A Common, Open Source Interface Between Earth Observation Data Infrastructures And Front-End Applications

No.	Participant organisation name		Country
1	Technische Universitaet Wien (Coordinator)	TU Wien	AT
2	Westfaelische Wilhelms-Universitaet Muenster	WWU	DE
3	Wageningen University	WUR	NL
4	Vlaamse Instelling Voor Technologisch Onderzoek N.V.	VITO	BE
5	Earth Observation Data Centre for Water Resources Monitoring GmbH	EODC	AT
6	Mundialis GmbH and Co. KG	Mundialis	DE
7	Sinergise Laboratorij Za Geografske Informacijske Sisteme Doo	Sinergise	SI
8	Academia Europea di Bolzano	EURAC	IT
9	Solenix Schweiz GmbH	Solenix	CH
10	Joint Research Centre of the European Commission	JRC	IT

TABLE of CONTENTS

PART B | 1

ABBREVIATIONS

A	ACF	Action Contre la Faim
	API	Application Programming Interface
	AWS	Amazon Web Services
B	BFAST	Breaks For Additive Season and Trend project
	BiDS	Conference on Big Data from Space
	BMLFUW	Austrian Federal Ministry for Agriculture & Forestry, the Environment and Water Resources
C	CA	Consortium Agreement
	CCI	Climate Change Initiative
	CEOS	Committee on Earth Observation Satellites
	CPU	Central Processing Unit
D	DIAS	Copernicus Data and Information Access Service
	DLR	Deutsches Zentrum für Luft- und Raumfahrt, German Space Agency
E	EARSEL	European Association of Remote Sensing Laboratories
	EC	European Commission
	EEA	European Environmental Agency
	EGI	European Grid Infrastructure
	EIONET	European Environment Information and Observation Network
	EO	Earth Observation
	EODC	Earth Observation Data Centre for Water Resources Monitoring
	ESA	European Space Agency
	ESRIN	European Space Research Institute (ESA)
	EUMETSAT	European Organisation for the Exploitation of Meteorological Satellites
F	FAO	Food and Agriculture Organization of the United Nations
	FIT	Frauen in die Technik
G	GDAL	Geospatial Data Abstraction Library
	GEE	Google Earth Engine
	GEO	Department of Geodesy and Geoinformation, TU Wien
	GEOSS	Global Earth Observation System of Systems
	GFOI	Global Forest Observations Initiative
	GOFC-GOLD	Global Observation for Forest Cover and Land Dynamics
I	ICIMOD	International Centre for Integrated Mountain Development
	ICT	Information and Communications Technology
	IFAD	International Fund for Agricultural Development
	INPE	National Institute for Space Research

	ISO	International Organization for Standardization
	ISPRS	International Society for Photogrammetry and Remote Sensing
	IT	Information Technology
J	JRC	Joint Research Centre (European Commission)
	JS	JavaScript
K	KO	Kick Off
M	MEP	Mission Exploitation Platform
	MINAM	Ministerio del Ambiente
N	NaN	Not a Number
	NASA	National Aeronautics and Space Administration
	NFMS	National Forest Monitoring Systems
	NGO	Non-Governmental Organisation
	NDVI	Normalized Difference Vegetation Index
	NOA	National Observatory of Athens
O	OGC	Open Geospatial Consortium
	OSGeo	Open Source Geospatial Foundation
P	PM	Project Management
	PROBA-V	Project for On-Board Autonomy – Vegetation
R	REST	Representational State Transfer
S	SAF	Satellite Application Facility
	SME	Small and medium enterprises
T	TRL	Technology Readiness Level
U	UDF	User Defined Function
V	VITO	Vlaamse Instelling Voor Technologisch Onderzoek NV –Flemish institute for technological research
W	WGISS	Working Group on Information Systems and Services
	WMS	Web Mapping Service
	WP	Work Package

1. EXCELLENCE

1.1 OBJECTIVES

As part of Europe's Copernicus programme, a series of Sentinel satellites has been launched that provide operational capabilities for observing the Earth across the whole measurement spectrum. Due to their advanced sensing concepts and outstanding spatio-temporal sampling characteristics, the Sentinel satellites collect more data than any Earth Observation (EO) programme before. While this is highly beneficial for all application domains, it implies that Copernicus faces significant Big Data challenges. In fact, the capacity of the Sentinel satellites to acquire data outstrips existing capacities to transmit, store, process, and analyse them. As a result, there is an urgent need to replace traditional workflows, which relied on distributing the data to thousands of users over the internet, with cloud computing approaches that **bring the users and their software to the data** instead. This is one of the central paradigms of the Big Data era. Many European organizations and initiatives have recognized this need at an early stage of the Copernicus programme, and are now working towards the establishment of back offices capable of storing and processing Petabytes of Sentinel data. Among these are the openEO partners EODC, VITO and EURAC Research who provide cloud platforms with access to worldwide Copernicus data[1]. Furthermore, European industry and academia lead the development of Sentinel-based front office services. To consolidate these activities, and to narrow the technological gap to the US, it is now of paramount importance - as identified in this H2020 call - to develop intermediate software layers allowing the exploitation of back office resources for the benefit of front office services.

Therefore, 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. The openEO interface will consist of **three layers of Application Programming Interfaces (APIs)** that connect applications of several front office clients with various back office drivers (see *Figure 1*). This will simplify the use of cloud-based EO processing engines, allow switching between cloud-based back office providers and comparing them, and enable reproducible, open EO science. Thereby, openEO reduces the entry barriers for the adaptation of cloud computing technologies by a broad user community and paves the way for the federation of EO data infrastructure capabilities.

The specific objectives of openEO are:

1. To establish openEO as an **open source initiative** by generating involvement of users and developers from both the EO and IT communities, using strong and open communication strategies (WP 2: User Involvement & Dissemination),
2. define and implement an open source **core API** for finding, accessing, and processing large EO datasets in cloud-based data processing environments (see *Figure 1;* WP 3: Core API),
3. develop **driver APIs** to connect to back offices operated by European and worldwide industry (WP 4: Driver APIs),
4. develop open source **client APIs** for analysing these datasets using R, Python and JavaScript (WP 5: Client APIs),
5. develop and publish use cases (WP 6: Use Cases), and
6. validate the openEO Interface (WP 7: Validation & Uptake).

As openEO aims to become a widely accepted standard, its applicability shall not be limited to a few selected back offices. Therefore, openEO will develop generic driver APIs that allow connecting to different types of back office services, ranging from simple IaaS (Infrastructure as a Service) accounts with access to raw Copernicus

[1] EODC provides Petabyte storage and supercomputing capabilities for global Sentinel-1, Sentinel-2 and Sentinel-3 data. VITO hosts worldwide SPOT Vegetation and Proba-V data and operates the PROBA-V Mission Exploitation Platform (MEP). EURAC Research is developing a cloud platform for the European Alps.

(and other) data to advanced PaaS (Platform as a Service) solutions providing pre-processed Copernicus data only through platform-specific API calls.

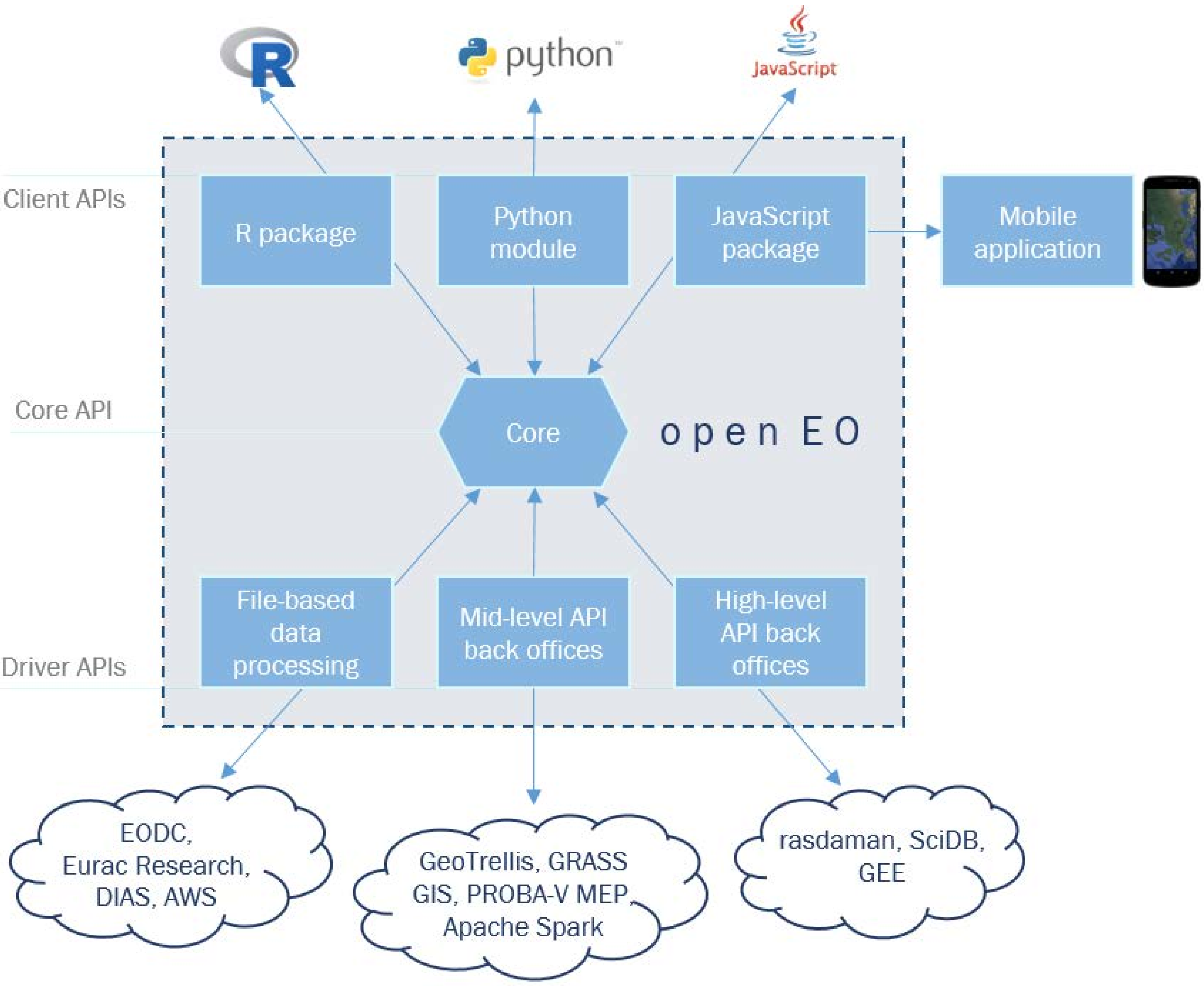

Figure 1: Scheme of openEO layered API approach. Several client APIs will provide connection to front offices (here R, Python, JavaScript; Various driver APIs will ensure connection to entrance points of different cloud service providers; the syntaxes of the aforementioned API results will be standardized in the Core API to allow end or intermediate users a standardized access to all user back office services.)

