Standardised and Scalable EO Workflows using openEO offered by Copernicus Data Space Ecosystem

Hans Vanrompay

dataspace.copernicus.eu

Introduction

dataspace.copernicus.eu

Continental projects powered by openEO

WorldCereal

Land Cover Forest Monitoring (LCFM)

Grasslandwatch

WS: Grasslands and Savannah diversity

Time:
Tuesday, 11/Feb/2025:
3:00pm - 4:30pm

World Ecosystem Extent Dynamics (WEED)

WS: EBVs for the GBF

WorldCereal

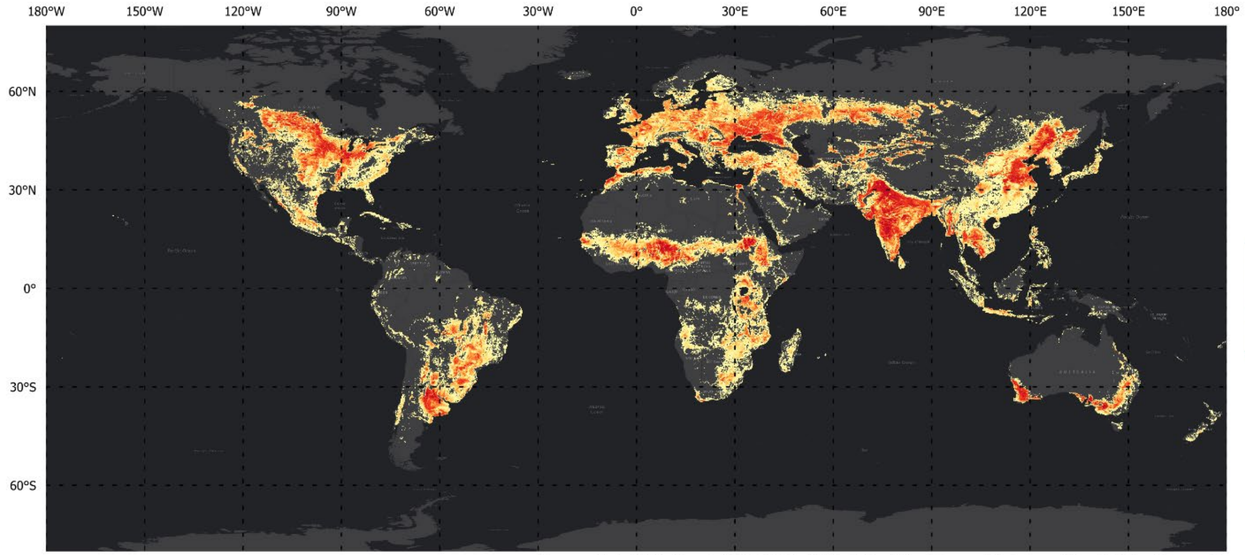
dataspace.copernicus.eu

Project objectives

Inputs for global monitoring of food and water security

1. Where is food crop production taking place?
2. Which crop is grown when and where? (focus on maize and cereals)
3. Which fields are being irrigated and when?

WorldCereal global temporary crop extent product 2021



Demonstrate scientific & technical feasibility of a global and season-specific crop monitoring system at 10m resolution

