Data Analysis made easy with the ENES Climate Analytics Service (ECAS)

Sofiane Bendoukha, Donatello Elia, Fabrizio Antonio, Sandro Fiore, Tobias Weigel

April 10, 2019 @ EGU 19
SC1.22

eosc-hub.eu

@EOSC_eu

Dissemination level: Public

EOSC-hub receives funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 777536.

Useful links

- Training materials
  - https://github.com/ECAS-Lab/ecas-training
- ECASLab / JupyterHUb
  - **ECASLab @ DKRZ** https://ecaslab.dkrz.de
  - **ECASLab @ CMCC** https://ecaslab.cmcc.it
- Ophidia framework documentation
  - http://ophidia.cmcc.it/documentation/users/index.html

ENES Climate Analytics Service (ECAS)

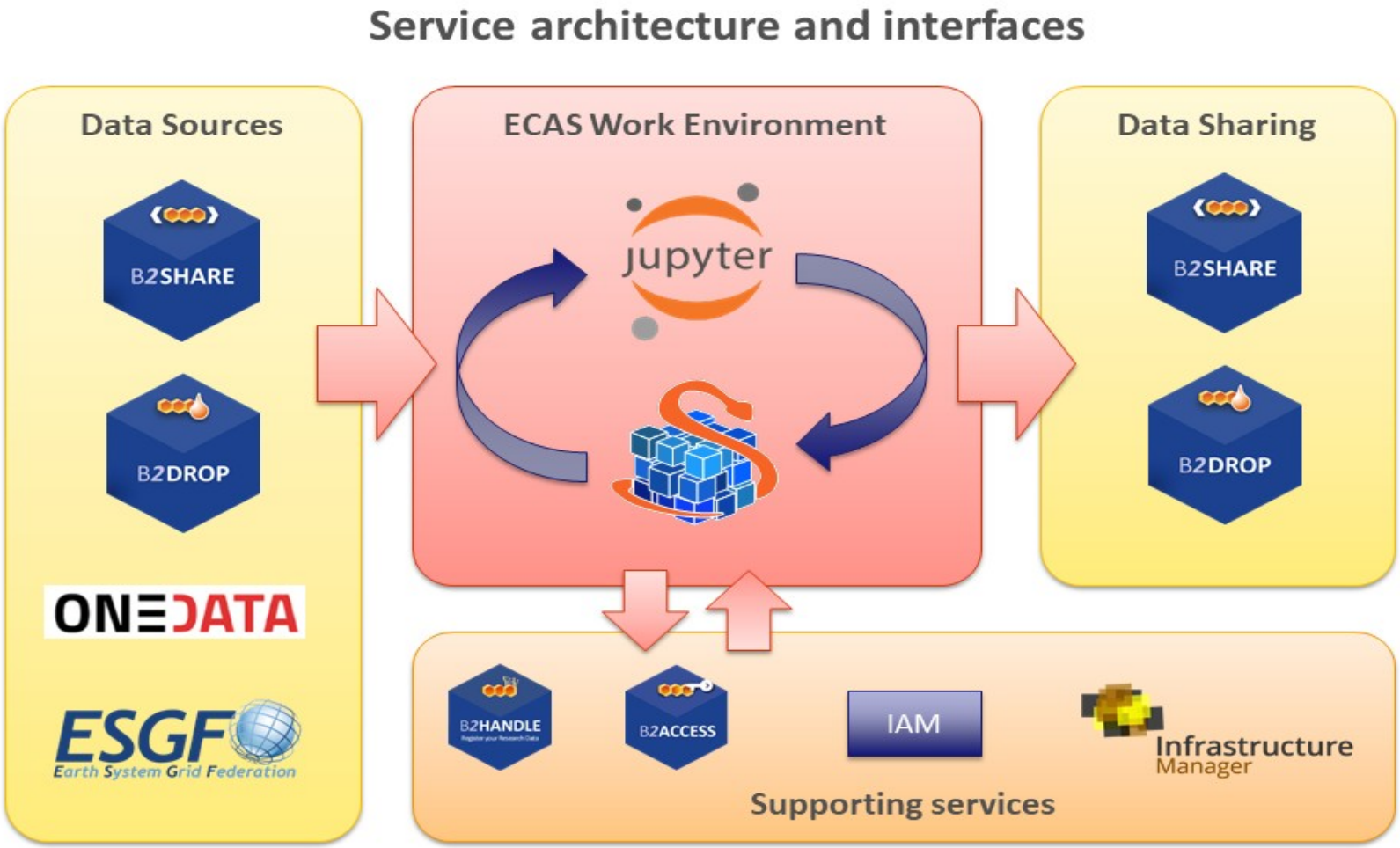
- ECAS is part of the **EOSC-HUB** service catalogue

