ENES Climate Analytics Service (ECAS)

Thematic Service, EOSC-HUB

Sofiane Bendoukha[1], Tobias Weigel[1], Sandro Fiore[2], Alessandro D'Anca[2]

[1]Deutsches Klimarechenzentrum (DKRZ), Hamburg, Germany
[2]Centro Euro-Mediterraneo sui Cambiementi Climatici (CMCC), Lecce, Italy

11. April 2018

EOSC-hub is EU's Horizon 2020 project under grant agreement 777536

ENES Climate Analytics Service (ECAS)

Sofiane Bendoukha[1], Tobias Weigel[1], Sandro Fiore[2], Alessandro D'Anca[2]

[1]Deutsches Klimarechenzentrum (DKRZ)
[2]Centro Euro-Mediterraneo sui Cambiementi Climatici (CMCC)

Objectives/Benefits

The **ENES Climate Analytics Service (ECAS)** will enable scientific end-users from climate data community and other disciplines to perform data analysis experiments on large volumes of climate data. It follows a **PID-enabled**, **server-side** and **parallel** approach.

Benefits:

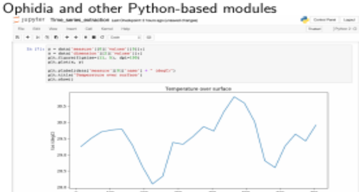
- Reduce the need for local data downloads
- Reduce the effort of maintaining client-side tools
- Reduce the complexity of workflows for simple users
- Encourage flexible and open data sharing
- Enable PID-based provenance support
- More efficient use of computing resources

ECASLab: User-friendly Scientific Data Analysis Environment

ECASLab is the gateway to **ECAS** features. It consists of several components:

- **Ophidia cluster:** a multi-node Ophidia framework instance with WPS-enabled interface accessible through the Ophidia Terminal[a] and any WPS-compliant client;
- **JupyterHub:** instance providing the user with a web-based system for creating, executing and sharing Jupyter notebooks (Python-based) supporting live-coding and visualization;
- A tool for the infrastructure monitoring (**Grafana**)

PyOphidia - the Ophidia Python bindings - allows to easily interact with Ophidia and other Python-based modules

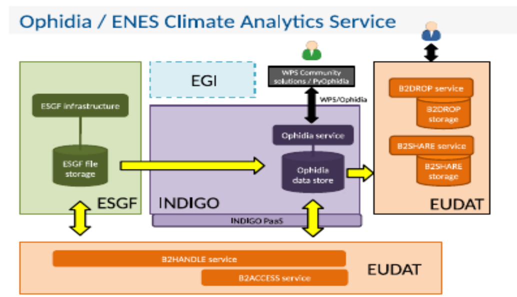


[a]The Ophidia Terminal is a robust, comprehensive, and user-friendly Ophidia client, developed with characteristics similar to the bash shell present in almost all Unix-like environments.

References

- **ESGF**, https://esgf.llnl.gov/.
- **EUDAT Service Suite**, https://sp.eudat.eu/ui/catalogue/services/.
- **Ophidia**, ophidia.cmcc.it.
- **INDIGO-DataCloud**, https://www.indigo-datacloud.eu/.
- **European Network for Earth System Modelling (ENES)**, https://verc.enes.org/.
- S. Fiore, A. D'Anca, D. Elia, C. Palazzo, D. N. Williams, I. T. Foster, and G. Aloisio. Ophidia: A full software stack for scientific data analytics. In *International Conference on High Performance Computing & Simulation, HPCS 2014, Bologna, Italy, 21-25 July, 2014*, pages 343–350. IEEE, 2014.

Service Architecture

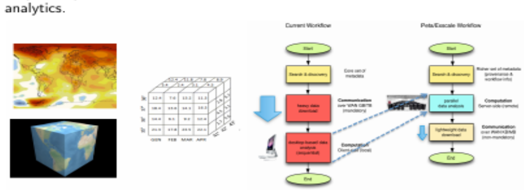


Key integration milestones:

- Enhance ENES/ESGF PID management services to be fully integrated into the EUDAT B2HANDLE service concept
- Provide/enable connectors to allow data transfer between ESGF nodes and Ophidia instances
- Integration of Ophidia with EUDAT B2HANDLE
- Make Ophidia final output available through EUDAT B2DROP and B2SHARE
- Setup and configuration of workflow repository for publishing and sharing workflows

Ophidia: Big Data Analytics Framework

Ophidia is a research effort carried out at the Euro Mediterranean Centre on Climate Change (CMCC) to address big data challenges, issues and requirements for climate change data analytics.



