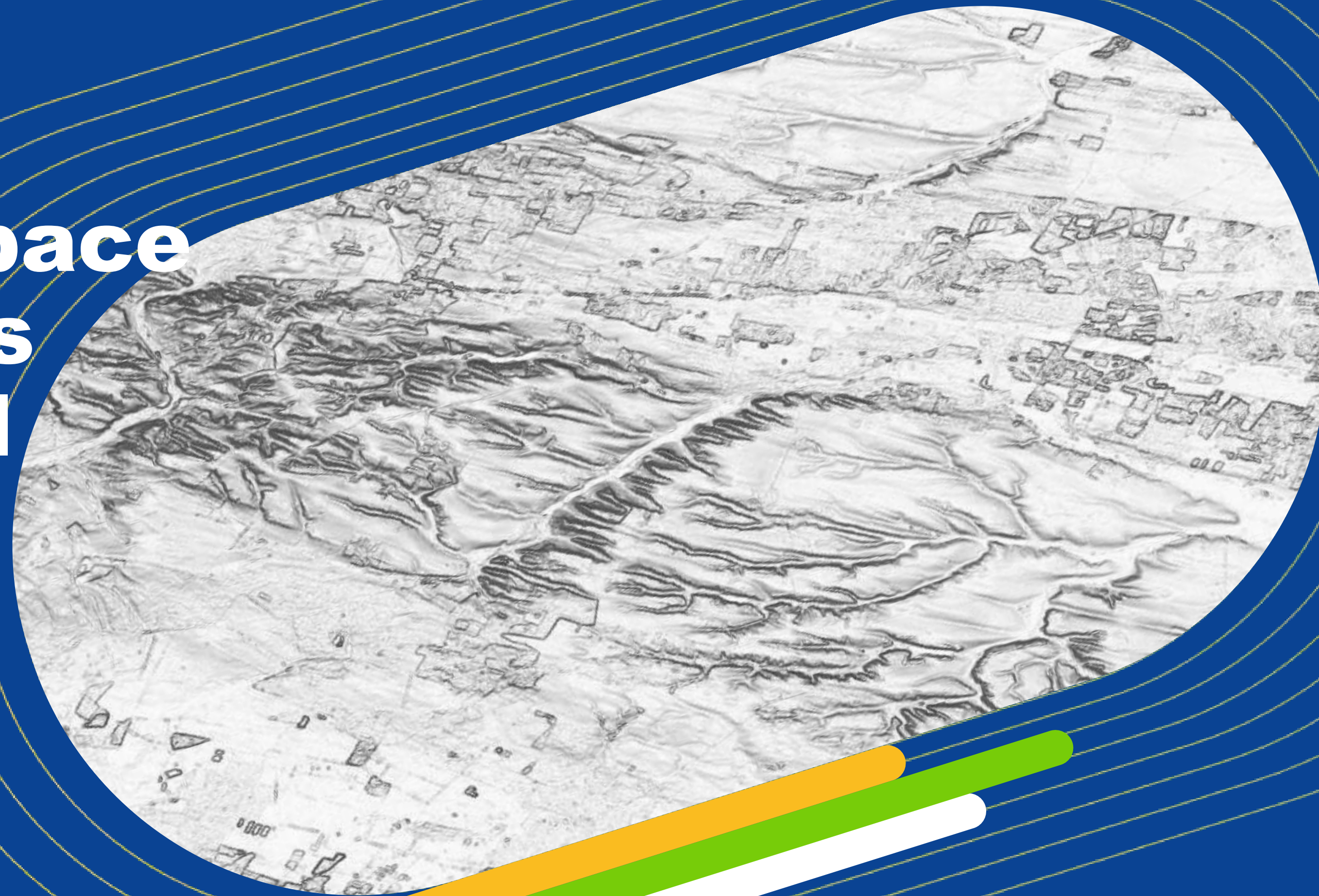




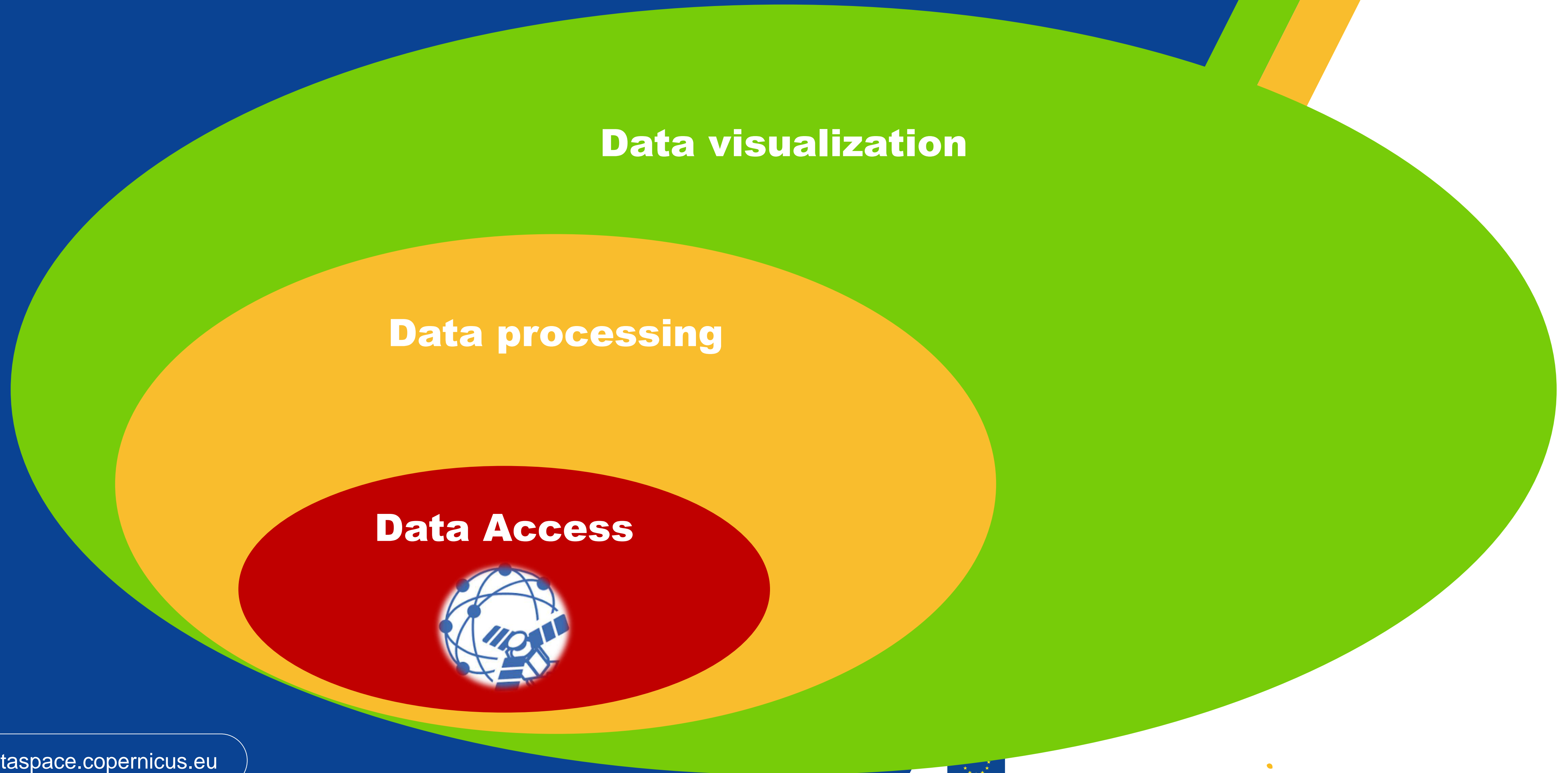
Copernicus Data Space Ecosystem provides free tools for global soil erosion risk estimation

