Exploring the FuseTS Toolbox: Fusing and Analyzing Multi-Source EO Time Series Data

Big Data from Space: BiDS

06/11/2023

Bram Janssen

Pratichhya Sharma

vito.be

AGENDA

PART 1

- Introduction to FuseTS
- Introduction to openEO
- Features of FuseTS

PART 2

- EOplaza Marketplace
- Hands-on Exercise

PART 3

- How to contribute?
- Q&A

Introduction to FuseTS

- About FuseTS
- Why FuseTS?
- What is included in FuseTS?
- Use Cases
- Challenges and solutions

vito SINERGISE

VNIVERSITAT DE VALÈNCIA IPL IMAGE PROCESSING LABORATORY

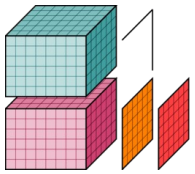
About FuseTS

- Open-source library
- Pre-Implemented Functions
- Data-Fusion Capabilities
- Focusing on Time-Series data analysis

To simplify the analysis of multi-dimensional time series data by providing a user-friendly interface open-source framework for detecting and predicting changes in land environments.

About FuseTS

An open-source service in the form of source code that focuses on simplifying and unifying Earth Observation data processing and analysis.



xarray

An open-source Python library for labelled, multi-dimensional array manipulation and analysis.

Why FuseTS?

- Advantages of openEO
  - Simple access to EO dataset, scalable, cross-platform system, reproducibility and independent of underlying technologies.
- Focusing on Time-Series data analysis
  - Designed for processing and analyzing multiple time series data, particularly focusing on Earth observation data.
- Extensive and easy to use
  - Built with a modular and extensible framework, which provides flexibility and customization options.
- Open-source
  - An active community surrounding FuseTS can provide ongoing support, updates, and contributions, making it a valuable resource.

Included in FuseTS

Time series analytics

```
from fusets import WhittakerTransformer
```

Time series smoothing methods take a single time-series (from a pixel or aggregated over an area) and smooth it over time. This reduces noise and allows the filling of gaps by interpolating along the smoothed curve that is fitted through the observations.

Time Series Fusion & Prediction

```
from fusets import MOGPRTransformer
```

Time series fusion methods take multiple input time series and produce a new, fused product, which tries to capture valuable information from the independent input sources.

Use Cases

- **Land Cover Change Monitoring**
  Detection of subtle changes to land cover (forest degredation and deforestation) using fused datasets
- **Cropland Phenology Indicators**
  Extraction of subsequent phenology metrics such as start-of-season, peak-of-season, end-of-season, …
- **Agriculture and Land Management Activities Identification**
  Dectection of land management activities (e.g. mowing) using fused datasets.

Challenges We Faced

- Understanding the shortcomings of the library
- Evaluating and improving the performance of implemented tools
- Understanding complex fusion and analytics methods to recognize areas of useful applications

Implemented Solutions

- Optimization of the included processes
- Versatility so that it can be used in different use cases
- Usability: both local and cloud versions

Introduction to openEO

- About openEO
- Why openEO?
- openEO Ecosystem
- openEO Workflow
- Future Plans

vito SINERGISE

VNIVERSITAT DE VALÈNCIA IPL IMAGE PROCESSING LABORATORY

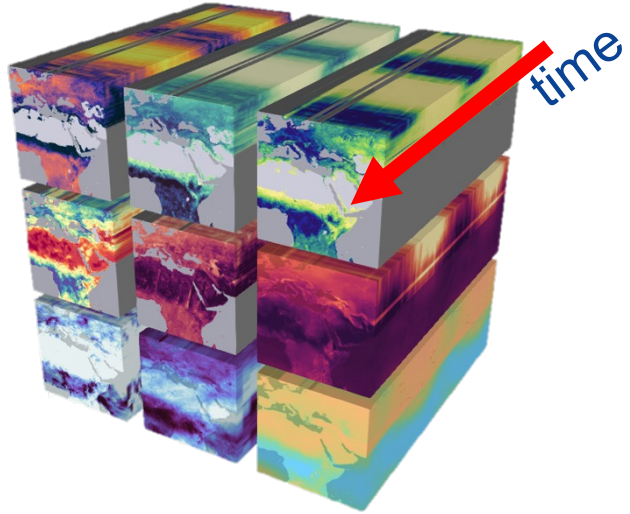
About openEO

Definition

It is an open-source service in the form of source code that focuses on simplifying and unifying Earth Observation data processing and analysis.

Datacubes

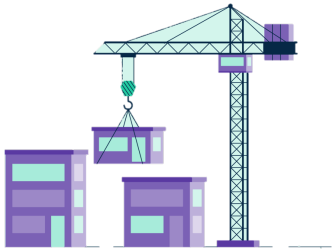
- multi-dimensional data structure
- Processes designed for efficient analysis



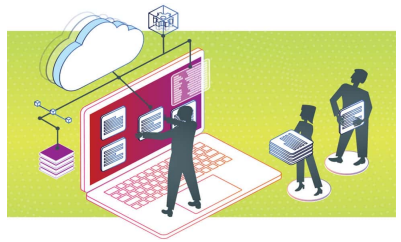
Why openEO?

- Unified and straightforward access to multiple Earth observation datasets

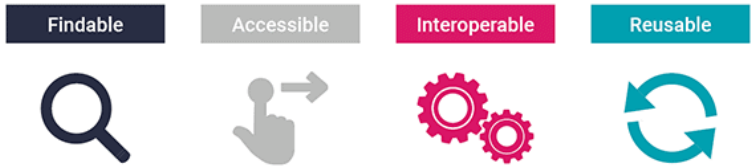
- Scalable and efficient processing capabilities