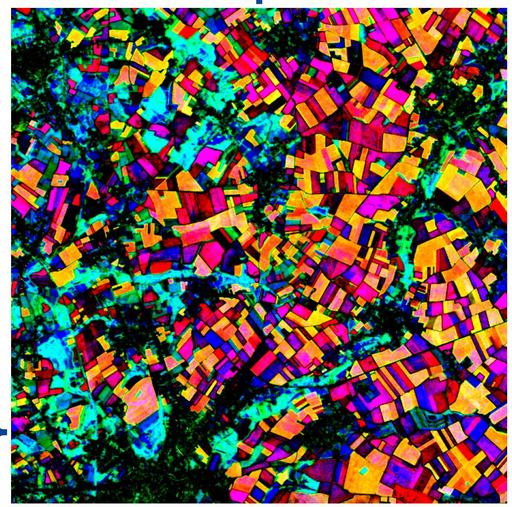
PROGRAMME OF THE EUROPEAN UNION

World Cereal

Choose region and season

Reference data + Pre-processed time series

Pre-trained AI Foundation Model



128 seasonal embeddings

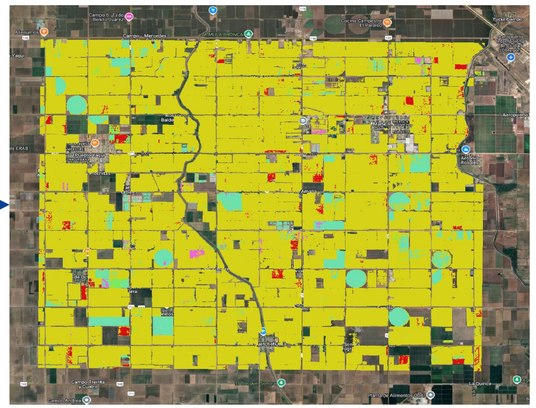
Choose crop types

Linear/MLP head

Custom crop type model

S3 bucket on CDSE

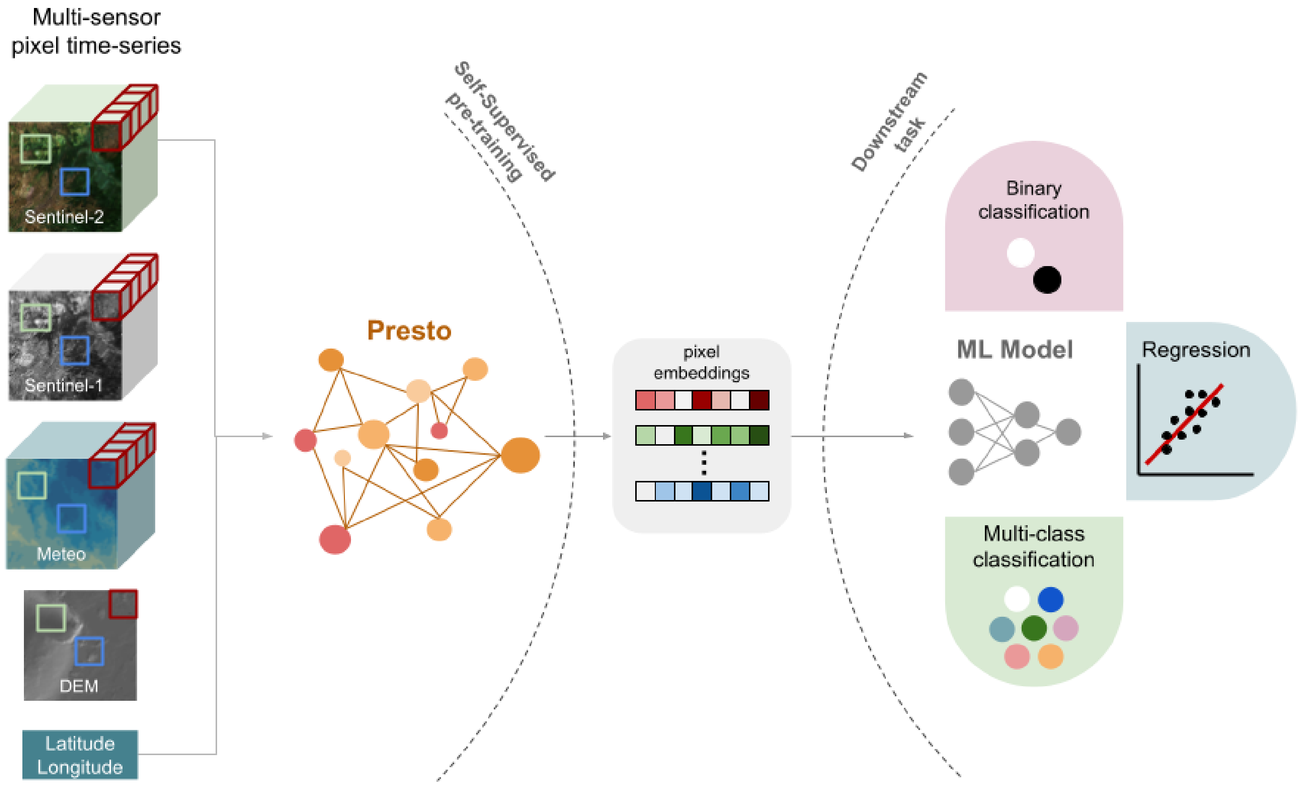
Full demo notebook



Custom crop type map

PROGRAMME OF THE EUROPEAN UNION

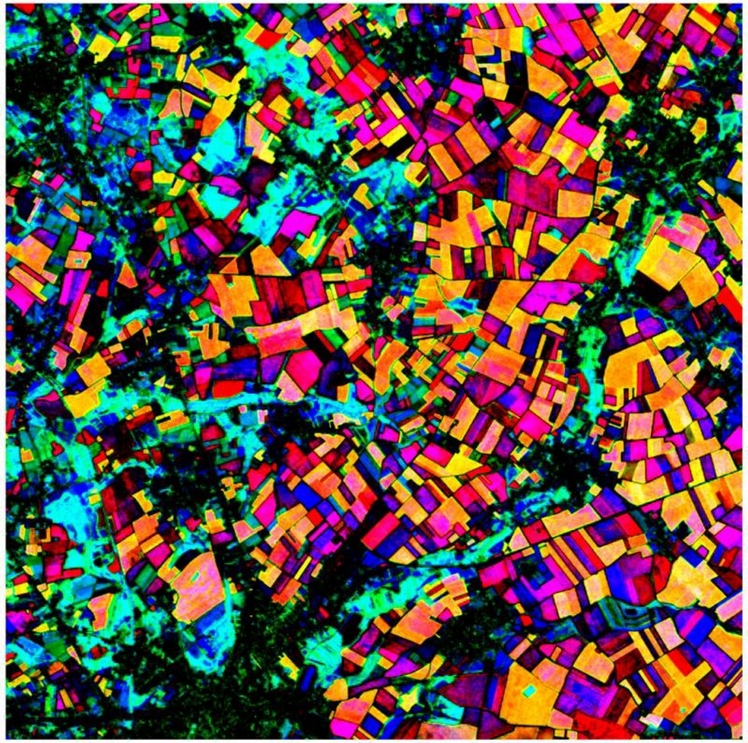
World Cereal



- **Developed by NASA Harvest**
- Learning from **millions of unlabeled data points**
- Representing multi-modal input time series as **one embeddings vector** with size **128**
- Downstream crop mapping training with **high generalizability**
- Updated for WorldCereal
- **Less labels needed**

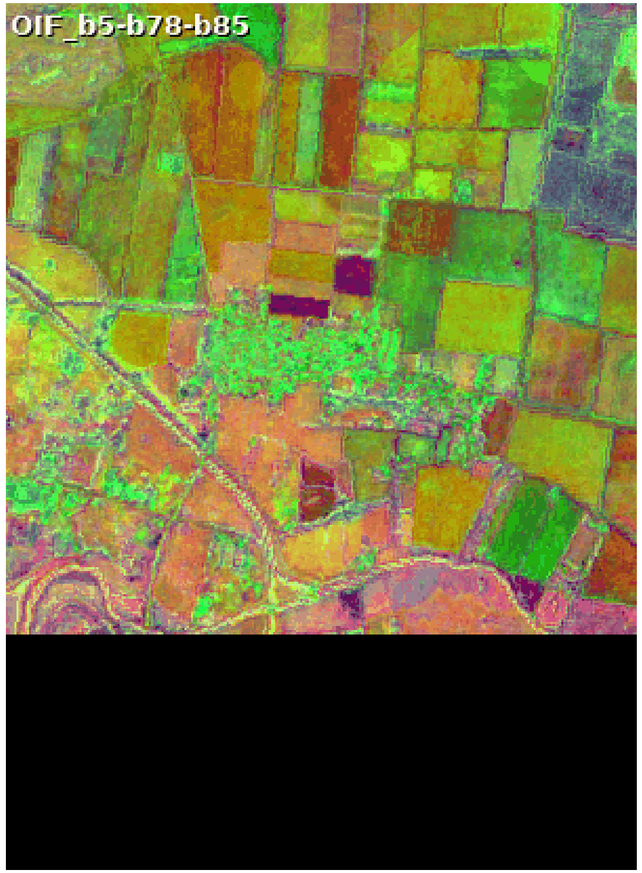
Tseng, G., Cartuyvels, R., Zvonkov, I., Purohit, M., Rolnick, D., & Kerner, H. (2023). Lightweight, pre-trained transformers for remote sensing timeseries. arXiv preprint arXiv:2304.14065.

