Destination Earth

A Notebook-Based approach to visualise data from DestinE Data portfolio

Authors: Patryk Grzybowski[1]; Marcin Ziółkowski[1]; Aubin Lambare[2]; Christoph Reimer[3]; Michael Schick[4]

Affiliations: [1]CloudFerro S.A., Warsaw, Poland; [2]CS Group, Le Plessis Robinson, France; [3]EODC, Vienna, Austria; [4]EUMETSAT, Darmstadt, Germany

Destination Earth Funded by the European Union Implemented by ECMWF esa EUMETSAT

STACK

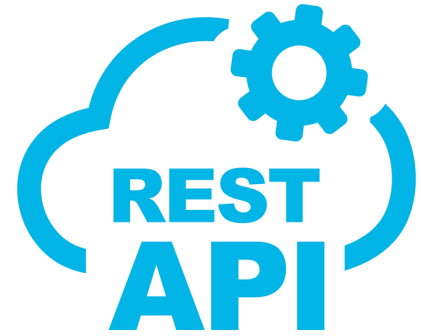
- RESTAPI
- EODAG
- PySTAC
- Xcube

Istlet

- Open Data Cube

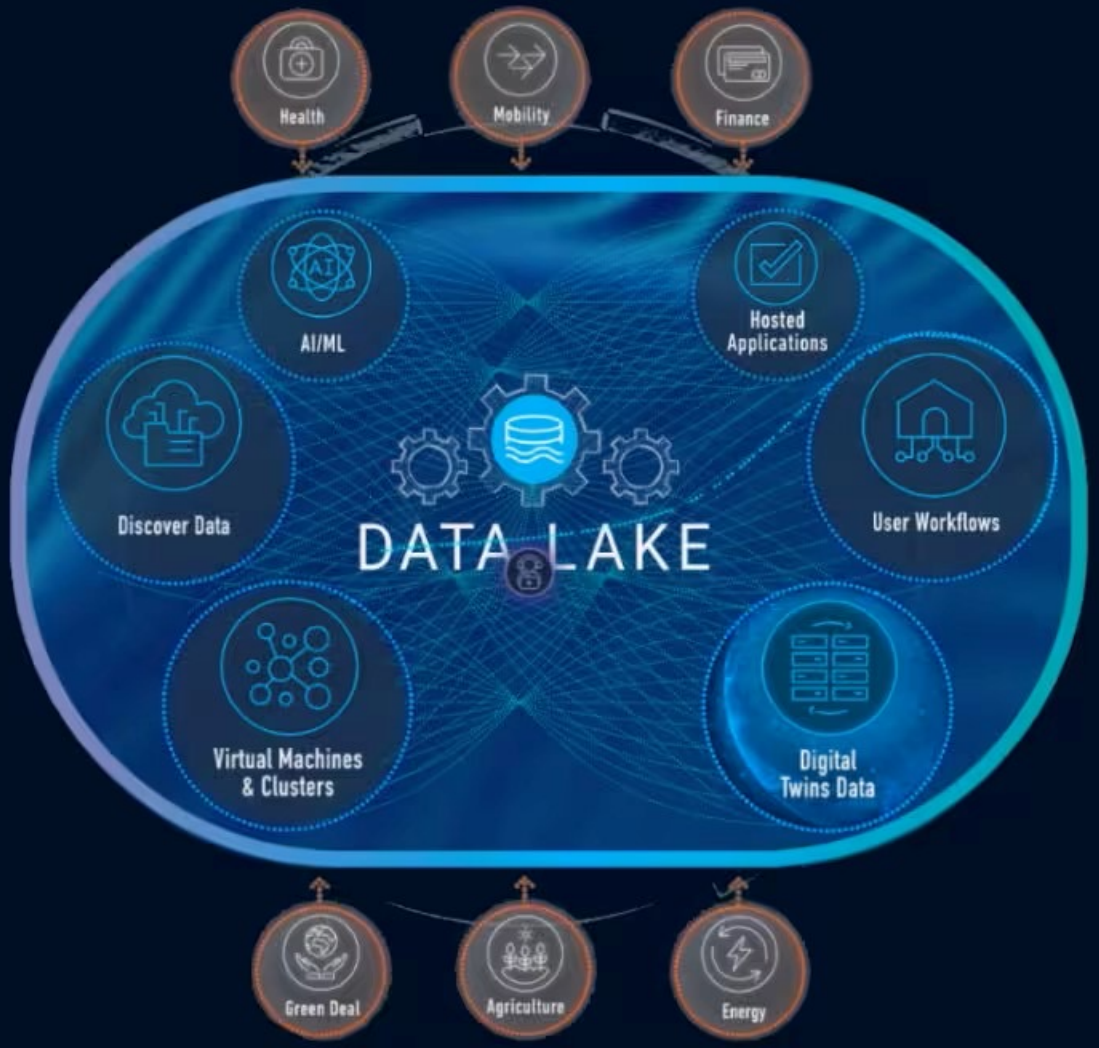
matplotlib

dask



Funded by the European Union Destination Earth IMPLEMENTED BY EUMETSAT esa ECMWF

Destination Earth



STACK

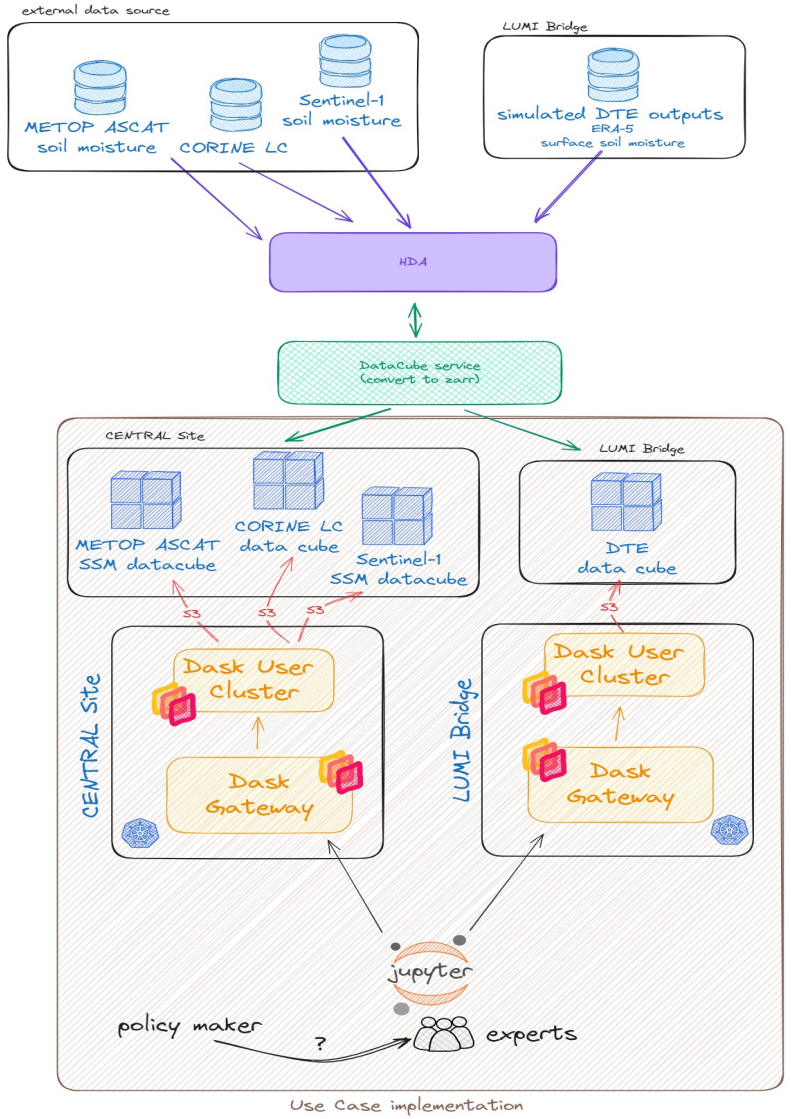
Data Bridges

STACK

