European Commission

ACRS Conference EOPEN open sessions
Daejeon Convention Center (DCC), Republic of Korea
October 16-17, 2019

EOPEN – A PLATFORM FOR DEVELOPING DISTRIBUTED APPLICATIONS

EOPEN

Bernard Valentin
Space Applications Services
Platform Development and Integration Lead

Overview

▷ Which platform should I use?

▷ Do I really need to choose?

EOPEN Developer Platform

▷ An environment-agnostic platform

▷ Application development life-cycle

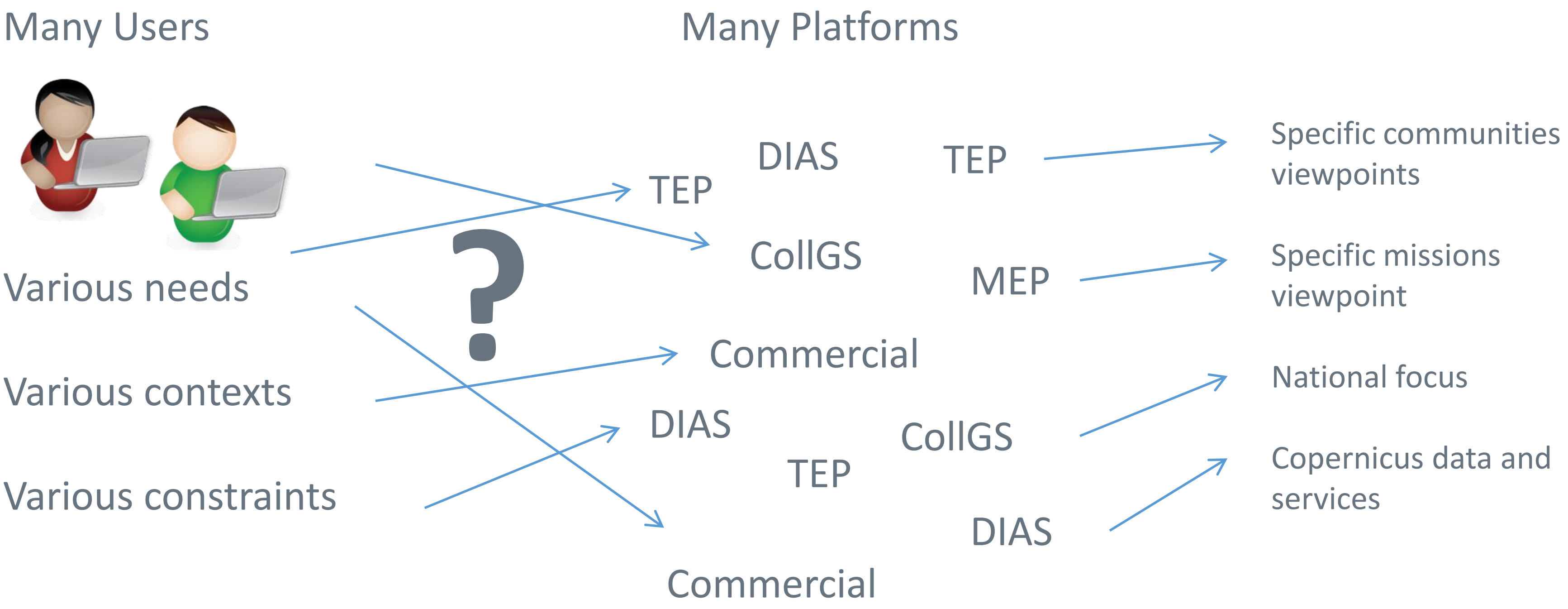
▷ Results publishing and access

EOPEN End-User Portal

▷ Customisable Web portal

▷ Visualisation of geo-temporal products

Which platform should I choose?



Many Users

Various needs

Various contexts

Various constraints

?

Many Platforms

DIAS

TEP

TEP

CollGS

MEP

Commercial

DIAS

CollGS

TEP

DIAS

Commercial

Specific communities viewpoints

Specific missions viewpoint

National focus

Copernicus data and services

Considerations when choosing a target environment

- Which is the best environment for my needs in terms of support, price, performances, data and services, …
- Which environment will best support the development and execution of my applications?
- Procedures to request for cloud resources
- Management interfaces, web portals, documentation, support
- Access to data and exchange of data between the platforms
- Do the chosen technologies allow to migrate between environments easily?

An environment-agnostic platform

- EOPEN helps developing and executing distributed applications

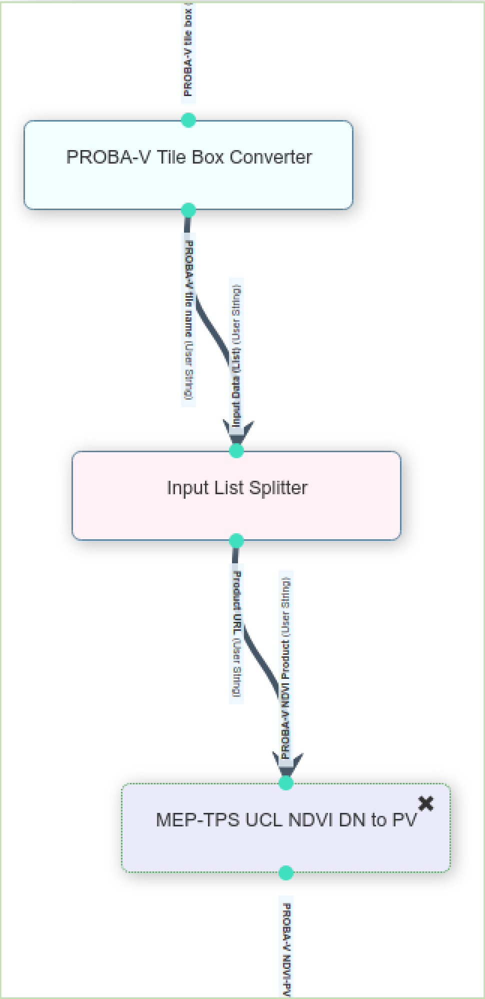
- It supports testing and migrating between the execution environments

It supports the application development cycle:

1. Import algorithm and create a dockerized process
2. Configure the process in a processor workflow
3. Processor selection and execution
4. Performance and results analysis
5. Processor sharing / publishing as an application

Main Technical Concepts

- A **Processor** is an application defined by a **Workflow** of **Processes**.
  - The Platform includes a graphical **Workflow Editor** for creating workflows interactively.
  - Processors can be executed on-demand, scheduled or externally triggered.
  - **Parameterization forms** are automatically generated.
  - Interfaces are included for monitoring and control, reporting and data access.
- A **Process** is a unit of execution.
  - A Process is implemented as a Dockerised **Algorithm**.
  - A Process has (typed) input and output parameters.
  - The Platform includes a tool for that automates the packaging and the registration of custom processes.

