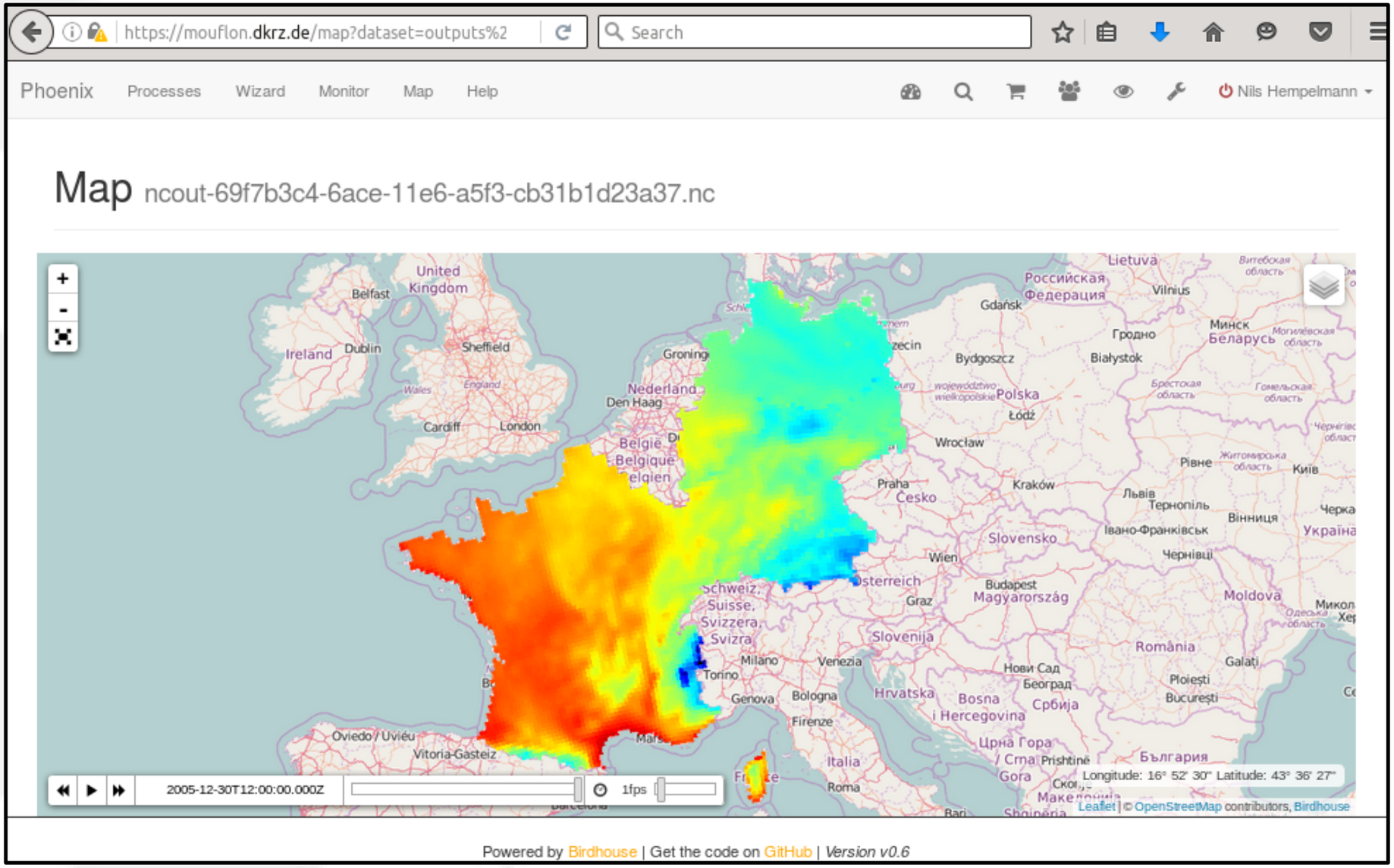
Solr Index for Data (**bird-feeder**)