ESA Symposium on Earth
Observation for Soil Protection
and Restoration

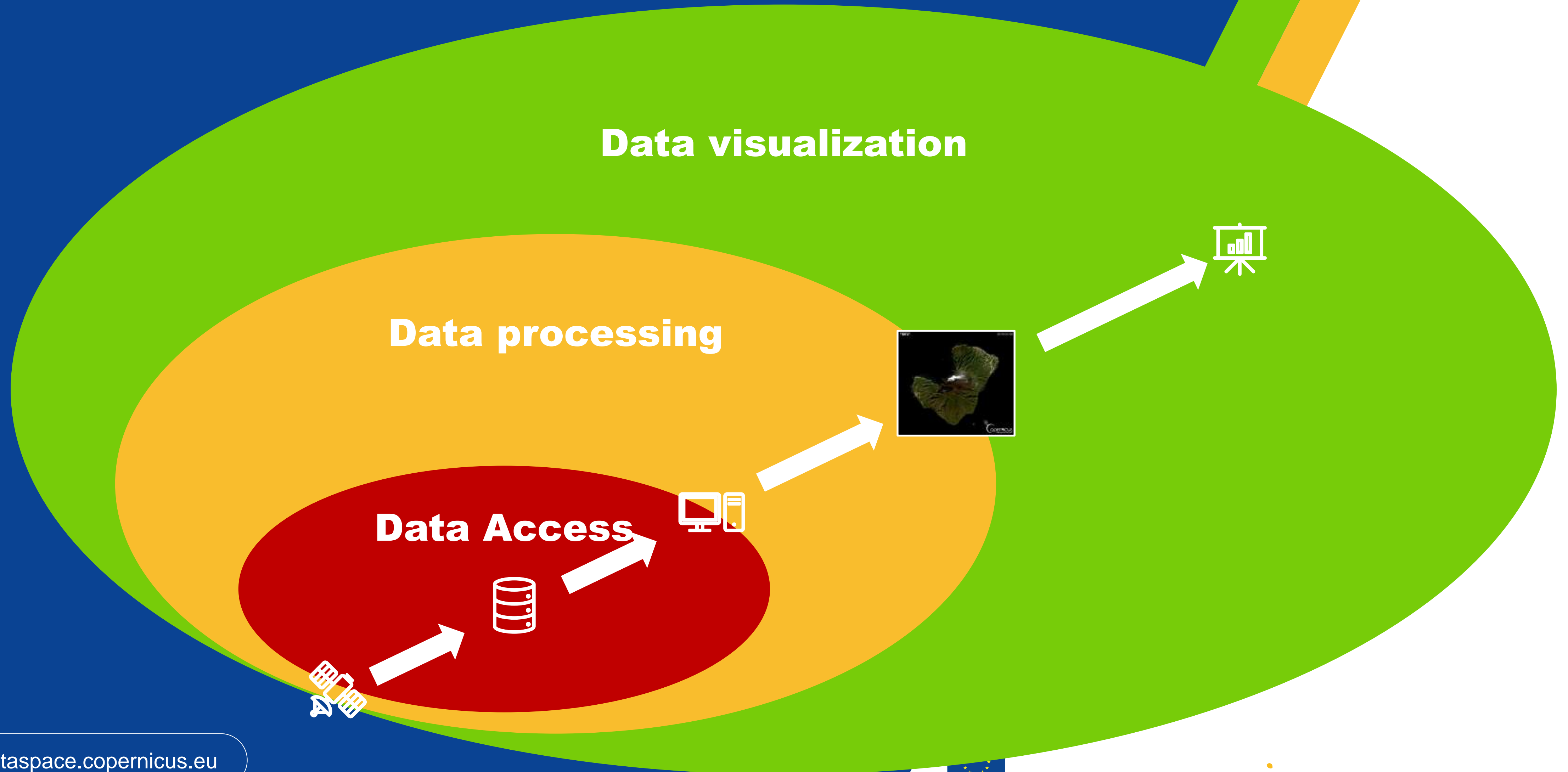
6-7.03.2024, Frascati, Italy



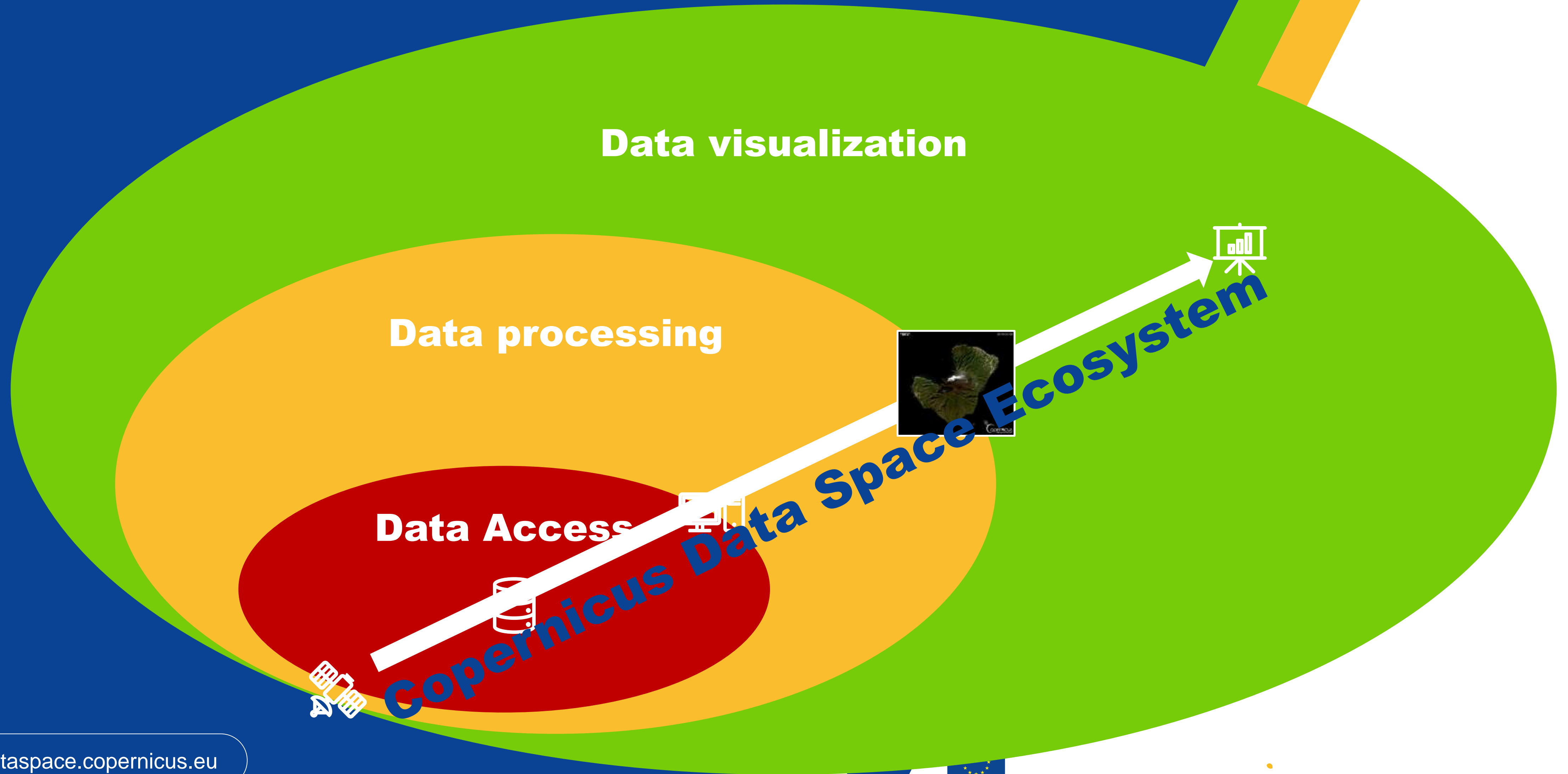
What is Copernicus Data Space Ecosystem?



