OPENEO HACKATHON

openEO | Grant agreement No. 776242

AGENDA

Monday, 25.06

09:00 – 10:00	**Registration**
10:00 – 10:30	**Welcome and Introduction to openEO**
10:30 – 12:00	**Test Cases**
12:00 – 12:45	Lunch
12:45 – 17:00	**Hacking**
17:00 – 17:30	**Day 1 Wrap-Up**
17:30 – 18:00	**Presentation of Findings from openEO User Requirements Survey**
18:00	Drinks & Networking

Tuesday, 26.06

09:00 – 13:00	**Hacking**
13:00 – 14:00	Lunch
14:00 – 14:30	**Day 2 Wrap-Up**
14:30 – 15:45	**Feedback from the Hackathon - How openEO Meets User Needs and Providing Feedback and Input Into openEO**
15:45 – 16:00	**Raffle and Closing Notes**

Icons of this presentation are made by Smashicons, Freepik and Roundicons from www.flaticon.com

WHO IS WHO?

Short **Introduction Round** *(max 30 Seconds!)*

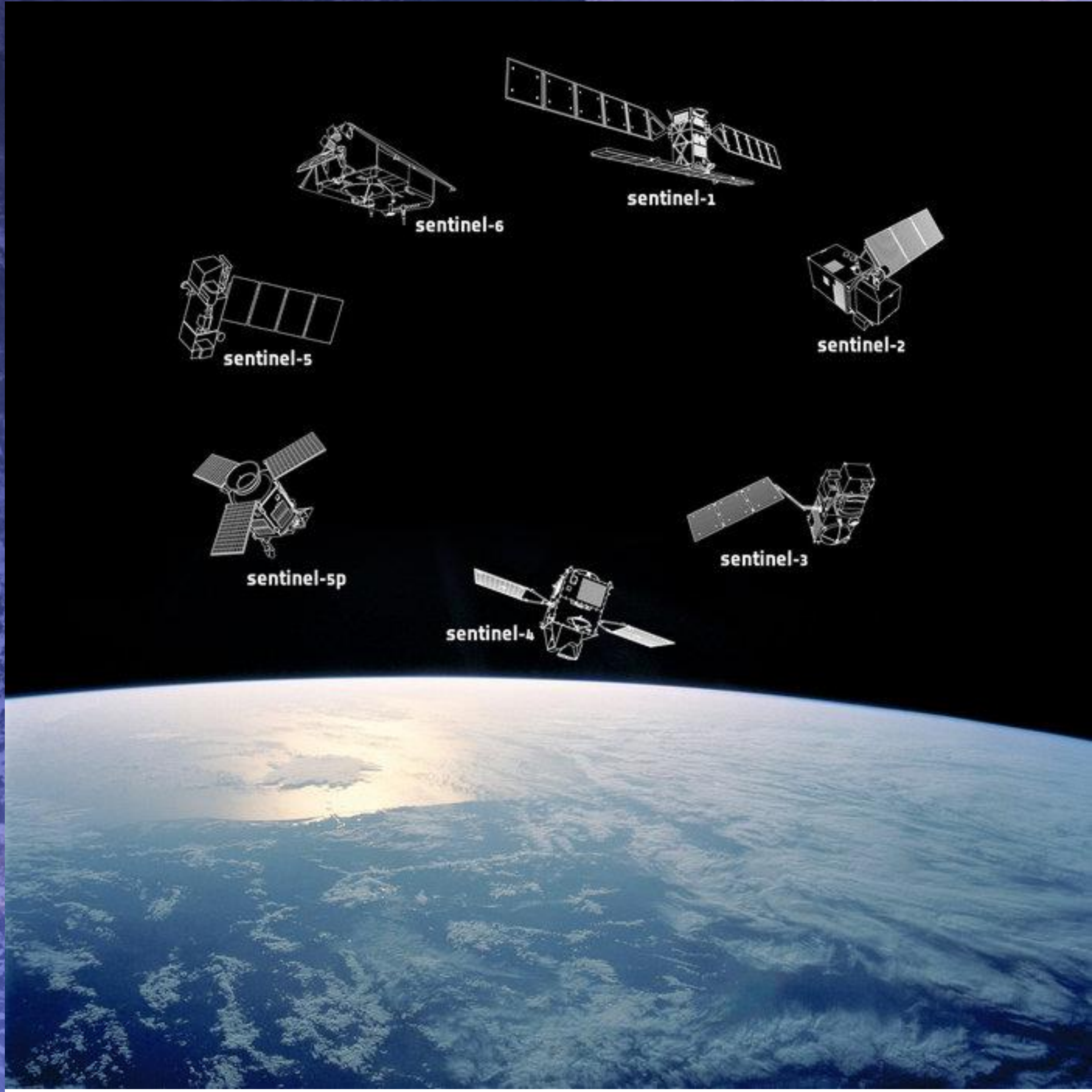
- Your **Name**
- Your **Profession**
- Your **Skills / Technological Stack**