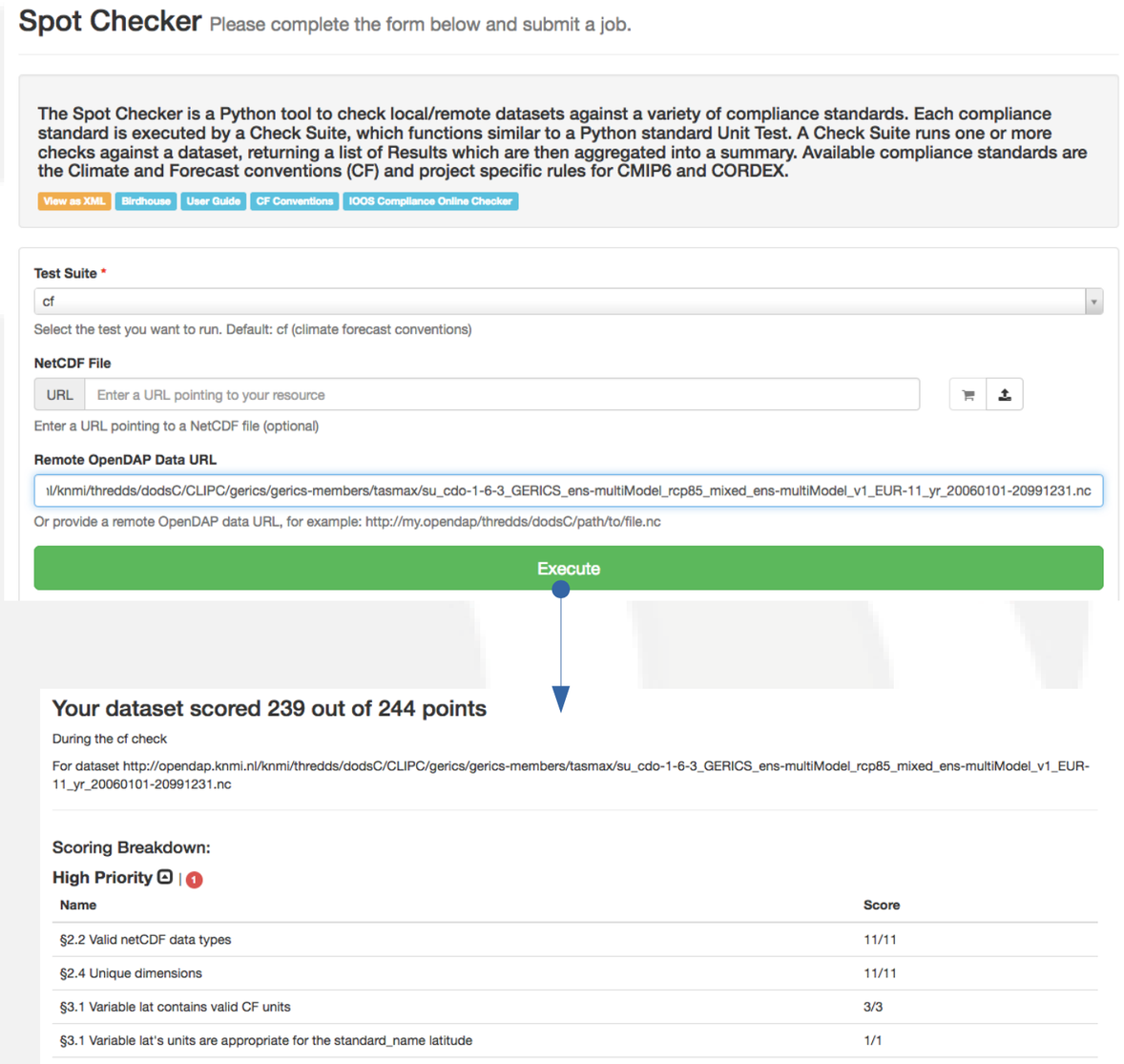
Run bird-feeder to create Solr search
Index for Thredds Data Catalogs and local data



Web Mapping Server



SpotChecker : Metadata Compliance Checks



Run SpotChecker on NetCDF file :
File URL, OpenDAP URL or uploaded File.

Performs compliance checks :
CF conventions, CORDEX, CMIP5, ...

Using IOOS Compliance-Checker and
DKRZ Quality Assurance Checker.

Check reports in HTML and YAML format.