What is Copernicus Data Space Ecosystem?



What is Copernicus Data Space Ecosystem?



Main features of Copernicus Data Space Ecosystem

All Copernicus Sentinel Data

- Copernicus Contributing Missions
- Federated datasets
- Complementary data

API-based open data distribution service

- Platform for building new software solutions
- Access to imagery and data products

Cloud computing with adjustable capacity and performance

- CloudFerro Cloud
- Open Telekom Cloud
- Third-party resources

Copernicus Browser

- Interactive visualization
- Custom processing and analysis
- Time series
- Image download

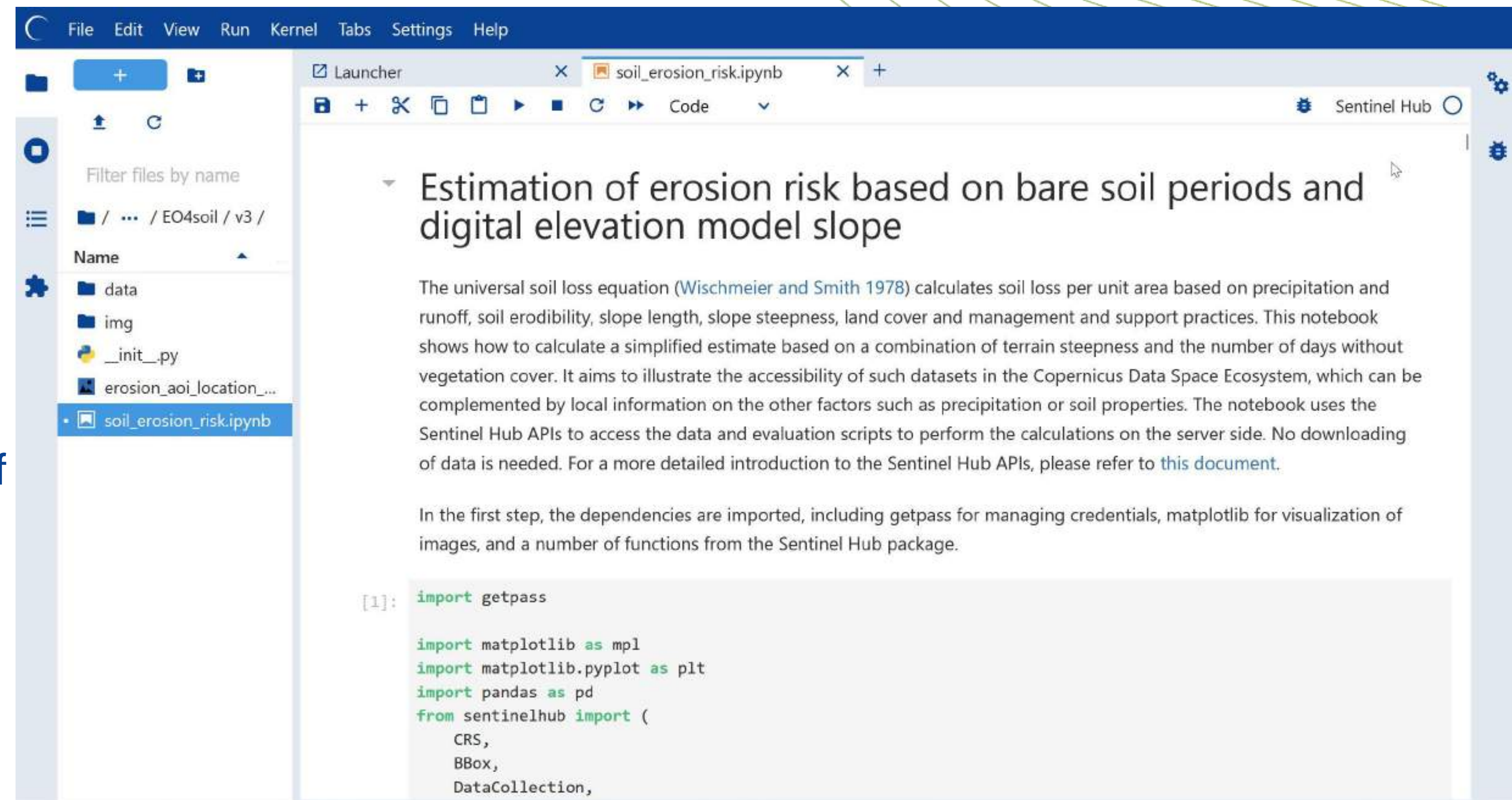
Onboard code lab and resources

- OpenEO
- Requests Builder
- Jupyter Notebooks

Open ecosystem

Use case example – global erosion risk estimation in JupyterLab

- JupyterLab has all Sentinel Data Collections, CDSE API-s and a set of EO Code libraries → no need to set up anything locally - https://github.com/eu-cdse/notebook-samples/blob/main/sentinelhub/soil_erosion_risk.ipynb
- jupyterhub.dataspace.copernicus.eu
- Start notebook, import dependencies
- Define study area (Somogybabod, Hungary)
- Calculate number of bare soil days from Sentinel-2
- Collect terrain slopes – pre-calculated with GDAL on CREODIAS and ingested with BYOC API
- Estimate erosion risk as product of slope and ratio of bare soil days
- Calculate time series of local estimated erosion risk using Statistical API



File Edit View Run Kernel Tabs Settings Help

Launcher x soil_erosion_risk.ipynb x + Sentinel Hub

Estimation of erosion risk based on bare soil periods and digital elevation model slope

The universal soil loss equation (Wischmeier and Smith 1978) calculates soil loss per unit area based on precipitation and runoff, soil erodibility, slope length, slope steepness, land cover and management and support practices. This notebook shows how to calculate a simplified estimate based on a combination of terrain steepness and the number of days without vegetation cover. It aims to illustrate the accessibility of such datasets in the Copernicus Data Space Ecosystem, which can be complemented by local information on the other factors such as precipitation or soil properties. The notebook uses the Sentinel Hub APIs to access the data and evaluation scripts to perform the calculations on the server side. No downloading of data is needed. For a more detailed introduction to the Sentinel Hub APIs, please refer to [this document](#).