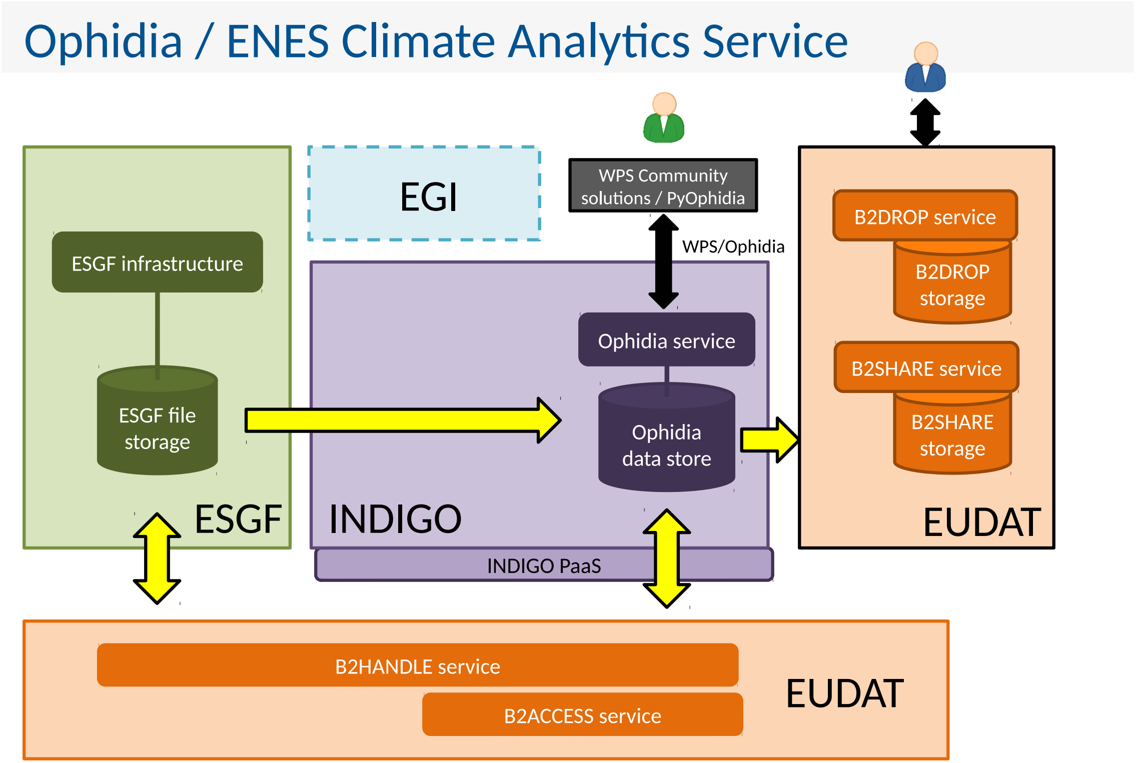
- Server-side, parallel and distributed
- Extensible
- Client application available as user interface
- Workflow support

ENES Climate Analytics Service (ECAS)

- **ECAS** is a contribution to the EOSC-hub project.
- It enables scientific end-users from climate data community and other disciplines to perform data analysis experiments on large volumes of (climate) data.
- **Server-side**, **PID[1]-enabled** and **parallel** approach
  - uplift existing computing capabilities of Ophidia framework to be available as an EOSC service
  - integrate with EOSChub services for ease of use
  - provide hands-on training to overcome adoption barriers

[1] Persistent Identifier

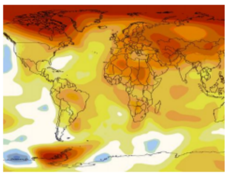
ENES Climate Analytics Service (ECAS)

