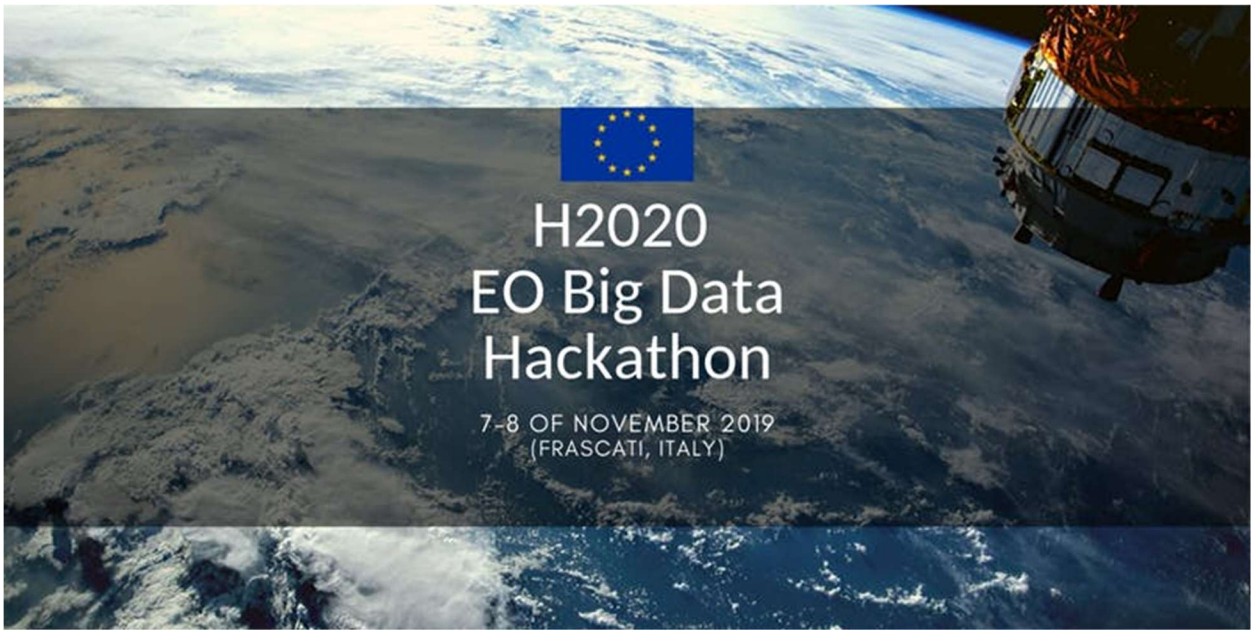


Programme

DAY 1 (9:00 - 17:30)

- 9:00-9:30 Welcome & Introductions (Local Hosts, Massimo Ciscato, Project Officer)
- 9:30-10:30 Project Introductions & Session Organisation
    - 5 projects: 10 minute overview of general project objectives, technology and results; and planned Hackathon exercise
    - Session Attendance intentions (sli.do poll) and Scribe Assignment

10:30-11:00 Coffee break & Villa tour

- 11:00-13:00 Parallel Sessions Day 1
    - Session A: PerceptiveSentinel & openEO
    - Session B: EOPEN

13:00-13:50 Lunch Break

- 13:50-14:50 Parallel Sessions Day 1 (continued)
    - Session A: PerceptiveSentinel & openEO
    - Session B: EOPEN

14:50-15:15 Coffee Break

- 15:15 - 17:15 Parallel Sessions Day 1 (continued)
    - Session A: PerceptiveSentinel & openEO
    - Session B: EOPEN

18:30 Cocktail & Networking Event

DAY 2 (09:00-14:00)

- 09:00-10:30 Parallel Sessions Day 2
    - Session C: BETTER
    - Session D: Candela