Figure 2: Scheme of openEO layered API approach. Several client APIs will provide connection to front offices (here R, Python, JavaScript; Various driver APIs will ensure connection to entrance points of different cloud service providers; the syntaxes of the aforementioned API results will be standardized in the Core API to allow end or intermediate users a standardized access to all user back office services.)

For concrete driver API implementations, openEO will focus on already existing European back office services operated by **EODC, VITO,** and **EURAC**. To ensure compatibility of the openEO interface with Google Earth Engine (GEE) and Amazon Web Services (AWS), driver APIs will be developed for these two well-established platforms as well. All developments will be undertaken with special consideration of the **Copernicus Data and Information Access Service** (DIAS) that is expected to go into operations in 2018. With the help of the European Space Agency (ESA) and the openEO Advisory Board, cases for adapting and implementing openEO on DIAS will be defined.

1.2 RELATION to the WORK PROGRAMME

The open EO project is fully in line with the scope of the call. In the following, we describe how openEO addresses the major objectives of the call as specifically mentioned by the work programme.

ADAPTATION of BIG DATA TECHNOLOGIES

Current mainstream big data technologies (e.g. map-reduce, hadoop, spark) are hard to deploy for EO data. This is because of their design to analyse large tables with independent observations, rather than collections of space- and time-referenced images taken by satellites. This forced back office providers to develop customized solutions, which in turn forces users to choose from competing back office services with very different, incompatible interfaces. openEO will start the development of a single standard for cloud-based EO data processing, implement this for a number of established cloud services, and provide client interfaces for popular environments used by data scientists and EO researchers (R, Python, JavaScript). This will not only make the life of researchers easier, but also enable a fair competition between small and large IT companies because cloud service offerings become comparable. It will also, finally, bring EO researchers the possibility to compare and cross-check analyses against different back offices, and by that foster open, reproducible EO science. Thereby, openEO will help adapting current and future big data technologies to Copernicus user scenarios, which is a core contribution of the project.

BENEFITING from STORAGE and PROCESSING SERVICE

Many intermediate and end users of EO data are hesitant to move their EO data analytics into cloud based processing platforms. The prime reason is in many cases not of a technical nature, but the fear of becoming too dependent on one particular back office provider. openEO will help to reduce concerns of potential lock-in situations by making it much easier to switch between providers or to use several of them. openEO will achieve this by creating a standard through its core API, and by demonstrating of how to develop driver APIs for a diversity of back offices. Thereby, any back office provider is put in a position to develop its own driver APIs and commit it to the openEO open source project. In this simple way, the back office becomes an integral part of a growing market, creating a win-win situation for both providers and users of EO storage and processing services.

CONCENTRATE on INTERMEDIATE LAYERS

Given that openEO *is* a stack of three intermediate layers (APIs) that connect user applications with EO data processing services, the project has a clear focus on intermediate layers. Over time, we anticipate that openEO becomes *the* intermediate API stack between cloud-based EO data processing services and end-user clients, providing standardized access to various back offices from different front office clients. Being open source, openEO can be used by Copernicus services and public and commercial users alike.

ALLOW CHAINING of VALUE-ADDING ACTIVITIES

As listed in ***Table 1***, the openEO core API will be made up of a bundle of microservices that cover all functions necessary for building value-added services. The granularity and object-oriented design of the software will allow for the chaining of value adding processing services, run on one or more back offices. Within the project, this capability will be demonstrated through the development of four use cases: (i) Radar Image Compositing, (ii) Multi-source Phenology toolbox, (iii) Optical-Radar Forest Monitoring, and (iv) Snow Monitoring. Concrete, customized implementations of these four use cases already exist. Within the openEO project, these four value-adding processing chains will be redesigned to build upon the openEO API stack. Furthermore, the new implementations will be tested on several back offices. These use cases will serve to prove to the larger community that the openEO concepts works, and are hence crucial for ensuring the success of the openEO initiative after the project's end.

Table 2: *Microservices and their tasks for the openEO API core layer*

1. Authentication • register • login • admin queries (credit, availability of CPU/memory/disc space)
2. User data management • upload (put) • download (get/fetch)
3. Data discovery • describe available data sets: scenes, composites, list of metadata • query properties of particular datasets • retrieve dimensions, coordinate reference system, measurement units, attributes
4. Process discovery • geometry modifying operations: pixel registration, mosaic, resample, join (fuse sensors) • operations working on scalars (zero dimensions), on one dimension (e.g. time series, spectral, on two dimensions (e.g. on single bands), on multiple dimensions (time series of bands, multispectral) • search operations matching a dataset name • search operations that can be carried out on a particular data set or subset
5. UDF (user-defined function) handling • on a particular dataset, describe how UDFs can be parallelised (by tiles, by time series etc.) • describe how (e.g. R, Python or JavaScript (JS)) UDFs are exposed to data, on the server side • validate (verify) a given UDF
6. Process execution • execute a process on a small subsample of the data; • execute a process on the full dataset
7. Process monitoring • request the status of a running process • subscribe to a message publishing service that notifies if a process is ready
8. Asynchronous user interaction • handle website updates, e.g. of a result map while computing takes place

RELY on OPEN SOURCE SOFTWARE/TOOLS/MODULES/PLUG-INS

openEO is to be released as an open source project with the permissive Apache license 2.0[2]. The code will be hosted on a public GitHub repository[3] to directly allow for the involvement of a wider user and development community beyond the project consortium. This has the benefit that a user and developer community will grow during the project, and that a larger group than the project consortium will contribute to consensus regarding

[2] https://www.apache.org/licenses/LICENSE-2.0
[3] https://GitHub.com/open-EO

the openEO design. In addition, more processes may be interfaced and implemented by users, and new commercial initiatives may reuse openEO after the project has been completed. A more heterogeneous user community may also lead to the implementation of more generic data processing commands in the future. Providing an open, transparent API will make it attractive for commercial activities to adopt this interface.

BRIDGE GAP between EO and IT SECTORS

openEO is designed as an interdisciplinary project, bringing together EO data scientists with IT specialists and innovative enterprises. The participating organizations have extensive experience with the design of software, the fostering of open source projects, large-scale computing of high-resolution EO data, deployment of IaaS and PaaS services, and the operation of Petabyte-scale storage and processing systems. This mix of expertise will imprint on the design of openEO and will shape its user and development community beyond the project's lifetime.

COMPLEMENATRITY to ICT ACTIVITIES in the AREAS of DATA MINING, OPEN LINKED DATA, WEB ONTOLOGY, DIGITAL EARTH

With its focus on delivering concrete solutions for the EO domain based upon solid IT (Information Technology) expertise, and benefiting from the computational power of major European research infrastructures, openEO is highly complementary to European initiatives and programmes in the ICT (Information and Communications Technology) domain. The tools for chaining source code will be delivered in the programming languages R, Python, and JavaScript. R and Python are the main platforms for data science, and major platforms for implementing and comparing data mining algorithms. For those back offices that allow this, openEO will let end-users program their computations on the cloud service in R or Python, and pass these user-defined functions in the call to the cloud service. This will make it possible to apply data mining algorithms available in R and Python directly on Copernicus data, in a scalable way. Existing open linked data vocabularies (such as the data cube vocabulary) and relevant web ontologies (e.g. the semantic sensor ontology) will be used where fit for the purpose. Linked data technology and the PROV ontology[4] will be used when provenance documents are created.

ADDRESS RELEVANT ASPECTS of EO DATA LIFECYCLE

Having a single openEO interface will make it possible to communicate, exchange, reproduce and publish complete EO data processing workflows addressing all relevant aspects of the EO data life cycle (see *Table 1*). openEO can be expected to become a flexible coverage and open processing standard. For example, an important part of the openEO interface is the discovery of the datasets available in a processing service. The tools for chaining up of use cases – on the client side R and Python scripts and on the driver side sequences of API service requests – will be provided and stored for provenance, so that the complete processing chains are understood by users of the front-end and managers of the back-end.

PARTICIPATION of INDUSTRY, in PARTICULAR SMEs

SMEs (small and medium enterprises) play a crucial role in the European EO sector. They are innovation drivers, have a good understanding of EO data user needs and often keep close ties to the scientific community. Hence, it is no surprise that several members of the openEO consortium are SMEs. The adoption of existing and new use cases to the openEO standard will allow them to use openEO commercially after the project's end. A strong involvement of end users in the project's decision-making processes will lower possible restraint of additional SMEs to use openEO in the future.

[4] https://www.w3.org/TR/prov-o/

ENGAGEMENT of INTERMEDIATE and END USERS

The interest of both intermediate and end users in openEO is demonstrated by the willingness of several user organizations to support the project. The easier access to Copernicus data and value-adding services via openEO will allow them (and other users) to use Copernicus data much easier than it is now. Thanks to the new openEO interface, the integration of Copernicus data in interdisciplinary research projects will be facilitated. The easier switch between various services (thanks to the standardised openEO syntax) decreases the barrier of a collaboration between diverse user communities. For commercial use cases, a further major advantage that the pricing of different processing services will become transparent and comparable. For scientific users, the fact that openEO will develop client APIs to R, Python or JavaScript is key. R and Python are today the main tools of data scientists, while JavaScript is a major language for developing web and mobile apps. In case the EO data processing service to be used is non-free, the only thing that has to be set up is a user account and where needed a payment method, with the service. This is a tremendously lower entry barrier compared to current practice, where (i) R and Python cannot be directly used, (ii) each cloud service not only has a different API but uses different terminology to describe its capabilities, and (iii) pricing models are not comparable.

1.3 CONCEPT and METHODOLOGY

(a) Concept

EO data collected today is so large that the legacy procedure of downloading the data to analyse them locally is being substituted by a workflow that involves accessing a cloud platform where the data has been downloaded and organized, and processing the data there. Setting up such a cloud platform is not particularly difficult, but involves skills that are rarely available to those who need to analyse EO data. In addition, it is an economic risk to set up a cloud platform, and it requires effort to create and sustain a large enough group of users.