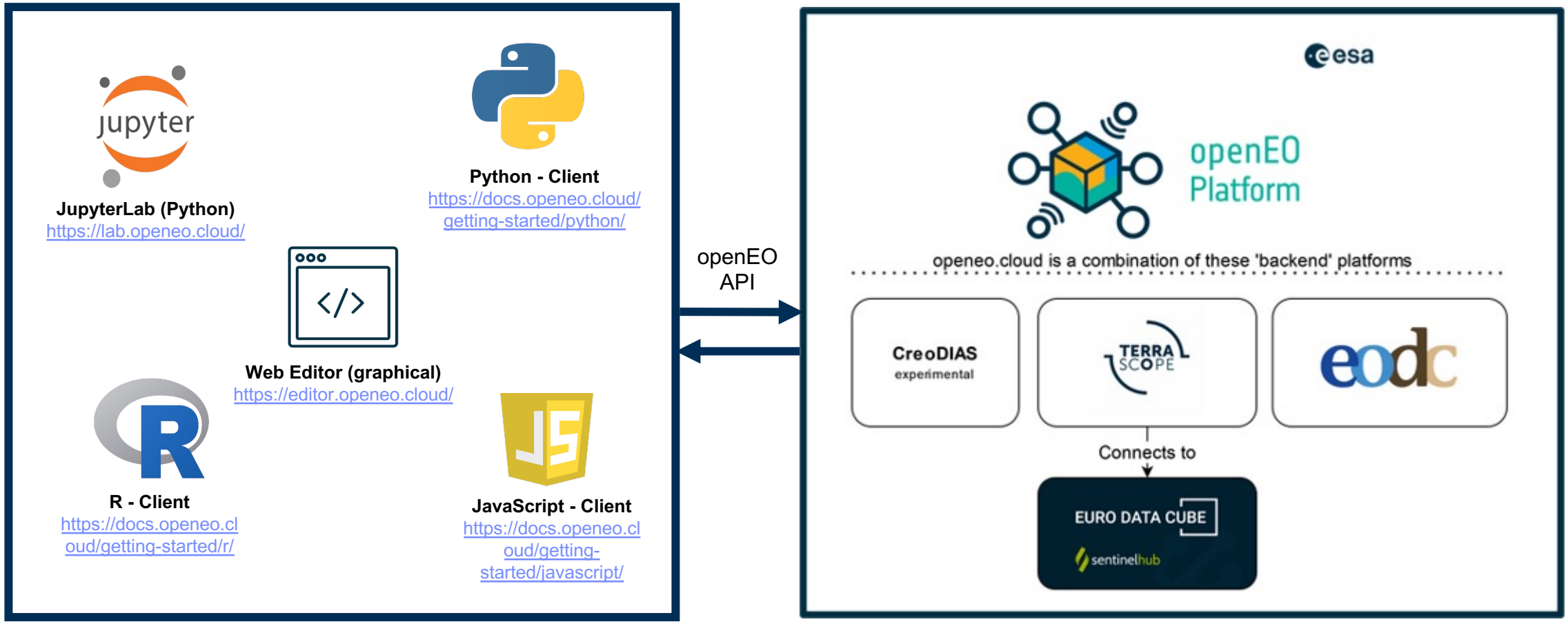
- A standardized system that works across different platforms



- Independent from underlying technologies and software libraries
- Supports principles of FAIR data and Open Science



openEO ecosystem



openEO ecosystem

JupyterLab (Python)
https://lab.openeo.cloud/

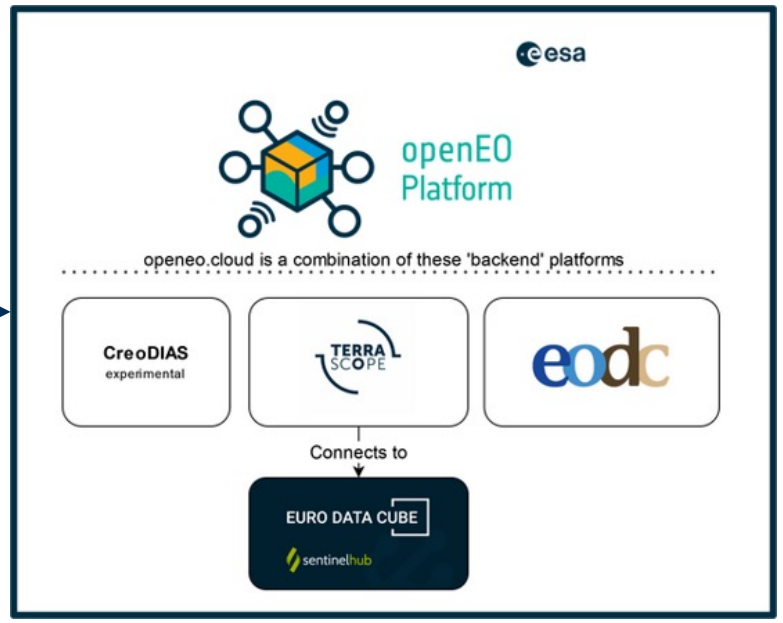
Python - Client
https://docs.openeo.cloud/getting-started/python/

Web Editor (graphical)
https://editor.openeo.cloud/

R - Client
https://docs.openeo.cloud/getting-started/r/

JavaScript - Client
https://docs.openeo.cloud/getting-started/javascript/

openEO API



openEO Workflow

#1

Authentication

#2

Search and check availability of the collection

#3

Download all the tiles for Aoi and interested time

#4

Preprocess and Resample

#5

Create a subset in space & time

#6

Apply algorithm

#7

Learn & assign for large job handling

#8

Save and deliver

Authenticate — #1

Load Collection — #2

Apply Algorithm — #3

openEO handles — #4

Download — #5

```
openeo.connect('
```

```
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"]
)
```