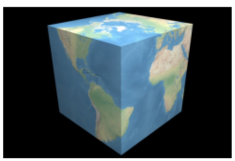


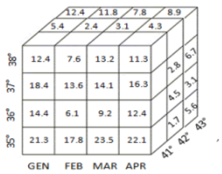
source: Tobias Weigel

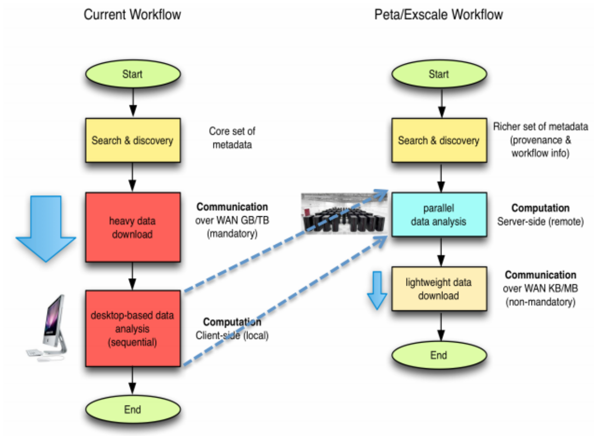
Ophidia: Big Data Analytics Framework

Ophidia is a research effort carried out at the Euro Mediterranean Centre on Climate Change (CMCC) to address big data challenges, issues and requirements for climate change data analytics.









- Server-side, parallel and distributed
- Extensible
- Client application available as user interface
- Workflow support