Being in the early days of the adoption of cloud-based EO data processing, we now see a very large heterogeneity in cloud platform offerings, not only in terms of the data they offer but also in terms of how the data are organized, which analysis methods are offered and how they are accessed, how the interface to the cloud platform works, and what the pricing is. This heterogeneity has a large number of adverse effects:

1. Users have a hard time to rationally choose between cloud services,
2. Cloud services providers find it difficult to make competitive offers,
3. Scientists have to work extremely hard to independently verify the correctness of published results,
4. Scientists have difficulty to present reproducible workflows, required for EO science to be an *Open Science*,
5. EO novices face big challenges trying to use EO data.
6. The main data science software clients used today (R, Python, JavaScript) lack interfaces to EO cloud services due to a missing unified interface.

Despite all these adverse effect, the pre-processing of downloaded data and analytical capability offered by cloud services are very similar, and typically include providing

1. a set of downloaded scenes that is continuous over space and covers a large time span and all spectral bands,
2. access to such collections using simple, standardised or well-known names,
3. simple ways to create cloud-free coverages, e.g. by median filters, or standard algorithms for cloud removal,
4. simple ways to implement algorithms operating on individual pixels, as well as on spatial, temporal and/or spectral dimensions.

The openEO project aims to counteract the adverse effects of the currently existing heterogeneity. It will do this by developing a simple layered API that connects cloud services (back offices) to clients (front offices). Process libraries will be implemented in R and Python, so users can use this API to integrate cloud-based EO analysis in their respective processing chains, essentially controlling the cloud-based processes by openEO-interfaced R and Python scripts, and retrieving the results from the analysis. An additional JavaScript client API will make it trivial to integrate openEO services in mobile and web applications.

Pilot use cases of R and Python processing chains will be implemented by adapting already existing workflows to the openEO syntax. In addition, workflows will be altered towards the use of the JavaScript client API. This serves several purposes. It will be a validation that users are able to connect already existing fully functional EO process chains to several back office service providers, will build up a first core user community, and will serve as template for future use cases of the EO community.

For boosting the generation of an EO user community, we will use multiple platforms to foster communication with and within the user community. The openEO source code will be developed in a public repository and therefore be accessible from day one, and all ideas as well as project progress will be communicated openly. This will allow the EO user community to take part in the project's decision-making process.

The software components that will be delivered will be brought to Technology Readiness Level (TRL) 6 – technology demonstrated in relevant environment (industrially relevant environment in the case of key enabling technologies) or 7 – system prototype demonstration in operational environment. This means that the software will work, will be tested and will be robust, and can be exploited in a commercial setting. Since it involves open source software for solving a very broad class of problems, it is expected that the development of the software will be continued by a community consisting of both EO data researchers and cloud providers. More drivers and clients are expected to be developed and more processes are expected to be interfaced.

The openEO project links to various national and international research and innovation activities. These include:

- GDAL[5] is possibly the largest open source project for spatial data formats; it provides read and write access to all possible file-based spatial data as well as selected SQL databases. openEO has a similar ambition, but interfaces cloud-based EO data access, and data processing. We carefully studied the reasons for the success of GDAL when designing openEO.
- OSGeo, the Open Source Geospatial Foundation[6] "supports the development of open source geospatial software, and promotes its widespread use". OSGeo focuses on long-term provenance of community geospatial software; we will adopt its guidelines to allow a smooth adoption of openEO as an OSGeo project by the time the project reaches its goals.
- The e-sensing project[7] run at INPE[8] has as its goal to "Conceive, build and deploy a new type of knowledge platform for organization, access, processing and analysis of big EO data." The group carrying this out has strong connections to the University of Muenster, where Gilberto Camara held a guest professorship from 2013-2015.
- R-spatial, the community of users and developers who use R for analysing geospatial data.
- Front-end libraries such as ESA-NASA WebWorldWind[9], Leaflet (an open-source JavaScript library for mobile-friendly interactive maps)[10] and similar.

[5] http://www.gdal.org/
[6] http://www.osgeo.org/
[7] http://www.esensing.org/
[8] http://www.inpe.br/
[9] https://nasaworldwind.github.io/
[10] http://leafletjs.com/

(b) Methodology

To reach the aims described above, we plan to address them in seven work packages (see section 3.1). Firstly, we will release a website and set up social media accounts to interact directly with the users on social media (WP 2) and in the code repository to promote the project's progress. This will allow us to gain information about the extent and orientation of interested future openEO users. We will use conferences and seminars to promote openEO and to expand the potential openEO user community. Special surveys among EO data users will give us a more detailed knowledge about specific questions that arise during the project.

We will build the openEO API, consisting of three different layers (WPs 3, 4, 5) of modular APIs to connect the EO community with back office service providers in standardized ways. The developers of the layers will collaborate intensively to get an optimal understanding of the way our users will work with the client APIs and of the specific back-end services of the different back offices addressed in this project.

Next to understanding the requirements from both sides (i.e. users and back offices) which will lead to the API definition, we will develop and release a Proof of Concept in the first six months of the project by intensive collaboration between developers of the different client APIs and back offices (see Figure 2). The Proof of Concept will consist of prototypes demonstrating the following core features:

- Submitting e.g. client side Python code to the back-end for executing,
- Defining processing chains that are evaluated on the back-end,
- Viewing of (intermediate) results either as raster data on a map, or plotted in a graph (histograms, time series), and
- Possible additional topics raised by users during the first months of the project (Task 2.1) or from the analysis of the different back offices (Deliverable 4.1)

These topics are chosen because we see them as the basic functionality, which have to be covered by the openEO API at the end of the project.

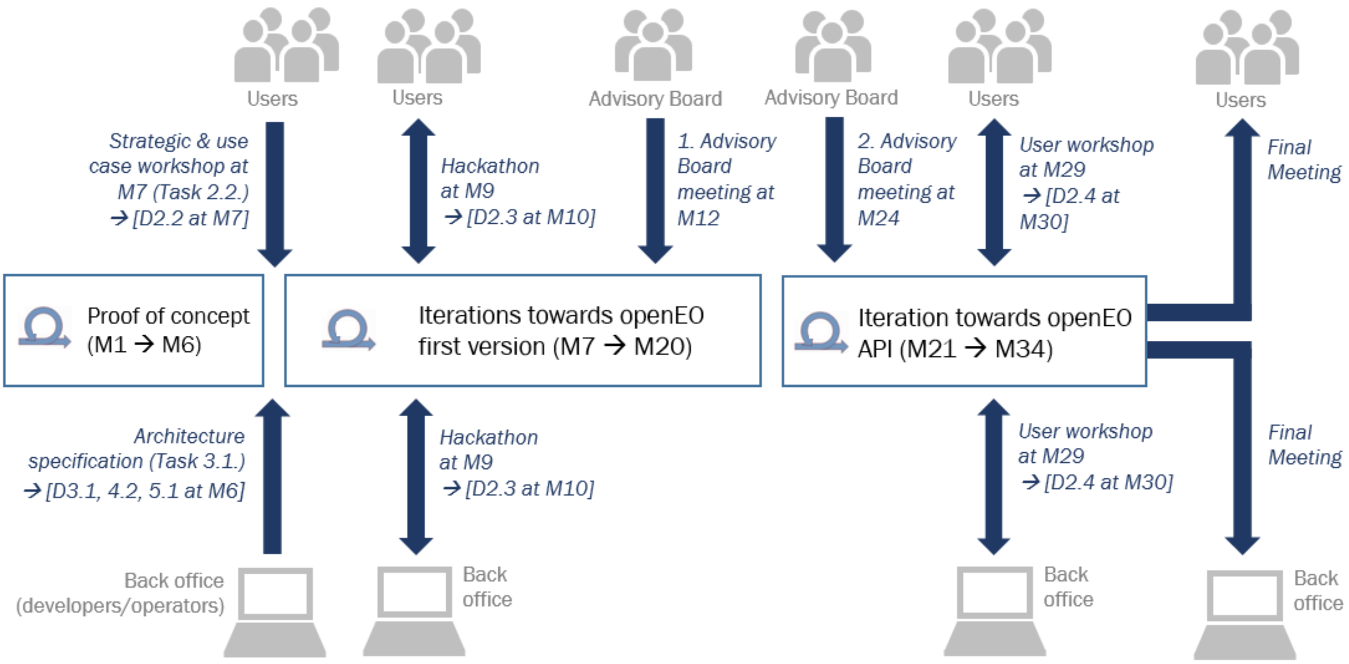

Figure 4: Project workflow and opportunities for direct communication with users and back office providers

Figure 5: 'Bayesian approach' for combining optical and RADAR time series and for near real-time deforestation detection.[1] Once a new observation from any sensor becomes available, deforestation events are indicated and the probability of deforestation is calculated. Newly acquired observations are used to iteratively update the probability of deforestation, and, thus to confirm or reject indicated deforestation events.Figure 6: Project workflow and opportunities for direct communication with users and back office providers

In month 6 the results from this Proof of Concept will be clear, together with the study of the back offices and the user requirements. The first user workshop is scheduled in month 7 to discuss the results of these three input streams to the project. Until the end of the first project year a product backlog of the most urgent features will be implemented as consolidated at this first user workshop; a hackathon in month 9 is organised to collect again feedback on what works and what should be improved, by real testing by external users. At the ends of the first and second project years we will hold a meeting with the Advisory Board, consisting of experts from EGI, ESA, and JRC. Goal is getting additional input from external experts. Until month 20 we will design, implement and test in an iterative approach a first version of the openEO API with a clear backlog. This will be done by gradually adding features (as user authentication, data discovery, data import/export, auxiliary data management, etc.; see *Table 1*) to the original Proof of Concept and will be aligned with the extraction of user requirements. The following use case implementation, based on the first version of openEO will be handed over to the pilot users at month 26 for further testing to identify and fix remaining flaws.

Until month 34 we will further improve openEO in an iterative approach towards a final version. A user workshop around month 29 openEO will be used to introduce openEO to a broad audience and to build up a post-project openEO Steering Committee.

The crucial aspects of openEO will be the definition of the Core API and its services (see *Table 1*). An application programming interface (API) can be seen as a contract between two computers, a client and a server: It specifies what the client can ask from the server and how it should do this, and what (and how) the server shall respond. A simplified example dialogue between client and server may be:

Client: “Which datasets do you have?”

Server: “Landsat 8, Sentinel 1A, Sentinel 1B, Sentinel 2A, Sentinel 2B”

Client: “Describe Sentinel 2A”

Server returns the description

Client: “Which operations can I do on this dataset?”

Server: “Derive a vegetation index”

Client: “Here is a polygon”

Server: “Thank you; I will call this Polygon1”

Client: “Here is a task: for all pixels of Sentinel 2A inside Polygon1, derive a time series and perform a trend analysis and return the distribution of the results”

Server: “Thank you; I will call this Task1”

Client: “How long will it take and how much will it cost to carry out Task1?”