- Server-based
  - Computation @ **CMCC** or **DKRZ** instances
  - Avoid data transfer (download)
- User-friendly
  - **ECASLab** provides a JupyterHub server
  - Interactive computation (programming) based on Jupyter Notebooks
- ECAS supports different Auth* providers
  - Local and external AAI providers supported (LDAP, B2ACCESS, EGI Check In)
  - Additional AAI providers can be integrated if needed

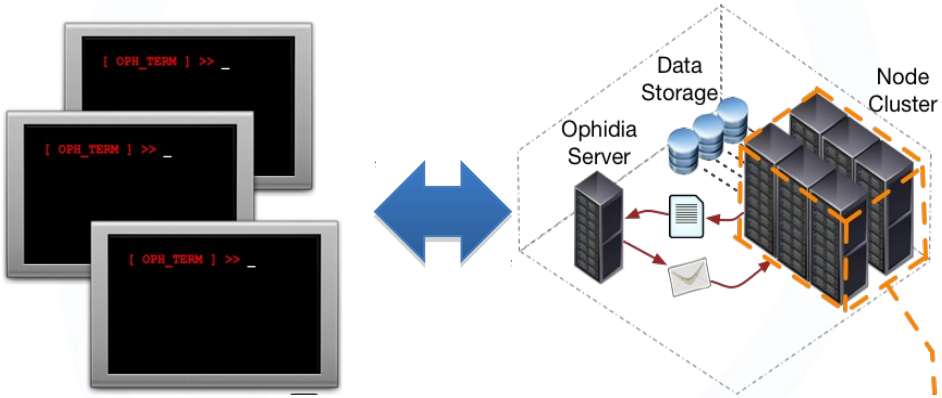
Data available

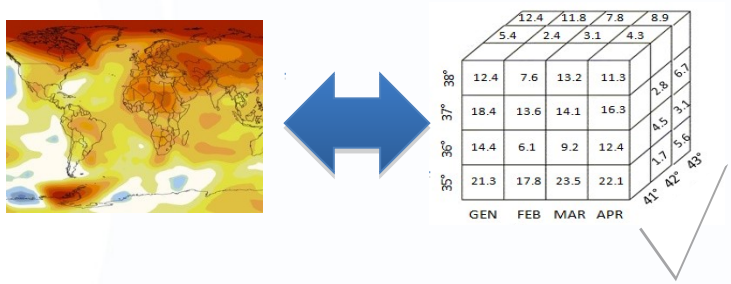
- ECAS provides data access via ESGF
- Coordinated Regional Climate Downscaling Experiment
  - ~ 100 Tbyte Cordex
- Coupled Model Intercomparison Project 5
  - ~ 1.2 Pbyte CMIP5 Data
- Coupled Model Intercomparison Project 6
  - ~ 250 Tbyte CMIP6 Data from the 1PByte published
- Other Data pools can be mounted on demand
  - MPI Grand ensemble (**MPI - GE**)