INTRODUCTION TO OPENEO

MOTIVATION

Copernicus Programme

- **World's largest** single EO programme
- Directed by the European Commission and ESA
- **Global**, **continuous**, **autonomous**, **high quality**, **wide range** Earth observation
- Data on **land surfaces**, **oceans** and **atmosphere**
- Development of a family of dedicated satellites, called the **Sentinels**
- Sentinels are capable to **measure earth across whole measurement spectrum**
- **Seven** planned Sentinel missions with a constellation of **two satellites** each
- Data acquired is systematically downlinked and processed to **operational user products**
- Users can download the data from **dedicated access points** (e.g. SentinelHub)
- Data is provided **free of charge** and **open**

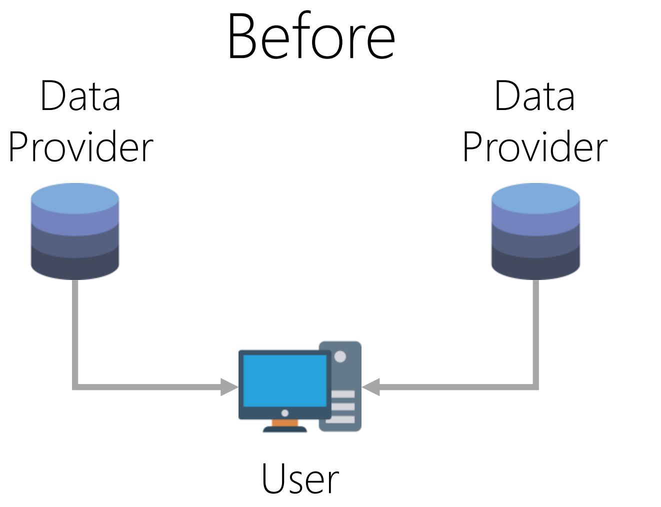


SOURCE: http://www.esa.int/spaceinimages/Images/2014/04/Sentinel_family

MOTIVATION

Big Data Challenge

- **Rapid growth** of EO data volume
- Capacity to **transfer data** falls increasingly **behind** the **capacity to store and process data**
- **Formation of new back-offices** that provide capacities for storing and processing EO data (Private, Public, Cooperation)
- Increasing number of **EO-services** and **applications**