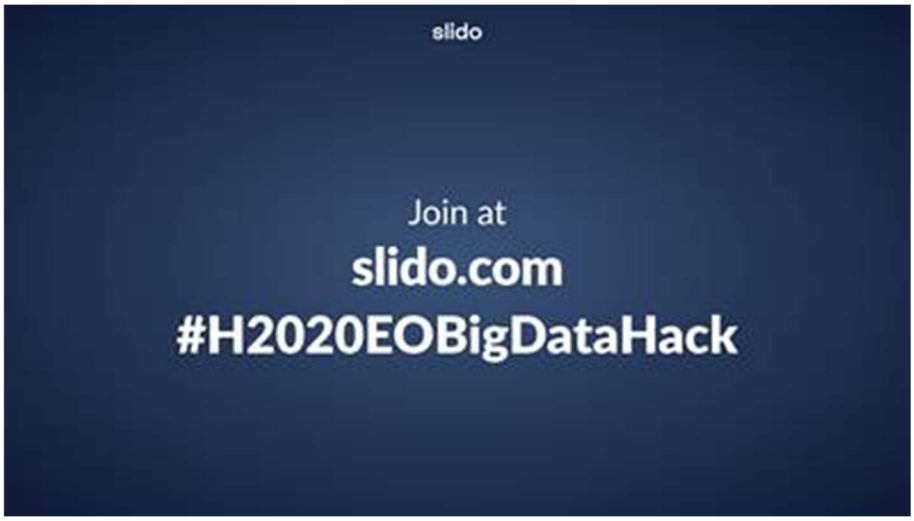
10:30-10:45 Coffee break

- 10:45-12:45 Parallel Sessions Day 2 (continued)
    - Session C: BETTER
    - Session D: Candela
- 12:45-13:15 Feedback and Questions, Reporting Exercise & Concluding Remarks

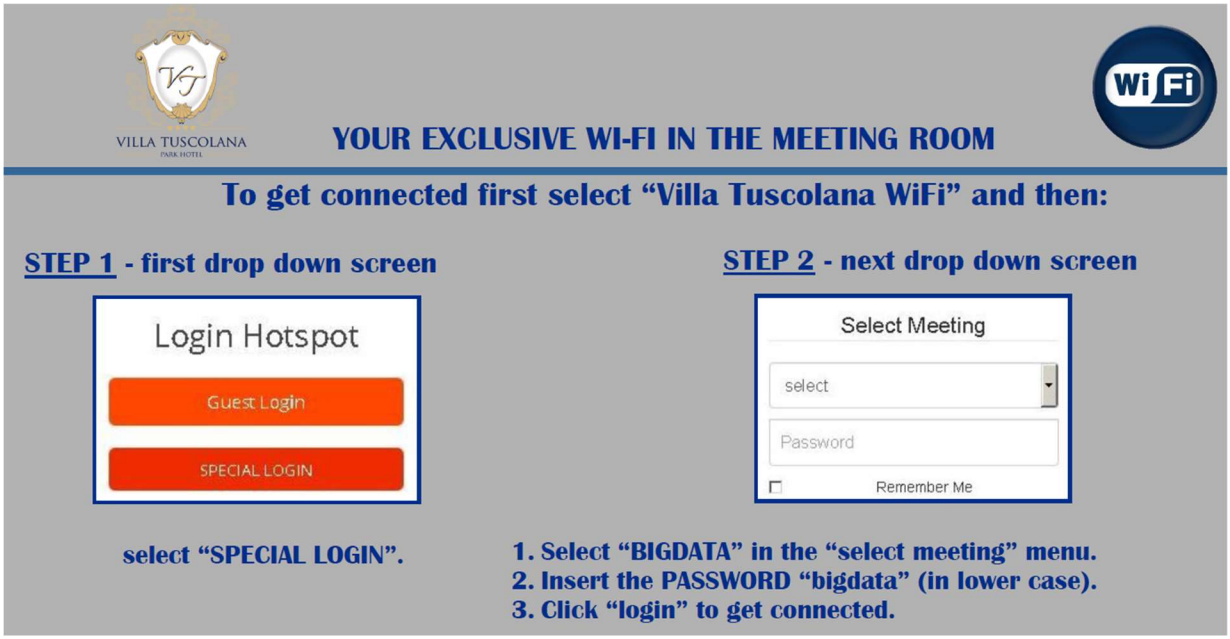
13:15 Lunch and Networking

Slido

Open Questions (anytime) & Predefined Polls will be open at end of sessions and end of event.



Wifi



Session Descriptions & Requirements

Session A-1: openEO

The attendees will work with the freely available openEO API from an end user perspective. openEO is a strategy to form standardised access points to various EO service providers, allowing users EO data processing at different cloud platforms via Python, R, or JavaScript. Being language neutral, the openEO API can further be enhanced by implementing APIs for e.g. other programming languages and applications. The openEO API is currently capable to connect to EO service providers from the EODC / Austria, VITO / Belgium, Eurac Research / Italy, Mundialis / Germany, Sinergise / Slovenia, JRC / Italy, and Google Earth Engine / Switzerland. An access point to the Sobloo DIAS is operational; connections to other DIAS are planned. Representing diverse infrastructures, these cloud providers are serving as templates for other back-ends to also connect to openEO.