Server: “1 hour, € 0.25”

Client: “Carry out Task1, and return results for downloading.”

Server carries out Task 1, returns results, reports remaining credit.

To implement this, RESTful[11] services will be used for communication, utilizing JSON as a data-interchange format; authentication protocols will include OAuth2. A pub/sub mechanism will be used to inform clients about the status of long-running, asynchronous compute jobs. It will be possible to test long-running compute jobs on smaller data samples (e.g. on a low spatial resolution), to speed up development and minimize the risk of wasting computing credits. Direct visualization of resulting computations e.g. in Leaflet html widgets will be developed for interactive visual feedback. Spatial, temporal and spectral sub setting will be supported, as well as band arithmetic. Executing user-defined functions defined as R, Python or JavaScript functions will be supported for parallel processing, if applicable and where supported by the back office. Lazy evaluation - postponing the computation of results until results are actually fetched - will be supported where possible. Coordinate reference system transformations, grid resampling and warping, and combining image coverages with different spatial and temporal resolution (e.g. from different satellites) will be supported as well as the handling of missing values in coverages or resulting from computations.
Within the Proof of Concept and the different iterations the developers of the different partners will collaborate intensively in joint bi-weekly sprints. These will be followed by videoconferences between the participants of work packages 3, 4, and 5 to coordinate the integration between the three openEO layers and to discuss the next steps. GitHub is used to share the source codes amongst the virtual team, which will meet regularly during the first six months (Proof of Concept) in ‘weeks of intense collaboration’ e.g. by bringing the developers together at the premises of one partner. Over the different sprints within an iteration we have continuous integration of features and adopt a test-driven development approach. This agile way of developing is used already today by the different partners in various projects and has demonstrated its usefulness.

To realize the Proof of Concept and later the final openEO API, a critical challenge will be to name and categorize the resources in the interface, where resources include:

[11] Representational State Transfer service type

- datasets, and their properties
- operations/functions/processes that can be carried out on datasets, and their parameters
- administrative resources like users, available computer time, disk space and compute credit

For naming and organizing datasets and analysis methods we will follow as far as possible the available best practices, and will carefully study example implementations, such as

- Currently available R or Python interfaces to EO data analysis
- The naming schemes adopted in Google Earth Engine
- The categorization of array manipulation operations provided by the computational database SciDB[12]

The ability for users of one of the clients to pass on a complete (R, Python or JavaScript) script with user-defined functions to the service back-end seems to solve one class of problems of having to search for the appropriate functions, but creates some new challenges too:

- The script has to know (and needs to be informed by the driver API) how the data it will receive on the service back-end is organized: it will have to work on data chunks, and in parallel -- the openEO project will develop the necessary standardisation in this respect;
- Not all service back-ends will allow for this option, because it potentially gives client-side users too large freedom over the service back-end; Google Earth Engine and SciDB without the stream extension are two examples where currently users are not allowed to run arbitrary scripts on the server side.

These challenges will be addressed in the different driver APIs (WP 4).

(c) Use cases

To ensure usability of the openEO API and to generate a first user community, four already operative use cases will be adapted to the openEO syntax in this project. These use cases will be validated by five pilot users, who will give feedback on user-friendliness, stability, and flexibility of the openEO API. Based on this feedback we will be able to improve the API in an iterative approach. The pilot users (i.e. BMLFUW, ACF, ICIMOD, FAO, Prov. Bolzano) cover a broad spectre of governmental, non-governmental, and environmental users and will validate the implemented processing chains in different regions in Europe, Africa, Asia, and Latin America. Thus, we expect to diversify this first user community to further serve as seeding points as diverse as possible. That will encourage a fast growing future openEO user community.

[12] http://www.paradigm4.com/

Pilot use case 1: Radar Image Compositing

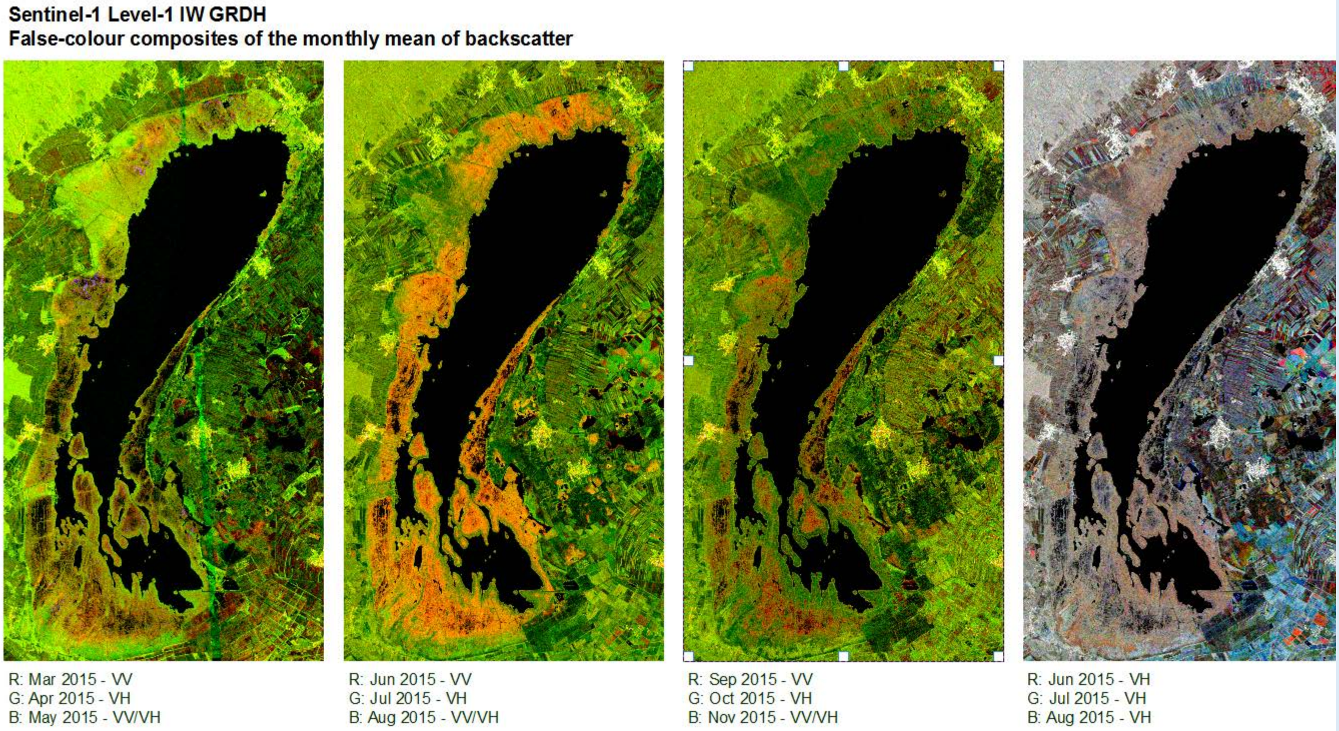

***Figure 7**: Image composites of Sentinel-1 data, showing the Neusiedler See, Austria*

The *Radar Image Compositing* use case will produce monthly and seasonal RGB composites of Sentinel-1 backscatter[13]. The workflow for producing these composites is the following:

1. Gather all Sentinel-1 observations for different polarizations (e.g. VV, VH) in the period of interest e.g. of one month
2. Perform quality checking and masking of NaN values in the input data
3. Calculate statistics like mean or median on the resulting stack of images
4. Store the products as RGB composites. There are several options to assign data to colour channels, one possibility is storing subsequent months in time as different colours e.g. R=June, G=July and B=August. Another is to make composite images for the same time span but using different polarizations e.g. R=VV_{July}, G=VH_{July}, B=VV_{July}/VH_{July}

Depending on the used compositing method these images (see ***Figure 3***) can be used for classification and crop monitoring[14,15]. This relatively simple use case provides a test for basic openEO functionality like querying data, data transformations, basic statistic, creation of multi-band images and export to a specific output format.

[13] Sabel, D.; Bartalis, Z.; Wagner, W.; Doubkova, M.; Klein, J.-P. (2012): *Development of a Global Backscatter Model in support to the Sentinel-1 mission design*, **Remote Sensing of Environment** 120, pp. 102-112.
[14] Nguyen, D.; Clauss, K.; Cao, S.; Naeimi, V.; Kuenzer, C.; Wagner, W. (2015): *Mapping Rice Seasonality in the Mekong Delta with Multi-Year Envisat ASAR WSM Data,* **Remote Sensing** 7, pp. 15808-15893.
[15] Naeimi, V., Hasenauer, S.; Cao, S.; Bauer-Marschallinger, B.; Dostalova, A.; Schlaffer, S.; Wagner, W. (2014): *Monitoring water resources using big data from Sentinel-1 satellites*, **Big Data from Space (BiDS'14)**, 12-14 November 2014, European Space Agency-ESRIN, Frascati, Italy, doi: 10.2788/1823, pp. 146-149.

The functionality of this use case on the EODC and GEE back offices will be tested on behalf of and by the *Austrian Federal Ministry for Agriculture and Forestry, the Environment and Water Resources* (BMLFUW) for regions in Austria.

Pilot use case 2: Multi-source Phenology toolbox

The *Multi-source Phenology Toolbox* use case will port available data fusion and phenology metrics tools (see ***Figure 3***) to the openEO platform. Data fusion techniques for Sentinel-2 data are e.g.

- merging with moderate resolution high temporal datasets (Walker et al. 2014[16])
- merging with radar datasets that can look through the clouds (Joshi et al. 2016[17])

This use case will increase the consistency of Sentinel-2 derived phenology parameters by combining Sentinel-2 time series with optical moderate resolution data (Sentinel-3 and Proba-V). The phenological metrics will be checked against ancillary datasets as is irrigation information, rainfall and soil moisture information.