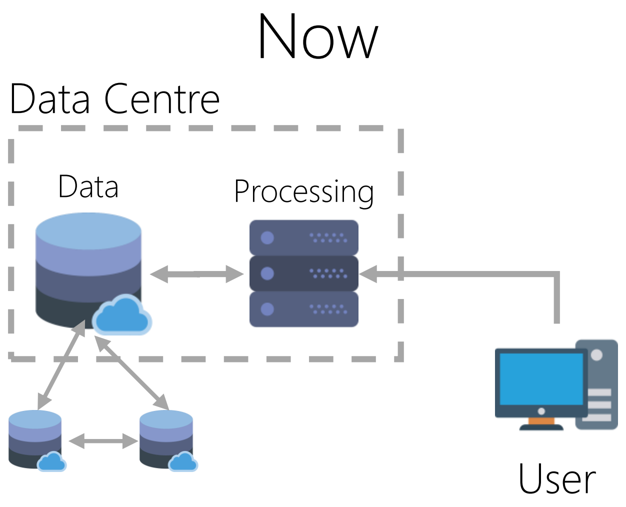


Before

Data Provider

Data Provider

User



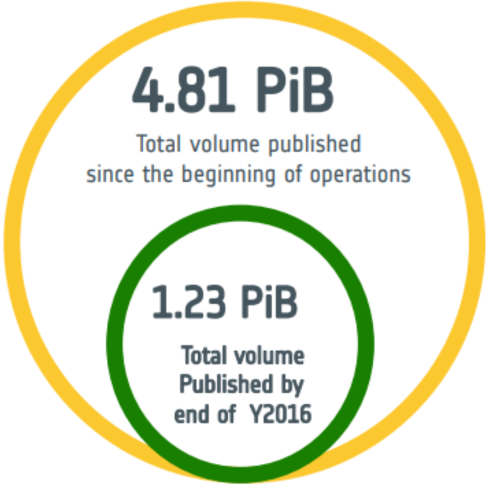
Now

Data Centre

Data

Processing

User



4.81 PiB

Total volume published since the beginning of operations

1.23 PiB

Total volume Published by end of Y2016

mission	Y2017-number of published products since start of operations	Y2017-volume of published products since start of operations (PiB)
S1[2]	1,978,597	3.05
S2	2,332,632	1.40
S3	859,246	0.36
ALL	5,170,475	4.81

SOURCE: Sentinel Data Access – Annual Report 2017

Icons of this presentation are made by Smashicons, Freepik and Roundicons from www.flaticon.com

OBJECTIVES

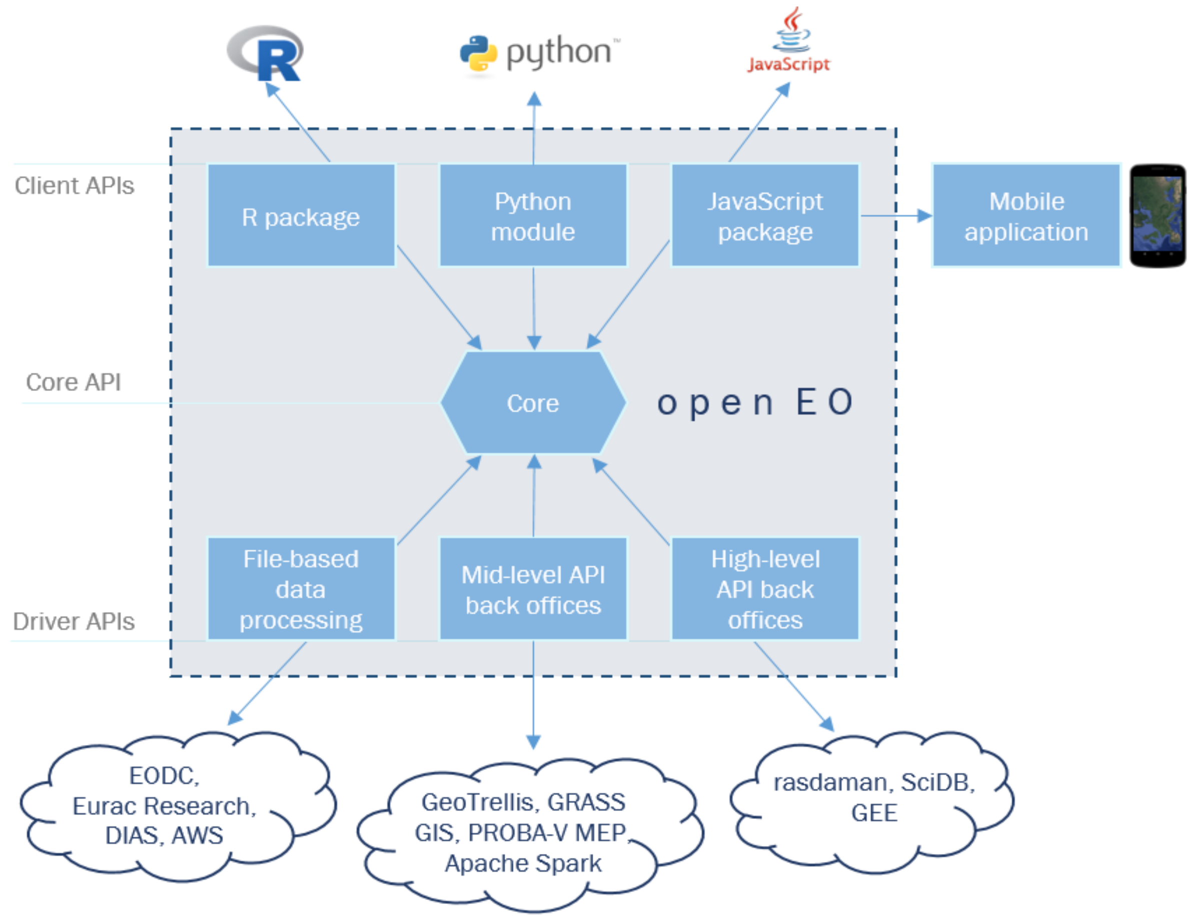
“The objective of the openEO consortium is to build a ***common****,* ***open source interface*** *that* ***facilitates standardized interchange*** *between* ***users*** *and* ***applications*** *of Copernicus and other EO data as hosted by an* ***increasing number of cloud providers****”* *(openEO proposal)*