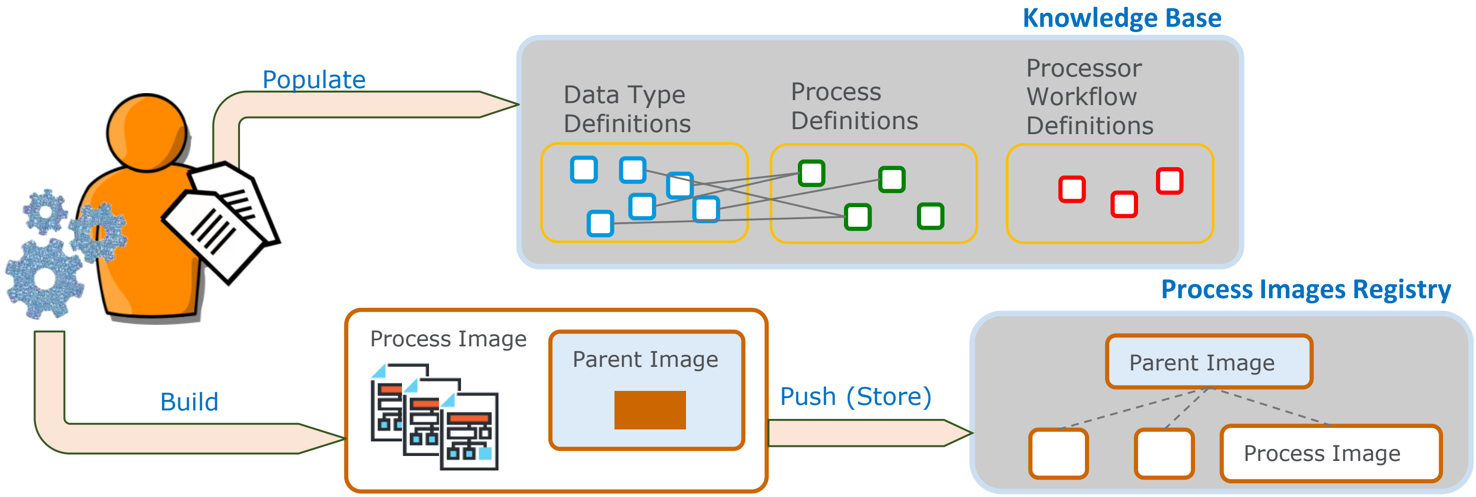

PROBA-V tile box
PROBA-V Tile Box Converter
PROBA-V tile name (User String)
Input Data (List) (User String)
Input List Splitter
Product URL (User String)
PROBA-V NDVI Product (User String)
MEP-TPS UCL NDVI DN to PV
✖
PROBA-V NDVI-PV

Development Cycle Step 1

The algorithm import mechanism

A user has a processing chain description and the algorithm files (artifacts) that implement the different operations of the chain

- The processing chain description is used to add knowledge into the framework: data types, parameters, processes and workflow definitions
- The algorithm files are packaged within Docker container images and stored in a Docker registry

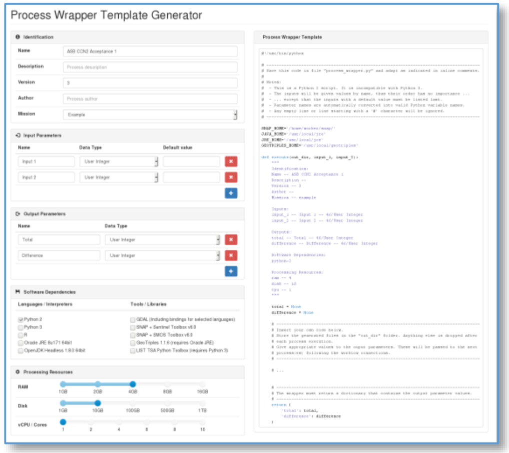


Development Cycle Step 1

The algorithm import mechanism

Import algorithm and create a dockerized process

- Ingest algorithms together with required libraries
- Programming language agnostic
- Automatically build and push Docker images, register in the system



Process Wrapper Template Generator

Identification

Name: ASB CCN2 Acceptance 1
Description: Process description
Version: 3
Author: Process author
Mission: Example

Input Parameters

Name	Data Type	Default value
Input 1	User Integer	
Input 2	User Integer	

Output Parameters

Name	Data Type
Total	User Integer
Difference	User Integer

Software Dependencies

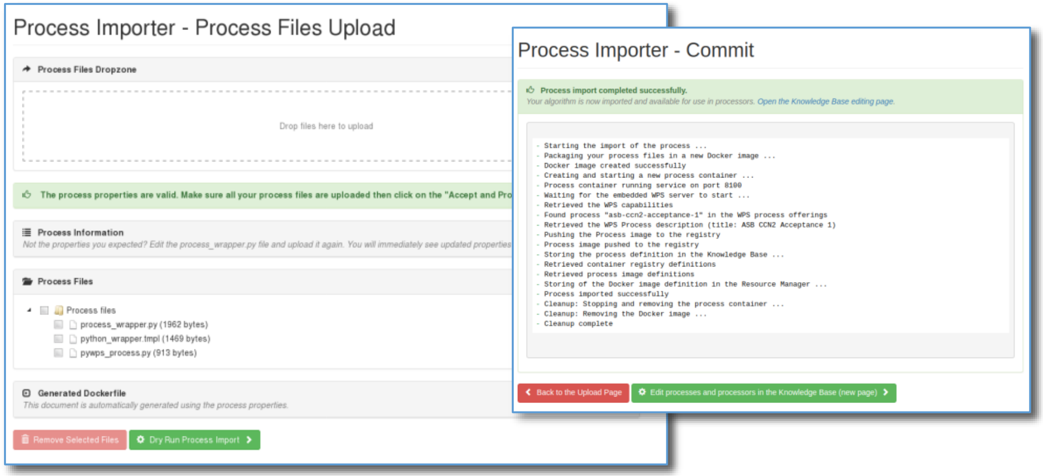
Languages / Interpreters	Tools / Libraries
☑ Python 2	□ GDAL (Including bindings for selected languages)
□ Python 3	□ SNAP + Sentinel Toolbox v6.0
□ R	□ SNAP + SMOS Toolbox v6.0
□ Oracle JRE 8u171 64bit	□ GeoTriples 1.1.6 (requires Oracle JRE)
□ OpenJDK Headless 1.8.0 64bit	□ LIST TSA Python Toolbox (requires Python 3)