Ophidia: Server Architecture

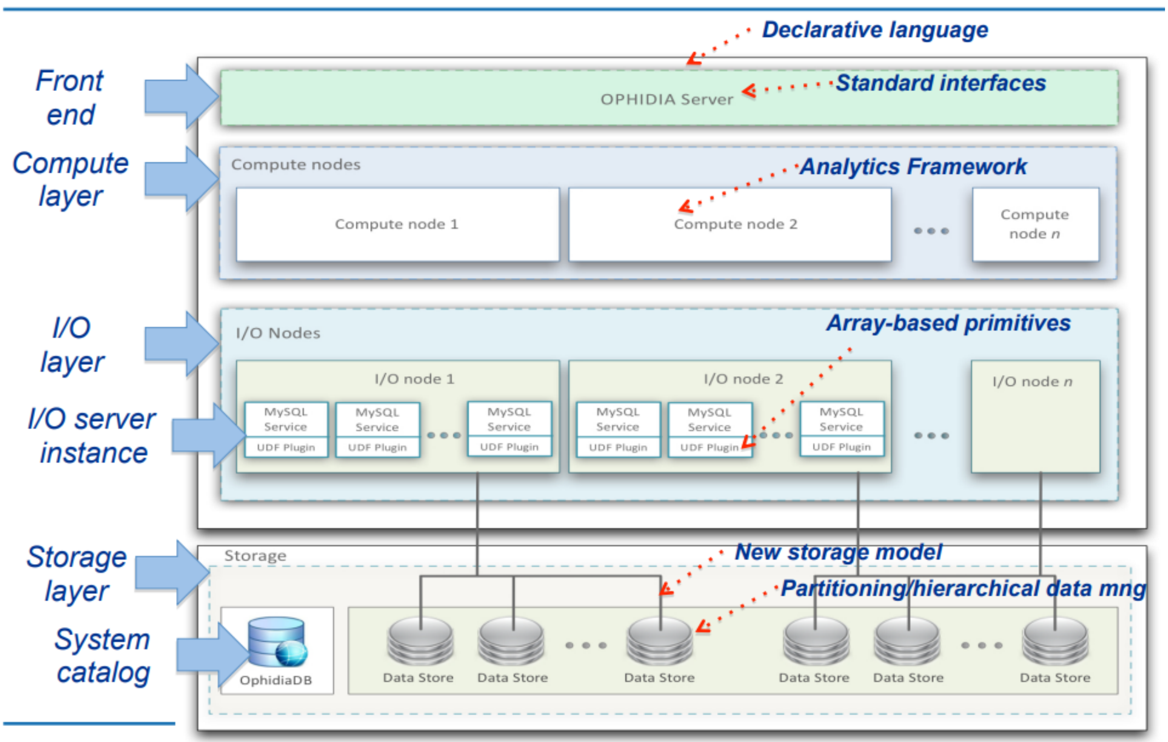


Figure: Sandro Fiore

Ophidia: Array based Primitives

- **Ophidia** provides a wide set of array-based primitives (about 100) to perform summarization, sub-setting, predicates evaluation, statistical analysis, etc
- Primitives comes as plugins and are applied on a single datacube chunk (fragment)
- **Primitives can be nested** to get more complex functionalities
- New primitives can be easily integrated as additional plugins

Ophidia: Data Cube Operators

OPERATOR NAME	OPERATOR DESCRIPTION
Operators "Data processing" – Domain-agnostic	
OPH_APPLY(*datacube_in, datacube_out, array_based_primitive*)	Creates the *datacube_out* by applying the *array-based primitive* to the *datacube_in*
OPH_DUPLICATE(*datacube_in, datacube_out*)	Creates a copy of the *datacube_in* in the *datacube_out*
OPH_SUBSET(*datacube_in, subset_string, datacube_out*)	Creates the *datacube_out* by doing a sub-setting of the *datacube_in* by applying the *subset_string*
OPH_MERGE(*datacube_in, merge_param, datacube_out*)	Creates the *datacube_out* by merging groups of *merge_param* fragments from *datacube_in*
OPH_SPLIT(*datacube_in, split_param, datacube_out*)	Creates the *datacube_out* by splitting into groups of *split_param* fragments each fragment of the *datacube_in*
OPH_INTERCOMPARISON (*datacube_in1, datacube_in2, datacube_out*)	Creates the *datacube_out* which is the element-wise difference between *datacube_in1* and *datacube_in2*
OPH_DELETE(*datacube_in*)	Removes the *datacube_in*

Data processing (parallel operators, MPI & OpenMP based)

Import/Export (parallel operators)

Data Access (sequential and parallel operators)

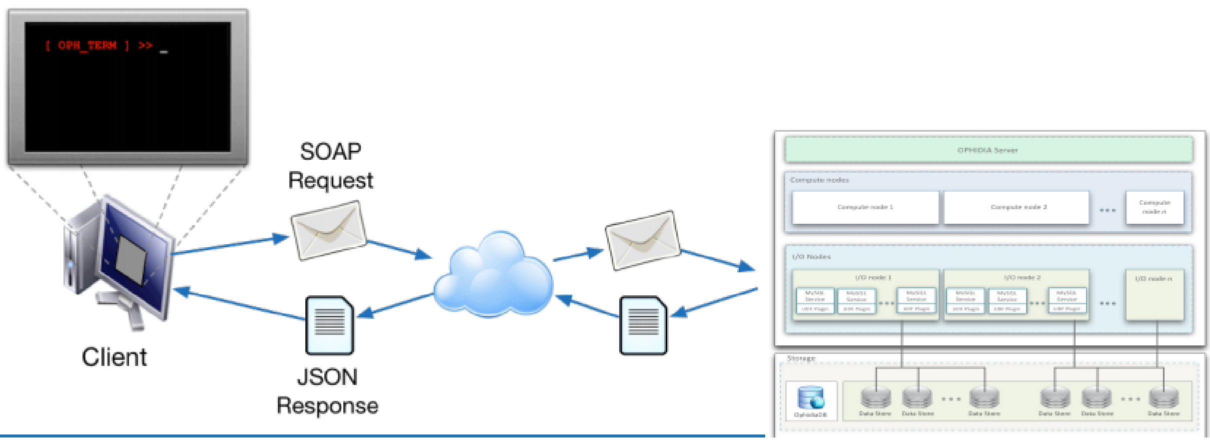
Metadata management (sequential and parallel operators)