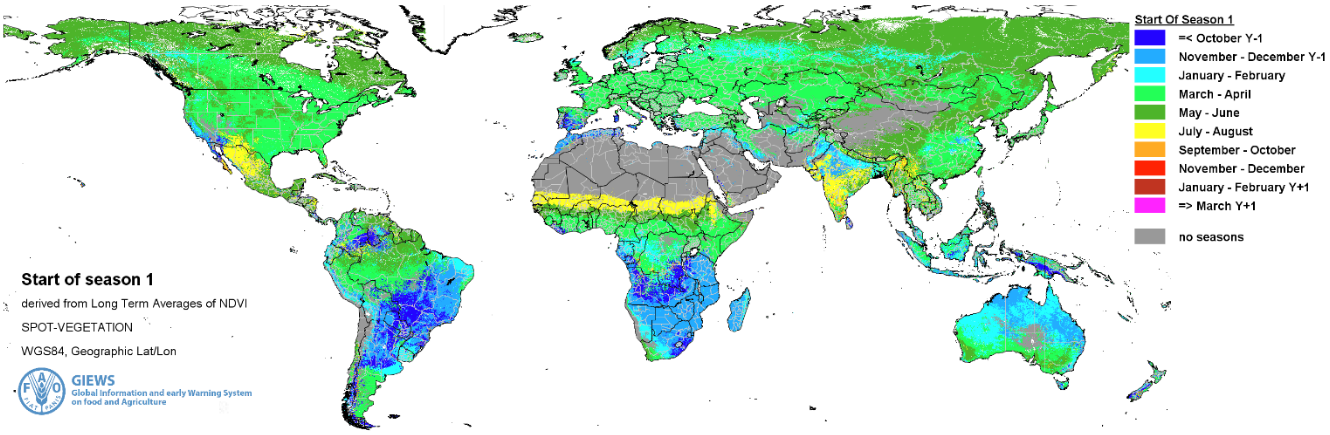

Figure 8: *Start of a growing season, derived by the Multi-source Phenology toolbox SPIRITS*[18]

A number of libraries shall be available through the Python openEO interface:

- Image pre-processing (scaling, atmospheric correction, cleaning, …)
- Band math operations
- Time-series extraction and metrics calculation (e.g. Harmonic Analysis of Time Series, aggregation)
- Regressions (e.g. regression weights for two time-series)
- Fusion (e.g. kalman, theano, keras)
- Plot functions

The functionality of this use case on the VITO and EODC back offices will be tested across Western-Africa for the *Action Contre la Faim* (ACF) as a pilot user, covering sparse vegetation and high density clouds. In addition, the *International Centre for Integrated Mountain Development* (ICIMOD) will test the toolbox to generate time-series of vegetation indices.

[16] Walker, J.J.; de Beurs, K.M.; Wynne, R.H. (2014): *Dryland vegetation phenology across an elevation gradient in Arizona, USA, investigated with fused MODIS and Landsat data*, **Remote Sensing of Environment** 144, pp. 85-97.
[17] Joshi, N.; Baumann, M.; Ehammer, A.; Fensholt, R.; Grogan, K.; Hostert, P.; Jepsen, M.R.; Kuemmerle, T.; Meyfroidt, P.; Mitchard, E.T.A.; Reiche, J.; Ryan, C.M.; Waske, B. (2016): *A review of the application of optical and radar remote sensing data fusion to land use mapping and monitoring*; **Remote Sensing** 8(1), pp. 1-23.
[18] Reiche, J.; de Bruin, S.; Hoekman, D.; Verbesselt, J. (2015): *A Bayesian Approach to Combine Landsat and ALOS PALSAR Time Series for Near Real-Time Deforestation Detection,* **Remote Sensing** 7(5), pp. 4973-4996

The following subtasks will be performed to release the use case:

- Port the SPIRITS and/or TIMESAT phenology tool (time series) using the openEO Python client API and test with moderate (Sentinel-3 and Proba-V) and high (Landsat-8, Sentinel-2) optical resolution datasets
- Port data fuse tools (e.g. Kalman filtering, Neural Networks) to Python using the openEO client API
- Generate multi-source phenology metrics
- Ingest ancillary datasets in SciDB through the client API
- Develop different correlation metrics to link phenological information with ancillary information

Pilot use case 3: Optical-Radar Forest Monitoring

The *Optical-Radar Forest Monitoring* use case will focus on testing and scaling-up the "Bayesian approach" (open-source, see ***Figure 4***) to combine Sentinel-1 and Sentinel-2 time series for near real-time tropical deforestation monitoring[19] (see ***Figure 5***). It provides a test for basic (querying data, data transformations, basic statistic, creation of multi-band images and export to a specific output format) and advanced openEO functionalities. The following subtasks will be performed to release the use case:

- Port the "Bayesian approach" using the openEO R and Python client APIs and test with Sentinel-1 and Sentinel-2 data
- Implement and test near real-time deforestation system using the openEO functionalities
- Test and compare the access of Sentinel-1 and Sentinel-2 data from different back-ends

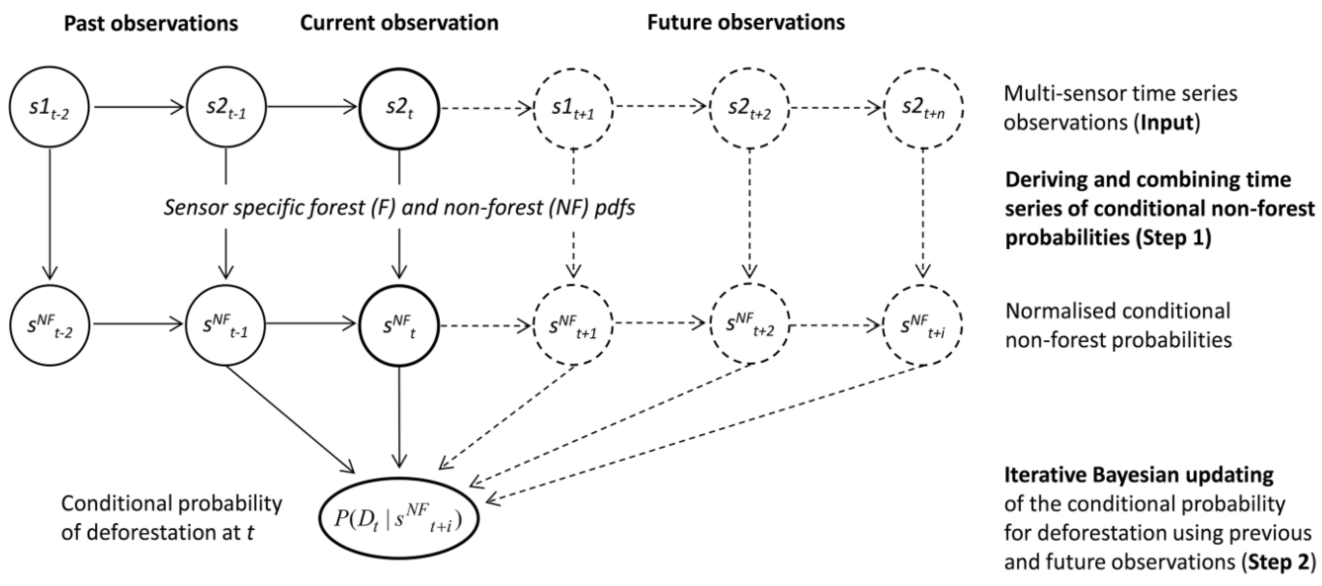

***Figure 9**: 'Bayesian approach' for combining optical and RADAR time series and for near real-time deforestation detection.[1] Once a new observation from any sensor becomes available, deforestation events are indicated and the probability of deforestation is calculated. Newly acquired observations are used to iteratively update the probability of deforestation, and, thus to confirm or reject indicated deforestation events.*

[19] Reiche, J. et al. (in review): *Near-real time deforestation detection in tropical dry forest combining Landsat, Sentinel-1 and ALOS-2 PALSAR-2 time series*, **Remote Sensing of Environment.**

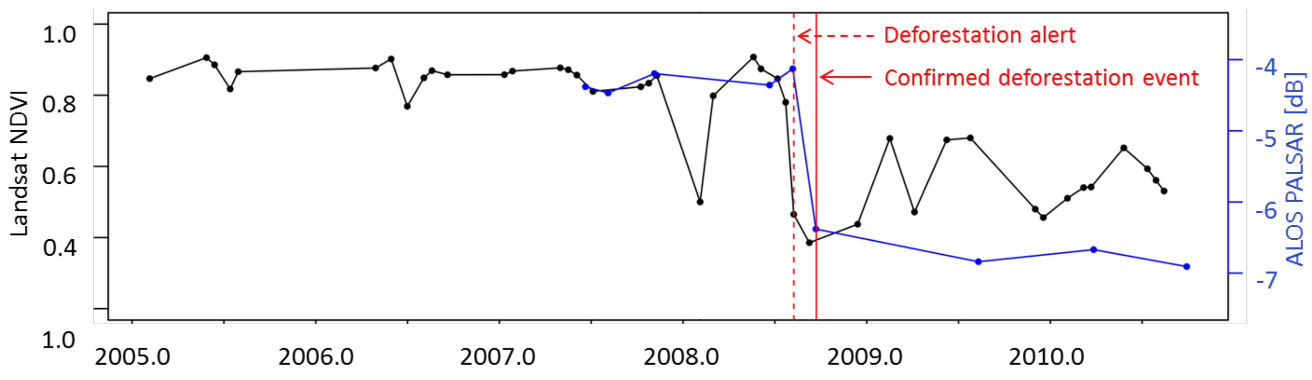

Figure 10: *Bayesian approach applied to a Landsat NDVI[20] (black dots) and ALOS PALSAR L-band backscatter (blue dots) single pixel time series to detect deforestation NRT. For the deforestation event in July 2008 an alert was provided using the first observation acquired after the event and the deforestation event was confirmed shortly after.* [21]

The use case will be tested in Madre de Dios, Peru on several back offices (e.g. EODC, GEE, AWS), whereby the *Food and Agriculture Organization of the United Nations* (FAO) will be the pilot user. Madre de Dios (85.000 km²) is situated in the southeast of Peru and is characterised by humid tropical forest. The region suffers from large-scale illegal deforestation activities and a rapid degradation of environment, e.g. due to mining[21]. For a test period, deforestation alerts will be provided in an operational framework to FAO. The results will be evaluated by jointly assessing the spatial[22;23] and temporal accuracy of the detected changes. FAO will provide some key reference data.

Pilot use case 4: Snow monitoring with Sentinel- 1 and Sentinel -3

The *Snow Monitoring* use case will focus on testing innovative algorithms for detecting snow cover and snow status based on the combined use of Sentinel-1 and Sentinel-3 time series[24, 25] (see ***Figure 7***). It provides a test for basic user functionalities (user authentication, data discovery, data management, data processing) and will be used to test the data cube back-end and the openEO specific drivers. The Hydrological Office of the Autonomous Province of Bolzano will be the pilot user. In particular, EURAC research is already running a snow monitoring NRT (near real time) operational service on cloud resources, but the technicians from the Hydrological Office need to get pushed dataset in their own system in order to feed their modelling and decision systems. The NRT service push via ftp daily snow maps at coarse spatial resolution (i.e. 250m).