1. Establish openEO as an **open source initiative**
2. Define a **Core API** to connect clients with back-offices
3. Develop **Driver APIs** to connect to back offices
4. Develop **Client APIs** for R, Python and JavaScript
5. Define and publish **Use Cases**
6. Software **validation** the openEO interface

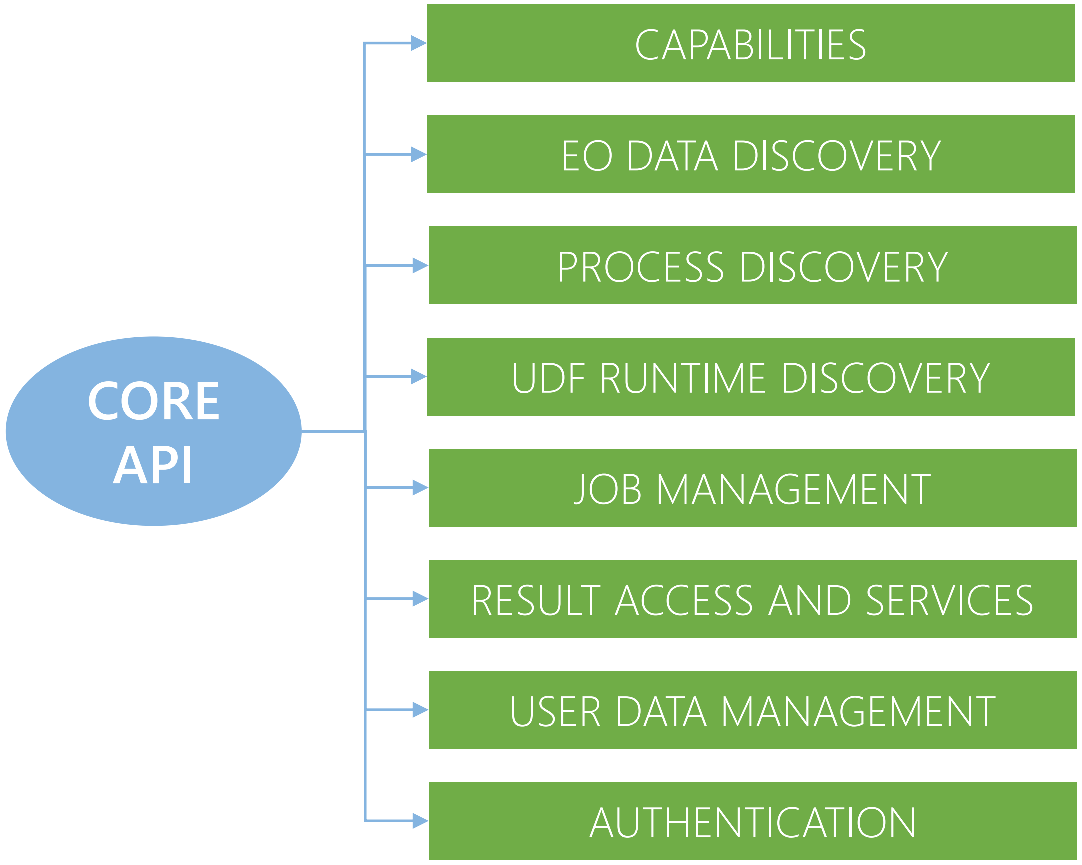
Becoming a **widely adapted standard** as **intermediate stack** between cloud-based EO data and end-users