OPERATOR NAME	OPERATOR DESCRIPTION
Operators "Data processing" – Domain-oriented	
OPH_EXPORT_NC (*datacube_in, file_out*)	Exports the *datacube_in* data into the *file_out* NetCDF file.
OPH_IMPORT_NC (*file_in, datacube_out*)	Imports the data stored into the *file_in* NetCDF file into the new *datacube_in* datacube
Operators "Data access"	
OPH_INSPECT_FRAG (*datacube_in, fragment_in*)	Inspects the data stored in the *fragment_in* from the *datacube_in*
OPH_PUBLISH(*datacube_in*)	Publishes the *datacube_in* fragments into HTML pages
Operators "Metadata"	
OPH_CUBE_ELEMENTS (*datacube_in*)	Provides the total number of the elements in the *datacube_in*
OPH_CUBE_SIZE (*datacube_in*)	Provides the disk space occupied by the *datacube_in*
OPH_LIST(void)	Provides the list of available datacubes.
OPH_CUBEIO(*datacube_in*)	Provides the provenance information related to the *datacube_in*
OPH_FIND(*search_param*)	Provides the list of datacubes matching the *search_param* criteria

source: Sandro Fiore

Ophidia client: The Ophidia Terminal

oph_term is a terminal-like commands interpreter serving as a client for the Ophidia framework.
Through the **oph_term** the user runs (*send*) commands (*operators*) to the Ophidia framework to manipulate datasets (*datacubes*).



Ophidia client: The Ophidia Terminal

```
[25..7214] >> ./src/ophidia-workflow-catalogue/indigo/precip_trend_analisys/precip_trend_analysis.json 2 CMCC-CM rcp85 day 0.9 1976_2006 2071_2101 30:45|0:40 r360x180
[JobID]:
http://127.0.0.1/ophidia/sessions/252514038013045837291501061213487214/experiment?430#1767

[Response]:
Workflow Status
---------------

OPH_STATUS_COMPLETED

Workflow Progress
-----------------
```

NUMBER OF COMPLETED TASKS	NUMBER OF SKIPPED TASKS	TOTAL NUMBER OF TASKS
17	9	26

Workflow Task List

OPH JOB ID	SESSION CODE	WORKFLOW ID	MARKER ID	PARENT MARKER ID	TASK NAME	TYPE	EXIT STATUS
http://127.0.0.1/ophidia/sessions/252514038013045837291501061213487214/experiment?430#1768	252514038013045837291501061213487214	430	1768	1767	Create Work Container	SIMPLE	OPH_STATUS_SKIPPED
http://127.0.0.1/ophidia/sessions/252514038013045837291501061213487214/experiment?430#1769	252514038013045837291501061213487214	430	1769	1767	Create Container Historical	SIMPLE	OPH_STATUS_SKIPPED
http://127.0.0.1/ophidia/sessions/252514038013045837291501061213487214/experiment?430#1770	252514038013045837291501061213487214	430	1770	1767	Create Container Scenario	SIMPLE	OPH_STATUS_SKIPPED
http://127.0.0.1/ophidia/sessions/252514038013045837291501061213487214/experiment?430#1771	252514038013045837291501061213487214	430	1771	1767	Import Type Selection Historical	SIMPLE	OPH_STATUS_COMPLETED
http://127.0.0.1/ophidia/sessions/252514038013045837291501061213487214/experiment?430#1772	252514038013045837291501061213487214	430	1772	1767	Import Type Selection Scenario	SIMPLE	OPH_STATUS_COMPLETED
http://127.0.0.1/ophidia/sessions/252514038013045837291501061213487214/experiment?430#1773	252514038013045837291501061213487214	430	1773	1767	Import Historical	SIMPLE	OPH_STATUS_COMPLETED
http://127.0.0.1/ophidia/sessions/252514038013045837291501061213487214/experiment?430#1775	252514038013045837291501061213487214	430	1775	1767	Import Scenario	SIMPLE	OPH_STATUS_COMPLETED
http://127.0.0.1/ophidia/sessions/252514038013045837291501061213487214/experiment?430#1777	252514038013045837291501061213487214	430	1777	1767	End Import Type Selection Scenario	SIMPLE	OPH_STATUS_COMPLETED
http://127.0.0.1/ophidia/sessions/252514038013045837291501061213487214/experiment?430#1778	252514038013045837291501061213487214	430	1778	1767	Subset JJA Scenario	SIMPLE	OPH_STATUS_COMPLETED