---

[20] Normalized Difference Vegetation Index

[21] Brack, A.; Ipenza, C.; Alvarez, J.; Sotero V. (2011): Minería Aurífera en Madre de Dios y Contaminación con Mercurio - Una Bomba de Tiempo, **Ministerio del Ambiente**, Peru

[22] Foody, G.M., (2002): Status of land cover classification accuracy assessment. **Remote Sensing of Environment**, 80(1), pp.185–201.

[23] Olofsson, P.; Foody, G.M.; Herold, M.; Stehman, S.V.; Woodcock, C.E.; Wulder, M.A. (2014): *Good practices for estimating area and assessing accuracy of land change*, **Remote Sensing of Environment** 148, pp. 42-57.

[24] Marin, C.; Callegari, M.; Notarnicola, C. (2016): *A novel multi-temporal approach to wet snow retrieval with Sentinel-1 images*, **SPIE Remote Sensing**, Edinburgh, 26-29 September 2016, Edinburgh, United Kingdom.

[25] Callegari, M.; Marin, C.; Notarnicola, C.; Carturan, L.; Covi, F.; Galos, S.; Seppi, R. (2016): *A multitemporal probabilistic error correction approach to SVM classification of alpine glacier exploiting Sentinel-1 images*, **SPIE Remote Sensing**, Edinburgh, 26-29 September 2016, Edinburgh, United Kingdom.

By performing the following subtasks, openEO will enhance the workflow for both EURAC research and the Hydrological Office:

- Data access, data discovery, data management within one single interface will be enable to the Hydrological Office via the openEO.
- EURAC research will be able to run their algorithms on any back-ends correctly interfaced with the OpenEO and where the input data are stored.

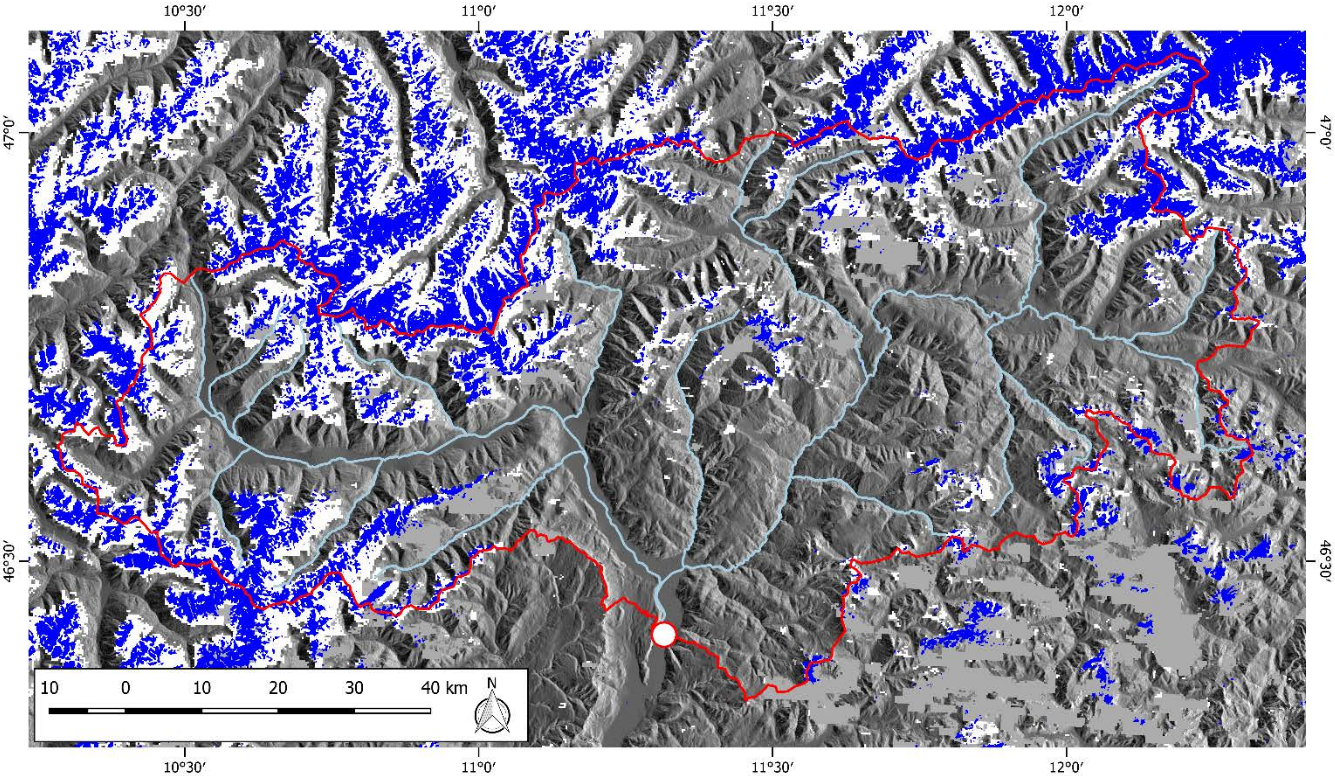

Figure 11: Snow maps over the South Tyrol region (Italy) indicating the snow cover extent (white) and the wet snow extent (blue)[24]. The wet snow area was obtained with Sentinel-1 images while the snow cover area with MODIS images in substitution of Sentinel-3 as at the moment a time series of Sentinel-3 data is not yet available for the fusion time span.

In this way, the snow monitoring service will be upgraded with the Copernicus program data, and the openEO will enable the upscale of the service for higher resolution snow monitoring products, enabling the user to access directly processed Sentinel data without having the need to reserve local dedicated resources.

The functionality of this use case on the EURAC and EODC platforms will be tested in the areas of South Tyrol region (Northern Italy). For a test period, the snow maps will be provided in an operational framework to the Province Office. The maps will be evaluated and validated by using the meteorological stations located on the areas and with dedicated field campaigns. Thus the openEO API will allow the Hydrologic office of the Province of Bolzano to enrich their downstream services.

(d) Gender equality

Within the last years, a lot of progress has been made within Europe to further increasing gender equality. Following the EC's *Strategic engagement of gender equality*[26], the openEO consortium will promote gender

[26] http://ec.europa.eu/justice/gender-equality/files/documents/160111_strategic_engagement_en.pdf

balance throughout the project duration. openEO will be open to all kinds of users, regardless of age, gender, race or origin and will be promoted via social media, thereby accessible and discussable for open public. There is no risk of stereotyping as no role models will be imposed and the open nature of the project does not facilitate the exclusion of certain groups. Within WP 2 (User Involvement & Dissemination) the diverse expertise, interests and needs of users will be taken into account and addressed within the design and development of the product. openEO will seek to compose a diverse team to ensure gender neutral coding and with this providing the opportunity to further extend the initial user group after project end.

All outcomes of openEO are equally important to women and men. The institutions involved in this project therefore take several measures in their work environment to address the improving of gender equality in careers. These measures involve e.g. flexible working hours, tele-working or support of child care (such as an in-house kindergarten or holiday programs). Specifically, TU Wien, WWU and WUR as (partly technical) educational institutions have a responsibility to support and increase the percentage of women in technical fields. One remarkable initiative of TU Wien is "FIT – Frauen in die Technik"[27], which enables female pupils to visit TU Wien and other partner institutions (universities as well as companies) for three days to thereby receive a valuable insight into different technical subjects. This event takes place every year in January and is organised by *genderfair!*[28], an initiative of TU Wien to promote equality of opportunities.

The Faculty of Mathematics and Geoinformation of TU Wien recently developed a strategy to address the advancement of women, which also includes the establishment of a mandatee for gender equality within the Department of Geodesy and Geoinformation. Alexandra von Beringe, who will be the administrative manager of openEO, is currently holding this position. The adequate addressing of eventual questions related to gender is therefore ensured.

Within the Remote Sensing group of TU Wien another initiative has been successfully implemented in the last years. Every summer a female student receives the Women4GEO[29] award to actively increase the proportion of women within the research group. With this award, the selected student has the possibility to work on a self-proposed project idea under supervision of the research group.

1.4 Ambition

(a) Interoperability

openEO will provide *interoperability* between data formats, back offices, and EO data processes of different programming languages. Venkatesh Raghavan from the OSGeo Foundation e.g. describes the overall concept as "... well reflecting the ideas of open science and interoperability ..."[30] A key geospatial example of practical interoperability is the *Geospatial Data Abstraction Library* (GDAL). It implements an API to read 152 and write 79 different raster formats. Any software working with raster data through this library can read and write data in all supported formats. Distributing raster data in a new format (as it was done e.g. for Sentinel 2) is not a problem when a GDAL driver becomes available for this format. The GDAL raster format does not map to a formal standard (in the sense of ISO[31] or Open Geospatial Consortium, OGC) yet makes it possible to work with different raster file formats. It is a *de facto* standard, defined by an implementation, just like R and Python are *de facto* standards.

An example of a format standard that *could be* of use to EO scientists is the Web Coverage Processing Standard. The reason why it is not adopted by the community is (i) its complexity, (ii) the lack of an open source implementation that supports distributed, scalable computing or user defined functions and (iii) lacking

[27] http://www.fitwien.at/
[28] http://www.tuwien.ac.at/en/services/gender_studies/genderfair/
[29] http://geo.TU Wien.ac.at/women4geo/
[30] See section 6, part B2 (Letters of Support)
[31] http://www.iso.org/iso/home.html

correspondence between processes implemented and processes needed by the EO community. As opposed to e.g. engineering and building standards where standards follow from safety regulation, in the geospatial world, the widely adopted formal standards have always followed broadly accepted open source implementations; examples are Web Map Services (WMS) and the simple feature access standard to specify common storage and access models of two-dimensional geometries. Many other standards were pushed by single parties for political or commercial reasons, and never found broad adoption or use.

Although GDAL is extremely relevant for EO data analysis because the basic element of distributing these data is the raster file (scene), it has its limitations when we want to use it for scalable, cloud-based processing of large scale imagery. Its limitations are: (i) Its data model has no information about the time an image was taken, the sensor that was used, or the spectral bandwidth of layers, (ii) it does (practically) not provide functions for processing data, and (iii) its model is tied to having the data at least partially in memory, instead of leaving open that the data is distributed over multiple computers. Also Noel Gorelick from Google Earth Engine recognises "... the shortcomings of the existing standards when they are applied to current and emerging petabyte-scale data catalogues and high-performance processing systems."[30]

(b) Merging standards for EO analytics

openEO has the ambition to become *a GDAL for EO analytics*[32].

To realise this ambition rather than start writing another standard, we will start by developing a working solution by

- developing software that binds a variety of client software, commonly used by data scientists to a variety of cloud-based computing services, by using a single API
- developing this to solve our own problems, but with the intention to solve the same problem for all other users with similar requirements and constraints
- starting openEO as an open-source, community-based project from day one.