World Cereal



- All satellite and other inputs are **summarized** into a 128 highly informative “embeddings”
- These are the basis for classifying crops in your region
- WorldCereal brings the power of Presto to the user
- You benefit from what has been learned in the past and **train a lightweight model for your needs based on your data**

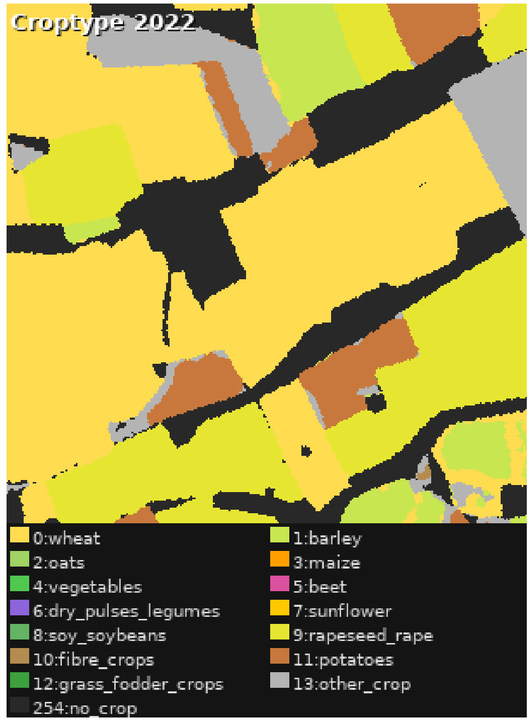
World Cereal

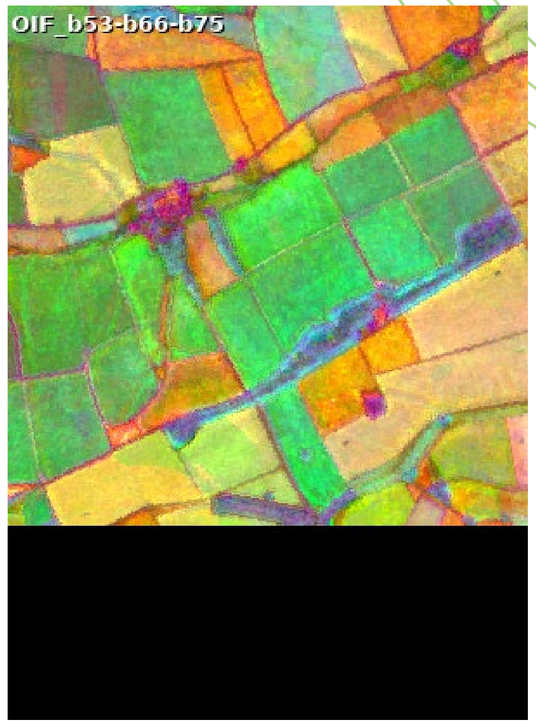


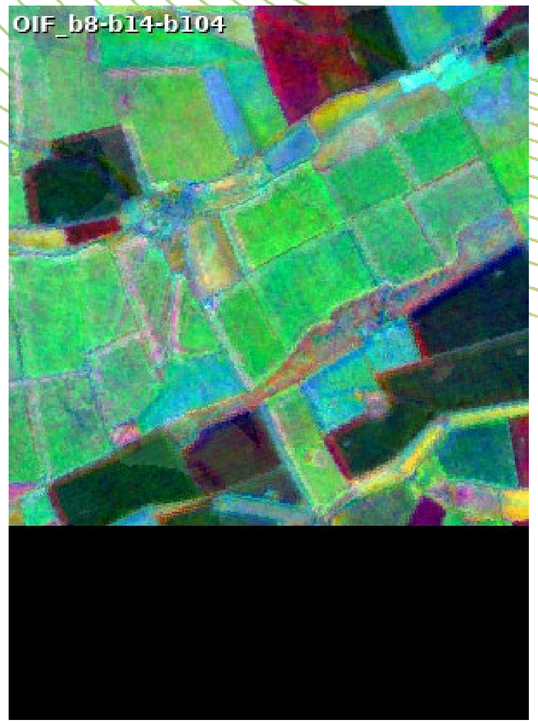
Geospatial Explorer

worldcereal-classification/notebooks/worldcereal_demo_embeddings.ipynb at main · WorldCereal/worldcereal-classification

World Cereal







[worldcereal-classification/notebooks/worldcereal_demo_embeddings.ipynb at main · WorldCereal/worldcereal-classification]

PROGRAMME OF THE EUROPEAN UNION

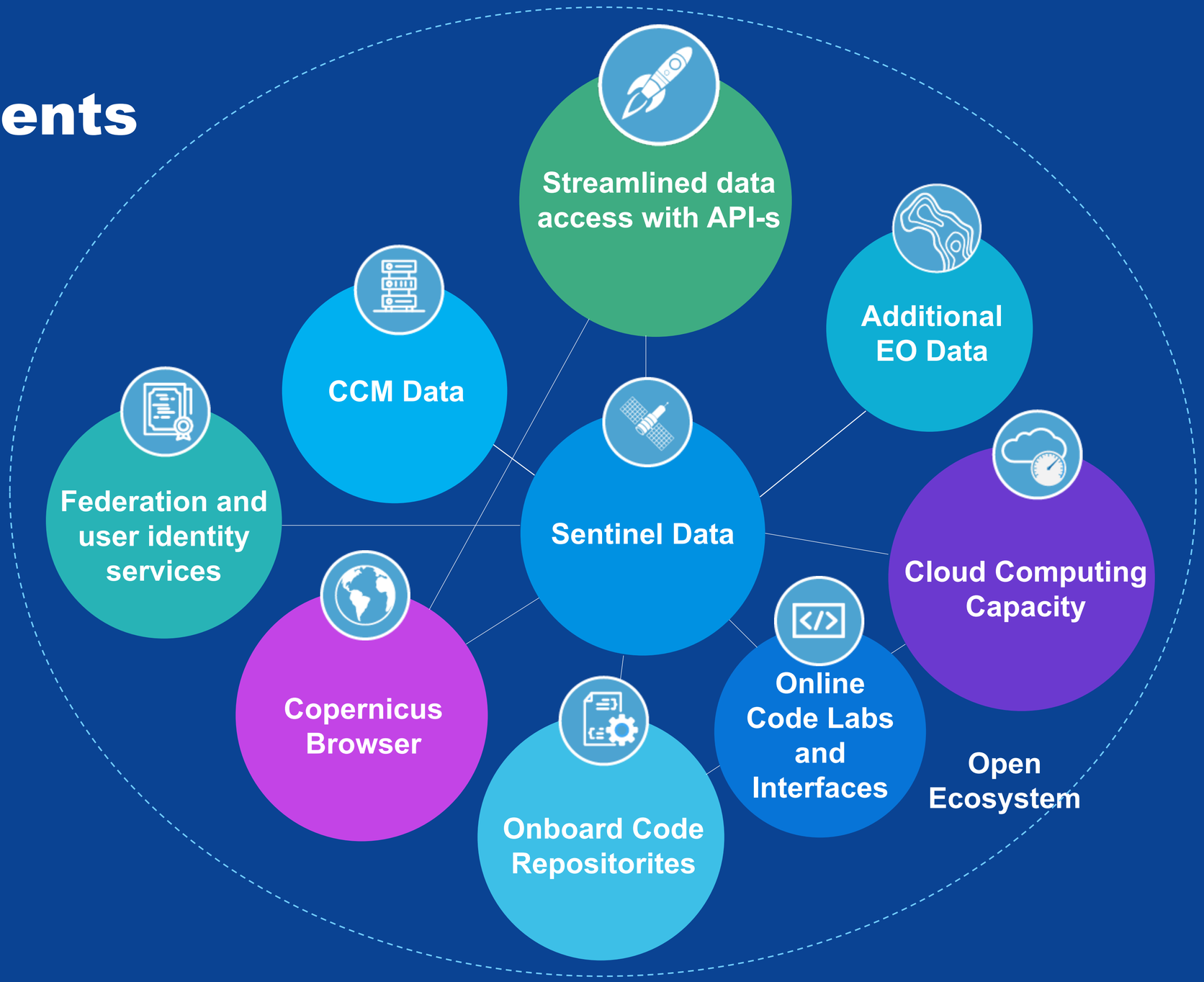
From project to product with UDPs in APEX

