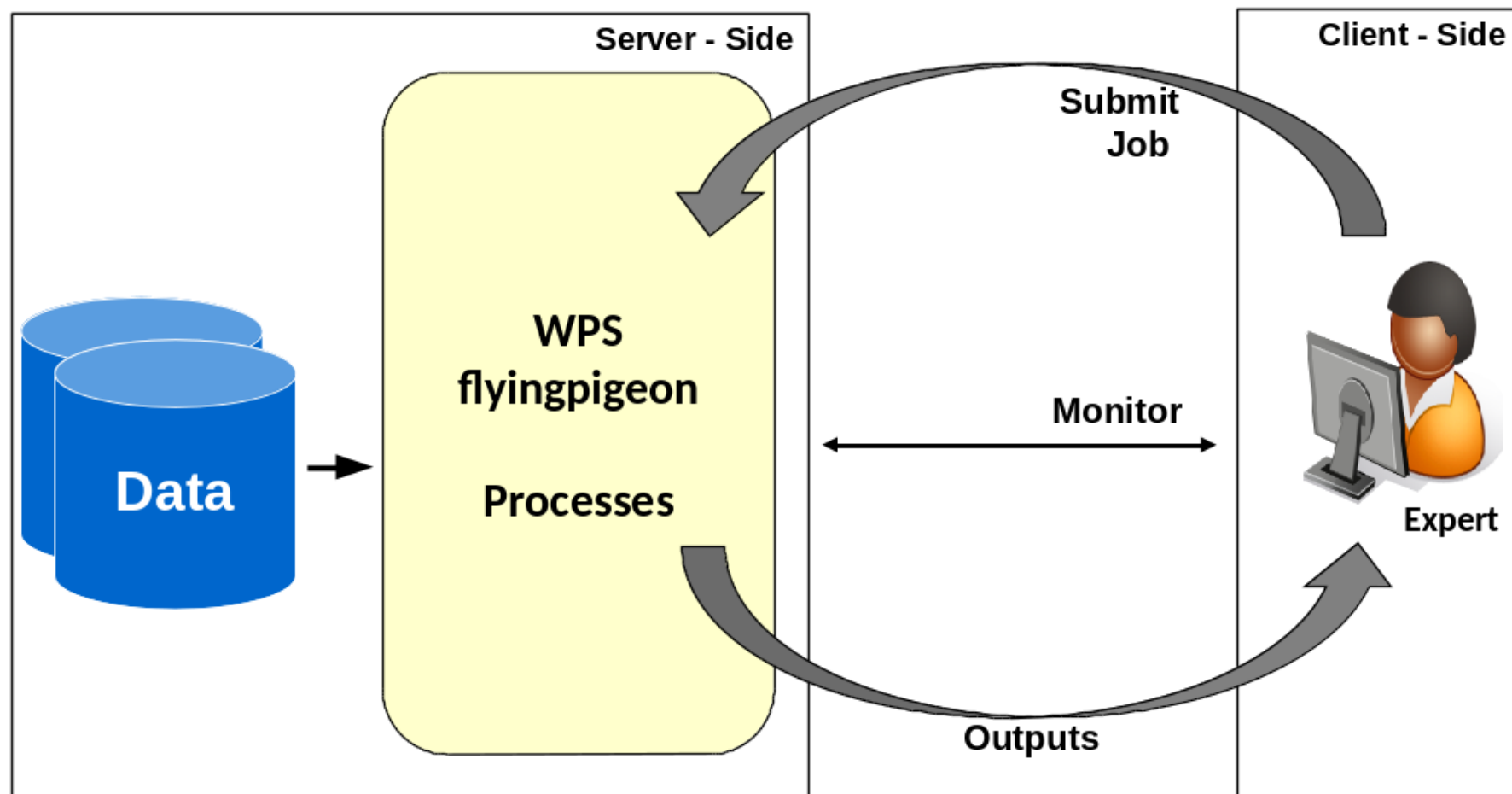




# **Birdhouse: Supporting Web Processing Services for Climate Data**



# Server-Client Side

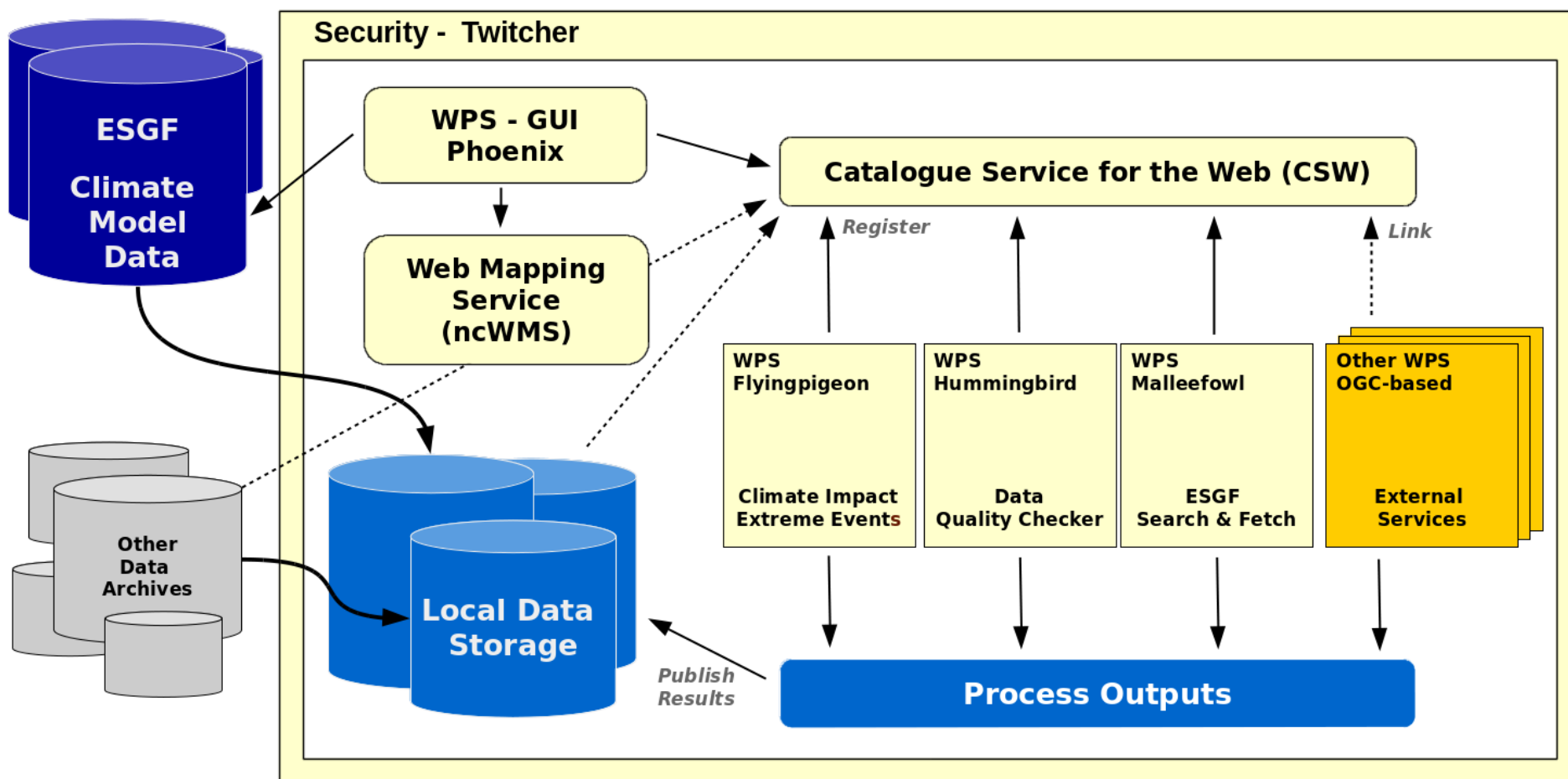


# What does Birdhouse provide?

- Customizable installation of Web Processing Services using conda, buildout and ansible.
- Provides WPS as Docker Container.
- Web-based and Terminal WPS clients.
- Security Proxy Twitcher for OGC/WPS services.
- Data Access: ESGF, Thredds, OpenStack, ...
- WPS for compliance checks and climate impact.
- Supports PyWPS 3. and 4.x ... but not restricted to it (others: Zoo, GeoServer, COWS, 52North).
- Supports to setup a WPS for your own use cases.