During the openEO related session users will have the chance to implement Python scripts via Jupyter, using openEO processes for EO data query, processing, and downloading on / from various service providers. A web editor, basing on JavaScript can be tested to generate own process graphs via a GUI as well as a QGIS plugin. Together with PerceptiveSentinel a transfer of user-defined functions (UDFs) to the VITO back-end will be tested during the hackathon, to serve as template for a strategy to broaden up the openEO's possibilities for the end users. The user experience of this exercise will be used by the openEO consortium to enable openEO compatible on-demand UDF processing on other back-ends.

Technical Requirements and Prerequisites

- Watch the recorded Webinar here: https://youtu.be/E0wtDvm2SfA
- Laptops with internet access
- Python 3.x installation
- Jupyter installation
- openEO library, installed from GitHub
- QGIS 3.x and openEO QGIS plugin, installed from GitHub or Plugin manager

Useful links

- GitHub repository for python client: https://github.com/Open-EO/openeo-python-client
- GitHub repository for QGIS plugin: https://github.com/Open-EO/openeo-qgis-plugin
- openEO processes: https://open-eo.github.io/openeo-api/v/0.4.0/processreference/
- openEO access points: http://hub.openeo.org/
- openEO web editor: https://editor.openeo.org/

Session A-2: PerceptiveSentinel

During the joint PerceptiveSentinel and openEO hackathon session the participants will get to know eo-learn Python package library. **eo-learn** is a collection of open source Python packages that have been developed to seamlessly access and process *spatio-temporal* image sequences acquired by any satellite fleet in a timely and automatic manner. **eo-learn** is easy to use, it's design modular, and encourages collaboration – sharing and reusing of specific tasks in a typical EO-value-extraction workflows, such as cloud masking, image co-registration, feature extraction, classification, etc. Everyone is free to use any of the available tasks and is encouraged to improve, develop new ones and share them with the rest of the community.

eo-learn *library acts as a bridge between Earth observation/Remote sensing field and Python ecosystem for data science and machine learning.* The library is written in Python and uses NumPy arrays to store and handle remote sensing data. Its aim is to make entry easier for non-experts to the field of remote sensing on one hand and bring the state-of-the-art tools for computer vision, machine learning, and deep learning existing in Python ecosystem to remote sensing experts.

Participants are challenged to:

- Run **eo-learn** on their own platforms.
- Adapt **eo-learn** surface water level extraction workflow to work with Sentinel-1 data as an input.
- Adapt **eo-learn** surface water level extraction workflow to detect deforested areas in Amazonia (i.e. Porto Velho, Brazil).
- Run ***s2cloudless*** cloud detection algorithm as User Defined Function within openEO.
- Run **eo-learn** workflow as User Defined Function within openEO.
- Implement **eo-learn** task that uses openEO's API to get satellite imagery from any of the openEO's backends.

Technical Requirements and Prerequisites

- Watch the recorded Webinar here: https://youtu.be/Rv-yK7Vbk4o
- Laptops with internet access
    - Participants are encouraged to install **eo-learn** following instructions at https://eo-learn.readthedocs.io/en/latest/install.html
    - Some challenges can be executed on Binder following instructions and link at https://github.com/sentinel-hub/eo-learn-workshop

Useful links

- https://eo-learn.readthedocs.io/
- https://github.com/sentinel-hub
- https://medium.com/sentinel-hub/
- http://www.perceptivesentinel.eu

Session B: EOPEN

The primary user community targeted by EOPEN is the application/services development community using EO and non-EO data. This can be both academic or commercial.

EOPEN is based upon the concept of lowering the barrier for use by reducing the coding effort and need to dive into the details of accessing and using data and distributing the processing across computing platforms. EOPEN extensions allow the use of additional data sources such as Twitter content from social media.

EOPEN supports its users by providing core capabilities to develop, manage and execute workflows, by providing specific capabilities for processing (algorithms) and including social media (Tweets) data for inclusion into workflows and integrating processes developed and available on other platforms.

The Hackathon has therefore two areas of interest for EOPEN:

1. **Challenge**: How do uninitiated users experience the EOPEN platform to create and execute a workflow that incorporates their own algorithms;
2. **Interoperability**: How easy is for EOPEN to integrate and use capabilities made available on other platforms or provide access for others to use EOPEN capabilities.