APEx Algorithm Catalogue

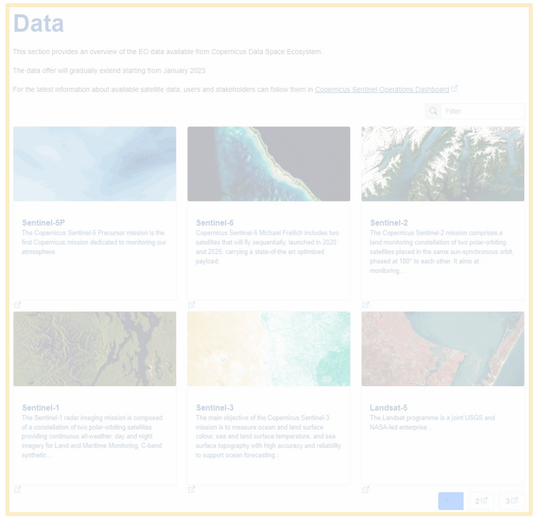
openEO - CDSE

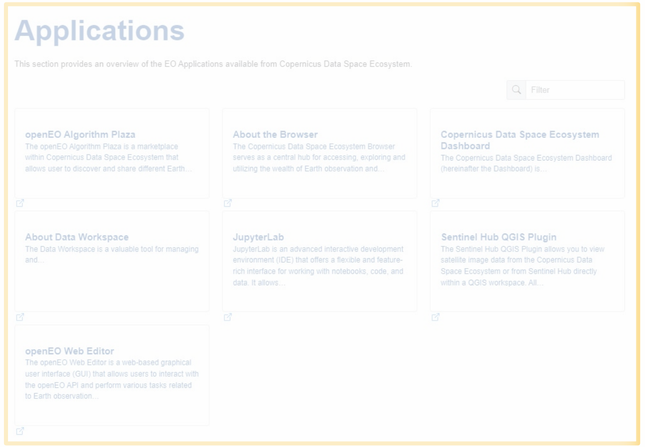
dataspace.copernicus.eu

CDSE components



Data Access and Processing API: openEO



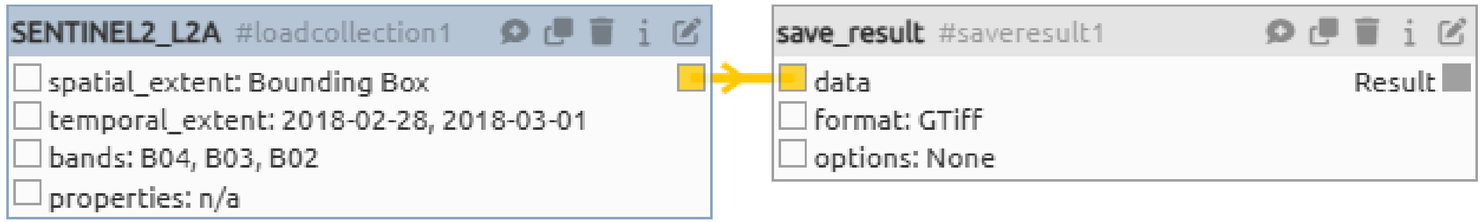


What is openEO?

*It is an **open-source API** that provides standardised access to EO data and simplifies the deployment of processing workflows as **scalable, cost-efficient services**.*

What is openEO?

```
{
 "process_graph": {
  "loadcollection1": {
   "arguments": {
    "bands": ["B04",  "B03",  "B02"],
    "id": "SENTINEL2_L2A",
    "spatial_extent": {"east": -76.19, "north": 3,  "south": 2.99, "west": -76.2},
    "temporal_extent": ["2018-02-28", "2018-03-01"]
   },
   "process_id": "load_collection"
  },
  "saveresult1": {
   "arguments": {
    "data": {
     "from_node": "loadcollection1"
    },
    "format": "GTiff",
    "options": {}
   },
   "process_id": "save_result",
   "result": true
  }
 }
}
```



vito

vito.b

Why openEO?