CONCEPT

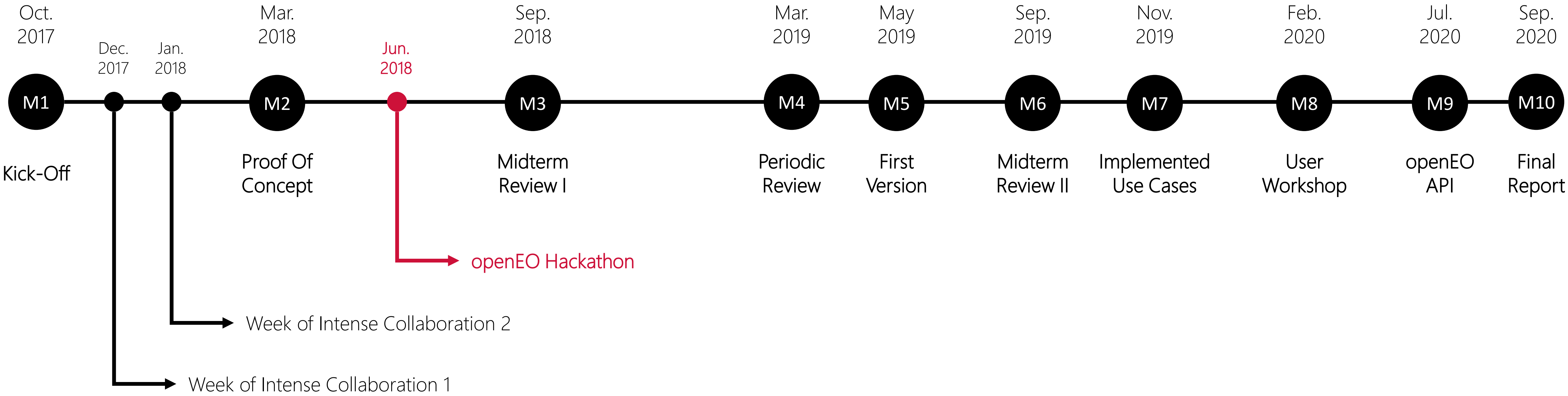
Layers of openEO


Client APIs
Core API
Driver APIs
R package
Python module
JavaScript package
Mobile application
Core
open EO
File-based data processing
Mid-level API back offices
High-level API back offices
EODC, Eurac Research, DIAS, AWS
GeoTrellis, GRASS GIS, PROBA-V MEP, Apache Spark
rasdaman, SciDB, GEE

Microservices


CORE API
CAPABILITIES
EO DATA DISCOVERY
PROCESS DISCOVERY
UDF RUNTIME DISCOVERY
JOB MANAGEMENT
RESULT ACCESS AND SERVICES
USER DATA MANAGEMENT
AUTHENTICATION

CURRENT STATE



- Oct. 2017 – M1 – Kick-Off
- Dec. 2017 → Week of Intense Collaboration 1
- Jan. 2018 → Week of Intense Collaboration 2
- Mar. 2018 – M2 – Proof Of Concept
- Jun. 2018 → openEO Hackathon
- Sep. 2018 – M3 – Midterm Review I
- Mar. 2019 – M4 – Periodic Review
- May 2019 – M5 – First Version
- Sep. 2019 – M6 – Midterm Review II
- Nov. 2019 – M7 – Implemented Use Cases
- Feb. 2020 – M8 – User Workshop
- Jul. 2020 – M9 – openEO API
- Sep. 2020 – M10 – Final Report

OPENEO HACKATHON

Aims of the Hackathon

Getting User Feedback

- **Strength and weaknesses** of openEO
- Missing **functionalities**
- Validate **design goals**