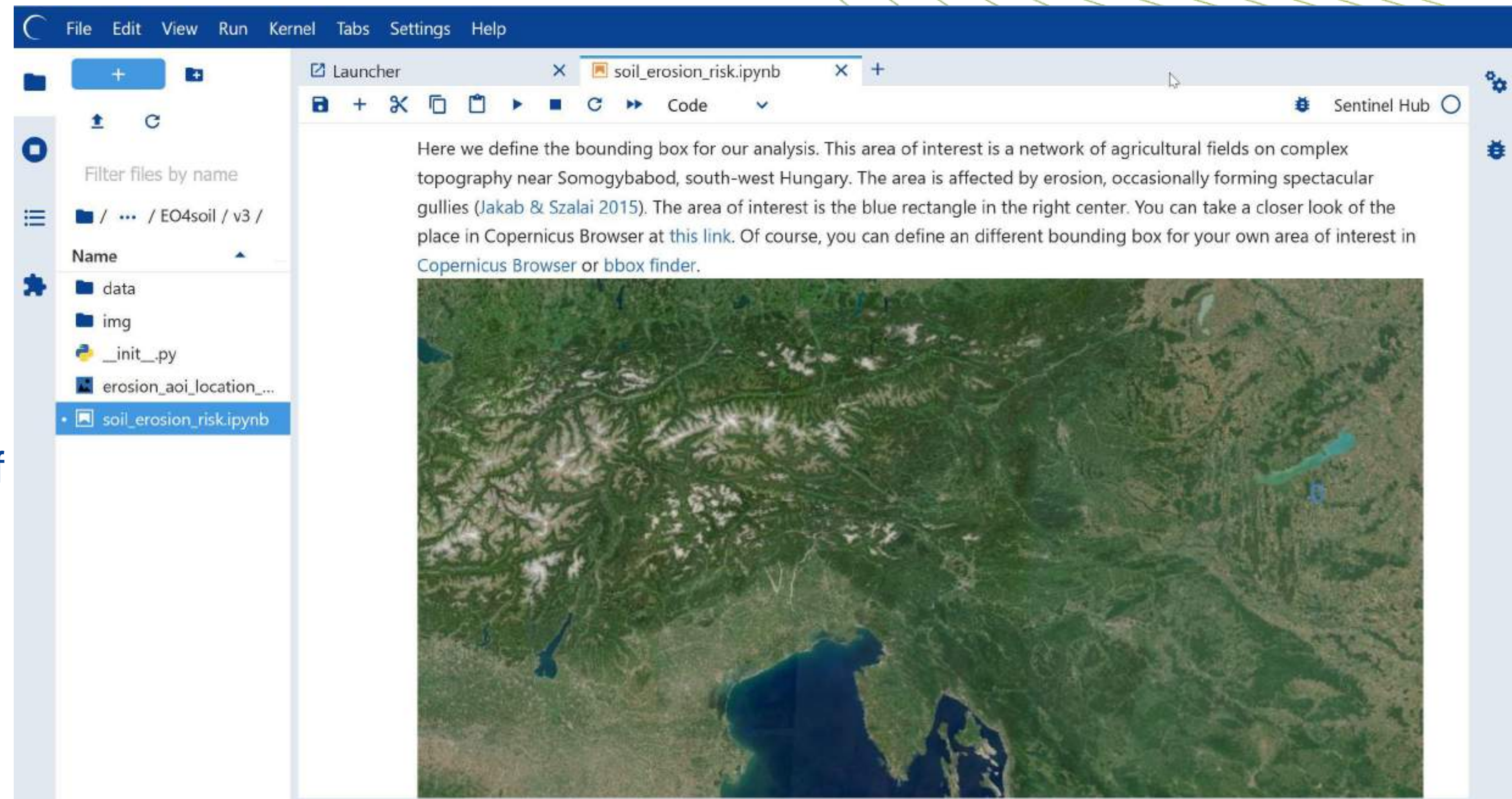
In the first step, the dependencies are imported, including getpass for managing credentials, matplotlib for visualization of images, and a number of functions from the Sentinel Hub package.

```
[1]: import getpass

import matplotlib as mpl
import matplotlib.pyplot as plt
import pandas as pd
from sentinelhub import (
    CRS,
    BBox,
    DataCollection,
```

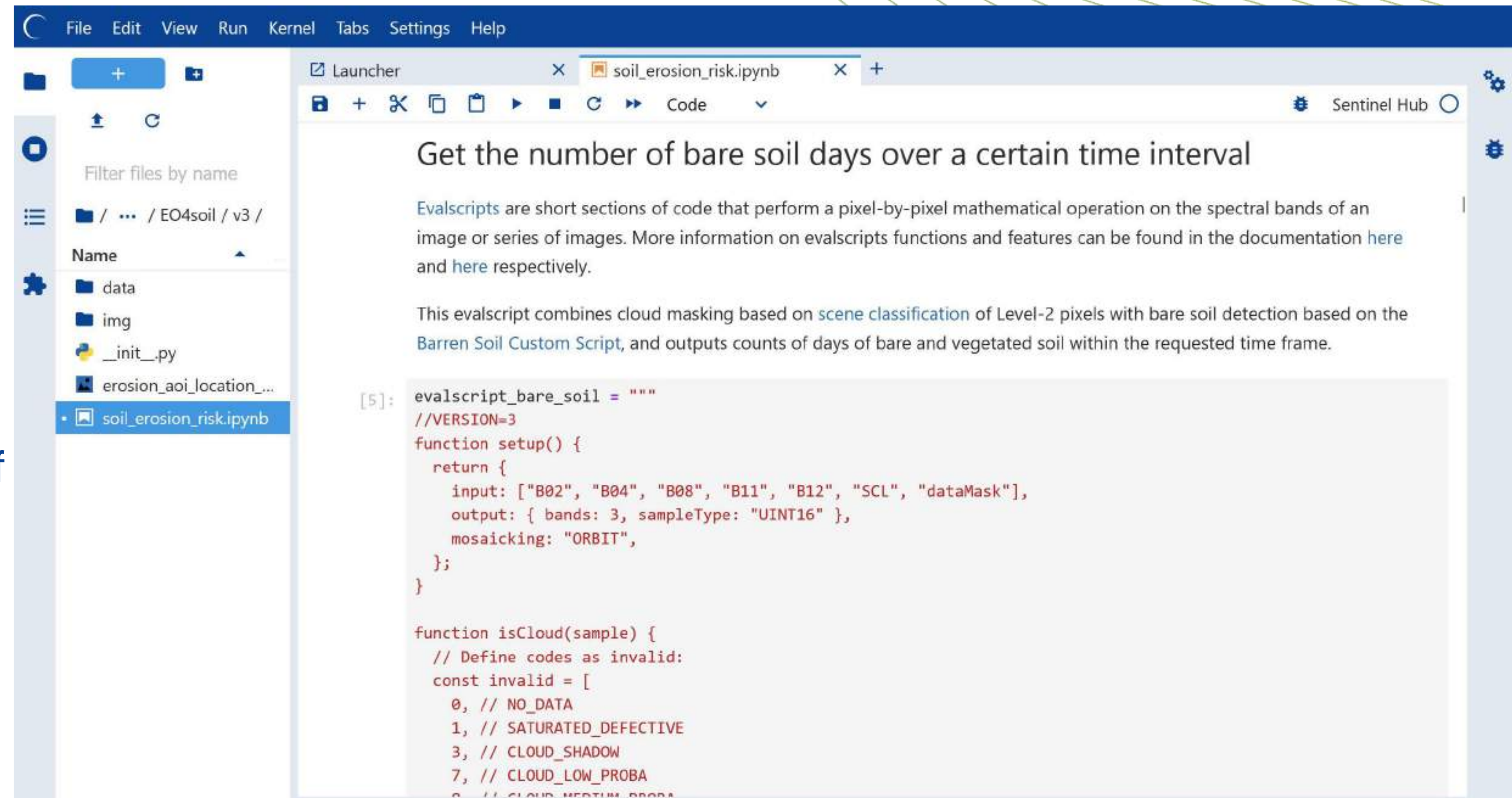
Use case example – global erosion risk estimation in JupyterLab

- JupyterLab has all Sentinel Data Collections, CDSE API-s and a set of EO Code libraries → no need to set up anything locally
- jupyterhub.dataspace.copernicus.eu
- Start notebook, import dependencies
- Define study area (Somogybabod, Hungary)
- Calculate number of bare soil days from Sentinel-2
- Collect terrain slopes – pre-calculated with GDAL on CREODIAS and ingested with BYOC API
- Estimate erosion risk as product of slope and ratio of bare soil days
- Calculate time series of local estimated erosion risk using Statistical API



Use case example – global erosion risk estimation in JupyterLab

- JupyterLab has all Sentinel Data Collections, CDSE API-s and a set of EO Code libraries → no need to set up anything locally
- jupyterhub.dataspace.copernicus.eu
- Start notebook, import dependencies
- Define study area (Somogybabod, Hungary)
- Calculate number of bare soil days from Sentinel-2
- Collect terrain slopes – pre-calculated with GDAL on CREODIAS and ingested with BYOC API
- Estimate erosion risk as product of slope and ratio of bare soil days
- Calculate time series of local estimated erosion risk using Statistical API



The screenshot shows a JupyterLab notebook titled "soil_erosion_risk.ipynb". The left sidebar displays a file browser with the following structure:

- Filter files by name
- / ... / EO4soil / v3 /
- Name
- data
- img
- __init__.py
- erosion_aoi_location_...
- soil_erosion_risk.ipynb

The main notebook area contains the following text and code:

Get the number of bare soil days over a certain time interval

Evals-scripts are short sections of code that perform a pixel-by-pixel mathematical operation on the spectral bands of an image or series of images. More information on evals-scripts functions and features can be found in the documentation [here](#) and [here](#) respectively.