Processing Resources

RAM: 4GB
Disk: 10GB
vCPU / Cores: 1

Process Wrapper Template

```
#!/usr/bin/python

# ------------------------------------------------------------------------------
# Save this code in file "process_wrapper.py" and adapt as indicated in inline comments.
#
# Notes:
#  - This is a Python 2 script. It is incompatible with Python 3.
#  - The inputs will be given values by name, thus their order has no importance ...
#  - ... except that the inputs with a default value must be listed last.
#  - Parameter names are automatically converted into valid Python variable names.
#  - Any empty line or line starting with a '#' character will be ignored.
# ------------------------------------------------------------------------------

SNAP_HOME='/home/worker/snap/'
JAVA_HOME='/usr/local/jre'
JRE_HOME='/usr/local/jre'
GEOTRIPLES_HOME='/usr/local/geotriples'

def execute(out_dir, input_1, input_2):
    """
    Identification:
    Name -- ASB CCN2 Acceptance 1
    Description --
    Version -- 3
    Author --
    Mission -- example

    Inputs:
    input_1 -- Input 1 -- 44/User Integer
    input_2 -- Input 2 -- 44/User Integer

    Outputs:
    total -- Total -- 44/User Integer
    difference -- Difference -- 44/User Integer

    Software Dependencies:
    python-2

    Processing Resources:
    ram -- 4
    disk -- 10
    cpu -- 1
    """

    total = None
    difference = None

    # --------------------------------------------------------------------------
    # Insert your own code below.
    # Store the generated files in the "out_dir" folder. Anything else is dropped after
    # each process execution.
    # Give appropriate values to the ouput parameters. These will be passed to the next
    # process(es) following the workflow connections.
    # --------------------------------------------------------------------------

    # ...


    # --------------------------------------------------------------------------
    # The wrapper must return a dictionary that contains the output parameter values.
    # --------------------------------------------------------------------------
    return {
        'total': total,
        'difference': difference
    }
```



Process Importer - Process Files Upload

Process Files Dropzone

Drop files here to upload

The process properties are valid. Make sure all your process files are uploaded then click on the "Accept and Pro

Process Information

Not the properties you expected? Edit the process_wrapper.py file and upload it again. You will immediately see updated properties

Process Files

- □ Process files
  - □ process_wrapper.py (1962 bytes)
  - □ python_wrapper.tmpl (1469 bytes)
  - □ pywps_process.py (913 bytes)

Generated Dockerfile

This document is automatically generated using the process properties.

Process Importer - Commit

Process import completed successfully.

Your algorithm is now imported and available for use in processors. Open the Knowledge Base editing page.

```
- Starting the import of the process ...
- Packaging your process files in a new Docker image ...
- Docker image created successfully
- Creating and starting a new process container ...
- Process container running service on port 8100
- Waiting for the embedded WPS server to start ...
- Retrieved the WPS capabilities
- Found process "asb-ccn2-acceptance-1" in the WPS process offerings
- Retrieved the WPS Process description (title: ASB CCN2 Acceptance 1)
- Pushing the Process image to the registry
- Process image pushed to the registry
- Storing the process definition in the Knowledge Base ...
- Retrieved container registry definitions
- Retrieved process image definitions
- Storing of the Docker image definition in the Resource Manager ...
- Process imported successfully
- Cleanup: Stopping and removing the process container ...
- Cleanup: Removing the Docker image ...
- Cleanup complete
```

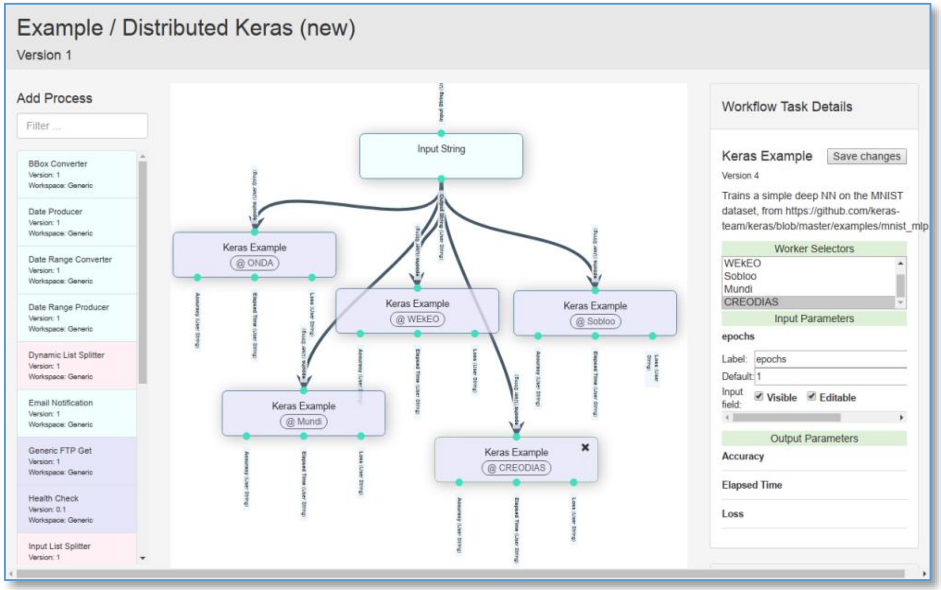
Back to the Upload Page
Edit processes and processors in the Knowledge Base (new page)

Development Cycle Step 2

Processor Workflow Configuration

Configure the process in processor workflow

- Graphically build processor workflows with built-in consistency checking of the dataflow between processes
- Customise input parameters (label, default values, etc.)
- Select target execution platform (DIAS, HPC, …) and required environment capabilities (GPU, data availability)



Example / Distributed Keras (new)

Version 1

Add Process

Filter ...