Identify Usability Problems

- Design **Inconsistencies**
- **Usability** problems (Clients / Core API)
- Setup / Comprehension / Usage problems

Implementing Ideas

- Involving the **users / experts from various domains**
- **Extend / Improve** the Core API / Clients
- **Build services** and **applications** on top of openEO

Mentors

- **Guide** the participants
- **Provide help** if needed
- **Collaboratively work together** to implement new features

Mentors

- **Bernhard Gößwein** (TU Wien)
- **Dainius Masiliunas** (WUR)
- **Jeroen Dries** (VITO)
- **Matthias Mohr** (WWU)
- **Gunnar Busch** (EODC)

License

All work that is produced at the Hackathon will be published under the **Apache 2.0 License** on openEO GitHub

TECHNICAL DETAILS

TERMS

Process

- The description of an **operation / filter** as provided by the back-end
- Similar to a **function** definition in programming languages."

Process Graph

- Includes specific **processes** (including **process arguments**)
- Chain **multiple processes**
- Arguments of processes in process graphs can be **again process graphs**, **input datasets**, **scalar** or **arrays**

Job

- Wraps the process graph and organizes its execution
  - **Batch job:** Job is submitted to back-end, but will retain inactive till the execution is called.
  - **Web services:** allow to change the viewing extent / level of detail. Computations may run on demand.
  - **Process graphs** can also be executed synchronously. Results are delivered with the request itself.

User Defined Function (UDF)

- Users are able to **upload custom code** and have it executed
- The openEO API defines **different UDF types**, e.g.:
  - apply_pixel()
  - aggregate_time()
  - …

EXAMPLE PROCESS GRAPHS

Process Graph

```
{
    "process_id": "min_time",
    "args": {
        "imagery": {
            "process_id": "/user/custom_ndvi",
            "args": {
                "imagery": {
                    "process_id": "filter_daterange",
                    "args": {
                        "imagery": {
                            "process_id": "filter_bbox",
                            "args": {
                                "imagery": {
                                    "product_id": "S2_L2A_T32TPS_20M"
                                },
                                "left": 652000,
                                "right": 672000,
                                "top": 5161000,
                                "bottom": 5181000,
                                "srs": "EPSG:32632"
                            }
                        },
                        "from": "2017-01-01",
                        "to": "2017-01-31"
                    }
                },
                "red": "B04",
                "nir": "B8A"
            }
        }
    }
}
```

Chained Process Graph

```
{
    "imagery": {
        "process_id": "union",
        "args": {
            "collection": [
                {
                    "process_id": "filter_bands",
                    "args": {
                        "imagery": {
                            "product_id": "Sentinel2-L1C"
                        },
                        "bands": 8
                    }
                },
                {
                    "process_id": "filter_bands",
                    "args": {
                        "imagery": {
                            "product_id": "Sentinel2-L1C"
                        },
                        "bands": 5
                    }
                }
            ]
        }
    }
}
```

OPENEO API

Live Slides web content

To view

Download the add-in.
liveslides.com/download

Start the presentation.

CLIENTS

Python Client

```python
import openeo

# Connect to GEE back-end
session = openeo.session(endpoint="http://giv-project8.uni-muenster.de")

# Retrieve capabilities
session.list_capabilities()

# ['/capabilities',
#  '/data',
#  ...]
```

R Client

```r
library(openeo)

# Connect to GEE back-end
session = connect("http://giv-project8.uni-muenster.de")

# Retrieve capabilities
session %>% listCapabilities()
# ['/capabilities',
#  '/data',
#  ...]
```

Back-End Driver

Currently available back-end driver:

Google Earth Engine (JavaScript / NodeJS)
https://github.com/Open-EO/openeo-earthengine-driver

Geotrellis / GeoPySpark (Python 3.5)
https://github.com/Open-EO/openeo-geopyspark-driver