Built-in OpenEO processes

User Defined Functions (UDF)

```
# batch processing in case of larger area
cropsar = cropsar.save_result(format="JSON")
job = cropsar.execute_batch(title="CropSAR FCOVER")
```

vito

SINERGISE

VNIVERSITAT DE VALÈNCIA

IPL IMAGE PROCESSING LABORATORY

vito.be

Wednesday 08/11 18:30 – 19:00

openEO Plaform Showcase

Features of FuseTS

- Smoothing
- Data Fusion
- Timne Series Analytics
- Future Plans

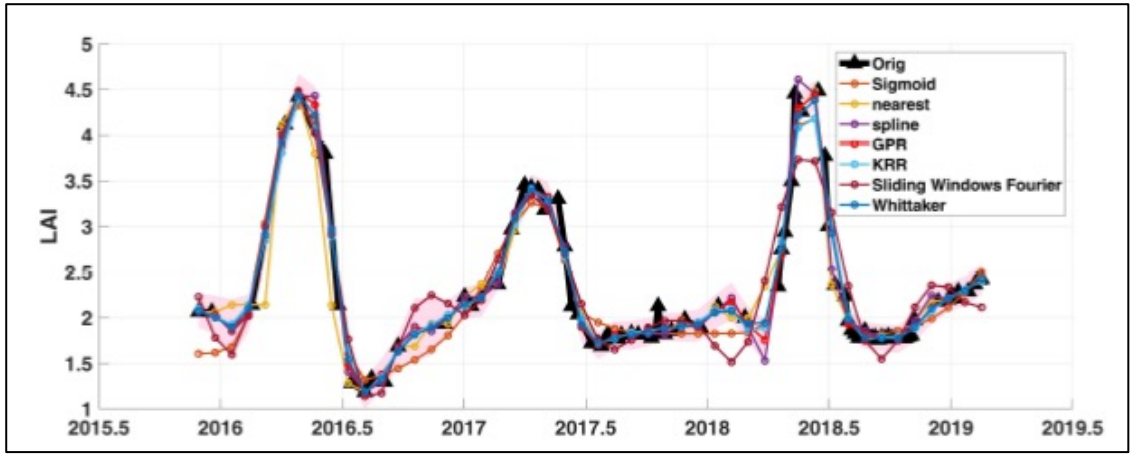
vito SINERGISE

VNIVERSITAT DE VALÈNCIA IPL IMAGE PROCESSING LABORATORY

Smoothing

- Usually, single-source gap-filling algorithms
- Useful when time series gaps are rather short (e.g., in the order of weeks)

Whittaker
Gap Filling Smoother



Original and reconstructed time series of LAI using several gap-filling techniques. Interpolated values of time series at a higher sampling frequency (every 25 days). The GPR uncertainties are shown in red shade areas.

Data Fusion

- Making use of multiple input sources
- Only beneficial in case of long temporal gaps
  (e.g., missing key seasonal events due to persistent cloud cover)
- Able to extract correlations from outputs
- Computational cost is high

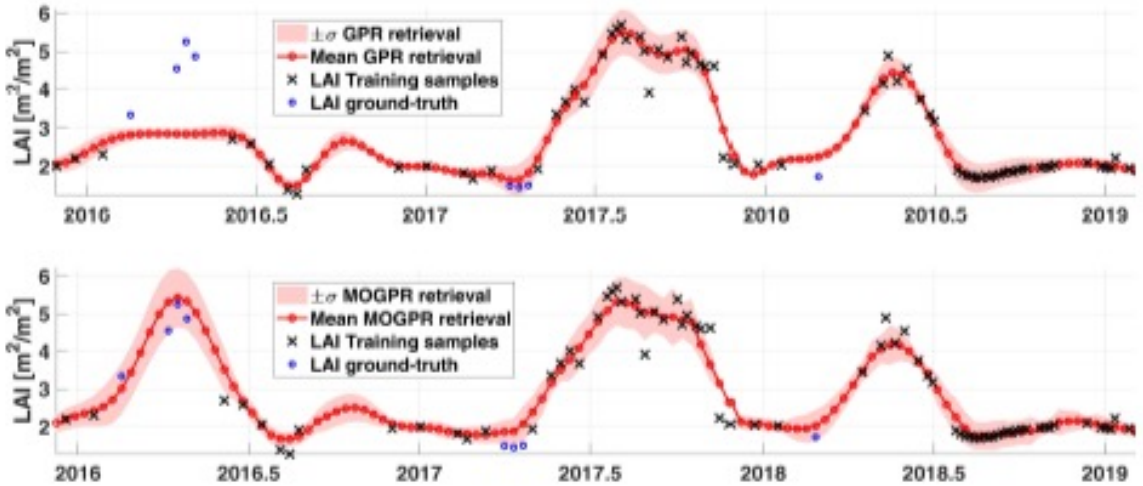
MOGPR
Multi-output Gaussian Process Regression

CropSAR
GAN-based neural network for S1 + S2 fusion

Data Fusion
What is MOGPR?

- Multi-output Gaussian Process Regression
- Learns correlations between inputs and produces new outputs
- Comes with uncertainties

(a)



(b)

GPR (top) and MOGPR (bottom) predictions (red) of S2 LAI

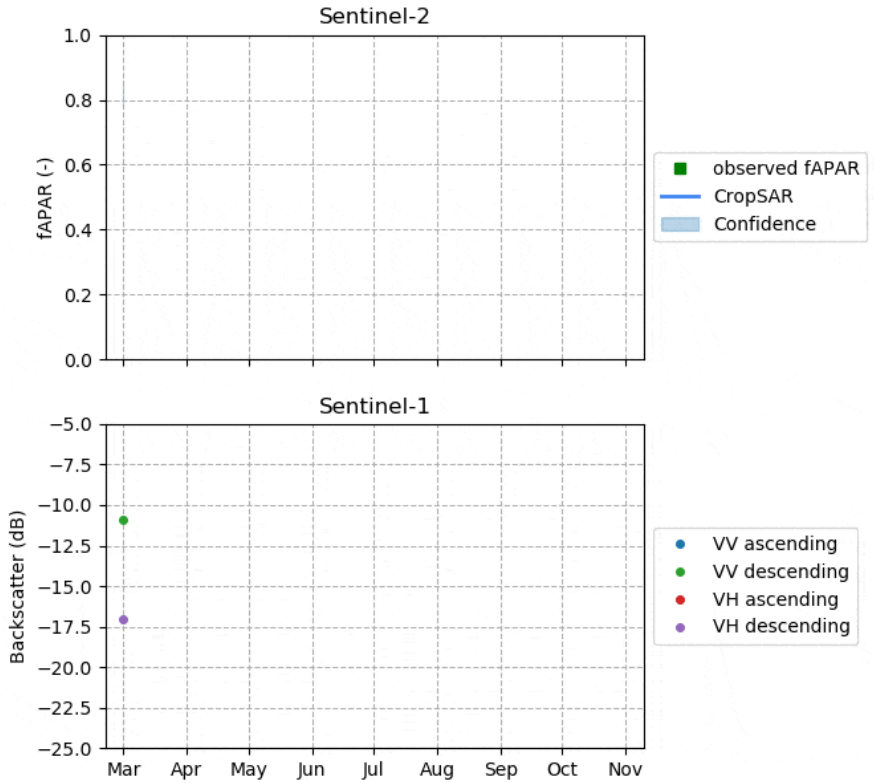
Black crosses represent data during training, while blue points are ground-truth data eliminated from the training step

MOGPR captures these eliminated points from correlations with S1 data

The ± σ prediction uncertainty is represented by the boundary shade (pink)

Data Fusion
What is CropSAR?