ECAS in a Nutshell



Server-side paradigm and datacube abstraction in Ophidia

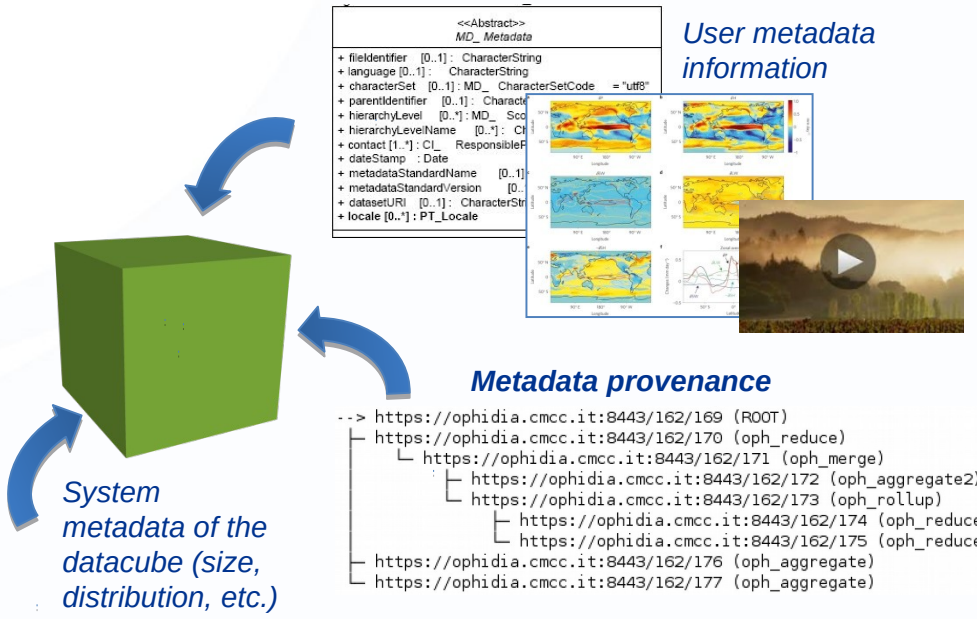




***Oph_Term**: a terlminal-like commands interpreter serving as a client for the Ophidia framework*

***Ophidia framework**: declarative, parallel server-side processing*

*Through the **oph_term** the user run ("send") commands ("operators") to the Ophidia framework to manipulate datasets ("datacubes")*



Array based primitives (about 100)

- *Ophidia provides a **wide set of array-based primitives** to perform data summarization, sub-setting, predicates evaluation, statistical analysis, compression, etc.*
- *Primitives come as plugins and are applied on a single datacube chunk (fragment)*
- *They are provided both for **byte**-oriented and **bit**-oriented arrays*
- ***Primitives can be nested** to get more complex functionalities*
- *Compression can be a primitive too*
- *New primitives can be easily integrated as additional plugins*

Ophidia 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)

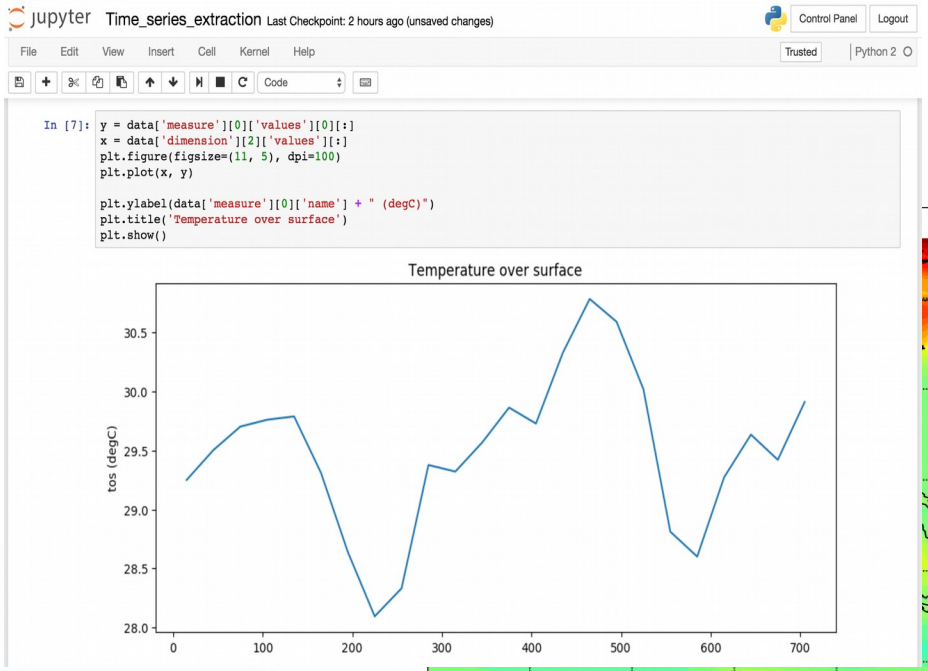
Data Access (sequential and parallel operators)