Subsetting : Region USA

Resource *

Resource

URL l-LR/historical/mon/atmos/Amon/r1i1p1/v20120315/tas/tas_Amon_MPI-ESM-LR_historical_r1i1p1_185001-200512.nc

Add Resource

NetCDF File

Region *

Region

USA

Add Region

Running 1 Finished 7 Matching 8 Sort

Status	User	Process	Service	Caption	Finished	Duration	Labels	
	Phoenix	subset_countries	flyingpigeon	???	???	0:00:31	dev single async edit labels	Details Restart

Log Inputs Outputs View as XML

no image

Subsets Parameter `output`, a WPS ComplexType

Tar archive containing the netCDF files

output-2045706c-b4d5-11e6-9959-868dacf6ed58.tar

application/x-tar

Download Share

no image

Subsets for one dataset Parameter `ncout`, a WPS ComplexType

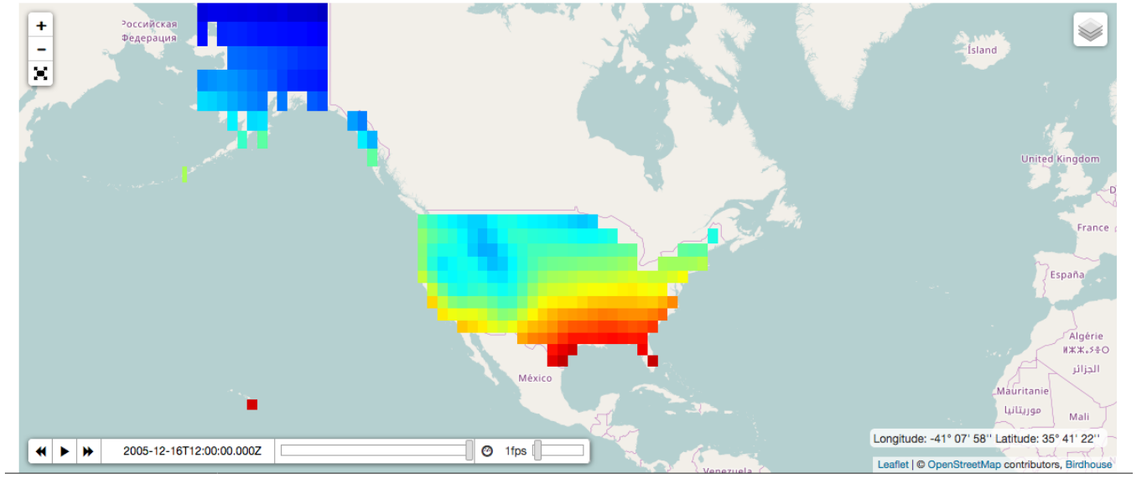
NetCDF file with subsets of one dataset.

ncout-2045706c-b4d5-11e6-9959-868dacf6ed58.nc

application/x-netcdf

Download Share Show on Map

Map ncout-2045706c-b4d5-11e6-9959-868dacf6ed58.nc



Powered by Birdhouse | Get the code on GitHub | *Version v0.6.2*

wps@dkrz.de

DKRZ cnrs

ESMValTool Diagnostics as Web Processing Service

Processes / esmvaltool

Description

WPS processes for ESMValTool.

XML Provider: EsmValTool

Processes

ESMValTool: surface contour plot for precipitation 1.0
Tutorial contour plot used in the doc/overview.pdf.

ESMValTool: tutorial diagnostic. 1.0
Tutorial diagnostic used in the doc/toy-diagnostic-tutorial.pdf.

Description

Tutorial contour plot used in the doc/overview.pdf.

XML Birdhouse ESMValTool

Run async *

Check this to run process async.

Model *
MPI-ESM-LR
Choose a model like MPI-ESM-LR.

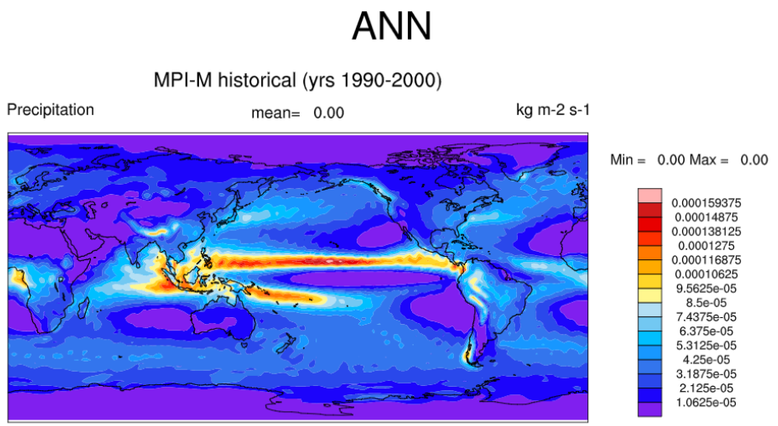
Experiment *
historical
Choose an experiment like historical.