- An AI-based tool to extract cloud-free time series of biophysical indices (NDVI, fAPAR, fCOVER)
- Based on joint radar Sentinel-1 and optical Sentinel-2 observations
- Works best over agricultural fields



Animation of CropSAR performing over a potato field. It builds upon any available cloud-free Sentinel-2 observations (green squares).

The long absence of valid Sentinel-2 data during crop emergence is here successfully bridged by CropSAR, thanks to the information contained in the Sentinel-1 signal.

Time Series Analytics

- Extract information from (fused) data streams
- Modelling and analytics

Phenology Metrics

Peak Valley Detection

Future Plans

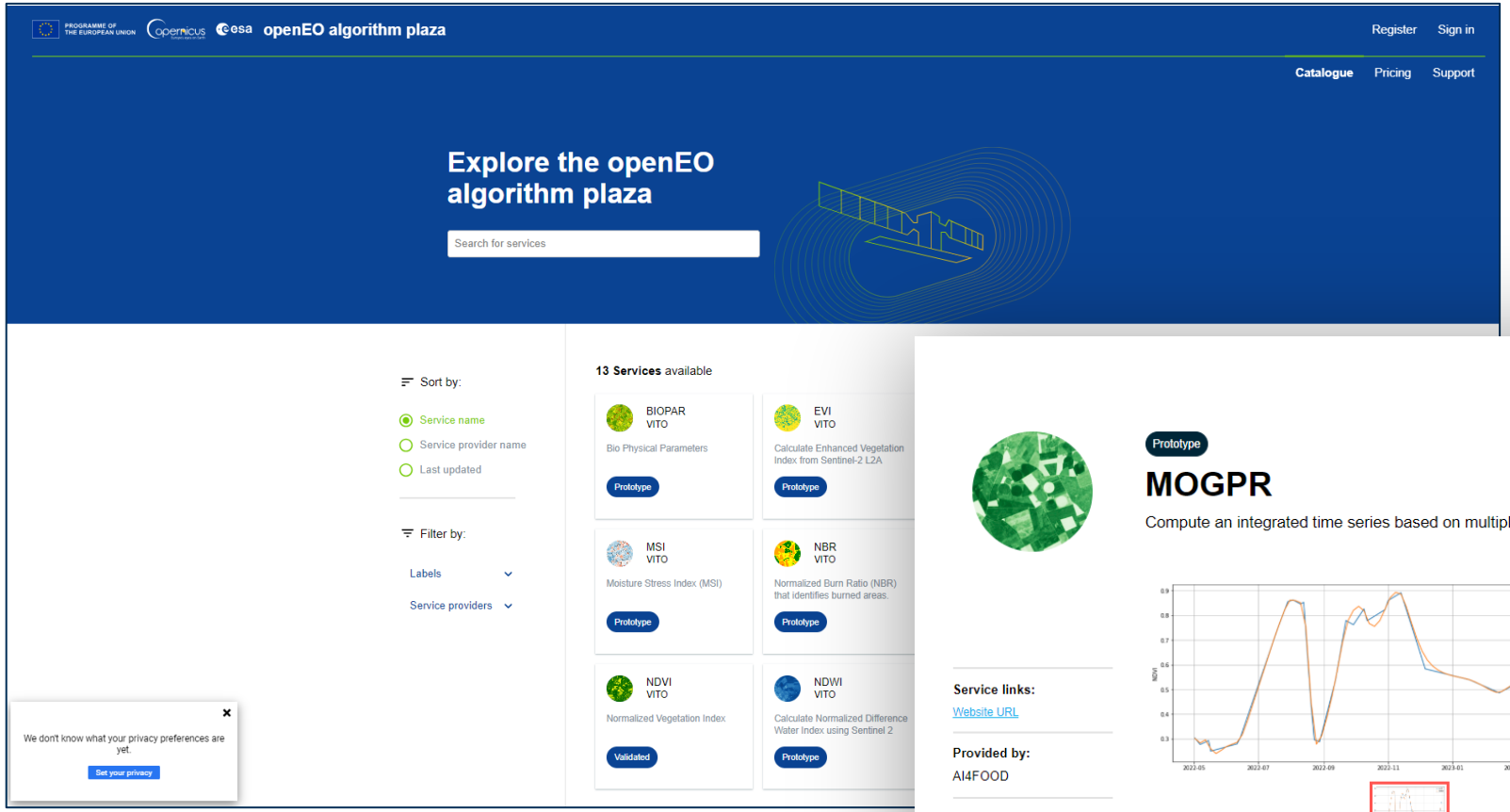
- Improve current features of the toolbox
- Integration of services into marketplaces
    - Copernicus Data Space Ecosystem
    - Network of Resources