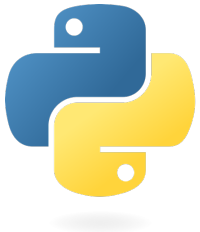
JupyterLab

DASK clusters

HDA: API
PySTAC
EODAG



Funded by the European Union
Destination Earth
IMPLEMENTED BY EUMETSAT esa ECMWF

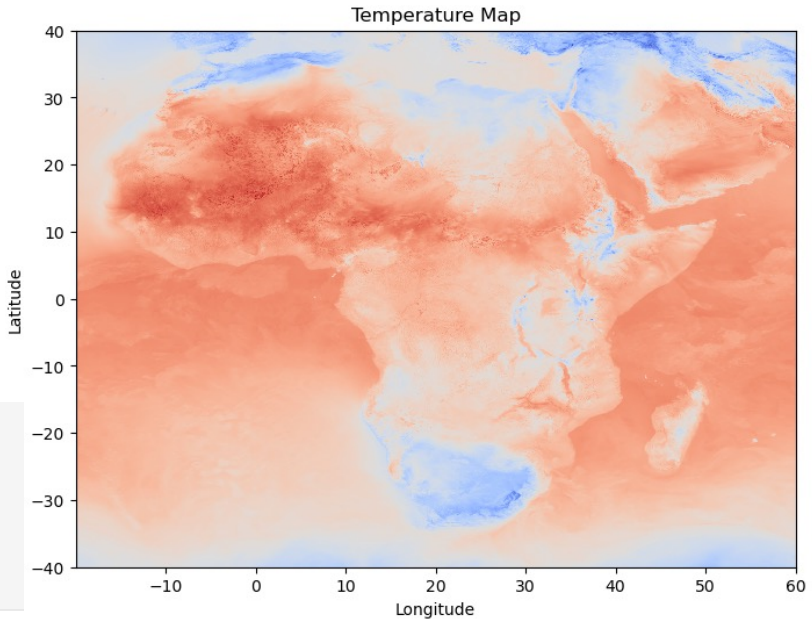


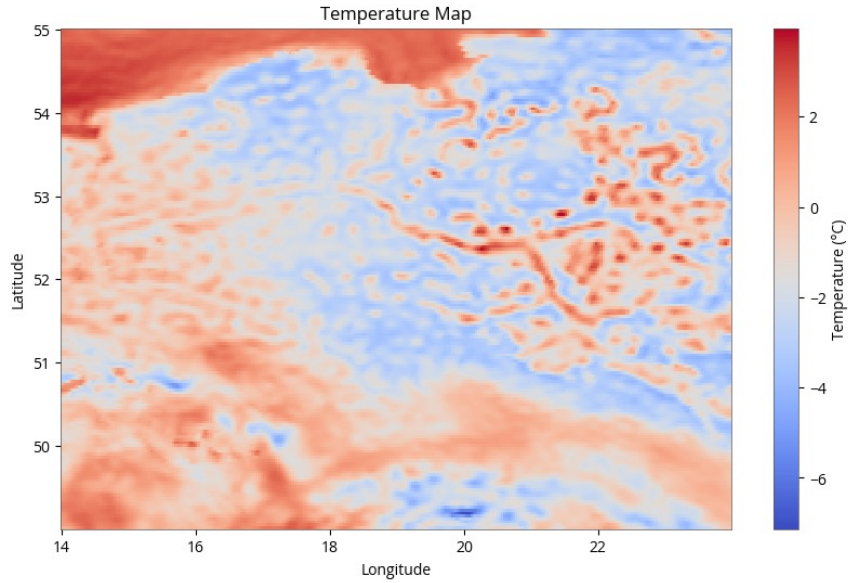
matplotlib

```
from IPython.display import JSON
import xarray as xr
import cfgrib
import numpy as np
from tqdm import tqdm
import time
import re
import json
import requests
```

```
import matplotlib as plt
```

```
plt.figure(figsize=(10, 6))
plt.pcolormesh(lon, lat, temperature, cmap='coolwarm')
plt.colorbar(label='Temperature (°C)')
plt.title('Temperature Map')
plt.xlabel('Longitude')
plt.ylabel('Latitude')
```





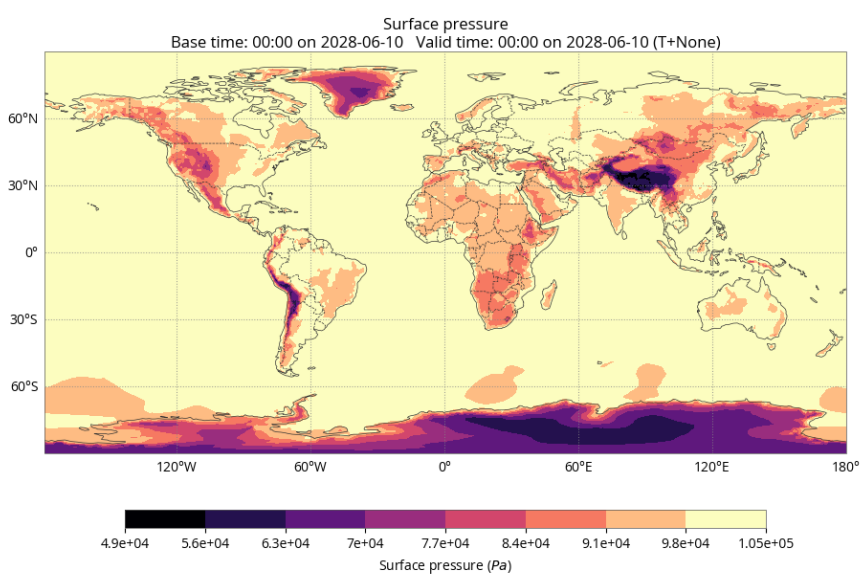
Funded by the European Union
Destination Earth
IMPLEMENTED BY EUMETSAT esa ECMWF