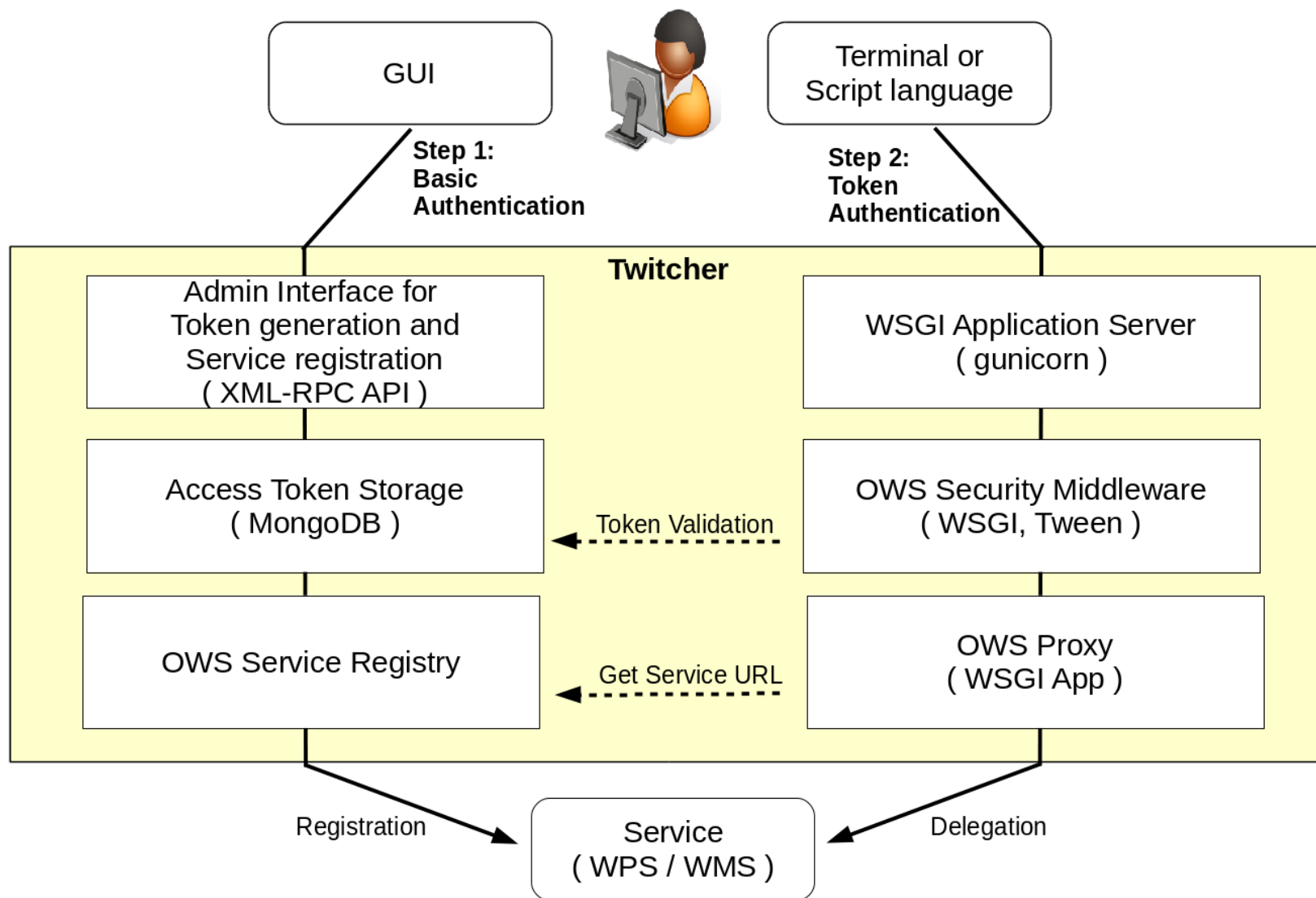
# Birdhouse - Ecosystem

<http://bird-house.github.io/>



# Client Side

# Security – Twitcher Security Proxy



# Security Token

Phoenix Processes Wizard Monitor Map Help

## Phoenix

Profile

Personal access token

C4I access token

ESGF access token

Group Permission

Personal access token

Generate Token

**Twitcher access token**

1907471138d44bec9ef71fc9ecd9d72e

**Expires**

2016-11-28 01:51:23 UTC

Powered by [Birdhouse](#) | Get the code on [GitHub](#) | Version v0.6.2

The token-based Security Proxy is similar to the Climate4Impact approach:  
<https://dev.knmi.nl/projects/impactportal/wiki/API>