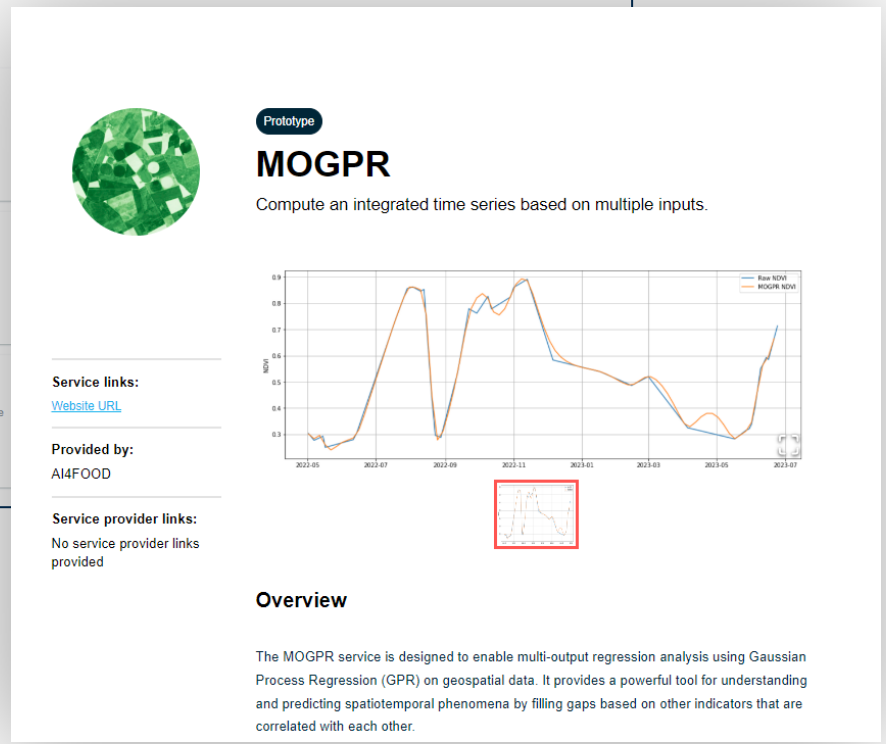
EOplaza Marketplace

- Introduction
- openEO platform credits
- Network of Resources
- Executing FuseTS services
- Onboarding your own service

vito SINERGISE

VNIVERSITAT DE VALÈNCIA IPL IMAGE PROCESSING LABORATORY





vito
SINERGISE
VNIVERSITAT DE VALÈNCIA
IPL IMAGE PROCESSING LABORATORY
vito.be

Level 1

Proof of Concept

- Service is provided ‘as-is’
- with a short description and possibly
- a reference to what it tries to implement (scientific paper, well know metric, …)

Level 2

Incubating service

- Quality of the service is documented with example requests (sets of parameters) and
- the corresponding output, as well as
- the resources required to generate that output

Level 3

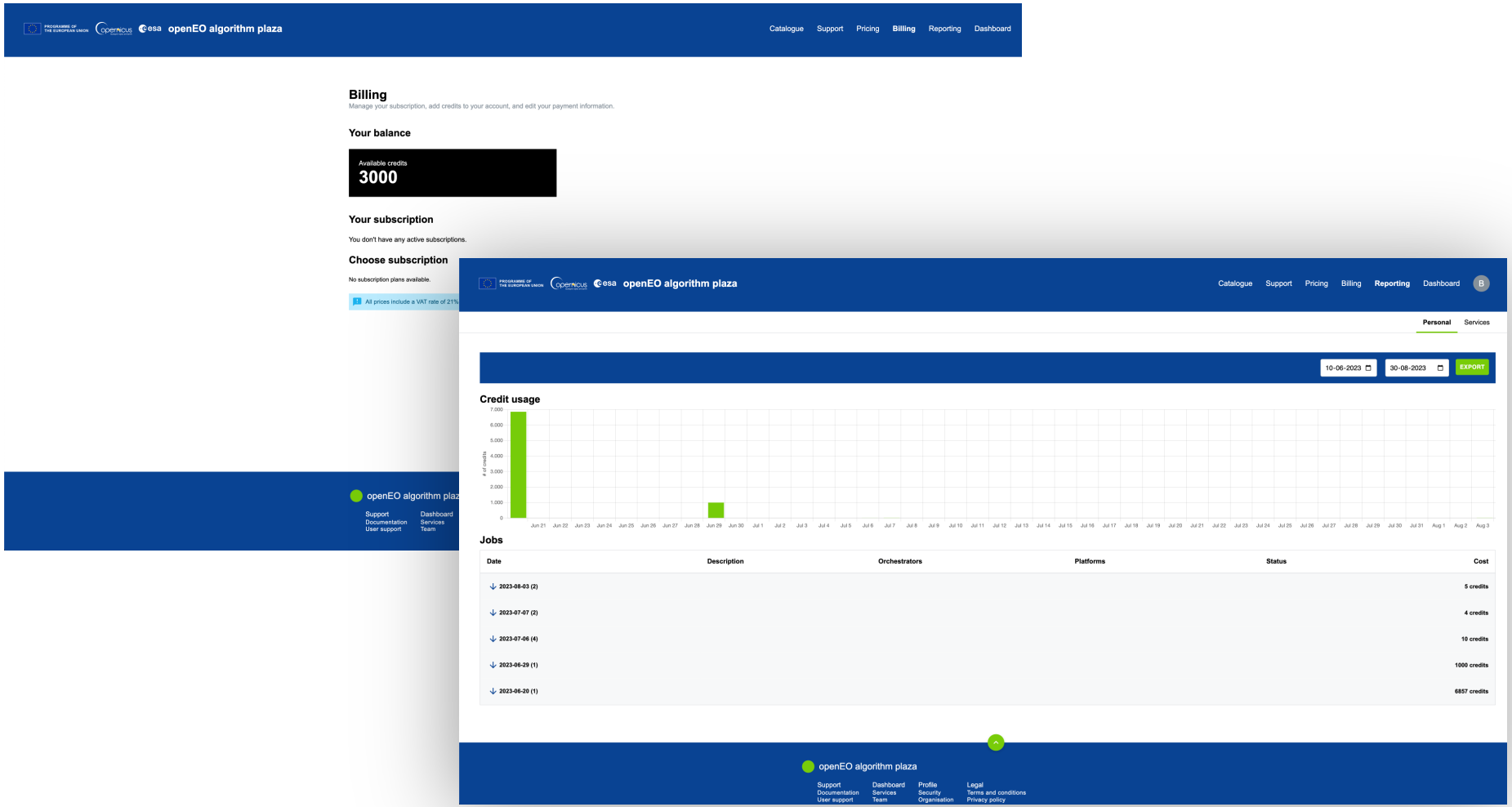
Validated / Verified service

- The service is validated, results & validation report available (optional)
- Basic integration testing
- Comprehensive functional tests
- Different logging levels
- Publications available