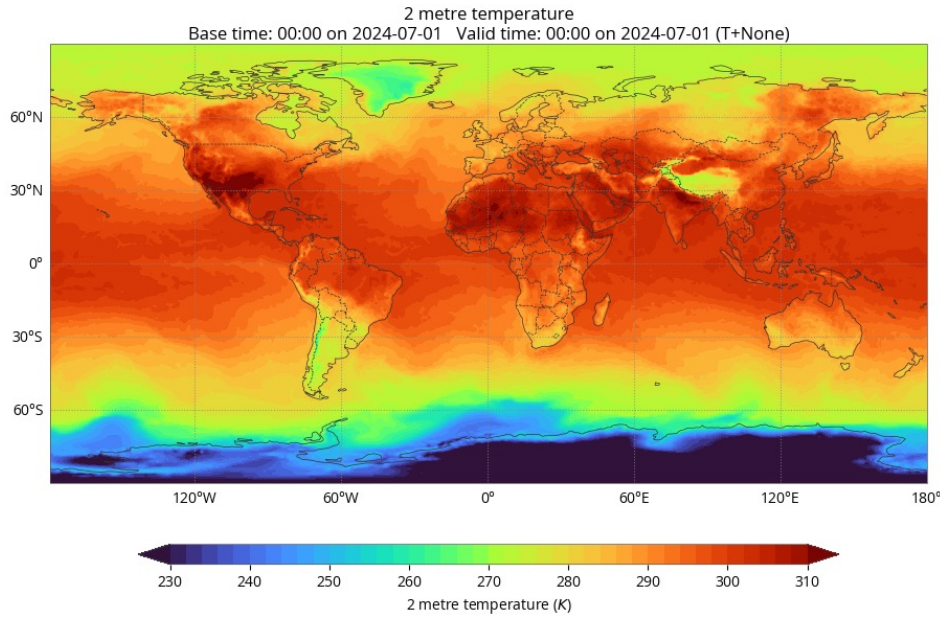
earthkkit

```
import requests
import json
import os
from getpass import getpass
import destinelab as deauth

import importlib.metadata
```

```
import earthkit.data
import earthkit.maps
import earthkit.regrid
```

```
data = earthkit.data.from_source("file", filename)
data.ls
earthkit.maps.quickplot(data,#style=style
                        )
```





Extremely useful with Digital Twin outputs!

Funded by the European Union Destination Earth IMPLEMENTED BY EUMETSAT esa ECMWF

STAC
SpatioTemporal
Asset Catalog

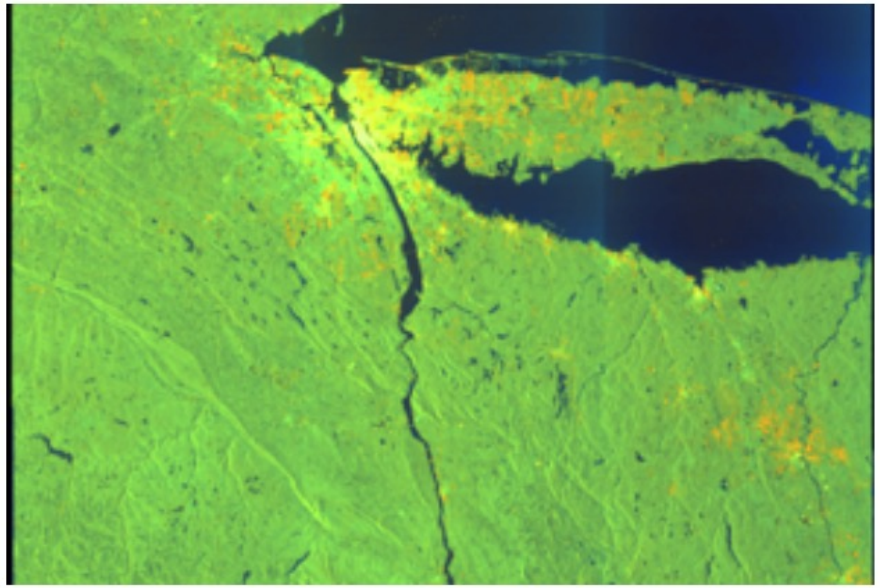
```python
import rich.table

selected_item = coll_items[3]

table = rich.table.Table(title="Assets in STAC Item")
table.add_column("Asset Key", style="cyan", no_wrap=True)
table.add_column("Description")
for asset_key, asset in selected_item.assets.items():
    table.add_row(asset_key, asset.title)

console.print(table)
```

Assets in STAC Item

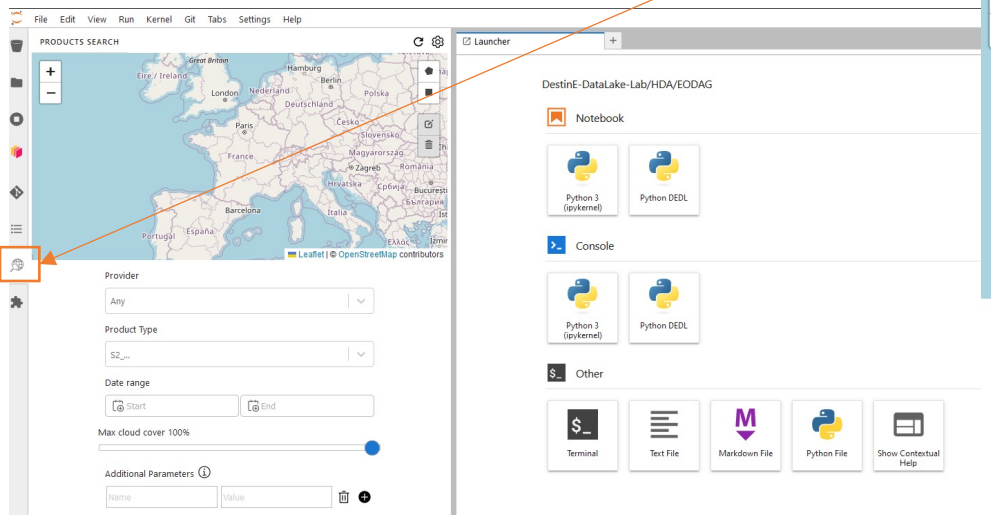
Asset Key	Descript…
downloadLink	Download link
S1A_IW_GRDH_1SDV_20230914T225135_20230914T225200_050330_060F40_9F54.SAFE-report-20230915T011838.pdf	S1A_IW_G…
calibration-s1a-iw-grd-vh-20230914t225135-20230914t225200-050330-060f40-002.xml	calibrat…
calibration-s1a-iw-grd-vv-20230914t225135-20230914t225200-050330-060f40-001.xml	calibrat…
noise-s1a-iw-grd-vh-20230914t225135-20230914t225200-050330-060f40-002.xml	noise-s1…
noise-s1a-iw-grd-vv-20230914t225135-20230914t225200-050330-060f40-001.xml	noise-s1…
rfi-s1a-iw-grd-vh-20230914t225135-20230914t225200-050330-060f40-002.xml	rfi-s1a-…
rfi-s1a-iw-grd-vv-20230914t225135-20230914t225200-050330-060f40-001.xml	rfi-s1a-…
s1a-iw-grd-vh-20230914t225135-20230914t225200-050330-060f40-002.xml	s1a-iw-g…
s1a-iw-grd-vv-20230914t225135-20230914t225200-050330-060f40-001.xml	s1a-iw-g…
manifest.safe	manifest…
s1a-iw-grd-vh-20230914t225135-20230914t225200-050330-060f40-002.tiff	s1a-iw-g…
s1a-iw-grd-vv-20230914t225135-20230914t225200-050330-060f40-001.tiff	s1a-iw-g…
logo.png	logo.png
map-overlay.kml	map-over…
product-preview.html	product-…
quick-look.png	quick-lo…
thumbnail.png	thumbnai…
s1-level-1-calibration.xsd	s1-level…
s1-level-1-measurement.xsd	s1-level…
s1-level-1-noise.xsd	s1-level…
s1-level-1-product.xsd	s1-level…
s1-level-1-quicklook.xsd	s1-level…
s1-level-1-rfi.xsd	s1-level…
s1-map-overlay.xsd	s1-map-o…
s1-object-types.xsd	s1-objec…
s1-product-preview.xsd	s1-produ…
thumbnail	Thumbnail