# 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,
)

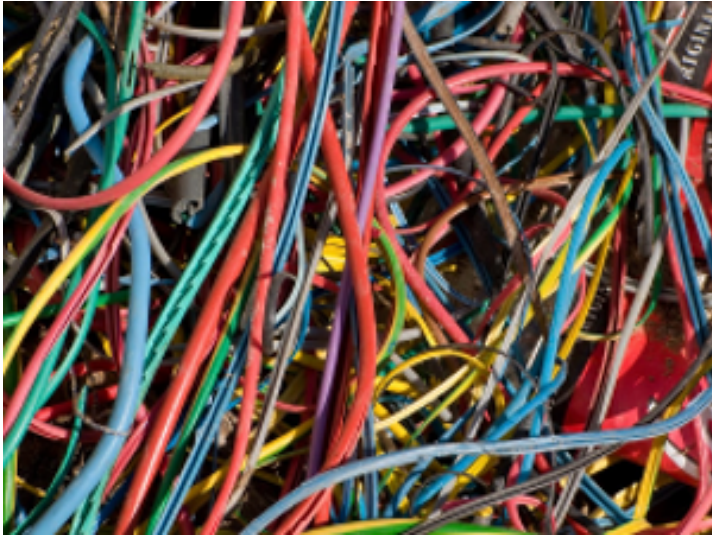
execute = wps.execute(
    identifier="niceprocess",
    inputs=[
        ("parameter_1", "argument"),
        ("parameter_2", "42"),
        # ("parameter_3", "0.987"), # use the default value
        ("file_identifier", "https://thredds/fileServer1/test/file1.nc"),
        ("file_identifier", "https://thredds/fileServer1/test/file2.nc"),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