GRASS GIS (Python 3.5)
https://github.com/Open-EO/openeo-grassgis-driver

OpenShift Origin (Python 3.5)
https://github.com/Open-EO/openeo-openshift-driver

R / plumber (R)
https://github.com/Open-EO/openeo-r-backend

Sentinel Hub (JavaScript / NodeJS)
https://github.com/Open-EO/openeo-sentinelhub-driver

rasdaman (Java)
https://github.com/Open-EO/openeo-wcps-driver

How To Develop

Develop Client Features or Applications

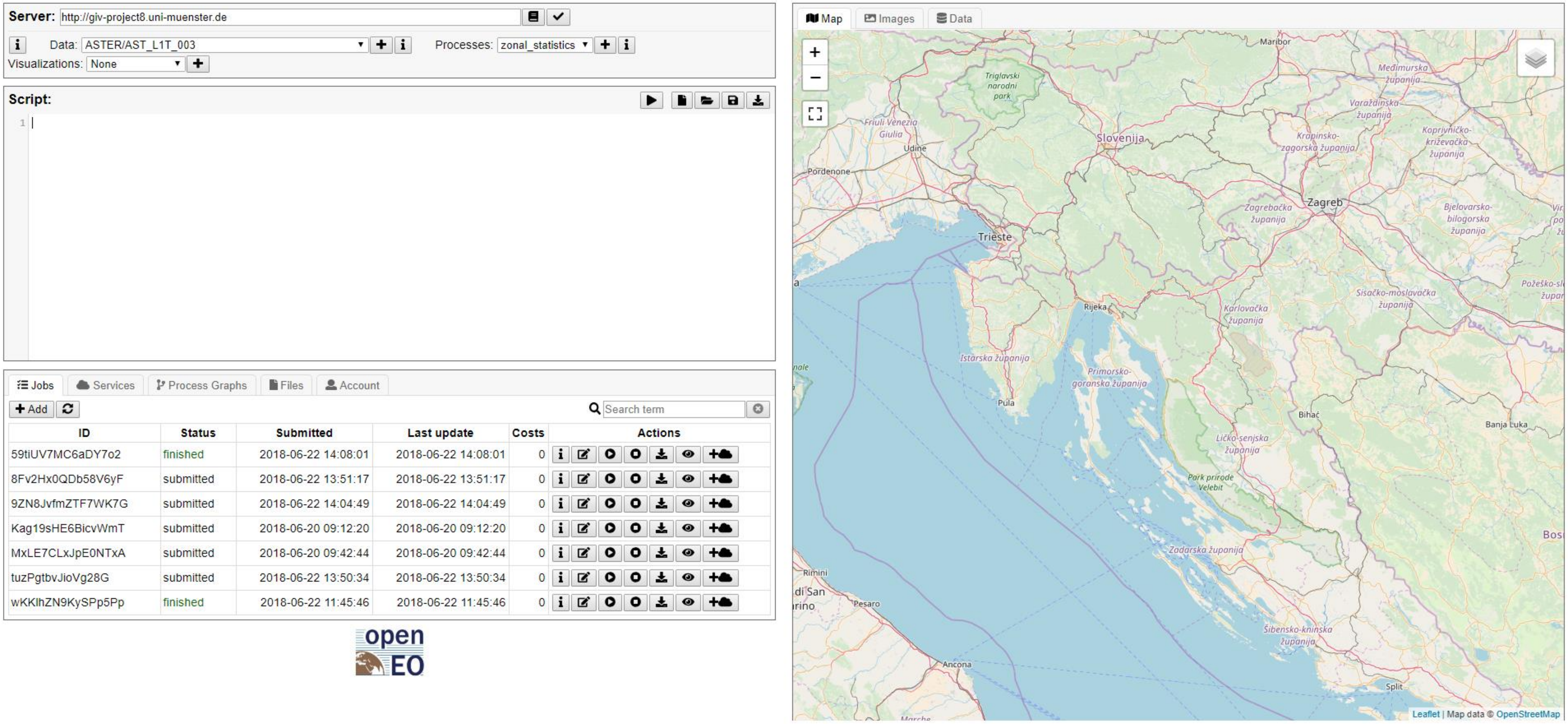
- Using the **Python** Client
- Using the **R** Client
- Using the **JavaScript** Client
- Work on **Jupyter** Notebook with openEO Python Client (ask Gunnar)