Ensemble *
r1i1p1
Choose an ensemble like r1i1p1.

Start year *
1990
Start year of model data.

End year *
2000
End year of model data.

Execute



Log Inputs Outputs

no image

Output plot Parameter `output`, a WPS ComplexType
Generated output plot of ESMValTool processing.
surfconplot_simple_pr_T2Ms_ANNjcrpMJ.pdf
application/pdf
Download Share

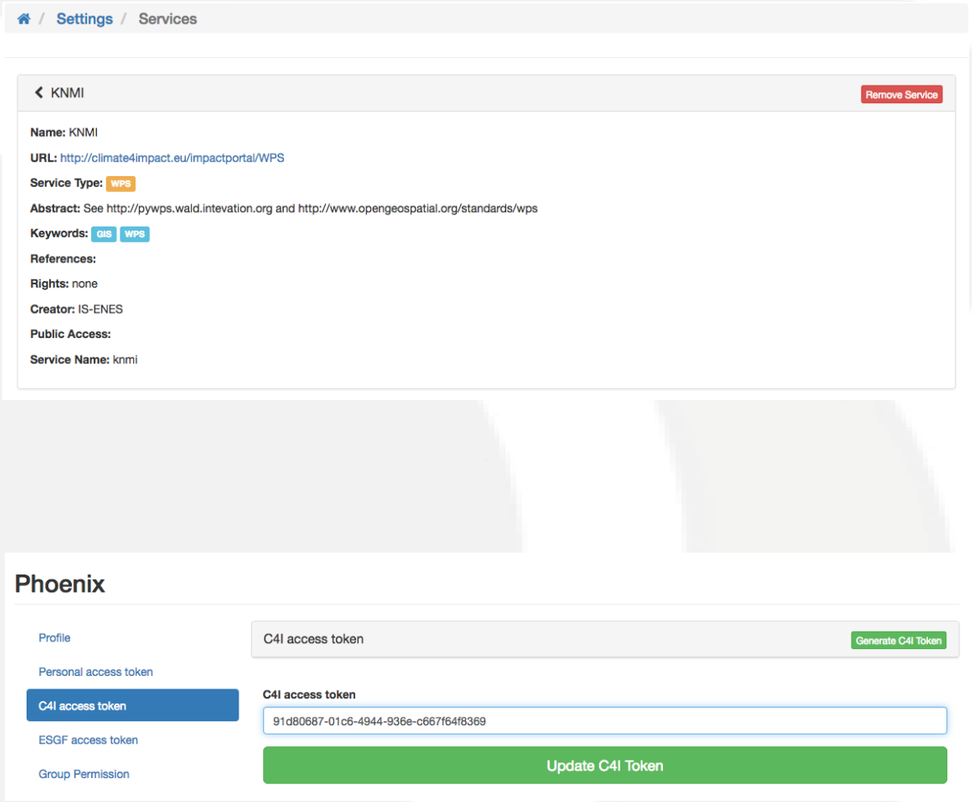
no image

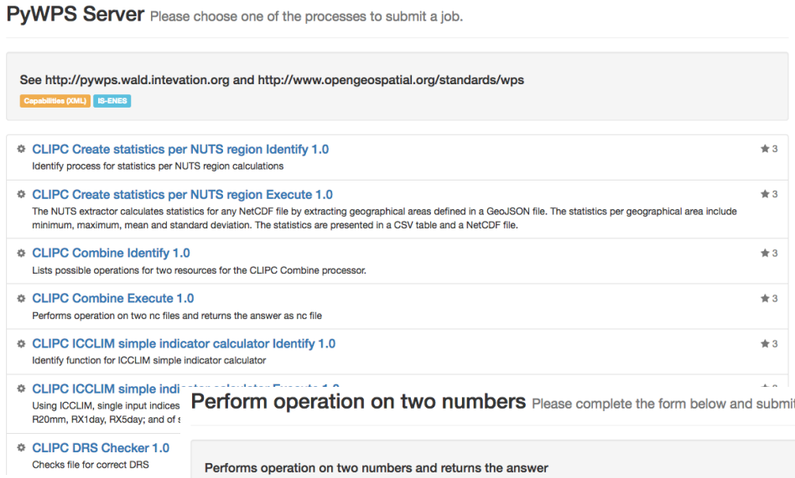
namelist Parameter `namelist`, a WPS ComplexType
ESMValTool namelist used for processing.
namelistvGe9v4.xml
text/plain
Download Share

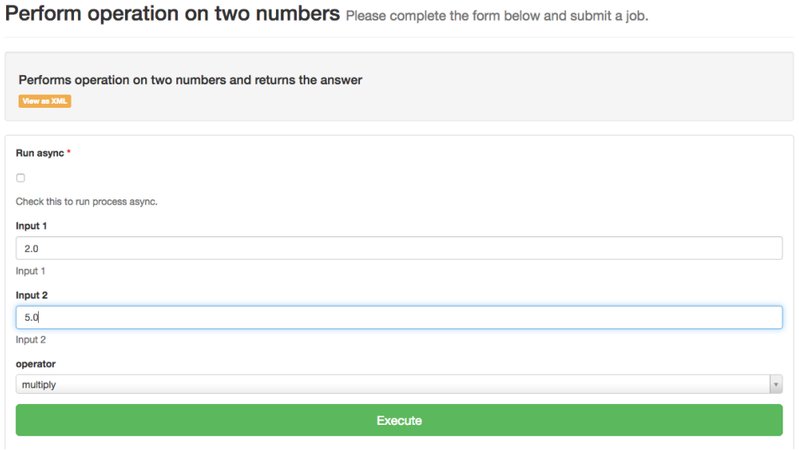
no image

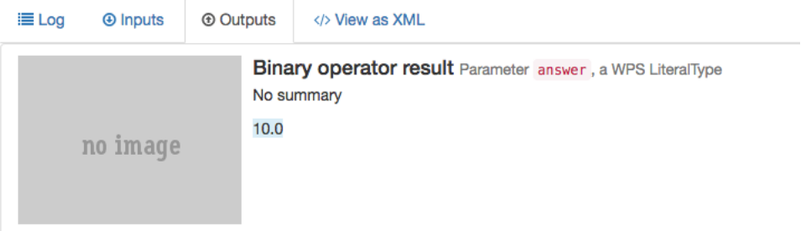
Log File Parameter `log`, a WPS ComplexType
Log File of ESMValTool processing.
loge1GhQG.txt
text/plain
Download Share

Accessing Remote WPS : Climate4Impact WPS at KNMI









Copernicus Extentions for WPS

- **Using PyWPS-4 : ready for WPS 2.0 (pause, resume, delete)**
- **Attach batch processing with SLURM etc ...**
- **Optionally run processes in Docker Container.**
- **Delegation to SLURM and Docker is handled internally of PyWPS (new feature for PyWPS-4).**
- **Process code and Process definition is not changed when run as batch process or in a docker container.**

- https://github.com/bird-house
- http://birdhouse.readthedocs.org/en/latest/

- https://gitter.im/bird-house/birdhouse
- https://lists.dkrz.de/mailman/listinfo/wps
- https://lists.dkrz.de/mailman/listinfo/wps-dev

- DEMO GUI: https://mouflon.dkrz.de

Contact :

wps@dkrz.de

Thanks to :

Carmen Alvarez-Castro, Katharina Berger, Patrick Brockmann, Carsten Ehbrecht, Wolfgang Falk, Nils Hempelmann, Heinz-Dieter Hollweg, Jörg Hoffmann, Nikolay Kadygrov, Stephan Kindermann, Florian Klemme, Nikolay Koldunov, Ben Koziol, Cathy Nangini, Sabine Radanovics, Seckmag, Robert Vautard, Pascal Yiou , …. , et. al.

Analogues of atmospheric Circulation