Exercise 1. Water body detection workflow on Sentinel-2 data obtained using the openEO Python client library.
Exercise 2. Water body detection workflow on Sentinel-2 data obtained using the BETTER API.
Exercise 3. Access Twitter content in either JSON or RDF semantic representation using an EOPEN service.
Exercise 4. Deployment and execution on ONDA DIAS (due to file-based access to Sentinel products). Other DIAS platforms are supported but require an additional download/transfer step.
Exercise 5. Import and execute your own process in the EOPEN Platform. Candidate processes to be discussed and selected during the session.

Technical Requirements and Prerequisites

- Watch the recorded Webinar here: https://eopen-project.eu/introductory-tour/
- Laptops with internet access
- Recent Web browser (Chrome or Firefox recommended)

Useful links

- https://eopen-project.eu/
- https://proto1.eopen.spaceapplications.com
  (Developer platform credentials will be provided during the Hackathon EOPEN session)

Session C: BETTER

BETTER is implementing a Big Data intermediate service layer focused on creating user-centric solutions, while addressing the full data lifecycle associated with EO data, to bring more downstream users to the EO market and maximise exploitation of Copernicus data and information services. These customized solutions, denominated as Data Pipelines, are driven by a large number of Data Challenges to be set by users deeply involved in addressing key Societal Challenges. Each Data Pipeline is reposnsible for the systematic production of output data streams delivered to users through standardised interfaces

BETTER will select leading challenges from the 1st cycle (9 challenges see here) and organise a hands-on session which includes the **deployment and access to the chosen data pipelines,** coupled with exercises where test users will have the role of consumers of those data pipelines in the scope of the development of applications. Three exercises are being planned to highlight these capabilities:

- **EO data discovery, processing and setting-up of data processing pipelines**, to learn how to discover input data for one of the pipelines, trigger the processing of this input data and visualise and analyse the results
- **Semantic Analysis**, to understand how to to perform a SANSA-based semantic correlation analysis between the output of one of the 1st cycle pipelines (CHIRPS aggregated precipitation) and external variable of Food Price data
- **Land cover changes and inter-annual vegetation performance analysis using ML algorithms**, to use the output of another BETTER pipeline in a machine learning crop classification application.

Technical Requirements and Prerequisites

- Watch the recorded Webinar here: https://youtu.be/w2OQ_s2XU5Q
- Laptops with internet access
  Part I: Web access
- **Part II: Semantic Analytics** Download Data & Scripts here. Requirements (Linux OS, Windows possible but would take too long for the Hackathon): Maven (Java SDK 1.7+), Python, Anaconda Packages: GDAL and Shapely, and Apache Spark.
- **Part III:** Web access; QGIS

Useful links

https://github.com/ec-better

Session D: CANDELA

Main idea during the session is to have users from the other projects and the other participants as **beta-testers of the tools already deployed in the CANDELA platform** in the CreoDIAS cloud.
Core exercises are related to the current (not final) analytic tools being developed: optical change detection, SAR (Synthetic Aperture Radar) change detection, and data mining for Sentinel-1 and Sentinel-2 data products; and will help as well to assess the usability of the platform (platform access, jupyter lab functionality and/or data access and management).

Step-by-step exercises will be presented in different jupyter-notebooks:
Exercise 1: **Change detection on Sentinel-2 data product**. Objective: Find the high-level of changes on a vineyard of Aquitaine (region of southwestern France) and the related information. This exercise not only shows how to run the change detection pipeline on Sentinel-2 images, but also presents CreoDIAS tool to access data, and introduce triplification and semantic search tools in order to interlink external datasets and query the database.
Exercise 2: **Change detection on Sentinel-1 data product**. Objective: Find all the changes over the Torun city (Poland) after a storm. This exercise shows how to access Sentinel-1 data and run the change detection tool on SAR images.
Exercise 3: **Data Mining** - Overall interactive and iterative process of discovering useful information in the Sentinel-1/-2 product or a collection of products. Objective: Discover in the EO products relevant and application-valuable scene structures; perform semantic annotation of selected structures, and generate a semantic catalogue for the observed area.

Technical Requirements and Prerequisites

- Watch the recorded Webinar here: https://youtu.be/8DOEo5aEEPo
- Laptops with internet access.
- Access to the platform will be via web browser (Firefox preferred) and login and password are needed. Credentials will be provided at the hackathon, either by generic accounts or after collecting participants' name/e-mail.
- Rights to download DLR front-end tool (around 10MB). To unzip the front-end file, compression software (e.g., WinZip, 7-zip) is needed; to run the tool, Java Runtime Environment (JRE) and command-line window is required.

Useful links

http://www.candela-h2020.eu/content/data-mining-v2
http://www.candela-h2020.eu/content/deep-learning-v2
http://www.candela-h2020.eu/content/semantic-search-v2