BBox Converter
Version: 1
Workspace: Generic

Date Producer
Version: 1
Workspace: Generic

Date Range Converter
Version: 1
Workspace: Generic

Date Range Producer
Version: 1
Workspace: Generic

Dynamic List Splitter
Version: 1
Workspace: Generic

Email Notification
Version: 1
Workspace: Generic

Generic FTP Get
Version: 1
Workspace: Generic

Health Check
Version: 0.1
Workspace: Generic

Input List Splitter
Version: 1

Workflow Task Details

Keras Example

Version 4

Trains a simple deep NN on the MNIST dataset, from https://github.com/keras-team/keras/blob/master/examples/mnist_mlp

Worker Selectors

WEkEO
Sobloo
Mundi
CREODIAS

Input Parameters

epochs

Label: epochs

Default: 1

Input field: ☑ Visible ☑ Editable

Output Parameters

Accuracy

Elapsed Time

Loss

Development Cycle Step 3

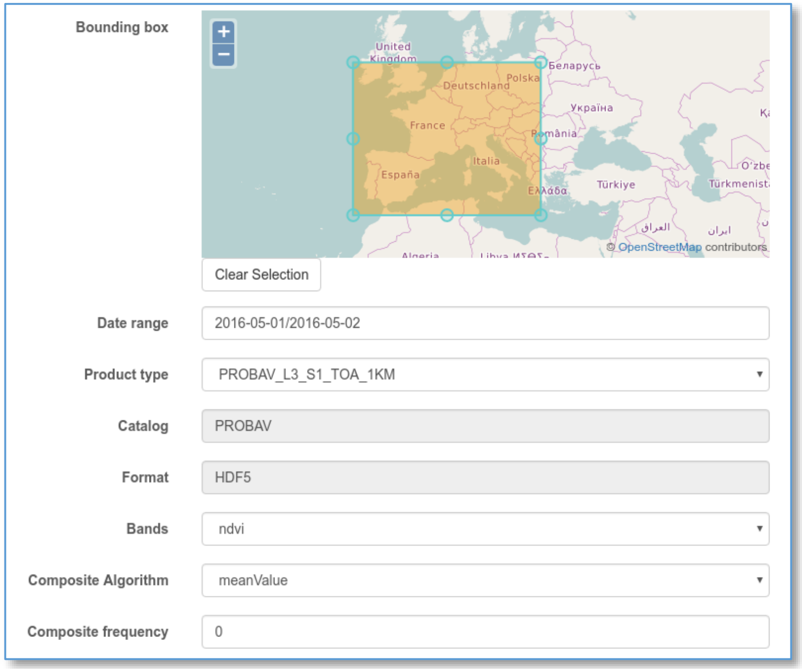
Processor Selection and Execution

Processor selection and execution

- Dynamically generated parameterisation forms based on unresolved inputs
- Input controls depend on parameter type (bbox, date, number, etc.)

Scheduling workflow executions

- At a given future date and time
- Scheduling regular executions at a given interval or month day(s)



Bounding box	
Date range	2016-05-01/2016-05-02
Product type	PROBAV_L3_S1_TOA_1KM
Catalog	PROBAV
Format	HDF5
Bands	ndvi
Composite Algorithm	meanValue
Composite frequency	0

Development Cycle Step 4

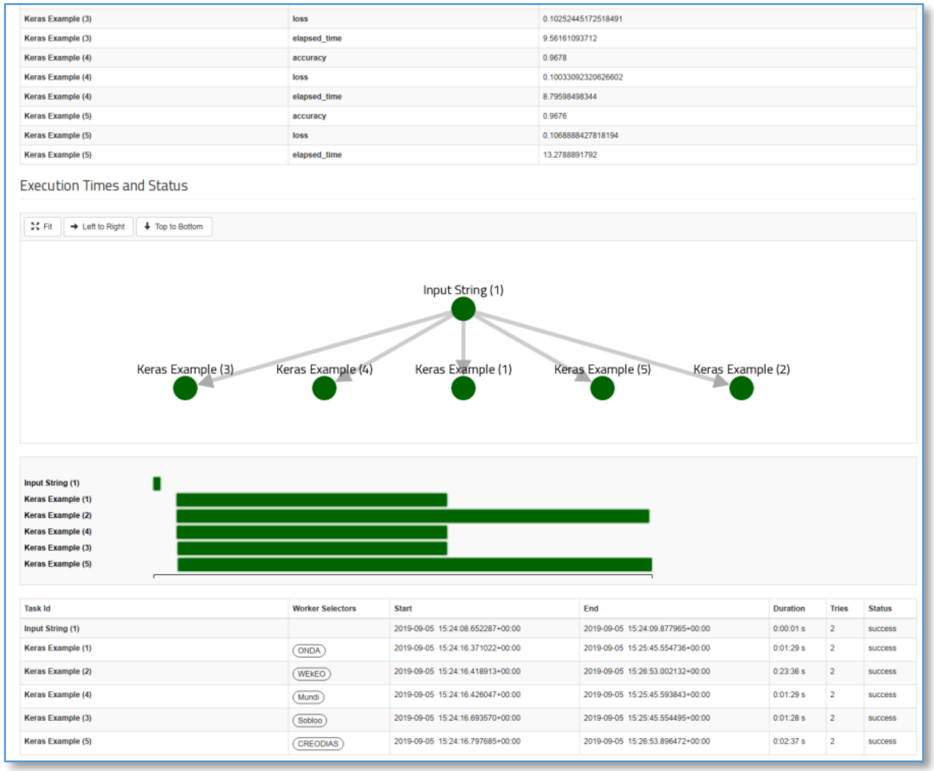
Performance and results analysis

- Each process in a workflow is deployed and executed on any available environment by the container orchestration tool

  or

- is deployed and executed in the environment selected by the user

➢ True global cross-environment orchestration



Keras Example (3)	loss	0.10252445172518491
Keras Example (3)	elapsed_time	9.56161093712
Keras Example (4)	accuracy	0.9678
Keras Example (4)	loss	0.10033092320626602
Keras Example (4)	elapsed_time	8.79598498344
Keras Example (5)	accuracy	0.9676
Keras Example (5)	loss	0.1068888427818194
Keras Example (5)	elapsed_time	13.2788891792