Level 4

Operational service

- The service has been shown to be fit for larger scale production and integration in operational systems
- Cost estimation based on large scale testing available



vito SINERGISE VNIVERSITAT DE VALÈNCIA IPL IMAGE PROCESSING LABORATORY

vito.be

openEO platform Credits

esa

openEO Platform

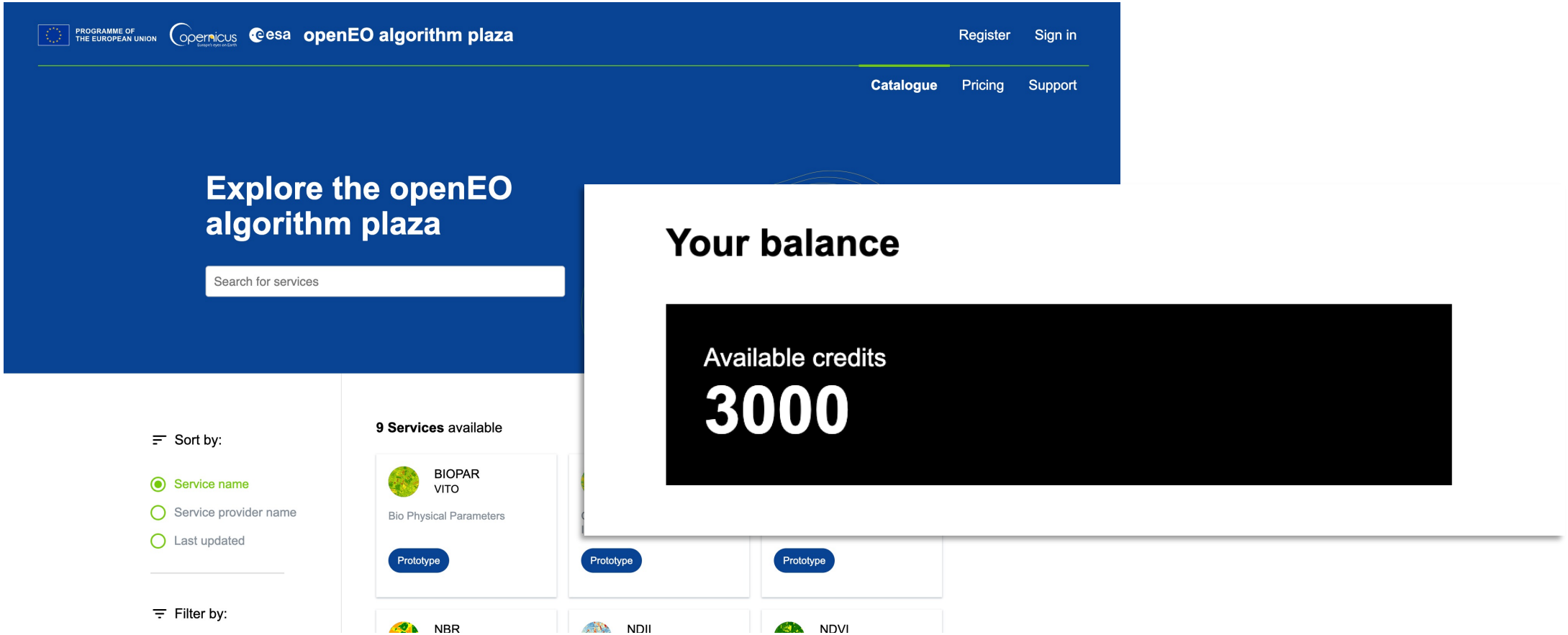
https://docs.openeo.cloud/join/free_trial.html

Free Trial	Network of Resources Sponsoring
free	free
You want to try and "play" with the Platform. You don't have a specific use case in mind and want to see how it works.	You want to use openEO Platform for longer running projects or get specific support from our development team for your workflows. Limited funding (5,000 EUR) for non-ESA projects.
Valid for: 30 days	Valid as per sponsoring request
Register	Apply

vito SINERGISE VNIVERSITAT DE VALÈNCIA IPL IMAGE PROCESSING LABORATORY

vito.be

Copernicus Data Space Ecosystem



https://marketplace-portal.dataspace.copernicus.eu/catalogue

vito

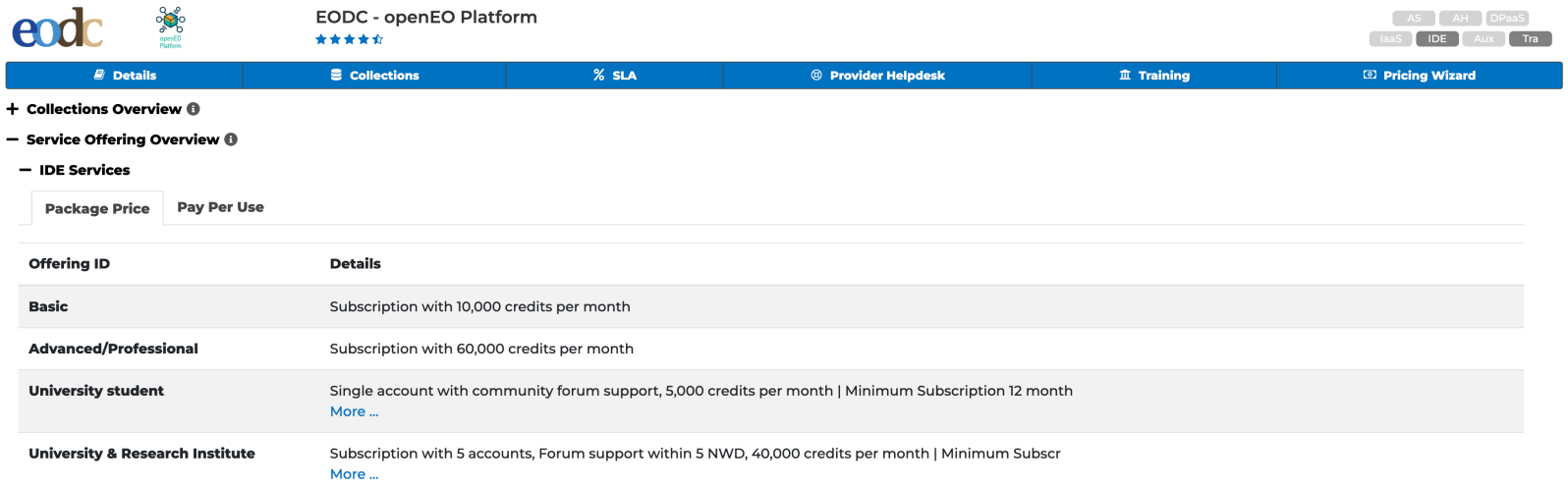
SINERGISE

VNIVERSITAT DE VALÈNCIA

IPL IMAGE PROCESSING LABORATORY

vito.be

Network of Resources (NoR)