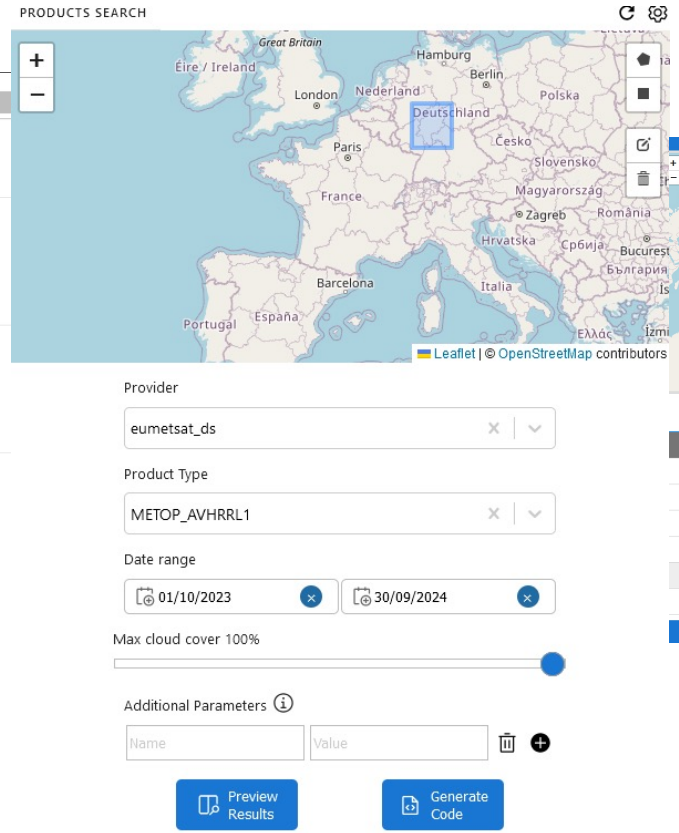
[16]:
```python
from IPython.display import Image

Image(url=selected_item.assets["thumbnail"].href, width=500)
```

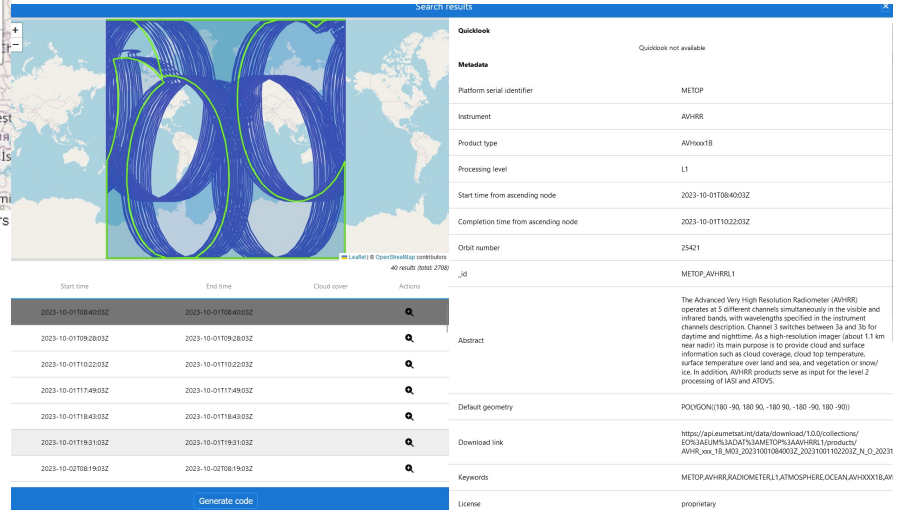
[16]:



Funded by the European Union
Destination Earth
IMPLEMENTED BY EUMETSAT esa ECMWF







Funded by the European Union
Destination Earth
IMPLEMENTED BY EUMETSAT esa ECMWF

Data Cubes

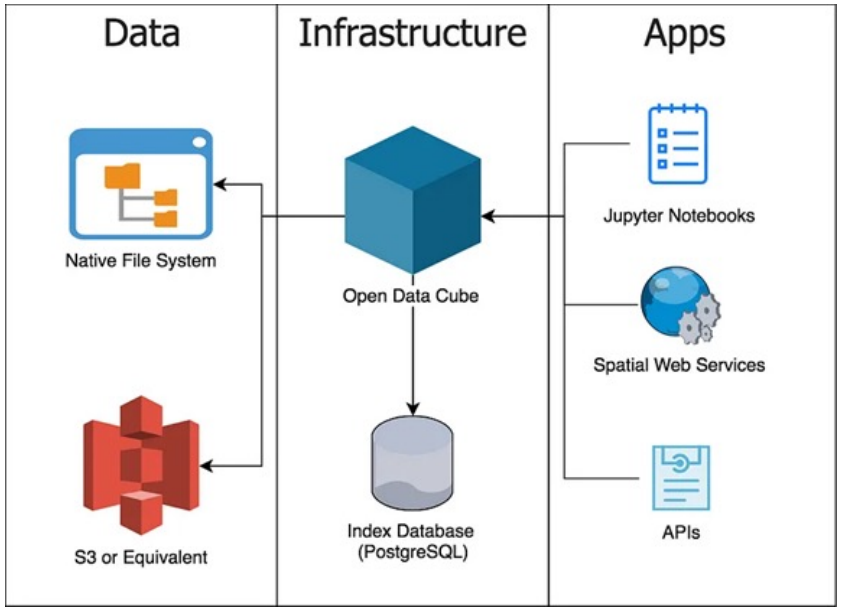
STACK/Islet/Locally

- Xarray and Xcube

```
<xarray.DataArray 'time' (time: 25)>
array(['2024-04-10T00:00:00.000000000', '2024-04-10T01:00:00.000000000',
       '2024-04-10T02:00:00.000000000', '2024-04-10T03:00:00.000000000',
       '2024-04-10T04:00:00.000000000', '2024-04-10T05:00:00.000000000',
       '2024-04-10T06:00:00.000000000', '2024-04-10T07:00:00.000000000',
       '2024-04-10T08:00:00.000000000', '2024-04-10T09:00:00.000000000',
       '2024-04-10T10:00:00.000000000', '2024-04-10T11:00:00.000000000',
       '2024-04-10T12:00:00.000000000', '2024-04-10T13:00:00.000000000',
       '2024-04-10T14:00:00.000000000', '2024-04-10T15:00:00.000000000',
       '2024-04-10T16:00:00.000000000', '2024-04-10T17:00:00.000000000',
       '2024-04-10T18:00:00.000000000', '2024-04-10T19:00:00.000000000',
       '2024-04-10T20:00:00.000000000', '2024-04-10T21:00:00.000000000',
       '2024-04-10T22:00:00.000000000', '2024-04-10T23:00:00.000000000',
       '2024-04-11T00:00:00.000000000'], dtype='datetime64[ns]')
Coordinates:
  * time     (time) datetime64[ns] 2024-04-10 2024-04-10T01:00:00 ... 2024-04-11
Attributes:
    axis:           T
    standard_name:  time
```