After we have implemented successfully a Proof of Concept, and when the development has stabilised sufficiently, we will take the steps necessary to make the API into a formal standard, and the software into an Open Source Geospatial (OSGeo) endorsed project. OSGeo endorsement would make the rules for source code governance clear, and keep them out of the hands of single individuals or individual organisation. Formal standardisation would ratify the standard by a standardisation body, which might e.g. help certain government agencies to adopt it.

Wanting to create a GDAL for EO analytics is of course an ambition of enormous magnitude, but one should not forget that GDAL also started small: in the early days it supported only raster, and few drivers. Working with a permissive open source license in order to remain attractive for industrial application, combined with the strong, permanent engagement of the open source geospatial community to control quality have made it into the comprehensive de-facto standard, which it is today.

(c) The right timing

Despite the magnitude of its ambition, the timing for starting openEO could not be better:

- Surfing the wave of big data and data science, cloud computing comes with strong and standardized offerings that involves dedicated tools for data analysis
- Several large cloud providers already offer, or will soon offer, fast access to large archives of satellite imagery with no need for downloading
- Given the availability of these archives, it has never been so easy to set up satellite imagery processing services

[32] See proposers' blog post of Nov 29, 2016 at http://r-spatial.org/2016/11/29/openeo.html

- The current situation, with no openEO standardized processing API, means that offerings and their pricing are maximally fragmented: they are completely incomparable, and EO analytics carried out on such clouds remain unverified, and unreproducible
- A strong group of users of open EO data and geoinformatics experts has formed to support this proposal
- Google Earth Engine, a large EO processing service with an easy interface, has expressed interest in joining the development, and potentially supporting a common, open API. The openEO architecture allows for working with both open-source and closed-sourced clients, or back-ends
- The European Commission's "Open Science Cloud" vision perfectly matches the ambitions of openEO to draw the practice of cloud-based EO data processing out of its engineering stage, into an open science phase
- The European Commission, with the "EO Big data shift" call, actively calls for an openEO initiative.

2. IMPACT

2.1 EXPECTED IMPACT

In the following the expected impacts according to the points listed in the call text are discussed.

Impact 1: *Enable value adding services on generic data and information storage and processing facilities, which can allow public and commercial users effective production environments to interact with and serve their user base without deploying their own storage and processing facilities.*

The first impact matches exactly the goal of openEO: By standardizing an API for cloud-based EO processing engines, both public and commercial users are empowered to use a standardised interface to carry out processing, without having to deploy storage or processing facilities. This of course assumes others deploy and offer these facilities, but given that they can do this under a standardized API, they will immediately reach a large customer base. To realize this goal during the project implementation phase, a number of open source openEO back-ends will be implemented and made available; involvement from users and developers outside the project consortium will be encouraged: road maps, project planning and development platform (GitHub) will all be public, from day one.

Impact 2: *Make access to the Copernicus data and information easy and user friendly through scalable dissemination and exploitation software based on international standards.*

The core idea of openEO is to standardize the interface of Copernicus cloud processing services, so that users with standard data science tools (R, Python, JavaScript) will be able to directly access these services and integrate them in their regular workflow. The scalability is taken care of on the service side, and not a worry of the end user. The client front-ends use *de facto* standardised software (R, Python), and it will be allowed to send R and Python scripts to the cloud service back-end. Other standards involved will concern the API (RESTful), authentication (OAuth), and existing data models (GDAL: Well-known-text for coordinate reference systems).

In this project, we will follow the standards recommended by the world's satellite EO data providers represented in the CEOS (Committee EO Satellites). The CEOS Working Group on Information Systems and Services (WGISS) has defined best practices for search services using OpenSearch with extension for EO data.

Impact 3: *Foster the establishment of interoperable access facilities to all EU Member States.*

openEO has the ambition to standardize a now highly fragmented market of non-interoperable services, and by that make them interoperable, and accessible from mainstream tools used by data scientists of Today. In doing so, it will not exclude any EU Member State.

Impact 4: *Link with other big data initiatives.*

Big data initiatives that will link to openEO include: R and Python (by generating client libraries, but also by creating cloud-based back-ends that can execute R or Python scripts on the big Copernicus data); GeoTrellis (a geographic data processing engine built on top of apache Spark), SciDB (an array-based data management and analysis system), and Google Earth Engine (a closed-source EO data processing system in the Google cloud). openEO will also serve as a demonstration, how additional back offices could be connected to form, from a user point of view, one joint meta platform.

Impact 5: *Provide user community tools including best-practices.*

Several media will be used during the project's progression to communicate with the project's pilot users as well as with external users and back office providers. So, social media accounts and web presences with discussion and comment functions will be created and managed. Additionally, we will build up other communication channels, which will also be available for the time after the project's end. So the openEO source code will be published in a GitHub public repository as well as in the open access repository Zenodo[33] to ensure resilience of the overall dissemination and exploitation system (see Impact 6: Ensure resilience of the overall dissemination and exploitation system.). In this context, we will provide openEO tutorials which can be used together with the implemented use cases (see *Pilot use case 1 - 4*) as templates to produce new front office and back office modules, new processes and new process chains.

User workshops during conferences will also help to generate momentum among potential users and to encourage communication.

Impact 6: *Ensure resilience of the overall dissemination and exploitation system.*

The openEO project will be set up as a community project that is 100% open from the start; expert users and developers from outside the consortium will be invited to participate and help to shape openEO. This will be done by communicating through a public GitHub organisation, which already exists at https://GitHub.com/open-EO.

Impact 7: *Optimise the use of Copernicus data by non-traditional user communities to meet societal challenges.*

By developing client libraries for R, Python and JavaScript, openEO will make cloud-based processing of Copernicus data very easy for those who already use one of these environments, and that constitutes the majority of ecologists, biologists, agricultural scientists, meteorologists, integrated modellers, and so on. Integrating Copernicus data in interdisciplinary studies, or in domain-oriented studies outside EO domain, will become trivial, and no longer require the strong technical skills that are now needed to work with Copernicus data.

Impact 8: *Impact on society and industry.*

Since openEO will be developed and distributed under a permissive open source license, both industry and society will benefit from participating together to form openEO: There will be no requirements or limitations to how the developed software can be used or redistributed. Permissive licenses do not control the license terms of future products, which build on openEO. Distributing openEO explicitly under the Apache license 2.0 will simultaneously guarantee the dissemination of the open source code without any changes in the original version.

[33] https://zenodo.org/

2.2 MEASURES to MAXIMISE IMPACT

(a) Dissemination and exploitation of results

We have foreseen a number of measures, which will ensure the highest possible impact:

1. The project's results will be directly usable for practically anyone working with Copernicus data - from a basic user, who will just want to download the data in a faster yet standard manner, researcher, being able to perform advanced analysis with a help of cloud infrastructure, to an application developer, being able to skip months of investment in building data processing libraries and avoiding large costs of data storage by facilitating open access on various back office infrastructures.
2. Developed software libraries will be completely open, making it possible for anyone to modify them to suit her needs.
3. We will provide four relevant pilot use-cases (see section 1.3), solving standard problems, which will serve as templates on how to use openEO API; advanced users will therefore be able to take these and adapt it for their specific case. These use-cases will also provide on-going results, relevant for wider public (e.g. snow cover, deforestation monitoring, etc.), which will ensure long-term visibility.
4. The project will engage ten beneficiaries, additional participating parties as DIAS providers, GEE or AWS and five pilot users (BMLFUW, ACF, ICIMOD, FAO, Prov. Bolzano) – already a representative EO community by itself. Interested parties will be able to participate at hackathons or via GitHub forums.
5. We will ensure the dissemination of the project results on several platforms, including project webpage, a GitHub organisation and social media. We will also propose to the GEO, GOFC-GOLD, and GFOI communities to register the openEO API to the GEOSS[34] and CEOS[35] platforms, providing a link to Copernicus data, which is currently missing.

These points are described in more details below.

OPEN STRATEGY and USABILITY of the RESULTS

This project does not deliver new data in the sense of the Horizon2020 definition but rather publish source code and demonstration use cases. However, the open strategy reflects the most important measure to maximize the impact of openEO. Therefore, we will implement and publish the openEO source code under the same restrictions mentioned in the H2020 call of being findable, accessible, interoperable, and reusable: (i) The development model will be open to external users, (ii) the implementations will be distributed under a permissive open source license, and (iii) after the project, openEO provenance will be handed over to a post-project openEO Steering Committee that will be selected during the project.

Publishing the code under the permissive Apache open source license 2.0 will allow the EO community to use the interface, adapt and enhance it, but also allows industry to integrate it in closed source solutions. New user communities - direct EO specialists, other researchers benefiting from using EO data as well as SMEs - gain access to a variety of back office service providers in one single step.

EXPLOITATION of the RESULTS

To evaluate whether the implemented processes (see *Table 1*) cover the complete data lifecycle as well as to demonstrate exploitation of the results, we will implement four pilot use cases, which are described in section 1.3 (c). All these implementations of the use cases already exist and are operational. The task of the project will be to transfer them to use the openEO API. FAO for example plans to validate openEO in Latin American test regions (see *Pilot use case 3*), which can serve as a template for many users in the regions. Dr. Inge Jonckheere describes the valuable impact of openEO as the following: "FAO Forestry works with developing countries to help them to support the design and implementation of their National Forest Monitoring Systems and expect

[34] http://www.earthobservations.org/geoss.php
[35] http://ceos.org/

great benefit from using openEO which facilitate the data transfer in developing countries."[30] Also, ICIMOD expects great impact of openEO "… for building open source based cloud computing interface which will greatly benefit us to customize innovative cloud computing solutions for Hindu Kush Himalayas region."[30]

Several back office service providers and intermediate SMEs are participants of this project, delivering the knowledge about existing strategies and standards. This will maximize the number of potential third party back office providers which can be connected to openEO with only few or no adaptations. Intermediate front office users in all EU Member States and worldwide, working with one of these back office services, may therefore easily change their workflows towards the openEO syntax. This flat learning curve also applies to the already existing broad user communities of the project's back office participants. By only changing their processes syntax and without having to get used to new data structures, they will be able to connect to several more Copernicus data distribution services in one single step. One institution that will make use of openEO directly from the beginning, is the Civil Protection Agency from the Autonomous Province of Bolzano – South Tyrol. Its director, Dr. Rudolf Pollinger, states that "… simple and efficient exploitation platforms are required to allow efficient access to the big data amounts collected and [that] the aim of the openEO project meets that need of the decision makers."