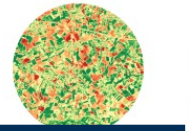


https://nor-discover.cloudeo.group/

vito.be

Service execution of FuseTS

CropSAR



ACCESS SERVICE

Service links:
No service links provided

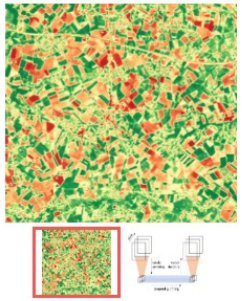
Provided by:
VITO

Service provider links:
No service provider links provided

Incubating

CropSAR 2D

Cloud-free monitoring using Sentinel satellites



Overview

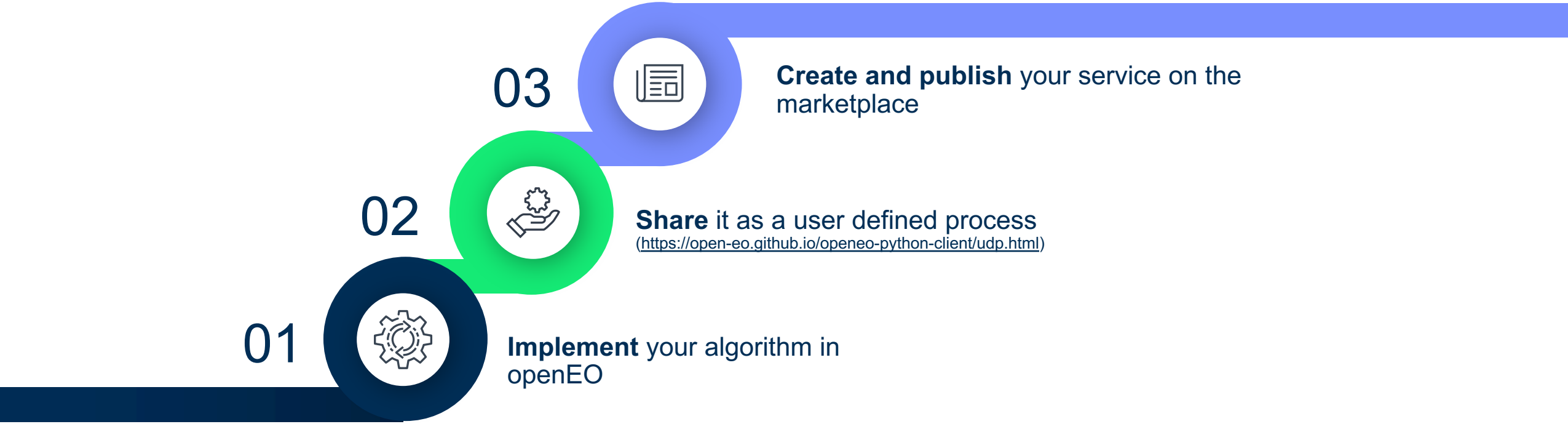
CropSAR is an innovative technique that uses Sentinel-1 radar observations to augment those of Sentinel-2 using advanced AI technology while relying on EO domain expertise. This version of the service works at image level and allows the user to generate consistent and cloud-free image time series of optical outputs at a temporal resolution of 5 days. It therefore differs from the CropSAR 1D service which focuses on field-averaged time series instead of images. The following outputs are supported by this service:

- NDVI
- FAPAR
- FCOVER
- RGB_NIR

MOGPR

```python
import openeo

## Setup of parameters
spat_ext = {
    "type": "Polygon",
    "coordinates": [
        [
            [
                5.170012098271149,
                51.25062964728295
            ],
            [
                5.17085904378298,
                51.24882567194015
            ],
            [
                5.17857421368097,
                51.2468515482926
            ],
            [
                5.178972704726344,
                51.24982704376254
            ],
            [
                5.170012098271149,
                51.25062964728295
            ]
        ]
    ]
}
temp_ext = ["2022-05-01", "2023-07-31"]

## Setup connection to openEO
connection = openeo.connect("openeo.vito.be").authenticate_oidc()
service = 'mogpr'
namespace = 'u:fusets'

## Create a base NDVI datacube that can be used as input for the service
base = connection.load_collection('SENTINEL2_L2A_SENTINELHUB',
                                  spatial_extent=spat_ext,
                                  temporal_extent=temp_ext,
                                  bands=["B04", "B08", "SCL"])
base_cloudmasked = base.process("mask_scl_dilation", data=base, scl_band_name="SCL")
base_ndvi = base_cloudmasked.ndvi(red="B04", nir="B08")

## Create a processing graph from the MOGPR process using an active openEO connection
mogpr = connection.datacube_from_process(
  service,
  namespace=f'https://openeo.vito.be/openeo/1.1/processes/{namespace}/{service}',
  data=base_ndvi
)

## Calculate the average time series value for the given area of interest
mogpr = mogpr.aggregate_spatial(spat_ext, reducer='mean')

# Execute the service as a batch process
mogpr_job = mogpr.execute_batch('./mogpr.json', out_format="json", title=f'FuseTS - MOGPR', job_options=
{   'udf-dependency-archives': [
        'https://artifactory.vgt.vito.be:443/auxdata-public/ai4food/fusets_venv.zip#tmp/venv',
        'https://artifactory.vgt.vito.be:443/auxdata-public/ai4food/fusets.zip#tmp/venv_static'
    ],
    'executor-memory': '7g'
})
```