Develop Back-End Features

- Local Back-end
  - **R Back-End**
  - **GEE Back-End** (with Google Account)
- **Team up** with back-end developers
  - **Geotrellis** / **GeoPySpark** (ask Jeroen)
  - **GEE** Back-End (ask Matthias)
  - **OpenShift** (ask Gunnar)
- Deploy **Docker Containers** on OpenShift with **EO Data Storage Access** (ask Gunnar)
- Work on **VMs** with **EO Data Storage Access** (ask Gunnar)

Please create and work in a directory in the **openEO Hackathon GitHub repository** (folder: **/hacking**)

Web Editor



Server: http://giv-project8.uni-muenster.de

Data: ASTER/AST_L1T_003

Processes: zonal_statistics

Visualizations: None

Script:

ID	Status	Submitted	Last update	Costs	Actions
59tiUV7MC6aDY7o2	finished	2018-06-22 14:08:01	2018-06-22 14:08:01	0	
8Fv2Hx0QDb58V6yF	submitted	2018-06-22 13:51:17	2018-06-22 13:51:17	0	
9ZN8JvfmZTF7WK7G	submitted	2018-06-22 14:04:49	2018-06-22 14:04:49	0	
Kag19sHE6BicvWmT	submitted	2018-06-20 09:12:20	2018-06-20 09:12:20	0	
MxLE7CLxJpE0NTxA	submitted	2018-06-20 09:42:44	2018-06-20 09:42:44	0	
tuzPgtbvJioVg28G	submitted	2018-06-22 13:50:34	2018-06-22 13:50:34	0	
wKKIhZN9KySPp5Pp	finished	2018-06-22 11:45:46	2018-06-22 11:45:46	0	

OPENEO GITHUB

Live Slides web content

To view

Download the add-in.

liveslides.com/download

Start the presentation.

RESSOURCES ON OPENEO

General

Website:	http://openeo.org/
GitHub:	https://github.com/Open-EO
YouTube:	https://www.youtube.com/channel/UCMJQil8j9sHBQkcSlSaEsvQ
Twitter:	https://twitter.com/Open_EO

Clients

Python:	https://github.com/Open-EO/openeo-python-client
R-Client:	https://github.com/Open-EO/openeo-r-client
JS-Client:	https://github.com/Open-EO/openeo-js-client

openEO API

Docs v0.0.2:	https://open-eo.github.io/openeo-api/
Docs v0.3:	https://open-eo.github.io/openeo-api/v/0.3.0/index.html
API Reference v0.0.2:	https://open-eo.github.io/openeo-api/apireference/index.html
API Reference v0.3:	https://open-eo.github.io/openeo-api/v/0.3.0/apireference/index.html

Test Cases

Test Cases

Instructions

1 Please open the **openEO Hackathon GitHub** repository: https://github.com/Open-EO/openeo-hackathon

2 In the directory **/test-cases** you will find 5 Test Case Tasks

3 Please try to complete the Tasks (by yourself or in a team) using the **Python or R Client** *(Task 5 is currently just supported by the Python Client)*

4 If you have questions, please **first use the Docs** (Python in Python Client Docs / Github, R using *help(keyword)*). If you can find the answer you can ask a mentor.

5 You will find the **solutions** for the tasks in the GitHub repository, but please first try to implement the tasks yourself.

Hacking

Hacking Ideas

Some ideas / inspirations:

- **OpenID Connect integration**
- **Process Graph Storage**
- **GraphQL Wrapper** for openEO
- Extending the **JavaScript Client**
- Implementing a framework for **back-end compliance testing**
- Extending the **Python Client:**
  - Implementing **Job Management**
  - Adding **Authentication**
  - Adding **Delete File Feature**
  - Adding a **User class**
  - Add **Exception Handling / Error Handling**
- Extend the **GEE back-end to support other GEE scripts**

(Details can be found in the GitHub repository)

Discussion

Discussion Topics

Topics to discuss:

- What are the first impressions of openEO?
- What are the Strengths, Weaknesses, Opportunities, and Threats for openEO?
- What are the one or two priority needs which would help each user in their work?
- Reflect on the outcomes from the user needs questionnaire?