# Deployment with conda and buildout



*“Manage the Chaos”*

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
$ cd emu
$ make clean install
$ make start
$ http://localhost:8094/wps
```

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

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

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

# Wizard: Using ESGF-Search

## 1. Use the Wizard to select Process and enter Parameters

Wizard / Literal Inputs

Quality Assurance Checker by DKRZ

The Quality Assurance checker QA-DKRZ checks conformance of meta-data of climate simulations given in NetCDF format with conventions and rules of climate model projects. At present, checking of CF Conventions, CMIP5, and CORDEX is supported. Development and maintenance for the QA checker is done by the German Climate Computing Centre (DKRZ). If you have suggestions for improvement then please contact Heinz-Dieter Hollweg at DKRZ (hollweg@dkrz.de).

[XML](#) [Birdhouse](#) [User Guide](#) [CF Conventions](#) [QA Checker Documentation](#) [Conda Package](#) [GitHub](#)

Literal inputs of Quality Assurance Checker by DKRZ

Project \*

CORDEX

Climate model data project to be checked: CORDEX or CMIP5

[Previous](#) [Cancel](#) [Next](#)

## 2. Choose ESGF as Input Source for NetCDF Files

Wizard / Choose Data Source

Choose Data Source for NetCDF File

Source \*

☒ Earth System Grid (ESGF)

☐ Birdhouse Solr Search

☐ Thredds Catalog Service

[Previous](#) [Cancel](#) [Next](#)

## 3. Select Datasets with ESGF Search Widget

Wizard / ESGF Search

ESGF Search \*

Datasets found: 7

Search Options

Freetext Search

Your keyword selections

variable:tas \* experiment:historical \* project:CORDEX \* time\_frequency:mon \* domain:EUR-44 \*

Categories

access data\_node driving\_model ensemble experiment\_family geo institute rcm\_name rcm\_version version

Keywords: institute

CLMcom DMI HMS KNMI MPI-CSC

Date

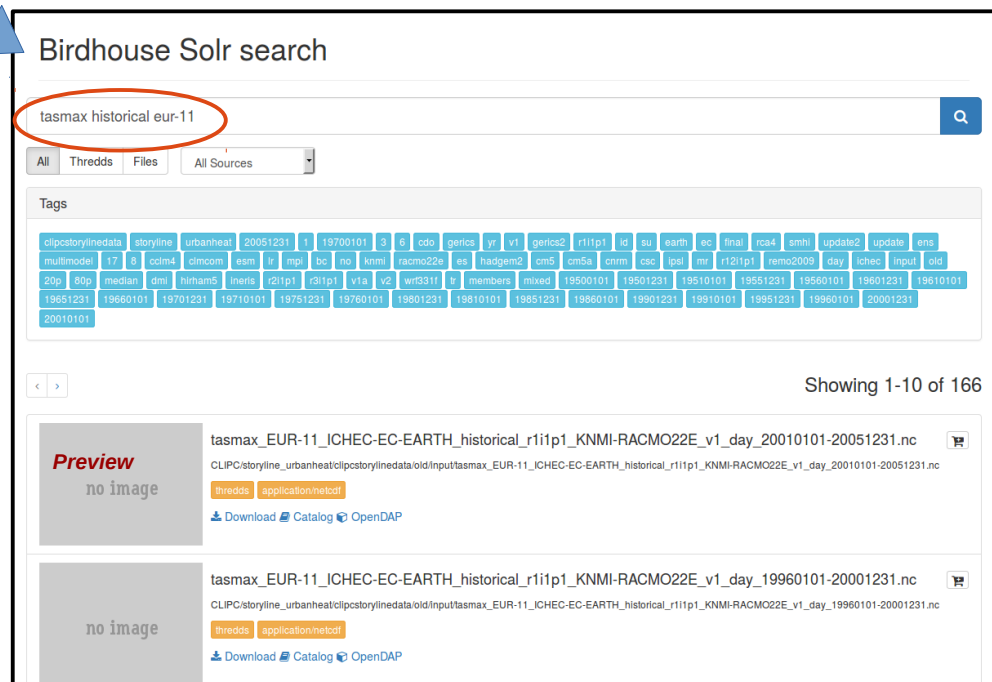
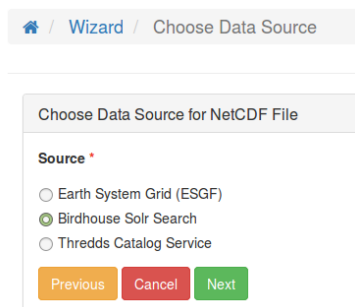
[Previous](#) [Cancel](#) [Next](#)

# Solr Index for Thredds Catalogs (bird-feeder)

Run bird-feeder to create Solr Search Index for Thredds Data Catalogs and local filesystems.



Wizard:  
Choose Solr Search  
of Thredds Catalog  
as Input Source



# Web Map Service: Show NetCDF Files on Map

## Browse Catalog

**Preview**  
no image

mslp.1979.nc

Datasets/ncp.reanalysis2/surface/mslp.1979.nc

[thredds](#)
[application/netcdf](#)

[Download](#)
[Catalog](#)
[OpenDAP](#)
[Show on Map](#)

## Process Inputs

[Log](#)
[Inputs](#)
[Outputs](#)
[View as XML](#)

no image

Resource Parameter `resource`, a WPS ComplexType

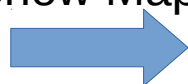
NetCDF File

tas\_Amon\_MPI-ESM-LR\_historical\_r11ip1\_185001-200512.nc

[application/x-netcdf](#)

[Download](#)
[Share](#)
[Show on Map](#)

Show Map



## Process Outputs

[Log](#)
[Inputs](#)
[Outputs](#)
[View as XML](#)

no image

Subsets Parameter `output`, a WPS ComplexType

Tar archive containing the netCDF files

output-2045706c-b4d5-11e6-9959-868dac6ed58.tar

[application/x-tar](#)

[Download](#)
[Share](#)

no image

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

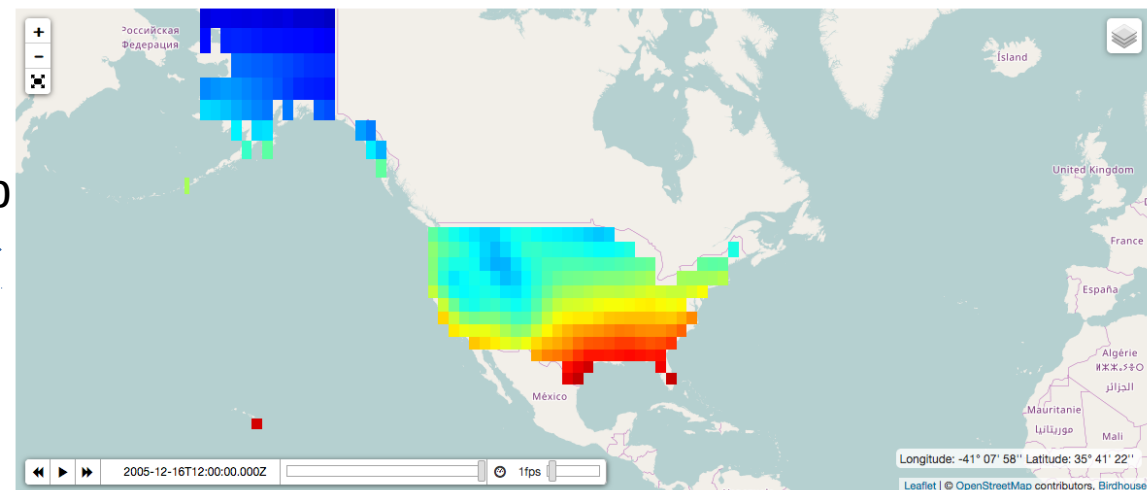
NetCDF file with subsets of one dataset.

ncout-2045706c-b4d5-11e6-9959-868dac6ed58.nc

[application/x-netcdf](#)

[Download](#)
[Share](#)
[Show on Map](#)

Map ncout-2045706c-b4d5-11e6-9959-868dac6ed58.nc