Execution Times and Status

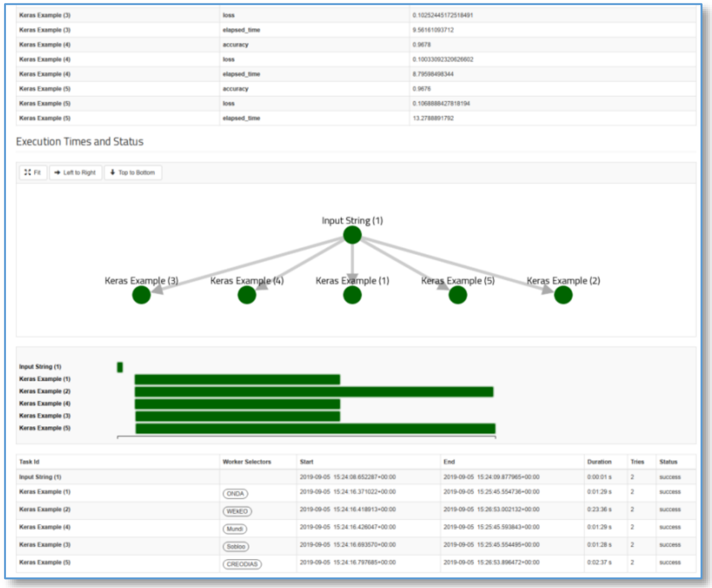
Task Id	Worker Selectors	Start	End	Duration	Tries	Status
Input String (1)		2019-09-05 15:24:08.652287+00:00	2019-09-05 15:24:09.877965+00:00	0:00:01 s	2	success
Keras Example (1)	ONDA	2019-09-05 15:24:16.371022+00:00	2019-09-05 15:25:45.554736+00:00	0:01:29 s	2	success
Keras Example (2)	WEkEO	2019-09-05 15:24:16.418913+00:00	2019-09-05 15:26:53.002132+00:00	0:23:36 s	2	success
Keras Example (4)	Mundi	2019-09-05 15:24:16.426047+00:00	2019-09-05 15:25:45.593843+00:00	0:01:29 s	2	success
Keras Example (3)	Sobloo	2019-09-05 15:24:16.693570+00:00	2019-09-05 15:25:45.554495+00:00	0:01:28 s	2	success
Keras Example (5)	CREODIAS	2019-09-05 15:24:16.797685+00:00	2019-09-05 15:26:53.896472+00:00	0:02:37 s	2	success

Workflow execution report (fragment)

Development Cycle Step 4

Access to generated outputs

- Execution reports link to Web-based Datastore folders
- Click to visualise in the browser or download locally

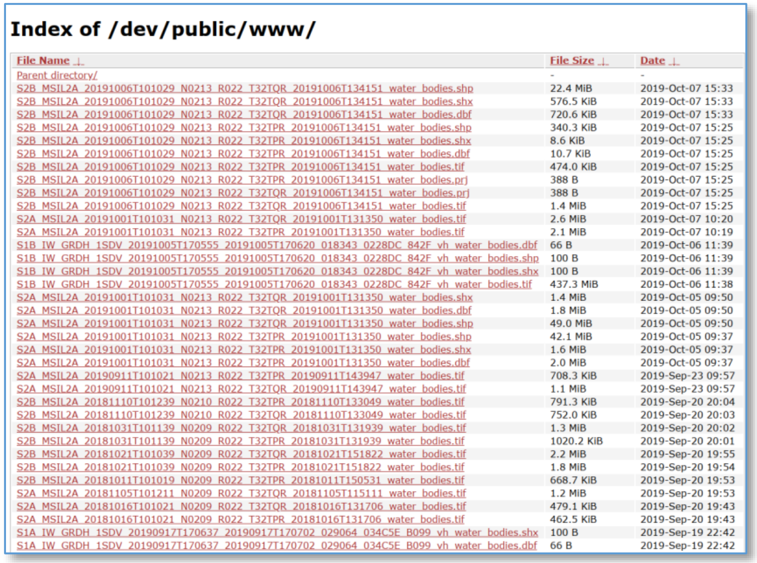


Keras Example (3)	loss	0.10252445172518491
Keras Example (3)	elapsed_time	9.56161093712
Keras Example (4)	accuracy	0.9678
Keras Example (4)	loss	0.10033092320626602
Keras Example (4)	elapsed_time	8.79598498344
Keras Example (5)	accuracy	0.9676
Keras Example (5)	loss	0.1068888427818194
Keras Example (5)	elapsed_time	13.2788891792

Execution Times and Status

Task Id	Worker Selectors	Start	End	Duration	Tries	Status
Input String (1)		2019-09-05 15:24:08.652287+00:00	2019-09-05 15:24:09.877965+00:00	0:00:01 s	2	success
Keras Example (1)	ONDA	2019-09-05 15:24:16.371022+00:00	2019-09-05 15:25:45.554736+00:00	0:01:29 s	2	success
Keras Example (2)	WEkEO	2019-09-05 15:24:16.418913+00:00	2019-09-05 15:26:53.002132+00:00	0:23:36 s	2	success
Keras Example (4)	Mundi	2019-09-05 15:24:16.426047+00:00	2019-09-05 15:25:45.593843+00:00	0:01:29 s	2	success
Keras Example (3)	Sobloo	2019-09-05 15:24:16.693570+00:00	2019-09-05 15:25:45.554495+00:00	0:01:28 s	2	success
Keras Example (5)	CREODIAS	2019-09-05 15:24:16.797685+00:00	2019-09-05 15:26:53.896472+00:00	0:02:37 s	2	success

Workflow execution report (fragment)