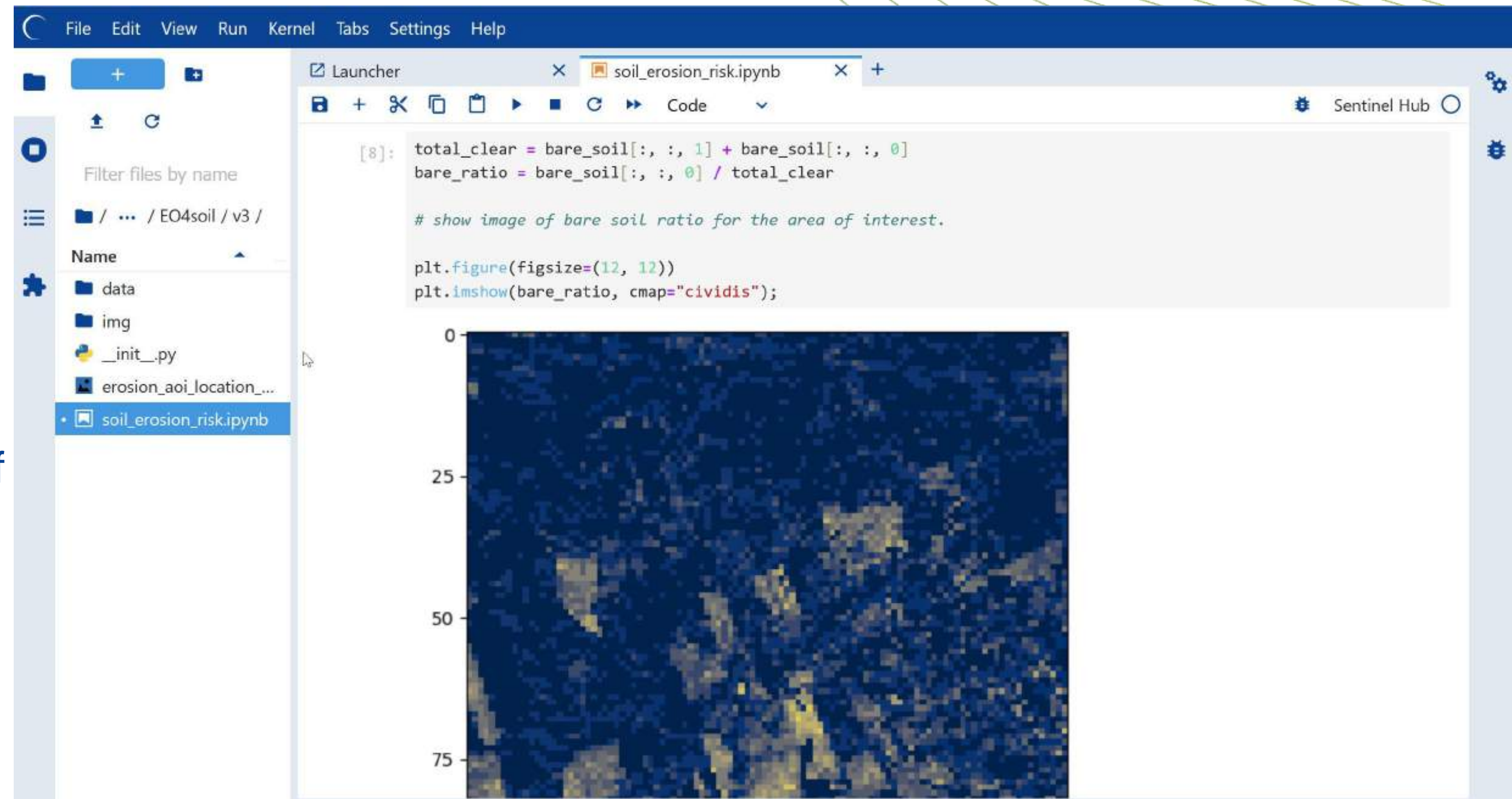
This evals-script combines cloud masking based on [scene classification](#) of Level-2 pixels with bare soil detection based on the [Barren Soil Custom Script](#), and outputs counts of days of bare and vegetated soil within the requested time frame.

```
[5]: evalscript_bare_soil = """
//VERSION=3
function setup() {
  return {
    input: ["B02", "B04", "B08", "B11", "B12", "SCL", "dataMask"],
    output: { bands: 3, sampleType: "UINT16" },
    mosaicking: "ORBIT",
  };
}

function isCloud(sample) {
  // Define codes as invalid:
  const invalid = [
    0, // NO_DATA
    1, // SATURATED_DEFECTIVE
    3, // CLOUD_SHADOW
    7, // CLOUD_LOW_PROBA
    8, // CLOUD_MEDIUM_PROBA
  ];
}
```


Use case example – global erosion risk estimation in JupyterLab

- JupyterLab has all Sentinel Data Collections, CDSE API-s and a set of EO Code libraries → no need to set up anything locally
- jupyterhub.dataspace.copernicus.eu
- Start notebook, import dependencies
- Define study area (Somogybabod, Hungary)
- Calculate number of bare soil days from Sentinel-2
- Collect terrain slopes – pre-calculated with GDAL on CREODIAS and ingested with BYOC API
- Estimate erosion risk as product of slope and ratio of bare soil days
- Calculate time series of local estimated erosion risk using Statistical API



Use case example – global erosion risk estimation in JupyterLab

- JupyterLab has all Sentinel Data Collections, CDSE API-s and a set of EO Code libraries → no need to set up anything locally
- jupyterhub.dataspace.copernicus.eu
- Start notebook, import dependencies
- Define study area (Somogybabod, Hungary)
- Calculate number of bare soil days from Sentinel-2
- Collect terrain slopes – pre-calculated with GDAL on CREODIAS and ingested with BYOC API
- Estimate erosion risk as product of slope and ratio of bare soil days
- Calculate time series of local estimated erosion risk using Statistical API

The screenshot shows a JupyterLab environment with a file browser on the left and a notebook editor on the right. The file browser shows a directory structure with files like 'data', 'img', and 'soil_erosion_risk.ipynb'. The notebook editor displays the following content:

Get terrain slope for the same area

Copernicus Data Space Ecosystem allows users to upload their own data in a Cloud Optimized Geotiff (COG) format, ingesting it with the Bring Your Own COG ("BYOC") API. Once a dataset is ingested, it can be made available privately or publicly, and other Sentinel Hub API requests can access the data collection via its BYOC collection ID ([example here](#))

Terrain slopes were calculated using the GDALDEM `slope` function, running on the full global 30 meter Copernicus DEM dataset. The following command was used:

```
`gdaldem slope input_dem.tif slope-byoc.tif -of COG -co COMPRESS=DEFLATE -co BLOCKSIZE=1024 -co RESAMPLING=NEAREST -co OVERVIEWS=IGNORE_EXISTING`
```

This dataset of slopes is made available in Copernicus Data Space Ecosystem as a public BYOC collection, making streamlined querying and processing possible.