ISLET

- Open Data Cube



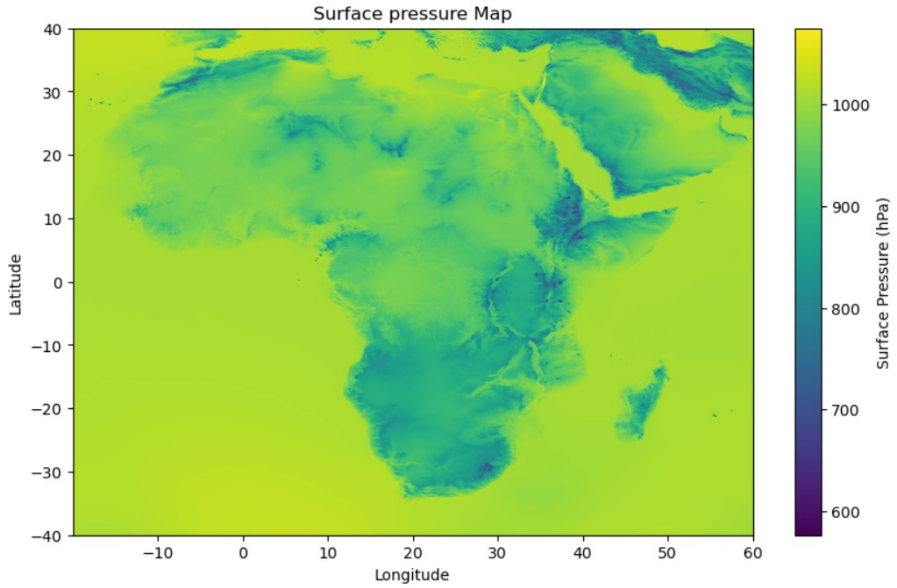
Funded by the European Union Destination Earth IMPLEMENTED BY EUMETSAT esa ECMWF

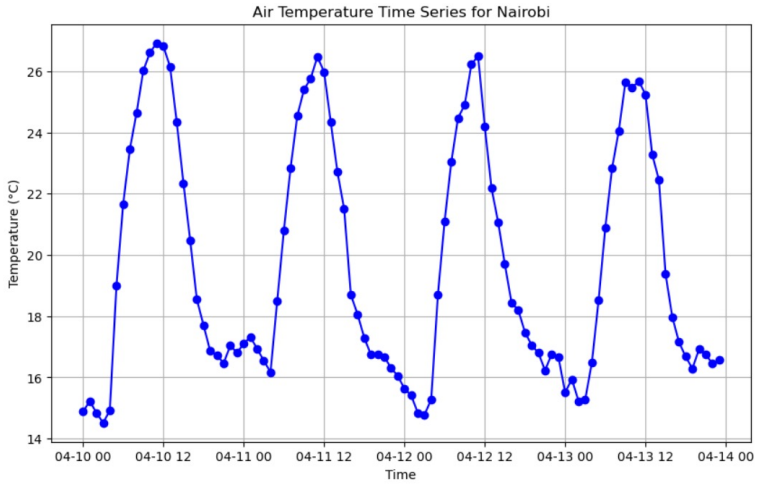
```
africa_bbox = [-20,     # West
               -40,     # South
               60,      # East
               40]      #North

africa_dt = xr.open_zarr(url)['sp'].sel(lon=slice(africa_bbox[0],
                                                  africa_bbox[2]),
                                        lat=slice(africa_bbox[3],
                                                  africa_bbox[1]),
                                        time=slice('20240410T000000', '20240411T000000')
                                                  )

print(africa_dt.var)
```

```
<bound method DataArrayAggregations.var of <xarray.DataArray 'sp' (time: 25, lat: 2276, lon: 2279)>
dask.array<getitem, shape=(25, 2276, 2279), dtype=float32, chunksize=(25, 417, 837), chunktype=numpy.ndarray>
Coordinates:
  * lat      (lat) float64 39.99 39.95 39.92 39.88 ... -39.92 -39.95 -39.99
  * lon      (lon) float64 -19.97 -19.94 -19.9 -19.87 ... 59.92 59.95 59.99
  * time     (time) datetime64[ns] 2024-04-10 2024-04-10T01:00:00 ... 2024-04-11
Attributes:
    CDI_grid_num_LPE:  2560
    CDI_grid_type:     gaussian
    long_name:         Surface pressure
    param:             0.3.0
    standard_name:     surface_air_pressure
    units:             Pa>
```

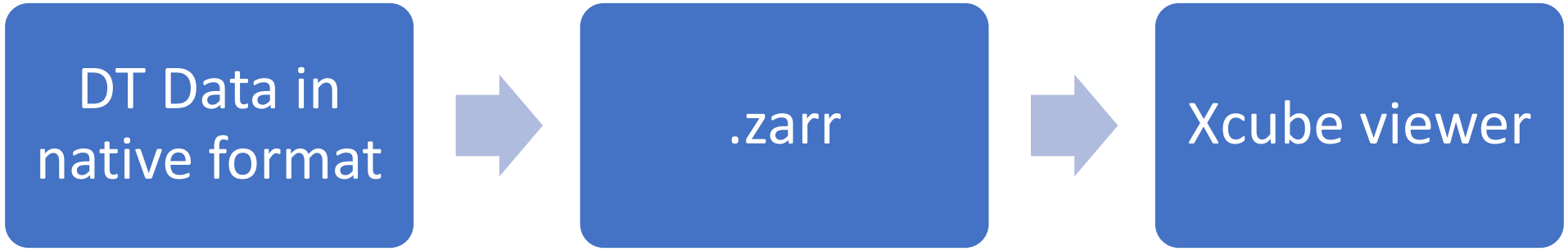
Plot map of surface pressure over Africa.

```
lon = africa_dt['lon']
lat = africa_dt['lat']
temperature = africa_dt[0] / 100    # Conversion to hectoPascals

plt.figure(figsize=(10, 6))
plt.pcolormesh(lon, lat, temperature, cmap='viridis')
plt.colorbar(label='Surface Pressure (hPa)')
plt.title('Surface pressure Map')
plt.xlabel('Longitude')
plt.ylabel('Latitude')
```