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

Possible Web Map Services:  
ncWMS, Adaguc (KNMI), sci-wms (planned)

# SpotChecker: Metadata Compliance Checks

**Spot Checker** Please complete the form below and submit a job.

The Spot Checker is a Python tool to check local/remote datasets against a variety of compliance standards. Each compliance standard is executed by a Check Suite, which functions similar to a Python standard Unit Test. A Check Suite runs one or more checks against a dataset, returning a list of Results which are then aggregated into a summary. Available compliance standards are the Climate and Forecast conventions (CF) and project specific rules for CMIP6 and CORDEX.

[View as XML](#) [Birdhouse](#) [User Guide](#) [CF Conventions](#) [IOOS Compliance Online Checker](#)

**Test Suite \***

cf

Select the test you want to run. Default: cf (climate forecast conventions)

**NetCDF File**

URL

Enter a URL pointing to a NetCDF file (optional)

**Remote OpenDAP Data URL**

Or provide a remote OpenDAP data URL, for example: <http://my.opendap/thredds/dodsC/path/to/file.nc>

**Execute**



Run SpotChecker on NetCDF File:  
File URL, OpenDAP URL or upload File

Perform Compliance Checks:  
CF Conventions, CORDEX, CMIP5, ...


Using IOOS Compliance-Checker and  
DKRZ Quality Assurance Checker.

**Your dataset scored 239 out of 244 points**

During the cf check

For dataset [http://opendap.knmi.nl/knmi/thredds/dodsC/CLIPC/gerics/gerics-members/tasmax/su\\_cdo-1-6-3\\_GERICS\\_ens-multiModel\\_rcp85\\_mixed\\_ens-multiModel\\_v1\\_EUR-11\\_yr\\_20060101-20991231.nc](http://opendap.knmi.nl/knmi/thredds/dodsC/CLIPC/gerics/gerics-members/tasmax/su_cdo-1-6-3_GERICS_ens-multiModel_rcp85_mixed_ens-multiModel_v1_EUR-11_yr_20060101-20991231.nc)

**Scoring Breakdown:**

**High Priority** 

Name	Score
\$2.2 Valid netCDF data types	11/11
\$2.4 Unique dimensions	11/11
\$3.1 Variable lat contains valid CF units	3/3
\$3.1 Variable lat's units are appropriate for the standard_name latitude	1/1

Check Reports in HTML and YAML  
Format.

# Subsetting: Region USA

**Resource**

Resource

URL

Add Resource

**NetCDF File**

**Region**

Region

USA

Add Region

Execute Job

Running 1 Finished 7 Matching 8 Sort

Status	User	Process	Service	Caption	Finished	Duration	Labels	
<input type="checkbox"/>	Phoenix	subset_countries	flyingpigeon	???	???	0:00:31	dev single async edit labels	Details Restart

Details

Log Inputs Outputs View as XML

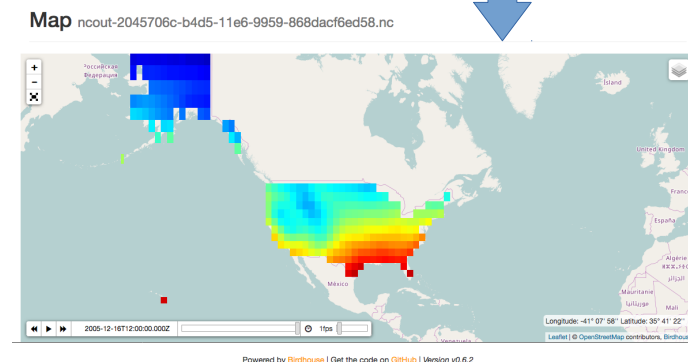
**Preview**

no image

**Subsets** Parameter `output`, a WPS ComplexType  
Tar archive containing the netCDF files  
`output-2045706c-b4d5-11e6-9959-868dac6ed58.tar`  
application/x-tar  
Download Share

**Subsets for one dataset** Parameter `ncout`, a WPS ComplexType  
NetCDF file with subsets of one dataset.  
`ncout-2045706c-b4d5-11e6-9959-868dac6ed58.nc`  
application/x-netcdf  
Download Share Show on Map

Show Map



# ESMValTool Diagnostics as Web Processing Service

## 1. Choose ESMVal Process

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.

## 2. Enter Input Parameters

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

## 3. Outputs: plot, namelist, log

Log Inputs Outputs

no image

Output plot Parameter output, a WPS ComplexType  
Generated output plot of ESMValTool processing.  
surfconplot\_simple\_pr\_T2Ms\_ANNprpMJ.pdf  
application/pdf  
Download Share

no image