Here we also define a simple evalscript that returns slope angle values directly from the dataset

```
[9]: evalscript = """  
//VERSION=3  
function setup() {  
  return {  
    input: ["slope", "dataMask"],  
    output: { bands: 1, sampleType: "FLOAT32" },  
  };  
}
```

Use case example – global erosion risk estimation in JupyterLab

- JupyterLab has all Sentinel Data Collections, CDSE API-s and a set of EO Code libraries → no need to set up anything locally
- jupyterhub.dataspace.copernicus.eu
- Start notebook, import dependencies
- Define study area (Somogybabod, Hungary)
- Calculate number of bare soil days from Sentinel-2
- Collect terrain slopes – pre-calculated with GDAL on CREODIAS and ingested with BYOC API
- Estimate erosion risk as product of slope and ratio of bare soil days
- Calculate time series of local estimated erosion risk using Statistical API

The screenshot displays a JupyterLab notebook titled 'soil_erosion_risk.ipynb'. The interface includes a file browser on the left showing a directory structure with files like 'data', 'img', and 'erosion_aoi_location...'. The main notebook area contains the following code and text:

```
responses=[SentinelHubRequest.output_response("default", MimeType.TIFF)],
bbox=bbox,
resolution=resolution,
config=config,
data_folder="./data",
)
```

We collect raster slope data of the area of interest and timeframe from the Process API request into a the variable `slope`

```
[11]: slope = slope_request.get_data(save_data=True)[0]
```

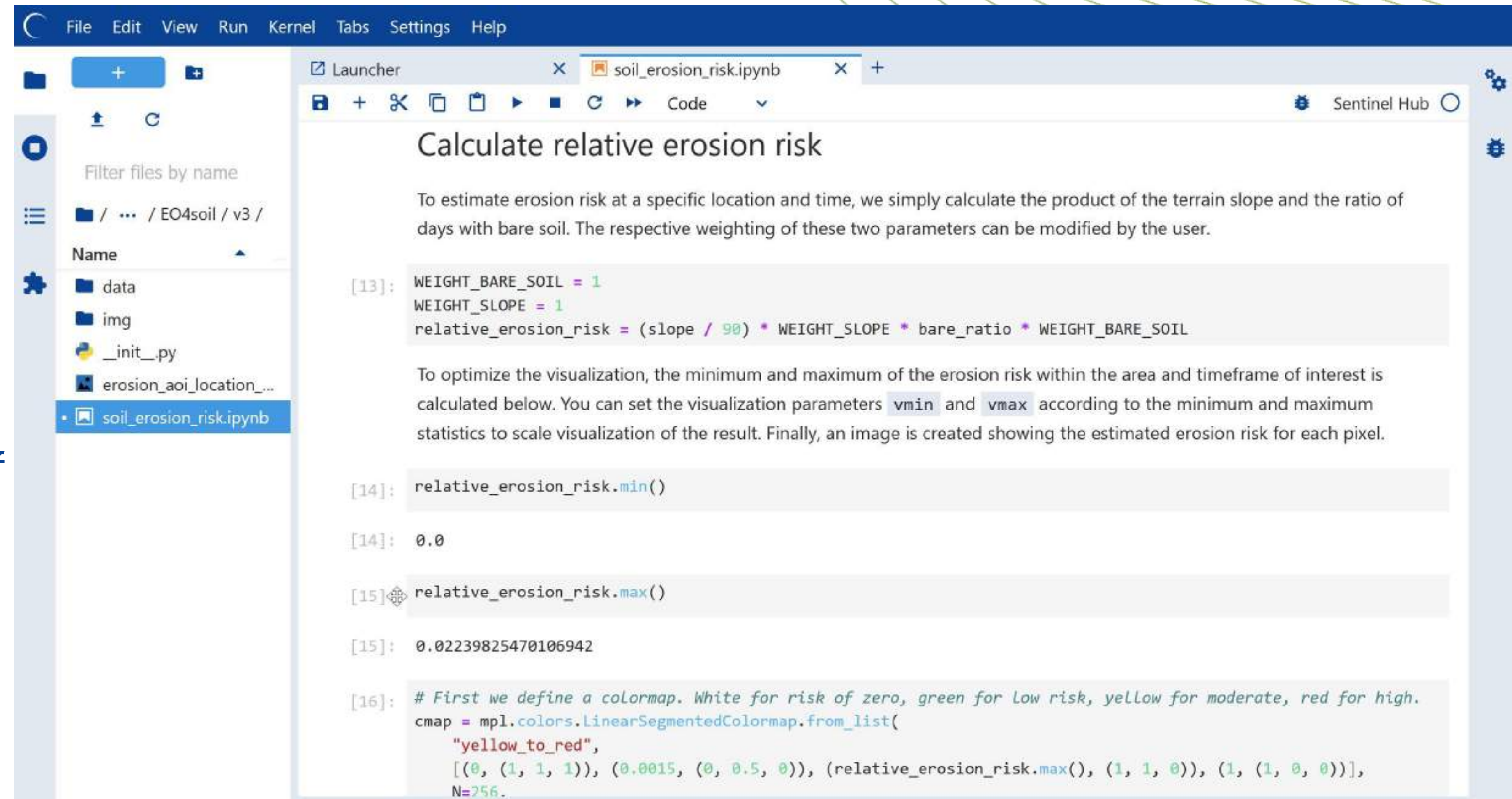
We visualize an image of the `slope` variable using pyplot again.

```
[12]: plt.figure(figsize=(12, 12))
plt.imshow(slope, cmap="cividis");
```

Below the code, a visualization of the slope data is shown as a heatmap. The y-axis is labeled with '0' at the top and '25' at the bottom. The plot shows a complex, textured pattern of colors ranging from dark blue to yellow, representing the slope values across the study area.