Ophidia client: PyOphidia

Python package to interact with the Ophidia framework

- Alternative to Oph Term
- Runs on Python 2.7, 3.3, 3.4 and 3.5
- Requires a running Ophidia instance
- It provides two main modules:
  - **client.py:** generic low level class to submit any type of requests (simple tasks and workflows), using SSL and SOAP with the client ophsubmit.py
  - **cube.py:** high level cube-oriented class to interact directly with cubes, with several methods wrapping the operators.

Ophidia client: PyOphidia

- Instantiate a client and submit a request

```
from PyOphidia import client

ophclient = client.Client("oph-user", "oph-password", "127.0.0.1", "11732")

ophclient.submit("oph_list level=2", display=True)
```

- Set a Client for the Cube class

```
from PyOphidia import cube

cube.Cube.setclient('oph-user','oph-password','127.0.0.1','11732')

#Export to NetCDF file
cube.Cube.createcontainer(container='test',dim='lat|lon|time',dim_type='double|double|double')

#Show a Cube structure and info
mycube1.info()

#Subset a Cube
mycube=mycube1.subset(subset_dims='lat|lon',subset_filter='1:10|20:30',subset_type='coord')
```

Workflow Support

```
{
        "name": "Example1",
        "author": "Foo",
        "abstract": "Simple workflow with three tasks",
        "exec_mode": "sync",
        "ncores": "1",
        "cube": "http://hostname/1/1",
        "tasks":
        [
                {
                        "name": "Extract maximum value",
                        "operator": "oph_reduce",
                        "arguments": [ "operation=max" ]
                },
                {
                        "name": "Extract minimum value",
                        "operator": "oph_reduce",
                        "arguments": [ "operation=min" ]
                },
                {
                        "name": "Evaluate max-min range",
                        "operator": "oph_intercube",
                        "arguments": [ "operation=sub" ],
                        "dependencies":
                        [
                                { "task": "Extract maximum value", "type": "single", "argument":
                                      "cube" },
                                { "task": "Extract minimum value", "type": "single", "argument":
                                      "cube2" }
                        ]
                }
        ]
}
```

Ophidia Use Cases

- Length of snow season
- Long-term monthly average of Snow Water Equivalent
- Monthly average of Sea Surface Temperature
- Daily Temperature Range
- Precipitation Trend Analysis

Precipitation Trend Analysis

Description: Precipitation trend analysis, on different spatial and temporal scales, has received notable attention during the past century due to its relations with global climate change stated by the scientific community. For this reason, a number of models for this atmospheric variable have been defined.

GitHub repository: https://github.com/OphidiaBigData/ophidia-workflow-catalogue/blob/master/indigo/precip_trend_analisys/optimized_precip_trend_analysis.json

OphidiaLab repository: ./workflows/precip_trend_analisys/optimized_precip_trend_analysis.json

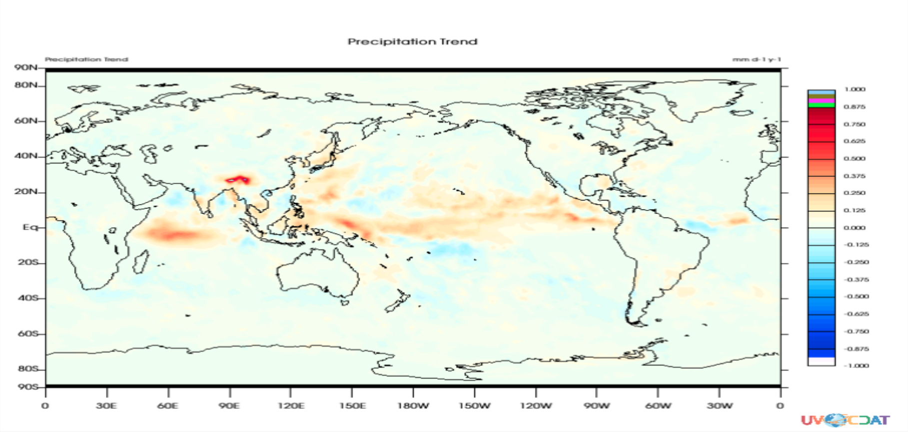
Arguments: number of cores; model name; spatial subset; time subset in the past; time subset in the future; output grid; import type (optional); I/O server type (optional)