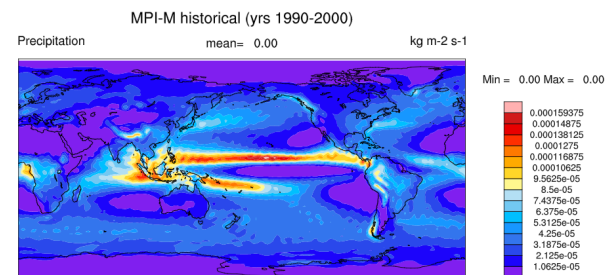
namelist Parameter namelist, a WPS ComplexType  
ESMValTool namelist used for processing.  
namelistvGe9v4.xml  
text/plain  
Download Share

no image

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

Show Plot

ANN



ESMVal namelist is generated by the WPS process (using the input parameters).  
Data is retrieved by the ESMVal ESGF coupling module.

# Accessing Remote WPS: Climate4Impact WPS at KNMI

## Register C4I WPS URL

🏠 / Settings / Services

< KNMI

Name: KNMI

URL: <http://climate4impact.eu/impactportal/WPS>

Service Type: **WPS**

Abstract: See <http://pywps.wald.intevation.org> and <http://www.opengeospatial.org/standards/wps>

Keywords: **GIS** **WPS**

References:

Rights: none

Creator: IS-ENES

Public Access:

Service Name: knmi

## Run a Process: operation on two numbers

**PyWPS Server** Please choose one of the processes to submit a job.

See <http://pywps.wald.intevation.org> and <http://www.opengeospatial.org/standards/wps>

Capabilities (XML) **IS-ENES**

- ⚙️ **CLIPC Create statistics per NUTS region Identify 1.0** ★ 3  
Identify process for statistics per NUTS region calculations
- ⚙️ **CLIPC Create statistics per NUTS region Execute 1.0** ★ 3  
The NUTS extractor calculates statistics for any NetCDF file by extracting geographical areas defined in a GeoJSON file. The statistics per geographical area include minimum, maximum, mean and standard deviation. The statistics are presented in a CSV table and a NetCDF file.
- ⚙️ **CLIPC Combine Identify 1.0** ★ 3  
Lists possible operations for two resources for the CLIPC Combine processor.
- ⚙️ **CLIPC Combine Execute 1.0**  
Performs operation on two no files and returns the answer
- ⚙️ **CLIPC ICCLIM simple indicator calculator I**  
Identify function for ICCLIM simple indicator calculator
- ⚙️ **CLIPC ICCLIM simple indicator calculator I**  
Using ICCLIM, single input indices of temperature TG, TX, R20mm, RX1day, RX5day; and of snowfall: SD, SD1, SDI
- ⚙️ **CLIPC DRS Checker 1.0**  
Checks file for correct DRS

**Perform operation on two numbers** Please complete the form below and submit a job.

Performs operation on two numbers and returns the answer

[View as XML](#)

Run async \*

☐

Check this to run process async.

Input 1

2.0

Input 1

Input 2

5.0

Input 2

operator

multiply

Execute

## Update C4I Access Token

Phoenix

Profile

Personal access token

**C4I access token**

ESGF access token

Group Permission

C4I access token

91d80687-01c6-4944-936e-c667f64f8369

Update C4I Token

Log Inputs Outputs [View as XML](#)

no image

**Binary operator result** Parameter **answer**, a WPS LiteralType  
No summary  
10.0



# Accessing Remote WPS: COWS WPS at CEDA

## Register COWS WPS URL

## Run a Process: Double It

[Home](#) / [Settings](#) / [Services](#)

**CEDA WPS Server** Please choose one of the processes to submit a job.

◀ COWS

**Name:** COWS

**URL:** <http://wps-web1.ceda.ac.uk/wps>

**Service Type:** WPS

**Abstract:**

**Keywords:** WPS CEDA COWS Web Processing Server

**References:**

**Rights:**

**Creator:** Centre for Environmental Data Archival (CEDA)

**Public Access:**

**Service Name:** cows


	Capabilities (XML)	Centre for Environmental Data Archival (CEDA)
<b>Double It None</b> ★ 3	The "DoubleIt" process is used to demonstrate how the WPS and the WPS User Interface work. The process accepts an integer or floating point number and returns some XML containing the input number double.	