Use case example – global erosion risk estimation in JupyterLab

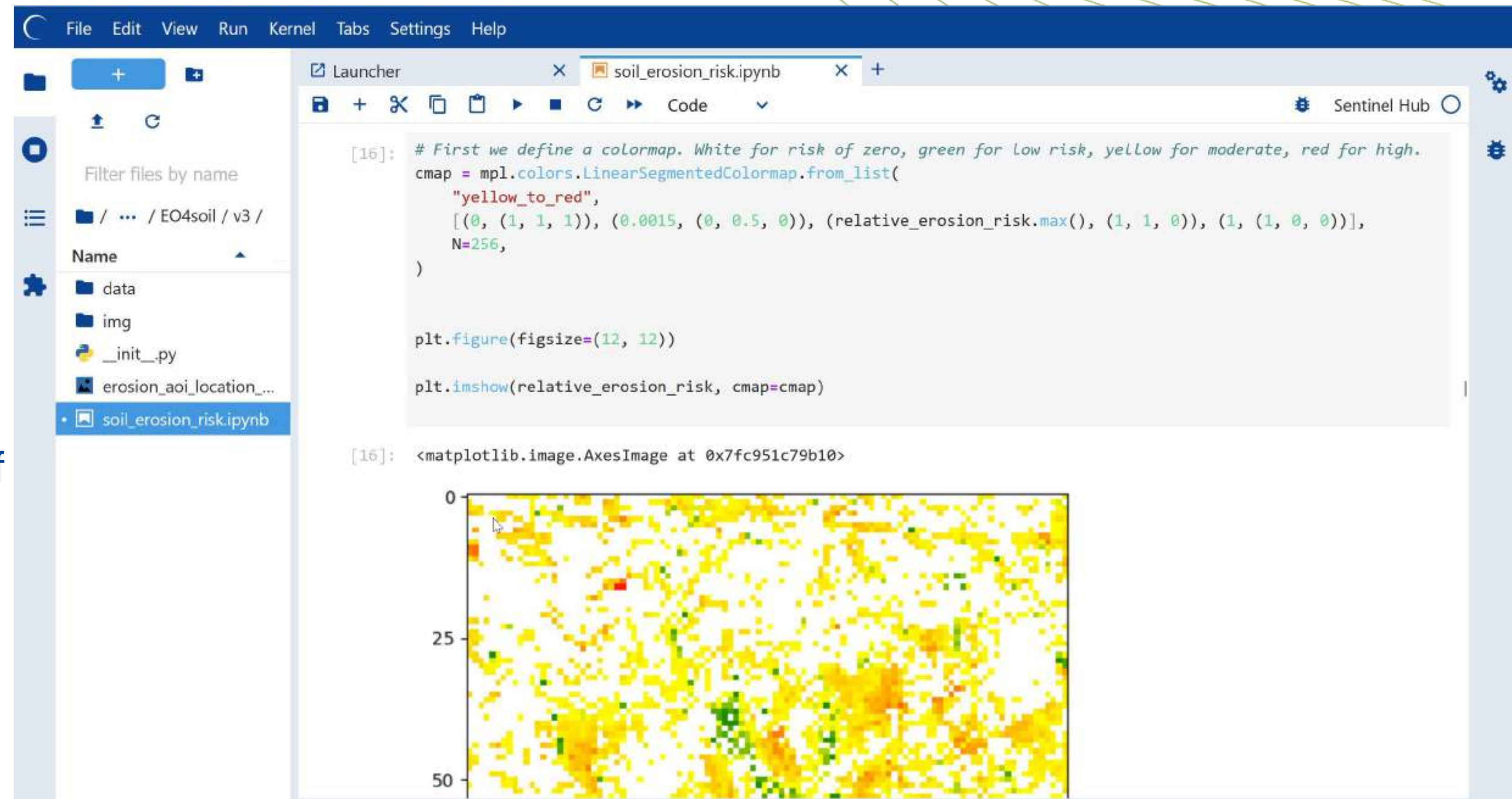
- JupyterLab has all Sentinel Data Collections, CDSE API-s and a set of EO Code libraries → no need to set up anything locally
- jupyterhub.dataspace.copernicus.eu
- Start notebook, import dependencies
- Define study area (Somogybabod, Hungary)
- Calculate number of bare soil days from Sentinel-2
- Collect terrain slopes – pre-calculated with GDAL on CREODIAS and ingested with BYOC API
- Estimate erosion risk as product of slope and ratio of bare soil days
- Calculate time series of local estimated erosion risk using Statistical API



```
File Edit View Run Kernel Tabs Settings Help
Launcher x soil_erosion_risk.ipynb x +
Code Sentinel Hub
Calculate relative erosion risk
To estimate erosion risk at a specific location and time, we simply calculate the product of the terrain slope and the ratio of days with bare soil. The respective weighting of these two parameters can be modified by the user.
[13]: WEIGHT_BARE_SOIL = 1
WEIGHT_SLOPE = 1
relative_erosion_risk = (slope / 90) * WEIGHT_SLOPE * bare_ratio * WEIGHT_BARE_SOIL
To optimize the visualization, the minimum and maximum of the erosion risk within the area and timeframe of interest is calculated below. You can set the visualization parameters vmin and vmax according to the minimum and maximum statistics to scale visualization of the result. Finally, an image is created showing the estimated erosion risk for each pixel.
[14]: relative_erosion_risk.min()
[14]: 0.0
[15]: relative_erosion_risk.max()
[15]: 0.02239825470106942
[16]: # First we define a colormap. White for risk of zero, green for low risk, yellow for moderate, red for high.
cmap = mpl.colors.LinearSegmentedColormap.from_list(
    "yellow_to_red",
    [(0, (1, 1, 1)), (0.0015, (0, 0.5, 0)), (relative_erosion_risk.max(), (1, 1, 0)), (1, (1, 0, 0))],
    N=256.
```

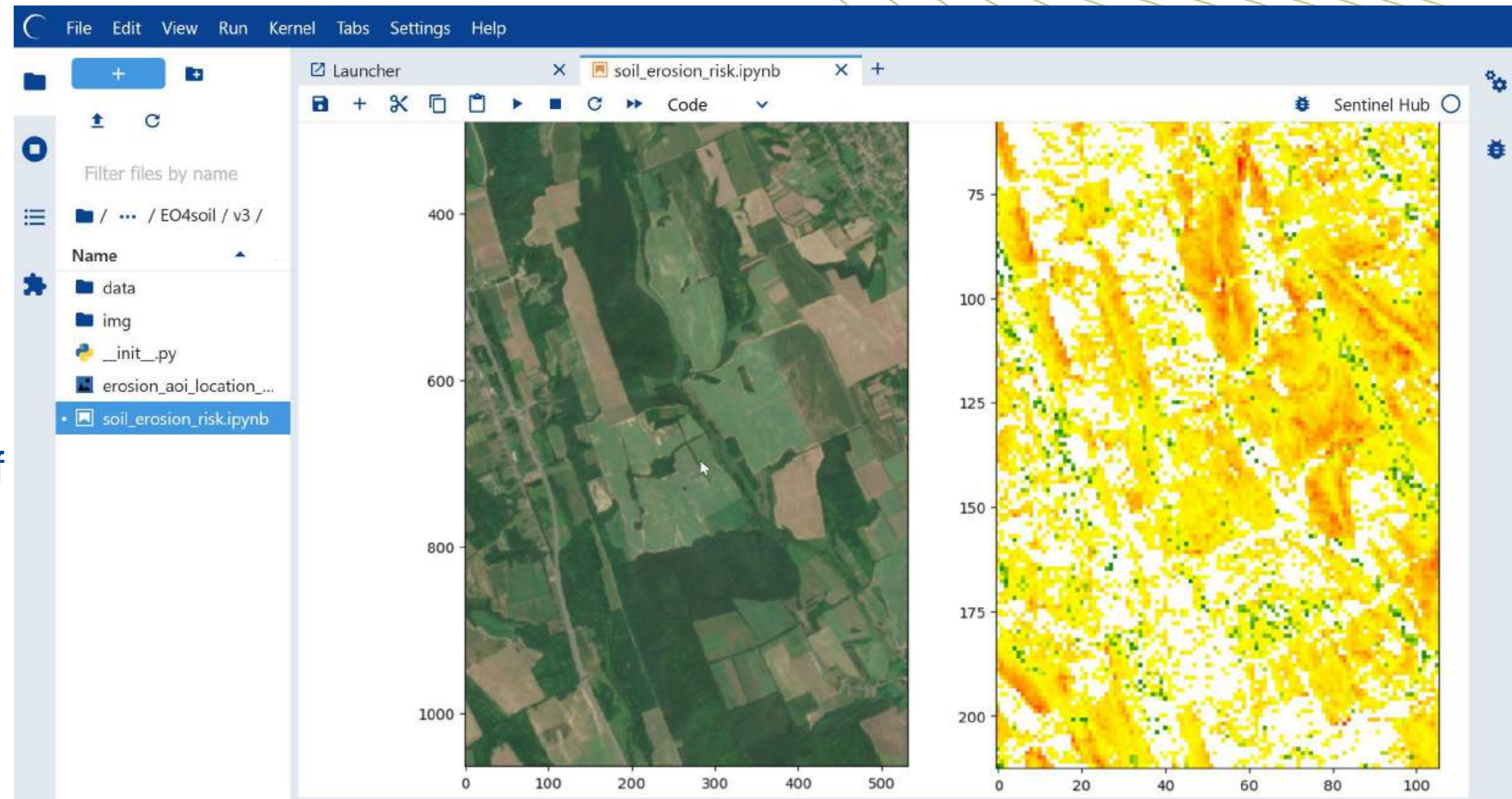
Use case example – global erosion risk estimation in JupyterLab