Create your own services



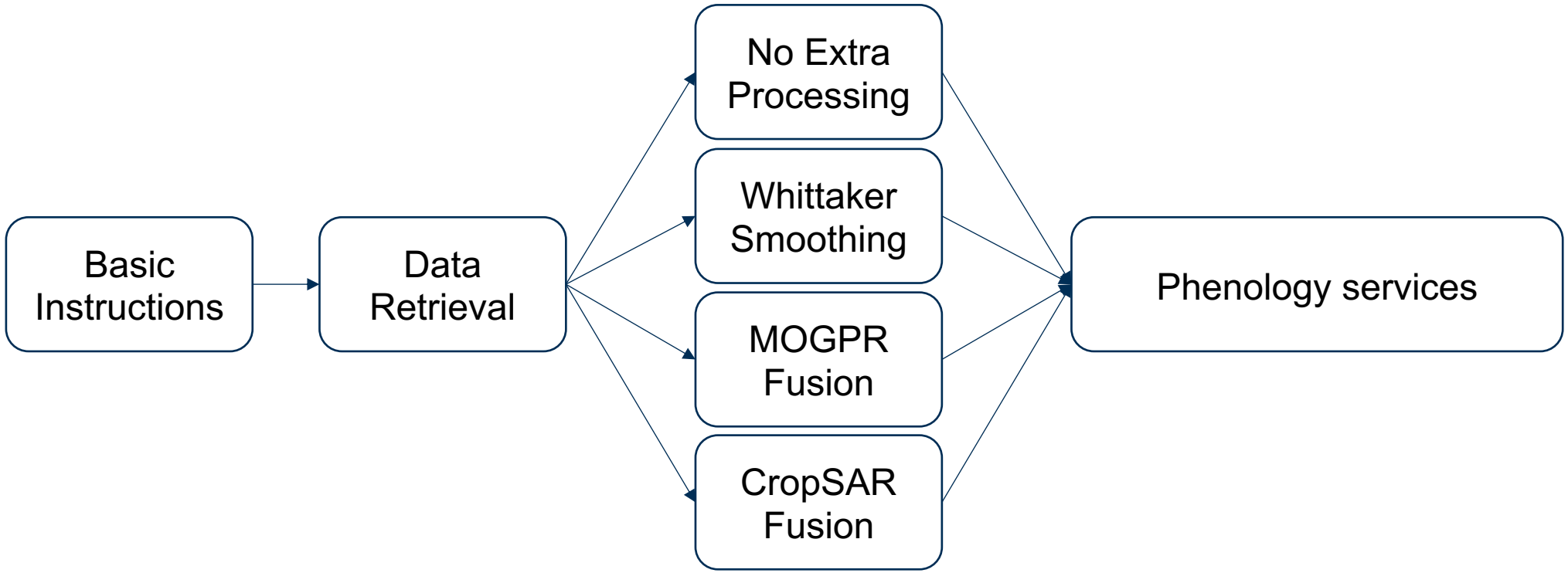
Hands-on Exercise

https://github.com/Open-EO/FuseTS/blob/main/workshops/BIDS/workshop/BiDS_Tutorial_FuseTS.ipynb

vito SINERGISE

VNIVERSITAT DE VALÈNCIA IPL IMAGE PROCESSING LABORATORY

Hands-on

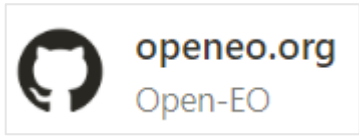

Basic Instructions
Data Retrieval
No Extra Processing
Whittaker Smoothing
MOGPR Fusion
CropSAR Fusion
Phenology services

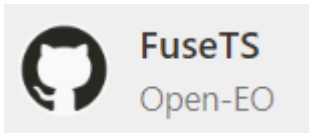
Contribution

- How to contribute?
- Create your own service

FuseTS

1. Familiarize yourself with the library

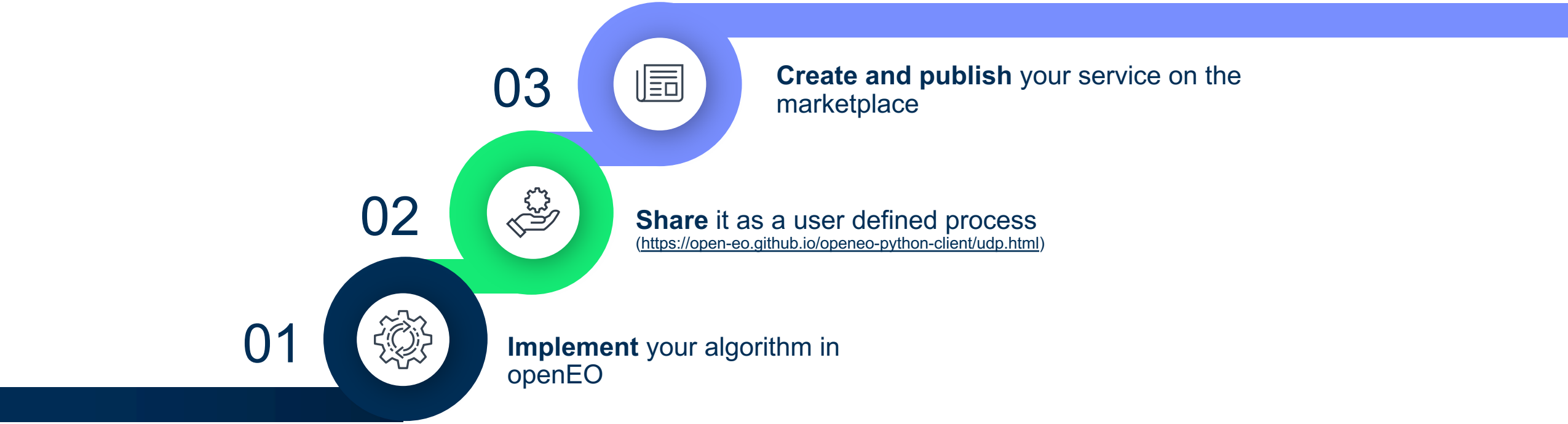




2. Fork the repository and submit a Pull Request

3. Collaborate and share your suggestions/feedback

Create your own services



QUESTIONS?

vito.be

Thank you

vito

vito.be