Funded by the European Union Destination Earth IMPLEMENTED BY EUMETSAT esa ECMWF

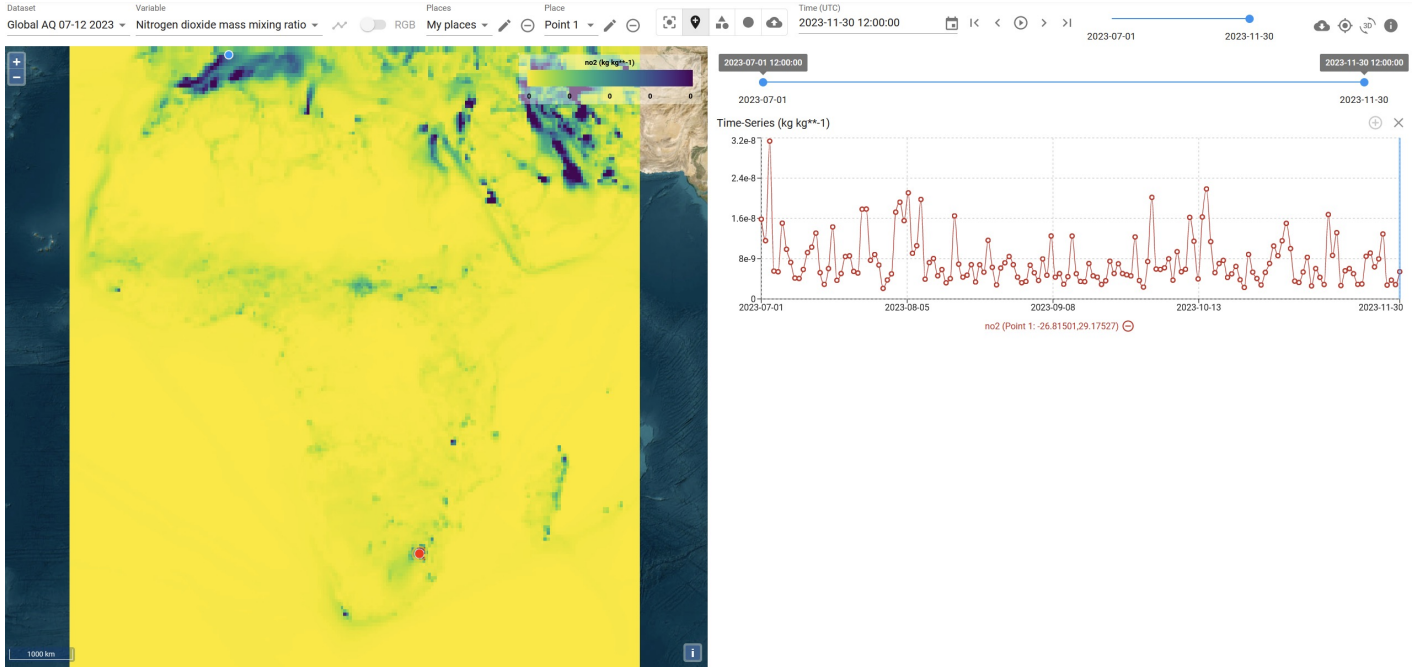


Funded by the European Union
Destination Earth
IMPLEMENTED BY EUMETSAT esa ECMWF

```
africa_bbox = [-20,
               -40,
               60,
               40]

africa_aq = xr.open_zarr("data_zar/global_aq_2023.zarr").sel(longitude=slice(africa_bbox[0],
                                                                             africa_bbox[2]),
                                                            latitude=slice(africa_bbox[3],
                                                                           africa_bbox[1]),
                                                            time=slice('2023-07-01', '2023-11-30')
                                                            )
```

```
from xcube.webapi.viewer import Viewer

viewer = Viewer(server_config={
    "Styles": [
        {
            "Identifier": "no2_legend",
            "ColorMappings": {
                "no2": {
                    "ValueRange": [0, 0.000000008],
                    "ColorBar": "viridis_r"
                },
                'go3': {
                    "ValueRange": [0, 0.00000007],
                    "ColorBar": "viridis"
                },
            }
        },
    ]
})
```

```
viewer.add_dataset(africa_aq,
                   style="no2_legend")

viewer.info()
```



Funded by the European Union
Destination Earth
IMPLEMENTED BY EUMETSAT esa ECMWF

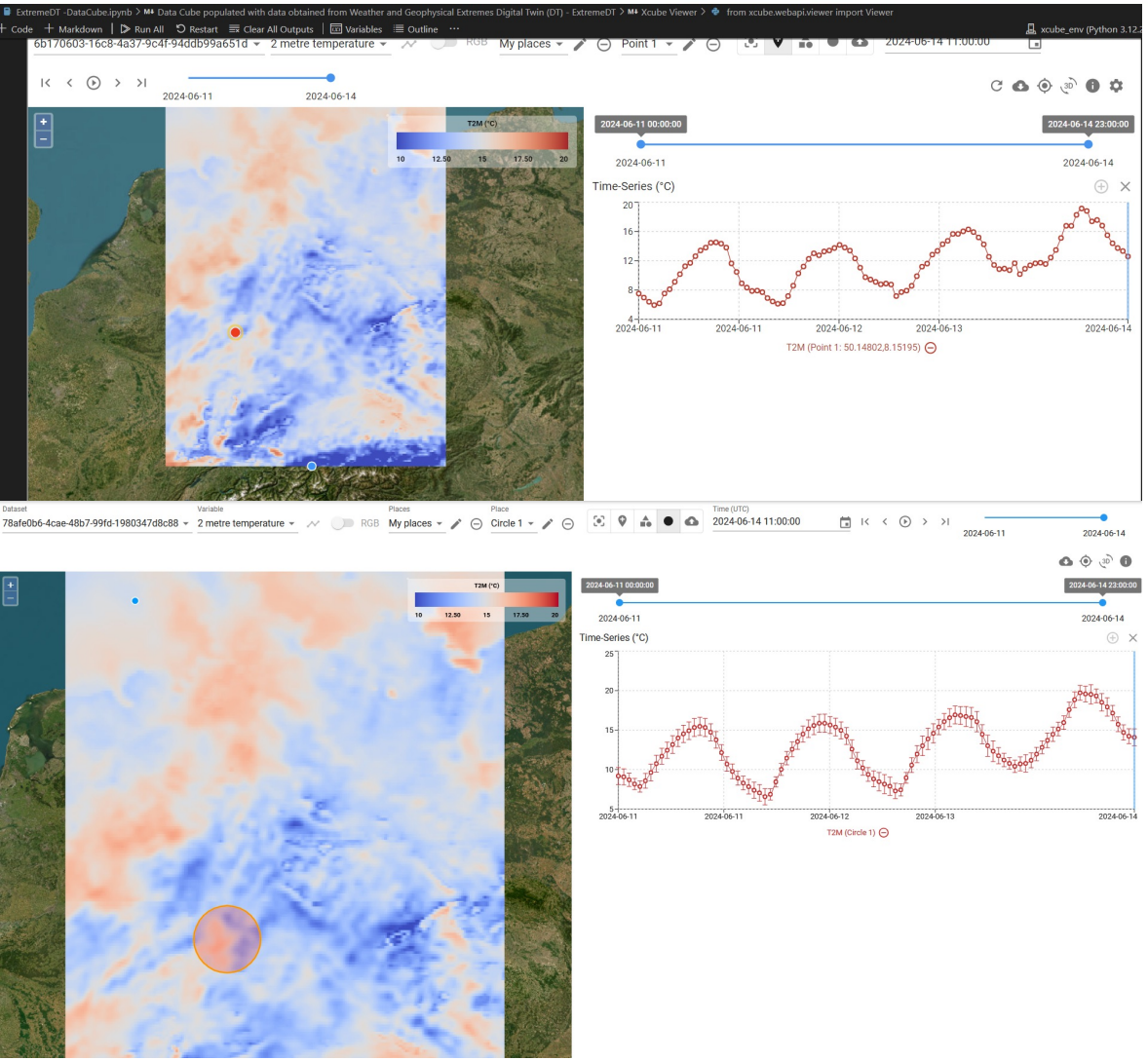
```python
from xcube.webapi.viewer import Viewer

germany_bbox = [5.866,    # West
                47.270,   # South
                15.042,   # East
                55.099]   # North

germany_dt = xr.open_zarr(url).sel(lon=slice(germany_bbox[0],
                                             germany_bbox[2]),
                                   lat=slice(germany_bbox[3],
                                             germany_bbox[1]),
                                             )
# Convert temperature from Kelvin to Celsius
germany_dt['T2M'] -= 273.15    # Conversion to Celsius degrees
germany_dt['D2M'] -= 273.15    # Conversion to Celsius degrees
# Convert pressure from Pascals to hectoPascals
germany_dt['SP'] /= 100        # Conversion to hectoPascals

# Set attribute units
germany_dt['T2M'].attrs['units'] = '°C'
germany_dt['D2M'].attrs['units'] = '°C'
germany_dt['SP'].attrs['units'] = 'hPa'
```