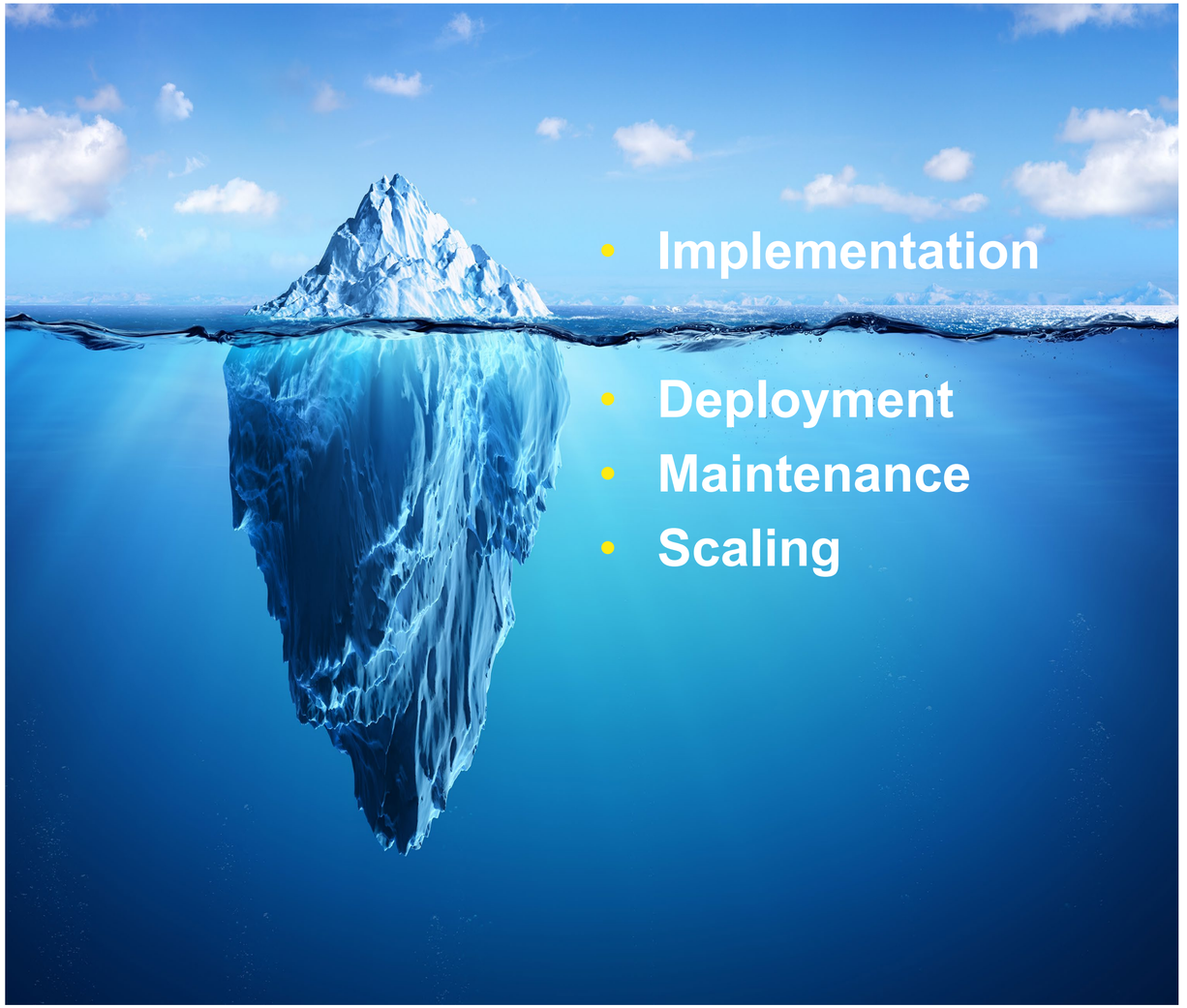
- **Simple data access & processing** for multiple Earth observation datasets
- Scalable and efficient **cloud computation**
- A **standardized system** that works across different platforms
- **Focus on the task at hand**, not the data engineering
- Supporting principles of **FAIR** (Findable, Accessible, Interoperable, and Reusable) data and **Open Science**

openEO instance in Copernicus Data Space Ecosystem

openEO


On the fly geospatial processing
Python
Scala
GeoTrellis
GDAL
Product catalogs
STAC
EO OpenSearch
Processing & Microservice middleware layer
APACHE Spark
Flask
Optional production components
KEYCLOAK
træfik proxy
Persistence
Just a database
ZooKeeper
In-Memory
Container Orchestration layer: anything that runs Spark
kubernetes
hadoop
Your Laptop
Shared Storage
Amazon S3
COG
GDAL supported formats

openEO

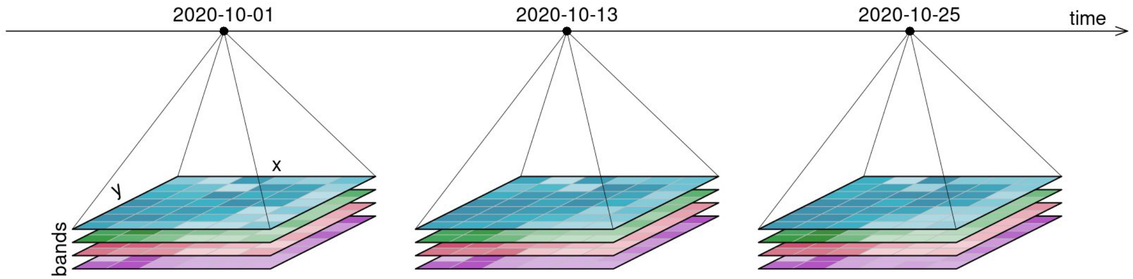


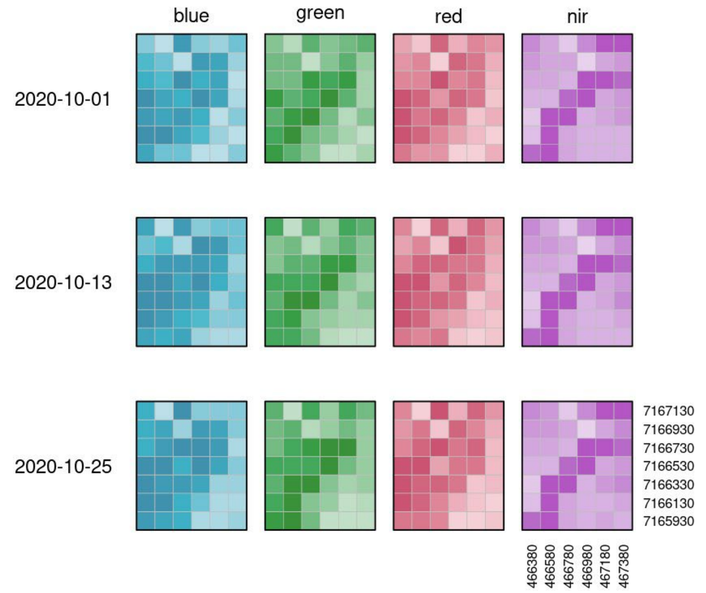
vito

vito.b

openEO concept: Datacube

- multidimensional arrays with one or more spatial or temporal dimensions
- Data in OpenEO is represented in this way
- Any representation of the data cube is fine (meaning – dimensions can be switched in display)





vito

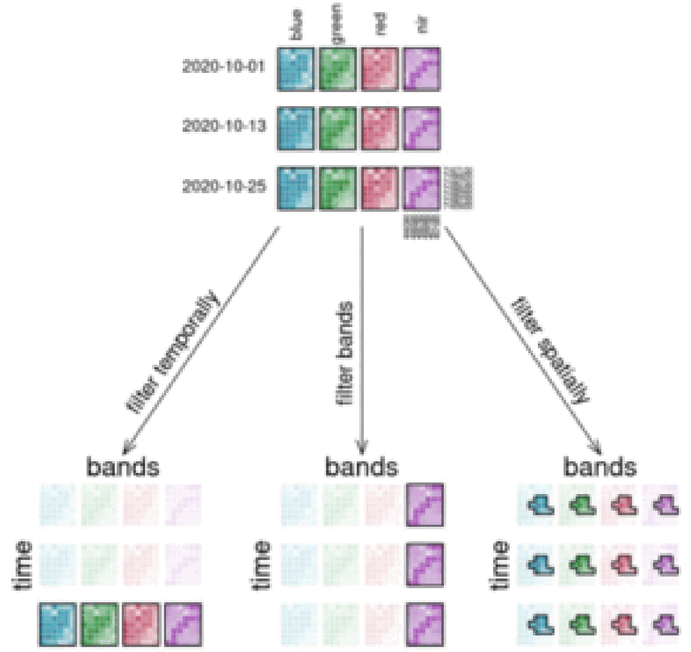
vito.b

openEO concept: Processes

Cubes and Clouds
https://eo-college.org/

Apply processes on your datacube

- Apply (values, kernels, temporal, …)
- Resample
- Filter
- Aggregate
- Reduce
- …