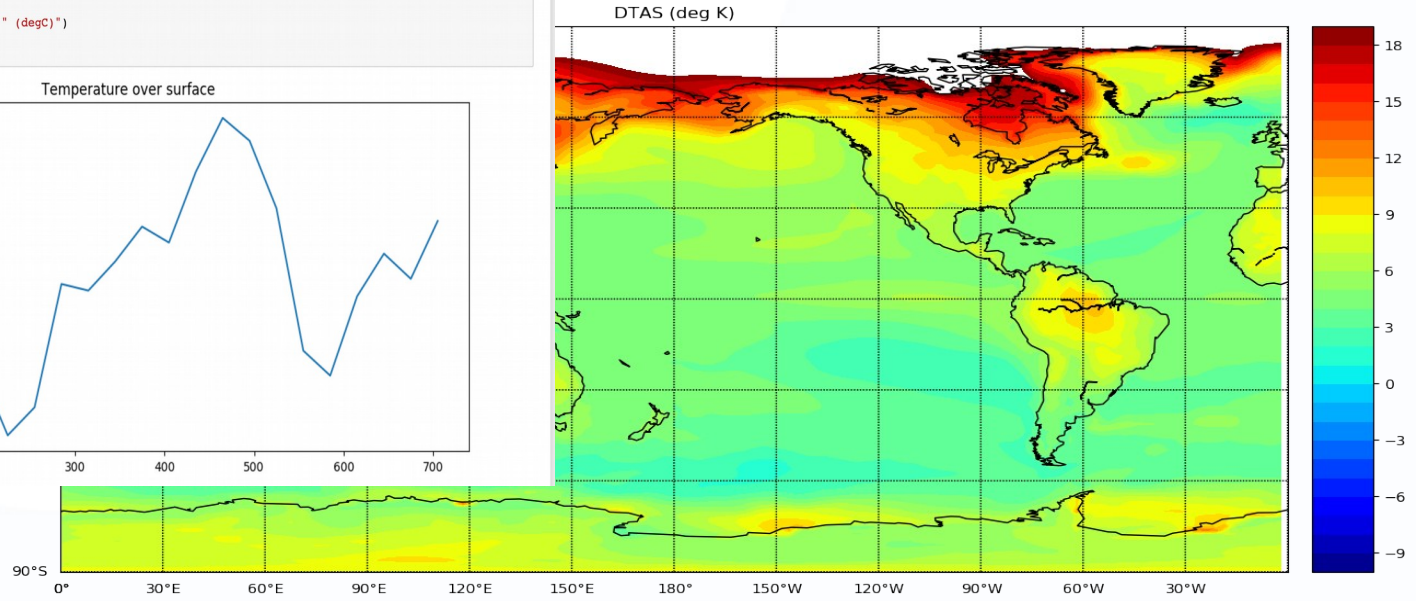
Metadata management (sequential and parallel operators)

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

ECASLab

- Provides a user-friendly scientific data analysis environment deployed at CMCC and DKRZ based on ECAS





Thank you for your attention!

Questions?

Sofiane Bendoukha bendoukha@dkrz.de
Donatello Elia donatello.elia@cmcc.it
Fabrizio Antonio fabrizio.antonio@cmcc.it

EOSC-hub

eosc-hub.eu @EOSC_eu

Useful links

- Training materials
  - https://github.com/ECAS-Lab/ecas-training
- ECASLab / JupyterHUb
  - **ECASLab @ DKRZ** https://ecaslab.dkrz.de
  - **ECASLab @ CMCC** https://ecaslab.cmcc.it
- Ophidia framework documentation
  - http://ophidia.cmcc.it/documentation/users/index.html