Index of /dev/public/www/

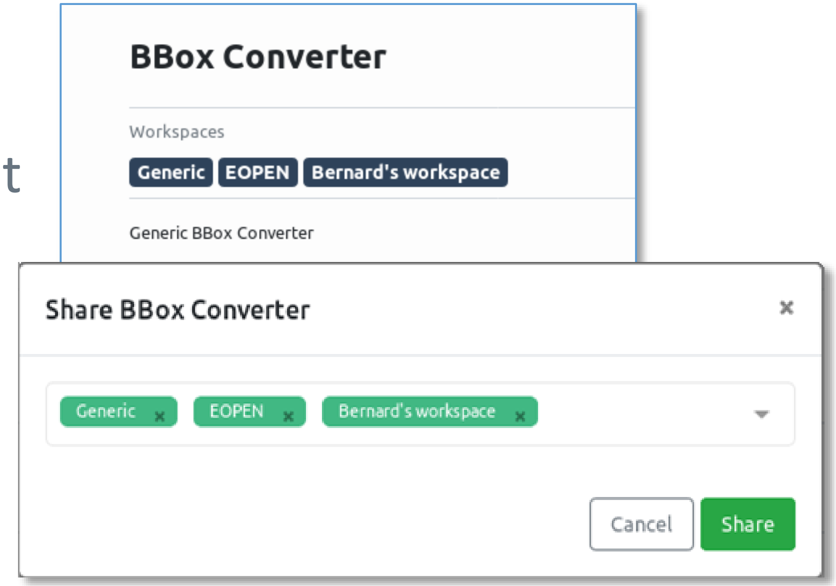
File Name ↓	File Size ↓	Date ↓
Parent directory/	-	-
S2B_MSIL2A_20191006T101029_N0213_R022_T32TQR_20191006T134151_water_bodies.shp	22.4 MiB	2019-Oct-07 15:33
S2B_MSIL2A_20191006T101029_N0213_R022_T32TQR_20191006T134151_water_bodies.shx	576.5 KiB	2019-Oct-07 15:33
S2B_MSIL2A_20191006T101029_N0213_R022_T32TQR_20191006T134151_water_bodies.dbf	720.6 KiB	2019-Oct-07 15:33
S2B_MSIL2A_20191006T101029_N0213_R022_T32TPR_20191006T134151_water_bodies.shp	340.3 KiB	2019-Oct-07 15:25
S2B_MSIL2A_20191006T101029_N0213_R022_T32TPR_20191006T134151_water_bodies.shx	8.6 KiB	2019-Oct-07 15:25
S2B_MSIL2A_20191006T101029_N0213_R022_T32TPR_20191006T134151_water_bodies.dbf	10.7 KiB	2019-Oct-07 15:25
S2B_MSIL2A_20191006T101029_N0213_R022_T32TPR_20191006T134151_water_bodies.tif	474.0 KiB	2019-Oct-07 15:25
S2B_MSIL2A_20191006T101029_N0213_R022_T32TPR_20191006T134151_water_bodies.prj	388 B	2019-Oct-07 15:25
S2B_MSIL2A_20191006T101029_N0213_R022_T32TQR_20191006T134151_water_bodies.prj	388 B	2019-Oct-07 15:25
S2B_MSIL2A_20191006T101029_N0213_R022_T32TQR_20191006T134151_water_bodies.tif	1.4 MiB	2019-Oct-07 15:25
S2A_MSIL2A_20191001T101031_N0213_R022_T32TQR_20191001T131350_water_bodies.tif	2.6 MiB	2019-Oct-07 10:20
S2A_MSIL2A_20191001T101031_N0213_R022_T32TPR_20191001T131350_water_bodies.tif	2.1 MiB	2019-Oct-07 10:19
S1B_IW_GRDH_1SDV_20191005T170555_20191005T170620_018343_0228DC_842F_vh_water_bodies.dbf	66 B	2019-Oct-06 11:39
S1B_IW_GRDH_1SDV_20191005T170555_20191005T170620_018343_0228DC_842F_vh_water_bodies.shp	100 B	2019-Oct-06 11:39
S1B_IW_GRDH_1SDV_20191005T170555_20191005T170620_018343_0228DC_842F_vh_water_bodies.shx	100 B	2019-Oct-06 11:39
S1B_IW_GRDH_1SDV_20191005T170555_20191005T170620_018343_0228DC_842F_vh_water_bodies.tif	437.3 MiB	2019-Oct-06 11:38
S2A_MSIL2A_20191001T101031_N0213_R022_T32TQR_20191001T131350_water_bodies.shx	1.4 MiB	2019-Oct-05 09:50
S2A_MSIL2A_20191001T101031_N0213_R022_T32TQR_20191001T131350_water_bodies.dbf	1.8 MiB	2019-Oct-05 09:50
S2A_MSIL2A_20191001T101031_N0213_R022_T32TQR_20191001T131350_water_bodies.shp	49.0 MiB	2019-Oct-05 09:50
S2A_MSIL2A_20191001T101031_N0213_R022_T32TPR_20191001T131350_water_bodies.shp	42.1 MiB	2019-Oct-05 09:37
S2A_MSIL2A_20191001T101031_N0213_R022_T32TPR_20191001T131350_water_bodies.shx	1.6 MiB	2019-Oct-05 09:37
S2A_MSIL2A_20191001T101031_N0213_R022_T32TPR_20191001T131350_water_bodies.dbf	2.0 MiB	2019-Oct-05 09:37
S2A_MSIL2A_20190911T101021_N0213_R022_T32TPR_20190911T143947_water_bodies.tif	708.3 KiB	2019-Sep-23 09:57
S2A_MSIL2A_20190911T101021_N0213_R022_T32TQR_20190911T143947_water_bodies.tif	1.1 MiB	2019-Sep-23 09:57
S2B_MSIL2A_20181110T101239_N0210_R022_T32TPR_20181110T133049_water_bodies.tif	791.3 KiB	2019-Sep-20 20:04
S2B_MSIL2A_20181110T101239_N0210_R022_T32TQR_20181110T133049_water_bodies.tif	752.0 KiB	2019-Sep-20 20:03
S2B_MSIL2A_20181031T101139_N0209_R022_T32TQR_20181031T131939_water_bodies.tif	1.3 MiB	2019-Sep-20 20:02
S2B_MSIL2A_20181031T101139_N0209_R022_T32TPR_20181031T131939_water_bodies.tif	1020.2 KiB	2019-Sep-20 20:01
S2B_MSIL2A_20181021T101039_N0209_R022_T32TQR_20181021T151822_water_bodies.tif	2.2 MiB	2019-Sep-20 19:55
S2B_MSIL2A_20181021T101039_N0209_R022_T32TPR_20181021T151822_water_bodies.tif	1.8 MiB	2019-Sep-20 19:54
S2B_MSIL2A_20181011T101019_N0209_R022_T32TPR_20181011T150531_water_bodies.tif	668.7 KiB	2019-Sep-20 19:53
S2A_MSIL2A_20181105T101211_N0209_R022_T32TQR_20181105T115111_water_bodies.tif	1.2 MiB	2019-Sep-20 19:53
S2A_MSIL2A_20181016T101021_N0209_R022_T32TQR_20181016T131706_water_bodies.tif	479.1 KiB	2019-Sep-20 19:43
S2A_MSIL2A_20181016T101021_N0209_R022_T32TPR_20181016T131706_water_bodies.tif	462.5 KiB	2019-Sep-20 19:43
S1A_IW_GRDH_1SDV_20190917T170637_20190917T170702_029064_034C5E_B099_vh_water_bodies.shx	100 B	2019-Sep-19 22:42
S1A_IW_GRDH_1SDV_20190917T170637_20190917T170702_029064_034C5E_B099_vh_water_bodies.dbf	66 B	2019-Sep-19 22:42

Datastore folder

Development Cycle Step 5

Process / Processor sharing

- The EOPEN Platform has the concept of workspaces
  - By default each user is given a personal workspace
  - Workspaces can have multiple users
  - For example, projects or teams can be assigned a workspace