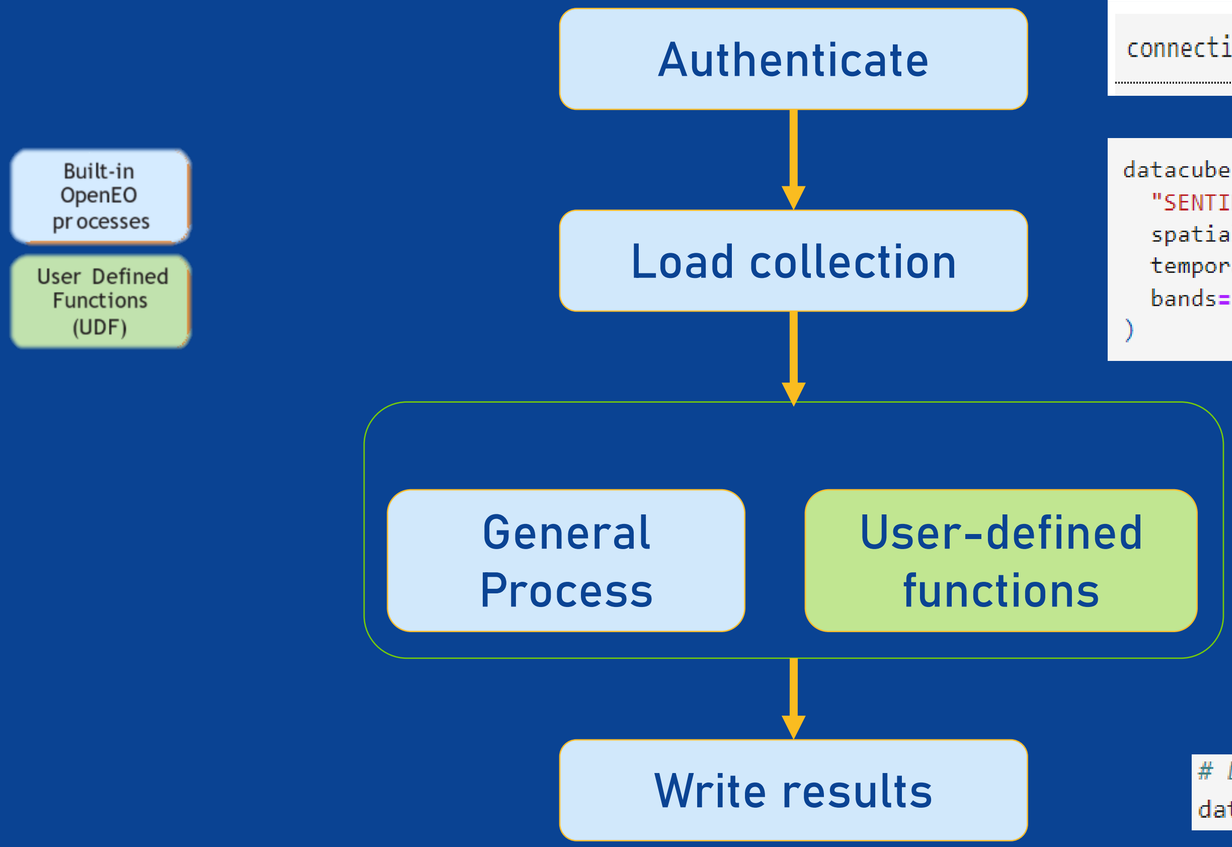


vito

vito.b

JupyterLab | Copernicus Data Space Ecosystem

openEO Workflow




```
connection = openeo.connect("openeo.dataspace.copernicus.eu").authenticate_oidc()
```

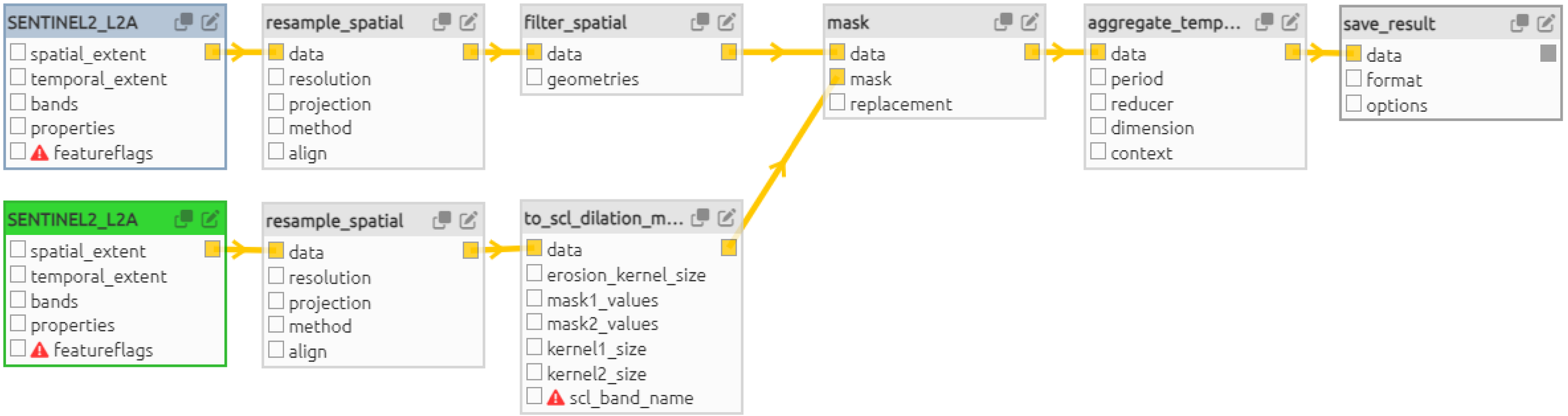
```
datacube = connection.load_collection(
  "SENTINEL1_GRD",
  spatial_extent={"west": 16.06, "south": 48.06, "east": 16.65, "north": 48.35},
  temporal_extent=["2017-03-01", "2017-04-01"],
  bands=["VV", "VH"]
)
```

https://processes.openeo.org/
https://open-eo.github.io/openeo-python-client/udf.html

```
datacube = datacube.min_time()
```

```
# Let's download the data
datacube.execute_batch(title="Simple execution", outputfile="Sentinel1.nc")
```