<b>CDO Calculate Single File Statistics With Argument None</b> ★ 3	Calls the Climate Data Operators (CDO) tool with the single file path and argument provided and uses the chosen operator to calculate statistics written to a NetCDF file. This process wraps the following CDO operators: ['runmin', 'runmax', 'runsum', 'runmean', 'runavg', 'runvar', 'runstd', 'ydrunmin', 'ydrunmax', 'ydrunsum', 'ydrunmean', 'ydrunavg', 'ydrunvar', 'ydrunstd']	
<b>CDO Calculate Single File Climate Indices None</b> ★ 3	Calls the Climate Data Operators (CDO) tool with the single file path provided and uses the chosen operator to calculate climate indices written to a NetCDF file. This process wraps the following CDO operators: ['eca_cdd', 'eca_cfd', 'eca_cwd', 'eca_id', 'eca_idf', 'eca_r10mm', 'eca_r20mm', 'eca_rr1', 'eca_sdi']	
<b>CDO Calculate Multiple File Statistics None</b> ★ 3	Calls the Climate Data Operators (CDO) tool with the file paths provided and uses the chosen operator to calculate statistics written to a NetCDF file. This process wraps the following CDO operators: ['ensmin', 'ensmax', 'enssum', 'ensmean', 'ensavg', 'ensvar', 'ensstd']	
<b>CDO Get Multiple File Info None</b> ★ 3	Calls the Climate Data Operators (CDO) tool with the file paths provided and uses the chosen operator to extract information that is written to a text file. This process wraps the following CDO operators: ['info', 'infov', 'sinfo', 'sinfov']	
<b>CDO Get Single File Info None</b> ★ 3	Calls the Climate Data Operators (CDO) tool with the single file path provided and uses the chosen operator to extract information that is written to a text file. This process wraps the following CDO operators: ['npar', 'nlevel', 'nyear', 'nmor', 'showyear', 'showmor', 'showdate', 'showtime', 'showtimestamp']	
<b>CDO Calculate Two File Climate Indices With Argum</b>	Calls the Climate Data Operators (CDO) tool with the two file path file. This process wraps the following CDO operators: ['eca_cwfi', 'eca_r99p', 'eca_r99ptot', 'eca_tg10p', 'eca_tg90p', 'eca_in10p', 'i	<div>View as XML</div> <div>None</div> <div>None</div>
<b>Simple Plot None</b>	The "SimplePlot" process demonstrates the visualisation capability user can choose a bounding box. The user can also decide on a interface can provide a quick look of the output if requested. This	<div>Run async *</div> <div>Check this to run process async.</div>


**Double It** Please complete the form below and submit a job.

The "DoubleIt" process is used to demonstrate how the WPS and the WPS User Interface work. The process accepts an integer or floating point number and returns some XML containing the input number double.

[View as XML](#) [code](#) [none](#)

---

**Run async** 

 Check this to run process async.

**Input Job Id**

The Id of a separate WPS Job used to provide input to this process.

**Number To Double**

**Execute**

# Copernicus Extensions for PyWPS

- 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 in PyWPS-4).
- WPS Process definition and code is not changed when run as batch job or in a docker container.

# Links

- <http://bird-house.github.io/>
- <http://birdhouse.readthedocs.io/en/latest/>
- <https://gitter.im/bird-house/birdhouse>
- <https://lists.dkrz.de/mailman/listinfo/wps>
- <https://lists.dkrz.de/mailman/listinfo/wps-dev>
- Demo: <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.

# Additional slides