BBox Converter

Workspaces

Generic EOPEN Bernard's workspace

Generic BBox Converter

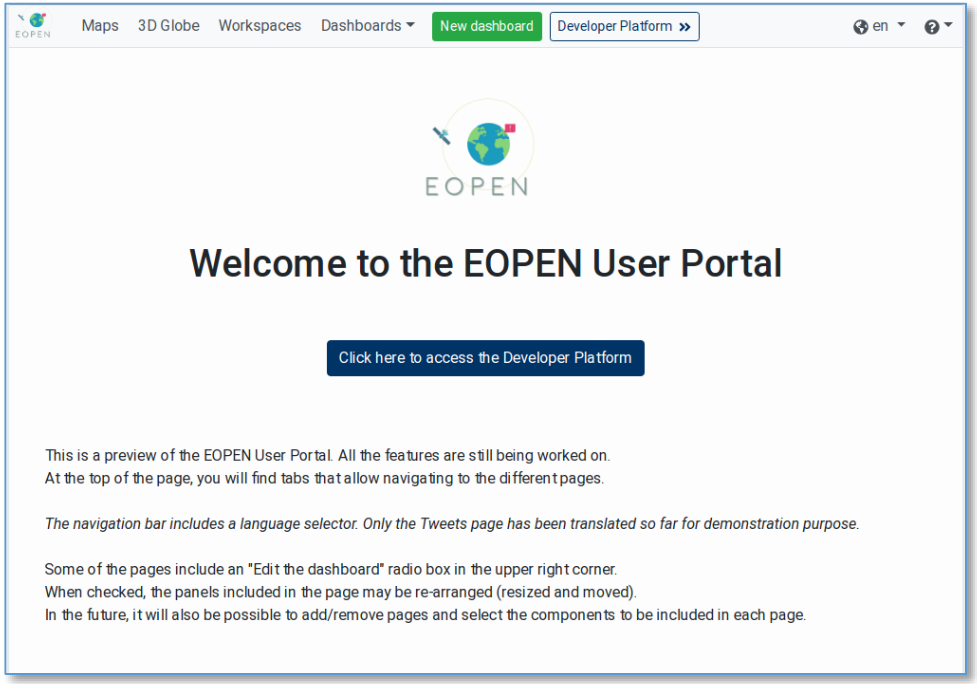
Share BBox Converter

Generic × EOPEN × Bernard's workspace ×

- Workspaces allow users to share algorithms, workflows, execution reports, generated outputs, etc.

➢ The possibility to share creates a collaborative co-development solution

EOPEN End-User Portal



Welcome to the EOPEN User Portal

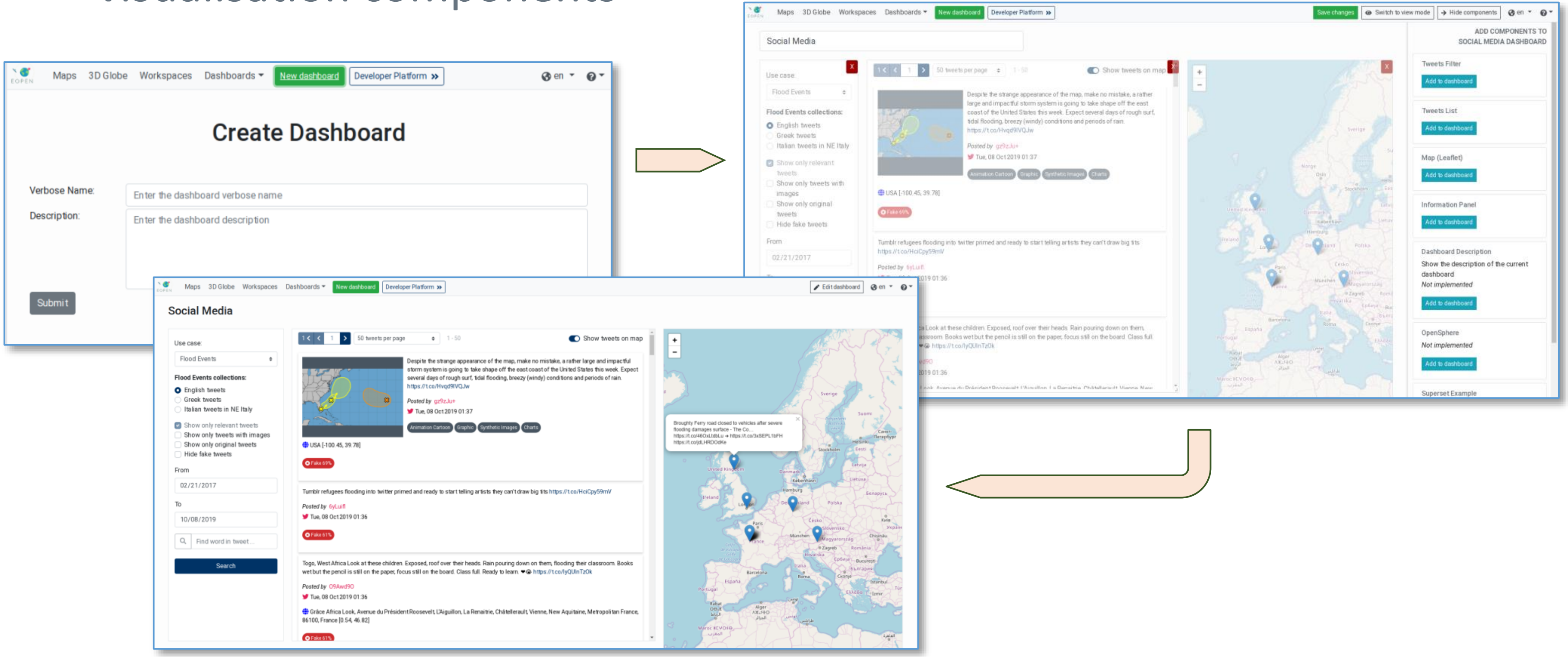
This is a preview of the EOPEN User Portal. All the features are still being worked on.
At the top of the page, you will find tabs that allow navigating to the different pages.

The navigation bar includes a language selector. Only the Tweets page has been translated so far for demonstration purpose.

Some of the pages include an "Edit the dashboard" radio box in the upper right corner.
When checked, the panels included in the page may be re-arranged (resized and moved).
In the future, it will also be possible to add/remove pages and select the components to be included in each page.

A Customisable Web-Portal

- New dashboard pages can be created and populated with available visualisation components



Create Dashboard

Verbose Name: Enter the dashboard verbose name

Description: Enter the dashboard description

Submit

ADD COMPONENTS TO SOCIAL MEDIA DASHBOARD

Tweets Filter — Add to dashboard

Tweets List — Add to dashboard

Map (Leaflet) — Add to dashboard

Information Panel — Add to dashboard

Dashboard Description — Show the description of the current dashboard *Not implemented* — Add to dashboard

OpenSphere — *Not implemented* — Add to dashboard

Superset Example

Social Media

Use case: Flood Events

Flood Events collections:

● English tweets
○ Greek tweets
○ Italian tweets in NE Italy

☑ Show only relevant tweets
□ Show only tweets with images
□ Show only original tweets
□ Hide fake tweets

From: 02/21/2017

To: 10/08/2019

Find word in tweet ...