Constructing a Process graph (LCFM)



Practical Exercise

openeo-community-examples/python/1. GettingStarted/GettingStarted.ipynb at main · Open-EO/openeo-community-examples

With openEO

- Simple data access & processing for multiple Earth observation datasets
- Scalable and efficient processing capabilities
- A standardized system that works across different platforms
- Independence from underlying technologies and software libraries
- Supporting principles of FAIR (Findable, Accessible, Interoperable, and Reusable) data and Open Science, e.g. transparent workflows

Advanced Workflows

dataspace.copernicus.eu

PROGRAMME OF
THE EUROPEAN UNION

Continental projects powered by openEO

WorldCereal

Land Cover Forest Monitoring (LCFM)

Grasslandwatch

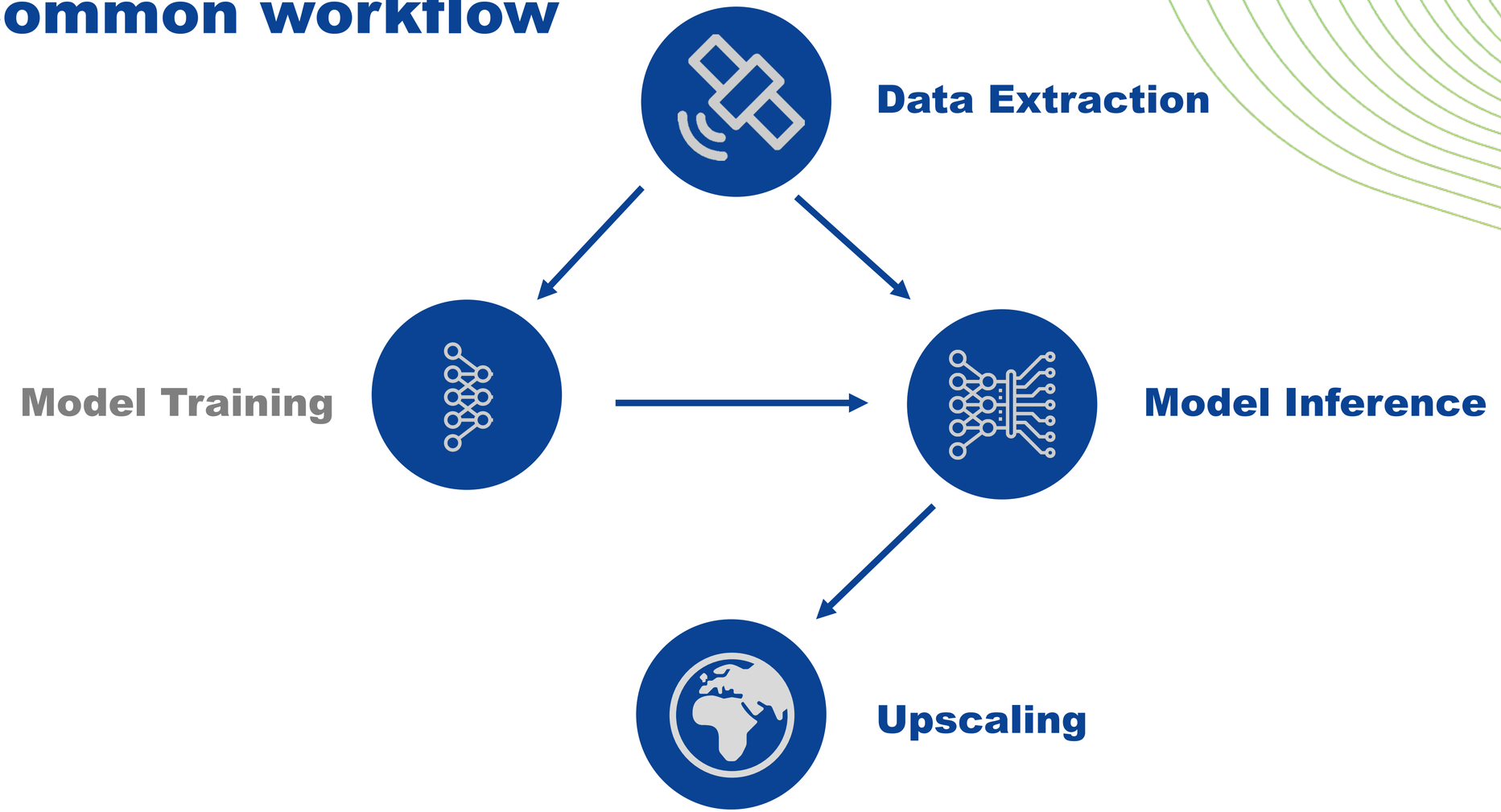
WS: Grasslands and Savannah diversity

Time:
Tuesday, 11/Feb/2025:
3:00pm - 4:30pm

World Ecosystem Extent Dynamics (WEED)

WS: EBVs for the GBF

Common workflow



Data Extraction

dataspace.copernicus.eu

PROGRAMME OF
THE EUROPEAN UNION

Data extraction

1. Create an openEO processing pipeline
2. Create a dataframe containing the input parameters for the areas of interest
3. Efficiently scale up the execution with openEO's MultiBackendJobManager

openeo-community-examples/python/ManagingMultipleLargeScaleJobs at main · Open-EO/openeo-community-examples

Model Inference

dataspace.copernicus.eu

Model Inference

1. Convert your model to ONNX
2. Create a Model Specific UDF
3. Create an openEO job which executes the UDF

openeo-community-examples/python/OnnxMLInference/Onnx_ML_Inference.ipynb at main · Open-EO/openeo-community-examples

User-Defined Function

```python
def apply_datacube(cube: xr.DataArray, context: Dict) -> xr.DataArray:
    """
    Run the inference on the given input data using the provided ONNX runtime session.
    This method is called for each timestep in the chunk received by apply_datacube.
    """
    # Load the ONNX model
    ort_session = _load_ort_session("test_model.onnx") # name of the model in the archive

    # Make sure the input dimensions are in the expected order and save the original shape
    input = cube.transpose("bands", "y", "x")

    # Get the underlying np.ndarray and reshape it to the expected shape.
    output =  ort_session(input)

    # Return the output as an xarray DataArray
    return xr.DataArray(
        output.reshape(input.shape),  # Reshape the output to the original shape (bands, y, x)
        dims=["bands", "y", "x"],
    )
```