Example: ./optimized_precip_trend_analysis.json 2 CMCC-CM|CMCC-CMS rcp85 day 0.9 1976_2006 2071_2101 -90:90|0:360 r360x180

Number of tasks: 77

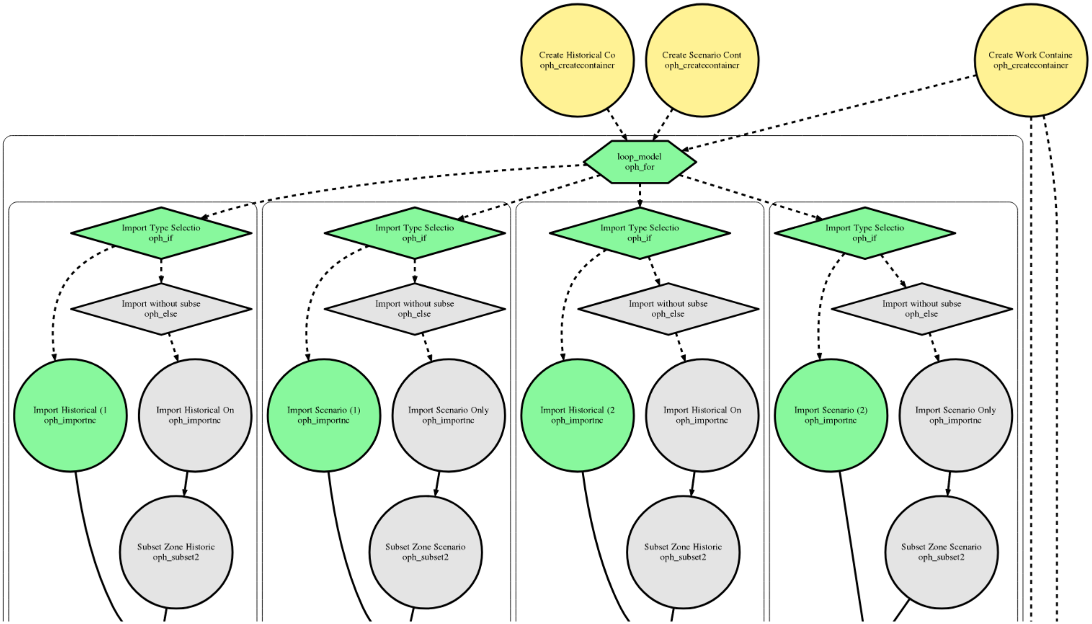
Ophidia Use Cases

Output: 5 png files: maximum temperature, minimum temperature, average, standard deviation, variance



Reference: S. Fiore, M. Plociennik, C. Doutriaux, C. Palazzo, J. Boutte, T. Zok, D. Elia, M. Owsiak, A. D'Anca, Z. Shaheen, R. Bruno, M Fargetta, M. Caballer, G. Moltó , I. Blanquer, R. Barbera, M. David, G. Donvito, D. N. Williams, V. Anantharaj, D. Salomoni, G. Aloisio - "Distributed and cloud-based multi-model analytics experiments on large volumes of climate change data in the Earth System Grid Federation eco-system" - in Proceedings of Workshop "Big Data Challenges, Research, and Technologies in the Earth and Planetary Sciences" within the 2016 IEEE International Big Data Conference, pp. 2911-2918, 5-8 December 2016, Washington D.C., USA.