Search

50 tweets per page 1 - 50 Show tweets on map

Despite the strange appearance of the map, make no mistake, a rather large and impactful storm system is going to take shape off the east coast of the United States this week. Expect several days of rough surf, tidal flooding, breezy (windy) conditions and periods of rain. https://t.co/Hvqd9lVQJw

Posted by gz9zJu+
Tue, 08 Oct 2019 01:37

Animation Cartoon · Graphic · Synthetic Images · Charts

USA [-100.45, 39.78]

Fake 69%

Tumblr refugees flooding into twitter primed and ready to start telling artists they can't draw big tits https://t.co/HciCpy59mV

Posted by 6yLuifl
Tue, 08 Oct 2019 01:36

Fake 61%

Togo, West Africa Look at these children. Exposed, roof over their heads. Rain pouring down on them, flooding their classroom. Books wet but the pencil is still on the paper, focus still on the board. Class full. Ready to learn. ❤ https://t.co/lyQUlnTzOk

Posted by O9Awd9O
Tue, 08 Oct 2019 01:36

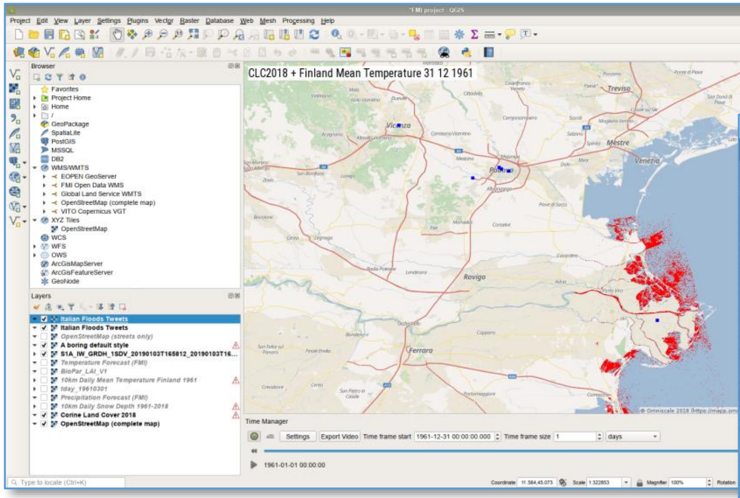
Grâce Africa Look, Avenue du Président Roosevelt, L'Aiguillon, La Renaitrie, Châtellerault, Vienne, New Aquitaine, Metropolitan France, 86100, France [0.54, 46.82]

Fake 61%

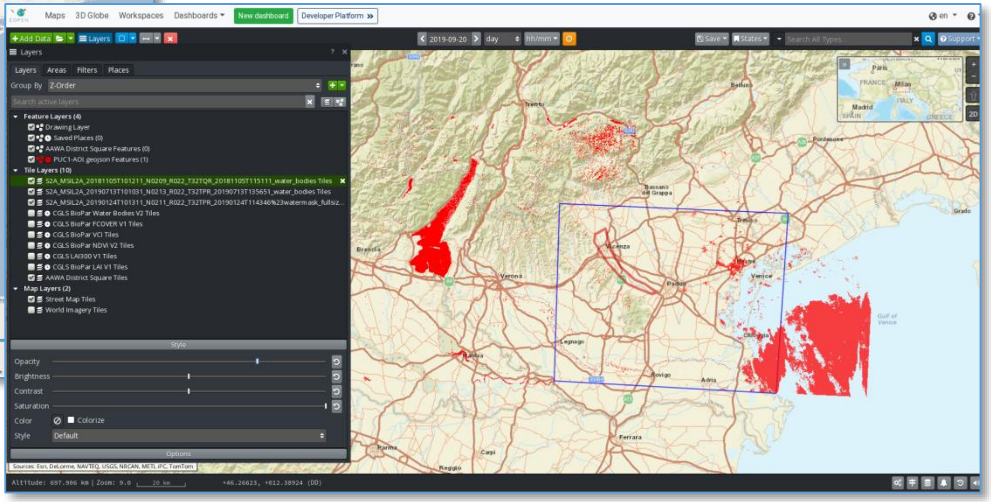
Broughty Ferry road closed to vehicles after severe flooding damages surface - The Co... https://t.co/46OxLtdbLu → https://t.co/3xSEPL1bFH https://t.co/jdLHRDOdKe

Visualisation of generated outputs – GIS Clients

- Geo-temporal data may be published in GIS Server (e.g. GeoServer)
- Visualisation in OGC compliant GIS Client (e.g. QGIS, OpenSphere)



QGIS (Desktop Client)


Maps
3D Globe
Workspaces
Dashboards
New dashboard
Developer Platform »
+ Add Data
Layers
Layers Areas Filters Places
Group By Z-Order
Search active layers
Feature Layers (4)
Drawing Layer
Saved Places (0)
AAWA District Square Features (0)
PUC1-AOI.geojson Features (1)
Tile Layers (10)
S2A_MSIL2A_20181105T101211_N0209_R022_T32TQR_20181105T115111_water_bodies Tiles
S2A_MSIL2A_20190713T101031_N0213_R022_T32TPR_20190713T135651_water_bodies Tiles
S2A_MSIL2A_20190124T101311_N0211_R022_T32TPR_20190124T114346%23watermask_fullsiz...
CGLS BioPar Water Bodies V2 Tiles
CGLS BioPar FCOVER V1 Tiles
CGLS BioPar VCI Tiles
CGLS BioPar NDVI V2 Tiles
CGLS LAI300 V1 Tiles
CGLS BioPar LAI V1 Tiles
AAWA District Square Tiles
Map Layers (2)
Street Map Tiles
World Imagery Tiles
Style
Opacity
Brightness
Contrast
Saturation
Color
Colorize
Style Default
Options
Sources: Esri, DeLorme, NAVTEQ, USGS, NRCAN, METI, iPC, TomTom
Altitude: 697.906 km | Zoom: 9.0 20 km +46.26623, +012.38924 (DD)
2019-09-20
day
Save
States
Search All Types
Support
Brescia
Verona
Vicenza
Padua
Venice
Rovigo
Ferrara

OpenSphere (Web Client)

Any questions?

Join us and become a user.

Contact:
Bernard.Valentin@spaceapplications.com