- JupyterLab has all Sentinel Data Collections, CDSE API-s and a set of EO Code libraries → no need to set up anything locally
- jupyterhub.dataspace.copernicus.eu
- Start notebook, import dependencies
- Define study area (Somogybabod, Hungary)
- Calculate number of bare soil days from Sentinel-2
- Collect terrain slopes – pre-calculated with GDAL on CREODIAS and ingested with BYOC API
- Estimate erosion risk as product of slope and ratio of bare soil days
- Calculate time series of local estimated erosion risk using Statistical API



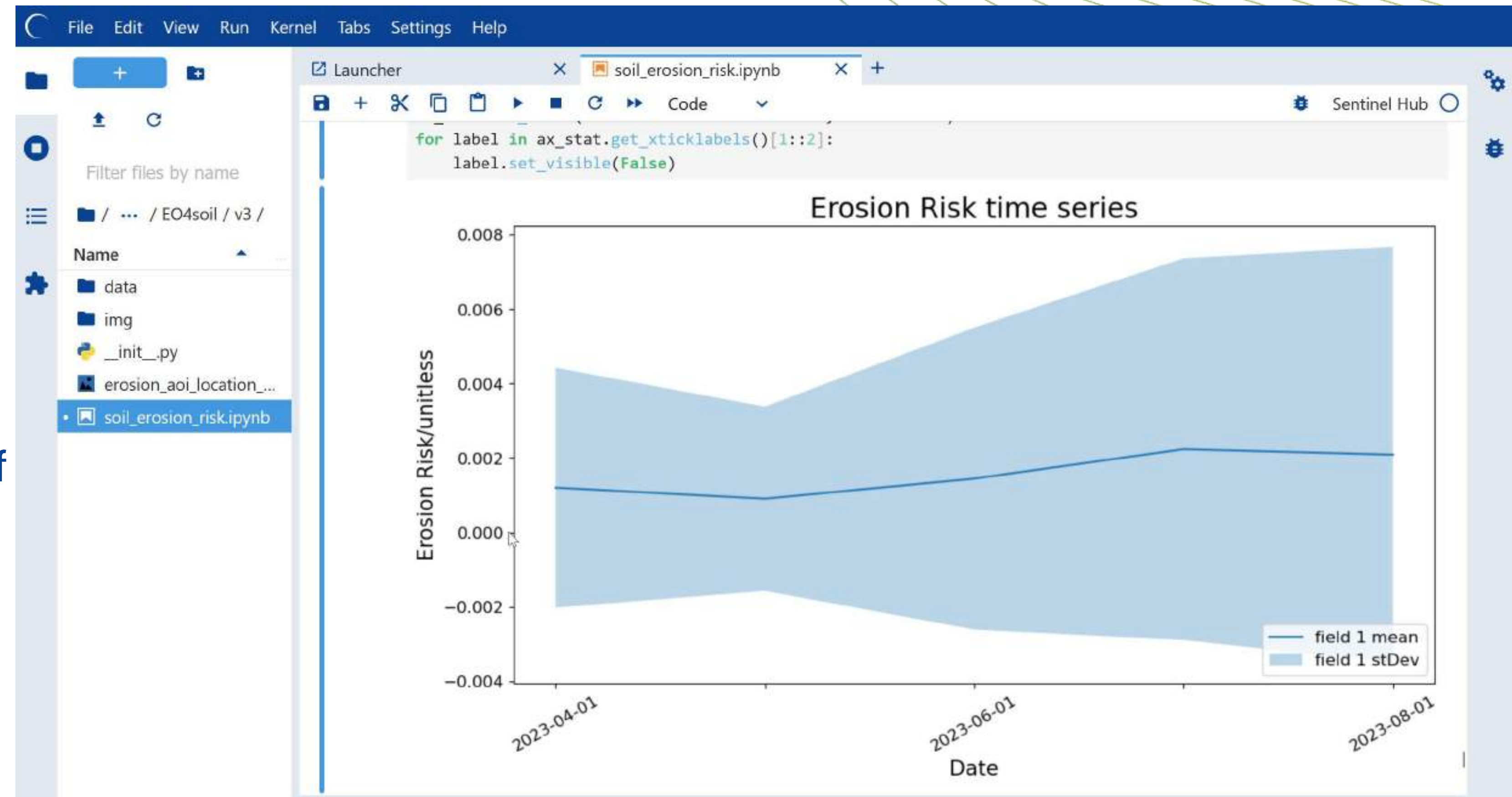
Use case example – global erosion risk estimation in JupyterLab

- JupyterLab has all Sentinel Data Collections, CDSE API-s and a set of EO Code libraries → no need to set up anything locally
- jupyterhub.dataspace.copernicus.eu
- Start notebook, import dependencies
- Define study area (Somogybabod, Hungary)
- Calculate number of bare soil days from Sentinel-2
- Collect terrain slopes – pre-calculated with GDAL on CREODIAS and ingested with BYOC API
- Estimate erosion risk as product of slope and ratio of bare soil days
- Calculate time series of local estimated erosion risk using Statistical API



Use case example – global erosion risk estimation in JupyterLab

- JupyterLab has all Sentinel Data Collections, CDSE API-s and a set of EO Code libraries → no need to set up anything locally https://github.com/eu-cdse/notebook-samples/blob/main/sentinelhub/soil_erosion_risk.ipynb
- jupyterhub.dataspace.copernicus.eu
- Start notebook, import dependencies
- Define study area (Somogybabod, Hungary)
- Calculate number of bare soil days from Sentinel-2
- Collect terrain slopes – pre-calculated with GDAL on CREODIAS and ingested with BYOC API
- Estimate erosion risk as product of slope and ratio of bare soil days
- Calculate time series of local estimated erosion risk using Statistical API



Roadmap overview – New datasets and functionality 2024

January 2024

- STAC – relation links
- openEO: generating netCDF larger than 1GB

February 2024

- On-Demand Production S-1, S-2 and S-3
- Availability of S-1 SLC and S-1 RAW EO compressed products in OData interface
- Additional Sentinel-3 products in openEO
- openEO: HR-VPP collections
- openEO: best available pixel compositing

March 2024

- Gradual start of CCM data access
- VHR 2021 and VHR 2018 in Copernicus Browser
- VHR 2015 (CCM)
- STAC - Support for Assets in newly published Sentinel products
- openEO: Random forest support
- openEO: Client-based-processing

April 2024

- Complete time series of Sentinel-2 Collection 1
- Monthly Sentinel -1 mosaics
- S-1 SLC Burst

May 2024

- VHR 2015 in Copernicus Browser
- STAC - availability of Assets in historical Sentinel products
- Access to historical AUX data for S-1, S-2 and S-3
- SAR Sea Ice (CCM)

June 2024

- Sentinel-3 Level 2 support in Copernicus Browser and Sentinel Hub API's



Copernicus data infrastructure is ready for upscaling of EO-based soil monitoring

- Building and selling/opening EO solutions is possible without major investment in data infrastructure
- Bandwidth, processing capacity or data storage is not a limit any more
- International initiatives can be supported by highly scaleable EO solutions

andras.zlinszky@sinergise.com