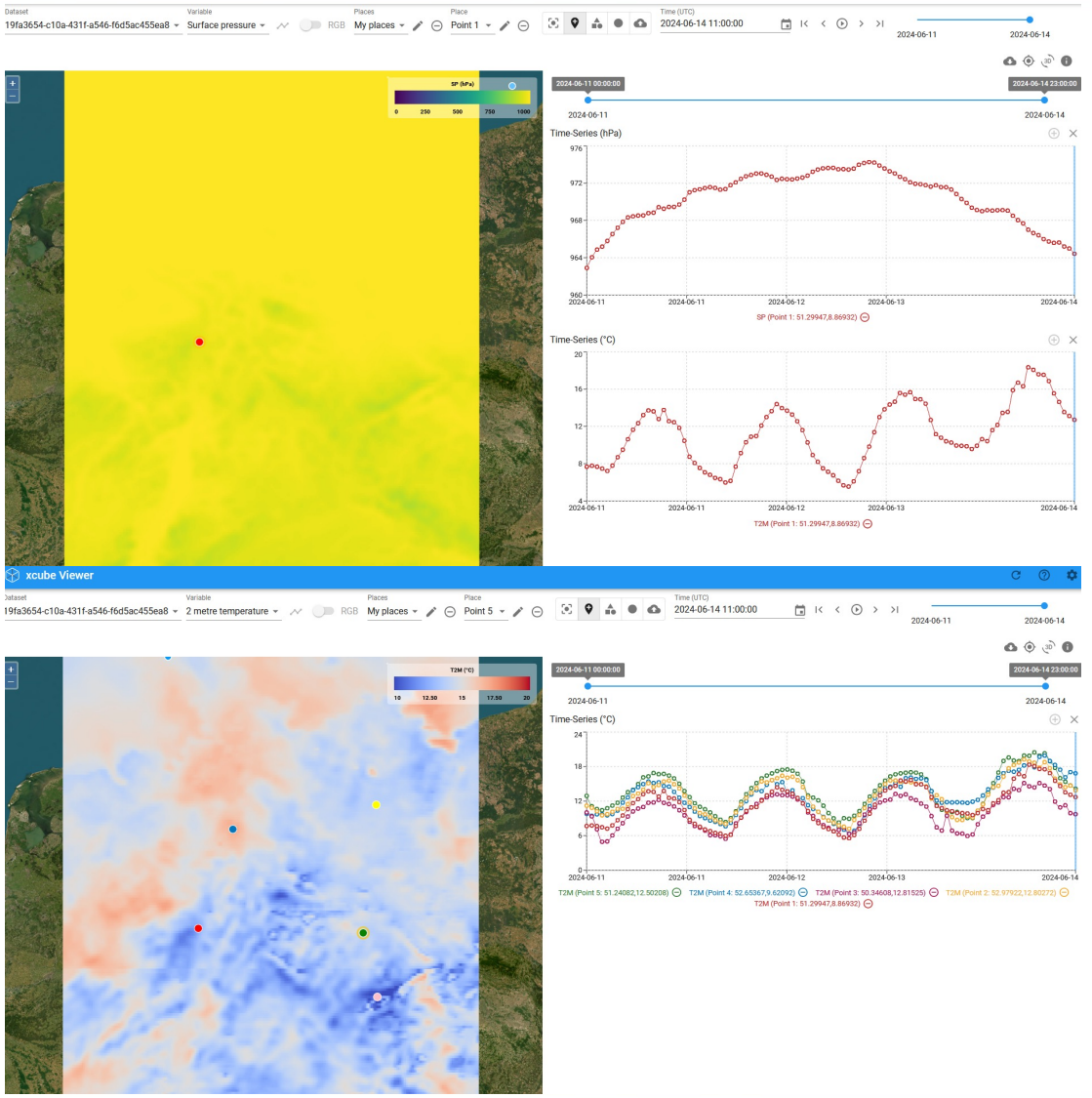
```python
viewer = Viewer(server_config={
    "Styles": [
        {
            "Identifier": "dt_legend",
            "ColorMappings": {
                "T2M": {
                    "ValueRange": [10, 20],
                    "ColorBar": "coolwarm",
                },
                "D2M": {
                    "ValueRange": [0, 10],
                    "ColorBar": "coolwarm",
                },
                'SP': {
                    "ValueRange": [0, 1000],
                    "ColorBar": "viridis"
                },
            }
        },
    ]
})
```

```python
viewer.show()
viewer.info()
```

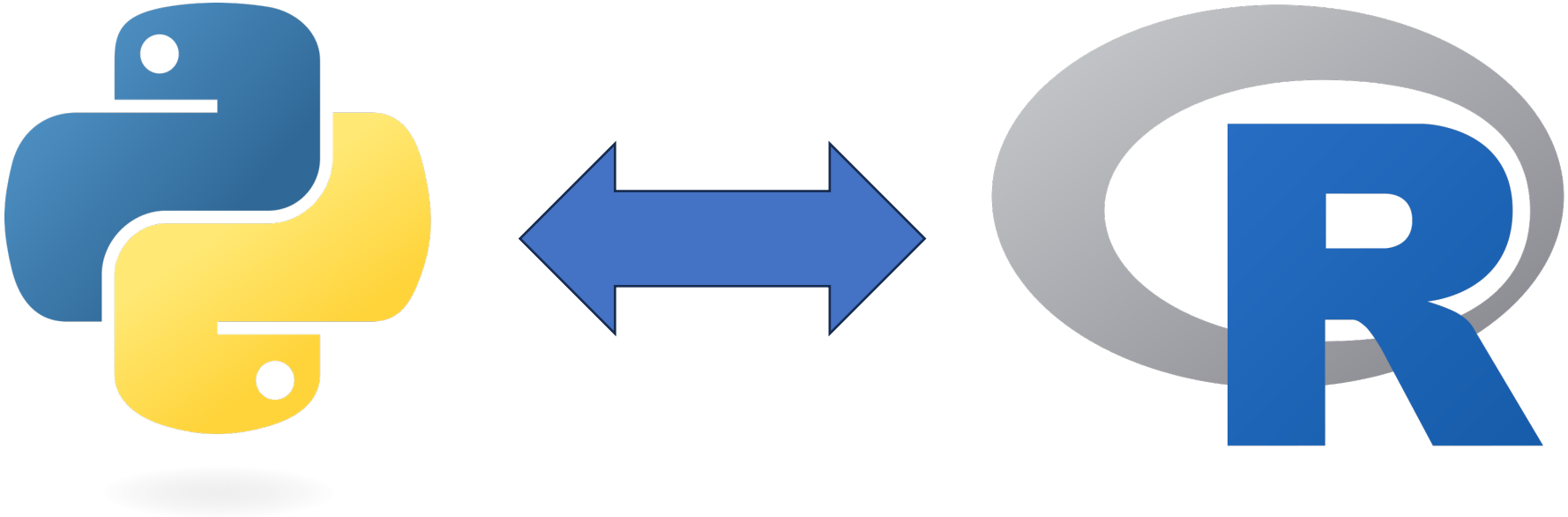
```
Server: http://localhost:8009
Viewer: http://localhost:8009/viewer/?serverUrl=http://localhost:8009
```