Managing Dependencies

- External Python packages may not be available in the base environment

→ Inline dependency specification (PEP 723)

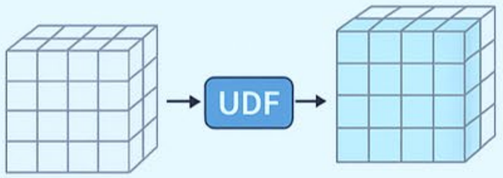
- Alternative: Create custom dependencies:
  - Create a virtual environment with required packages.
  - Upload the zip file to an accessible URL.
  - Specify the archive in the job options

```
# /// script
# dependencies = [
#   "numpy",
#   "scikit-learn",
# ]
# ///
```

Applying the UDF

apply

Applies a UDF to each pixel individually




```
datacube = datacube.apply(my_udf)
```

apply_dimension

Applies a UDF along a specified dimension


```
datacube = datacube.apply
dimension(dimension='t', process
=my_udf)
```

reduce_dimension

Applies a UDF along a specified dimension, after which it collapses


```
datacube = datacube.reduce_
dimension(dimension='t', reducer
=my_udf)
```

apply_neighborhood

Applies a UDF to each pixel considering its neighborhood


```
datacube = datacube.apply
neighborhood(process=my_udf
size=[...],
overlap=[...]
```

Standardisation

dataspace.copernicus.eu

PROGRAMME OF
THE EUROPEAN UNION

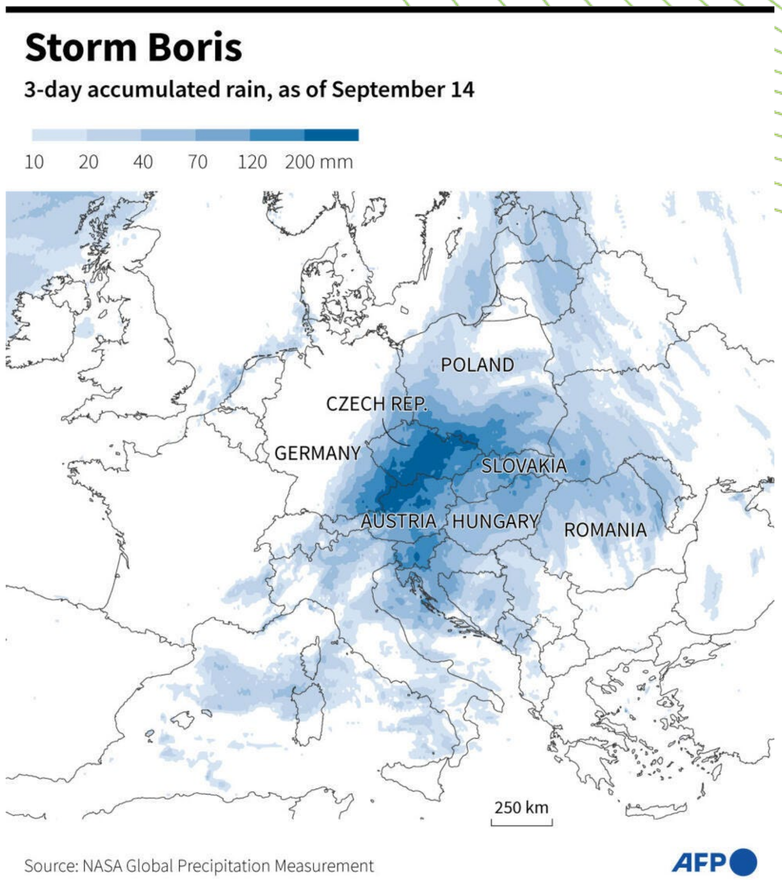
From project to product with UDPs in APEX

APEx Algorithm Catalogue

What’s Next?

Introduction

You are a rescue worker with less than 12 hours to decide; to evacuate or not?



What if we could go seamlessly from data to decision?

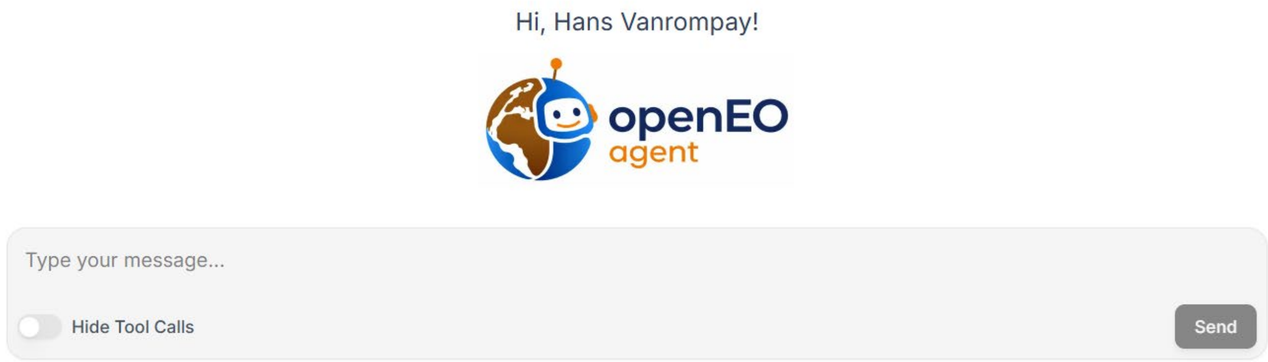
PROGRAMME OF THE EUROPEAN UNION

Copernicus

esa

Solution

- **AI powered agent that:**
  - Translates user-intent into openEO code
  - Validates and executes the code
  - Explainability of open-source code
  - Returns on demand maps



Solution

PROGRAMME OF
THE EUROPEAN UNION

AI meets openEO

PROGRAMME OF THE EUROPEAN UNION

Thank You

PROGRAMME OF
THE EUROPEAN UNION

Relevant links:

- **Forum**: https://forum.dataspace.copernicus.eu/
- **Documentation**: https://documentation.dataspace.copernicus.eu/Home.html
- **Submit a ticket**: https://helpcenter.dataspace.copernicus.eu/hc/en-gb/requests/new
- **openEO endpoint**: “openeo.dataspace.copernicus.eu”
- **openEO web-editor**: https://openeo.dataspace.copernicus.eu/
- **openEO Python client documentation**: https://open-eo.github.io/openeo-python-client/index.html
- **openEO community examples**: https://github.com/Open-EO/openeo-community-examples/tree/main/python
- **openEO Algorithm Plaza**: https://marketplace-portal.dataspace.copernicus.eu/catalogue
- **Youtube videos**: https://www.youtube.com/@copernicusdataspaceecosystem
- **FAQ**: https://documentation.dataspace.copernicus.eu/FAQ.html