Ophidia Use Cases



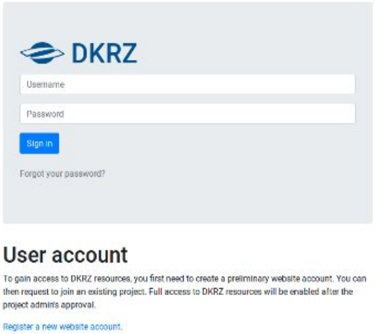
ECASLab

A user-friendly scientific data analysis environment.
ECASLab consists of:

- **Ophidia cluster**
- JupyterHub instance
  - Jupyter notebook
  - Ophidia client (PyOphidia, Ophidia terminal)
- Set of pre-installed Python libraries for running data manipulation, analysis and visualization
- Data publication service

How to use it?

(1)



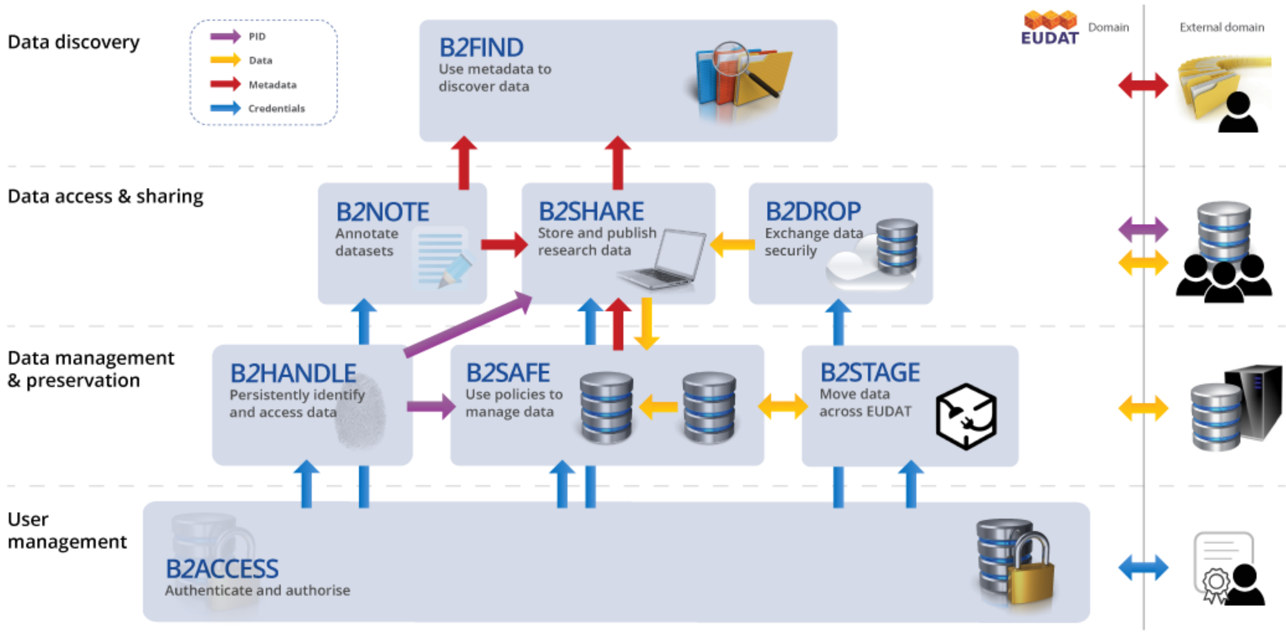
(2)

(3)

1- Register*
2- Login to JupyterHub
3- Perform data analysis

* At CMCC or DKRZ

Integration with EUDAT Services



source: https://eudat.eu/

Integration with EUDAT Services

- **B2DROP:** Through B2DROP, ECAS provides a secure and trusted data exchange service to store and exchange processed data outputs
- **B2SHARE:** ECAS offer users to receive, store and publish processed data outputs
- **B2HANDLE:** Through the B2HANDLE service, ECAS assigns Persistent Identifiers (PID) to Ophidia output and records kernel information according to a community profile

For more informations

- **Thursday 16:45, oral presentation on ECASLab. Room M2.**
- EOSC-HUB project `http://eosc-hub.eu/`
- EUDAT services `www.eudat.eu/services`
- Ophidia `http://ophidia.cmcc.it/`
- Ophidia use cases `https://github.com/OphidiaBigData/ophidia-workflow-catalogue`
- Get in touch ecas-support@dkrz.de