Funded by the European Union Destination Earth IMPLEMENTED BY EUMETSAT esa ECMWF

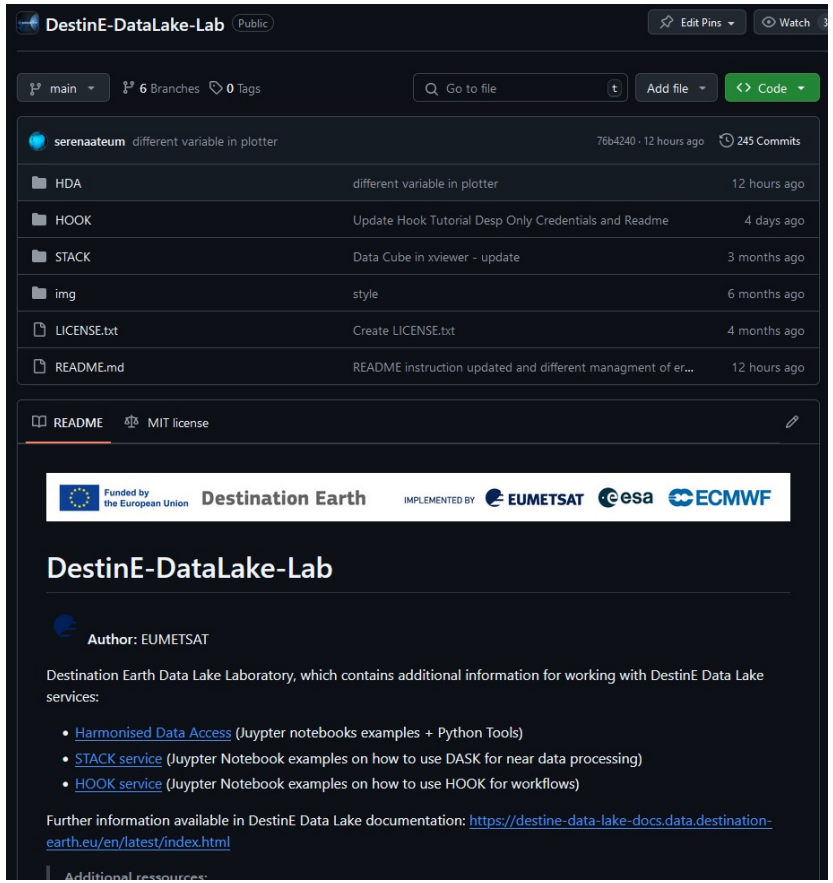


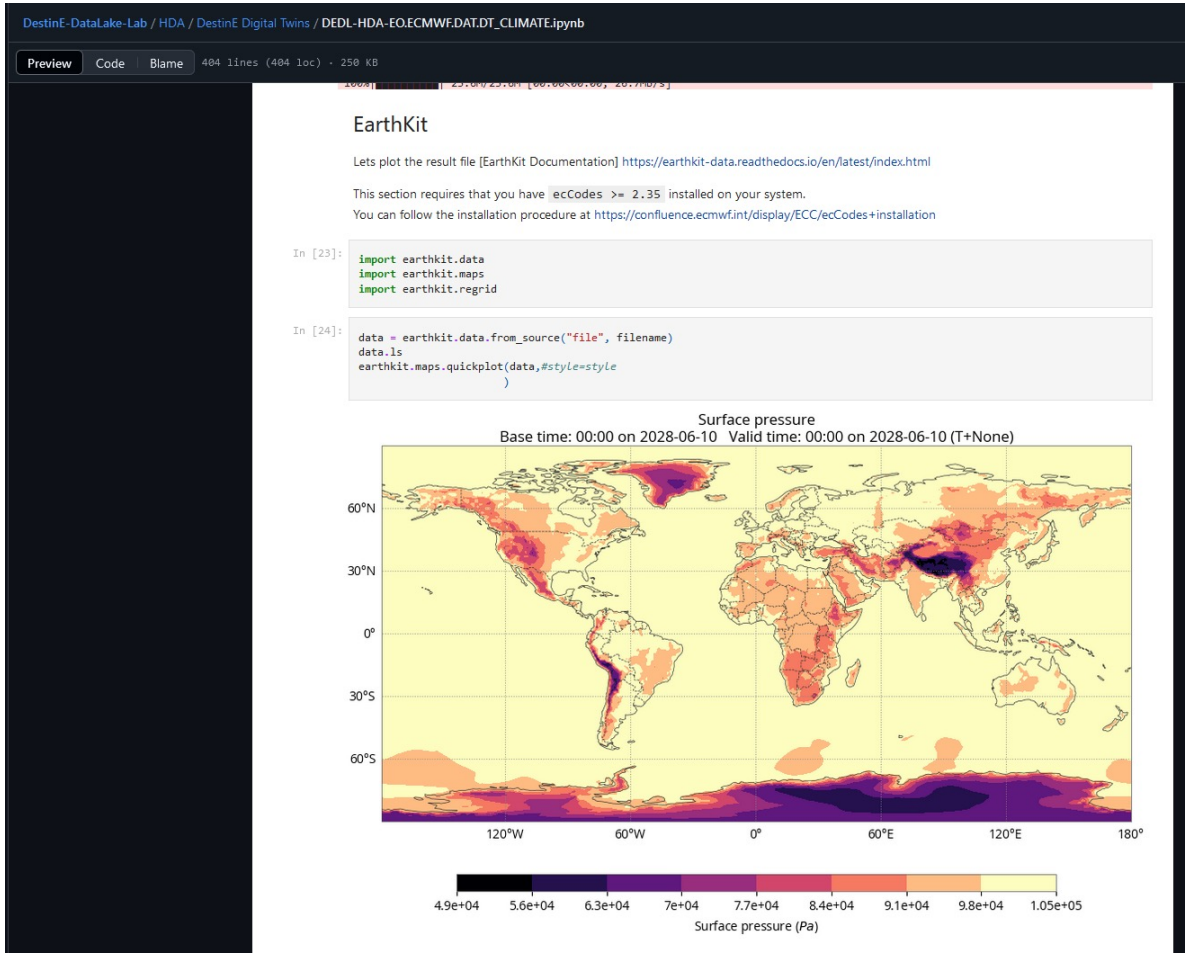


Funded by the European Union
Destination Earth
IMPLEMENTED BY EUMETSAT esa ECMWF



https://github.com/destination-earth/DestinE-DataLake-Lab





Destination Earth

Thank you!

pgrzybowski@cloudferro.com

https://github.com/destination-earth/DestinE-DataLake-Lab

Destination Earth Funded by the European Union Implemented by ECMWF esa EUMETSAT