We will set up a mini-cloud environment by preparing container images for a mini cloud back office. Users will be able to run this on a desktop, which will help the community to easily reproduce the processes shown in the tutorials, without having to connect to EO data back offices, thus further lowering barriers for adopting scalable cloud solutions.

ENGAGING COMMUNITY

openEO will engage a pretty large community during the lifetime of the project - the project partners themselves represent already more than 300 EO experts. A diverse nature of pilot partners will add the chance of reaching out to users in the governmental, non-governmental, and environmental sectors. Altogether, it generates a large and representable 'core' community. Project participants as WUR are already actively collaborating with the R&D community (SME's, research institutes, and other universities) while working on the innovative edge of Copernicus satellite data exploration and algorithm development. The resulting knowledge about the needs of the EO data community takes a key role in maximizing the impacts of openEO.

openEO will be developed in an already created GitHub organization[3], which will provide a user forum for open discussions between the openEO participants and potential users. This way, we will be able to consider suggestions or constraints of potential future users as e.g. intermediate SME users, developers or researchers. The direct and immediate feedback will also help us to adapt our strategy and make openEO adaptable to additional back office services.

In an early stage of the project, we will provide a first basic use cases in the GitHub repository as a Proof of Concept. This will give potential users an idea about the access to back office drivers and their Copernicus data, and stimulate them to get involved. We will also publish four use case workflows (*see* ***Pilot use case 1 - 4***) for five pilot users based on openEO APIs.

Tutorials, which will be added to the repository, will represent together with the published pilot use cases a template that can be used by future developers to easily add new back office services and processes, or chain up new use cases for additional intermediate SMEs. In addition, additional big data initiatives can add new driver APIs to openEO, which will act as their entrance points.

Workshops and short courses during conferences (e.g. the yearly EODC Forum or the Conference on big data from space (BiDS)) or as stand-alone events during the project will help third party intermediate and end users to adopt openEO for processing their own use cases. At the end of the project, the forthcoming and future development of the interface will be guided by the post-project openEO Steering Committee, which will steer the direction of the openEO standard and software, after the project.

This committee will ensure that the openEO GitHub organization and its user discussion forums can remain active after the funding period.

EXPLOITATION by the PROJECT PARTICIPANTS

The potential community represented by the pilot users adds up to the effort, which several project participants who 'operate' a relevant back-end (e.g. VITO, EODC, WUR, Mundialis) will do to integrate the openEO interface to their structure. Applying the API on existing back-ends which serve large communities of both scientific and commercial users is an ideal approach to exploit the results of this open-source interface to a maximal extend. Several specific measures will be integrated in the exploitation plan:

- **VITO** and **WUR** will further exploit the results of the project in the operational Mission Exploitation Platform for the Copernicus Contribution Mission Proba-V[36] and in the Copernicus Global Land Service[37]. In these services our users have a need to use complementary data offered by EODC (e.g. Sentinel-2 and Sentinel-3) and data which many users serve via GRASS GIS which is integrated as well in the Proba-V MEP.
- **Solenix** is supporting the Thematic Exploitation Platforms (TEP) at ESA with two members in the TEP's Core Team. This team is in charge of identifying, assessing and evaluating new approaches and concepts for the exploitation of EO data. Solenix will promote the work done in openEO in the TEP team and reach out to a wider group of users via the different TEP projects. Hence, promoting openEO we will reach the ESA community working on the different exploitation platforms (Thematic TEPs, Mission MEPs, Regional REPs).
- **EURAC** Research will use openEO to improve their EO data cloud infrastructure and services, which are devoted to local decision makers and research communities. Further, EURAC back-end will use the openEO to grant access to several partner to its EO cloud platform. Building on this cloud making available the functions of openEO via an interface that can be accessed also via mobile devices will allow the users and third parties to make use of openEO and embed it in their own web and mobile applications. openEO will generate interoperability with other clouds (in particular EODC) and will allow to users and researchers to interface with larger EO infrastructures in a transparent way, making collaboration on a larger scale easier and efficient.
- **Sinergise** is planning to use openEO to further expand its Sentinel Hub services business. On one side these services should get wider audience, as they would be integrated in a common interface. In addition, Sinergise is hoping to integrate additional Copernicus datasets as Sentinel-1 in Sentinel Hub - datasets, which were so far inaccessible but will become available over openEO interfaces.
- **VITO** will integrate the openEO interface in the NextGEOSS DataHub for land related data and discuss in this project the potential of the interface with other important 'data provider' partners in this H2020 NextGEOSS project e.g. DLR[38] and NOA[39] for Sentinel data, CLS for marine data, MeteoSwiss and DLR for Atmosphere related data etc.
- openEO can as well link the **VITO** Proba-V MEP back-end with Google Earth Engine, which can further boost the synergy between the two platforms for the global vegetation community. This can be exploited in several ongoing activities in the frame of Agriculture Monitoring for FAO, IFAD[40], etc.
- **JRC**, in the context of its Earth Observation and Social Sensing Big Data pilot project, is developing the JRC Earth Observation Data and Processing Platform (JEODPP) to serve the needs of a series of JRC projects heavily relying on the analysis of geospatial data and in particular Sentinel 1, 2, and 3 images. The openEO interface will be adopted wherever possible on the JEODPP. This will contribute to the

[36] https://proba-v-mep.esa.int
[37] http://viewer.globalland.vgt.vito.be/tsviewer/
[38] http://www.dlr.de/dlr//en/desktopdefault.aspx/tabid-10002/
[39] http://www.noa.gr/index.php?lang=en
[40] https://www.ifad.org/

promotion of openEO among JEODPP users and partners by securing the portability of relevant project workflows on any platforms having the necessary data and processing capability offerings.

- **Solenix** and **VITO** will develop a mobile application interfacing openEO and based on the NASA-ESA's open source 3D globe WebWorldWind[41]. This will facilitate the promotion of openEO amongst a wider audience and provide users with a broad range of data and analytics at their fingertips. This first experience with the openEO project can open the door for the general public to EO data, their use and relevance in today's world. In addition, making available the functions of openEO via an interface that can be accessed via mobile devices will allow the community to make use of openEO and embed it in their own web and mobile applications. The application will be published on the Proba-V MEP platform and promoted as a user-oriented tool to improve the uptake of EO data by the general public.
- **Solenix** is leading a frame contract at ESA/ESRIN for the development and promotion of EO based applications. openEO will be an ideal candidate to use in conjunction with other ongoing projects, such as the Proba-V mobile app. Solenix will also support the further promotion of the project within the ESA community.
- **WUR** developed several open-source algorithms like for example BFAST[42] that are widely used for satellite data based change monitoring. The algorithms are currently being optimised for and integrated within several key back offices to be applied across large amounts of satellite images like Sentinel Copernicus data, the full Landsat Archive, in combination with in-situ non-EO field data for validation and accuracy assessment. Within this context, several key parties (e.g. Committee on EO Satellites (CEOS), FAO, ESA, Google) are currently supporting us to apply our algorithms on their cloud platforms and linked back offices:
  - Proba-V MEP testing and development within the context of the Copernicus Land Monitoring service project[43]
  - Near real-time deforestation detection with Google Earth Engine via two successful Google Research Awards
  - Active collaboration with FAO, on the integration of BFAST open-source algorithm within their Amazon AWS based cloud computing platform (SEPAL[44]) for automated forest change monitoring. Here, the goal is to provide ways for capacity building so that countries (e.g. governments, local NGOs) can assess forest cover change themselves.
  - CEOS Data Cube platform[45] towards integration of BFAST for generic change monitoring and trend analysis of satellite image time series
- **TU Wien, WWU** and **WUR** will benefit from the results of the project in all research efforts. It will open up more opportunities for scientific collaboration since a standardized API removes many initial obstacles when working with a new partner. openEO will help them in the publication of reproducible scientific results. If openEO becomes a working standard, they will adopt and integrate the openEO software and tutorial materials in undergraduate and graduate courses on EO analytics taught at these universities.

(b) Communication and dissemination activities

User involvement and public outreach is one of the main project's goals. We believe that the success of openEO after the period of grant is based on a strong interaction with the potential user community from the first day of the project. Therefore, we want to connect with several target audiences.

A **web page** will be created, explaining the idea, actual state, potential, and working use case examples to a broad audience of researchers, EO intermediate SMEs, and decision makers. This will help creating interest in the use of and active participation on openEO and thus in building up a user community for communication

[41] https://webworldwind.org/
[42] http://bfast.r-forge.r-project.org/
[43] http://land.copernicus.eu/
[44] https://sepal.io/
[45] https://software.nasa.gov/featuredsoftware/ceos2

during the project and openEO usage after it. The use of additional **social media** (as Twitter and Facebook) and other communication channels to interact with the EO community will increase the acceptance of openEO among researchers and SMEs and will help setting up broadly accepted standards in the project.

A broad **online user survey** will be executed to ensure potential user's participation on specific critical aspects of the project. Including the general user requirements and suggestions into the decision-making process will rise the chance of acceptance and usage of in the EU Member States and worldwide. We will be using newsletter to inform users about changes in the preliminary openEO source code and the achievement of milestones.

Scientific publications coming out of this project will be published under "gold" open access conditions, meaning that they will be made available open access to all readers without any time delay. They will be uploaded to appropriate web sites and get linked from the GitHub and Zenodo repositories to help researchers, SMEs, and policy makers to understand the potential and operation of openEO.

To encourage the formation of a stable and self-organizing **openEO user and developer community** after the period of grant, we will build up a post-project openEO Steering Committee, which will guide further implementations or projects to enhance openEO possibilities and distribution after the project. During the last part of the project, we will also organize user workshops, which will be held during conferences and in conjunction with the EODC Forum to increase the user's resonance.

We will also promote openEO at scientific conferences and in scientific papers. This also includes the integration and uptake of openEO by key organisations as EARSeL[46] by initiating e.g. an openEO Special Interest group.

The integration of an **Advisory Board** - consisting of external colleagues from the *European Grid Infrastructure* (EGI), the *European Space Agency* (ESA), and the *Joint Research Centre* (JRC) - will help us to get additional technical feedback for openEO. Therefore, we are planning two Advisory Board meetings to introduce and explain the openEO interface. The suggestions of the following reviews we will integrate in our decision-making processes of all work packages.

3. IMPLEMENTATION

3.1 WORK PLAN – WORK PACKAGES, DELIVERABLES

The openEO project is divided into seven work packages that stretch thematically from front office user services to back office providers and concentrate on the intermediate layer and potential user involvement (see ***Figure 2***). The actual openEO layers will be built in *work packages 3, 4, and 5* in an iterative approach. We will connect the openEO interface on the front office side to key programming languages R, Python, JavaScript and to a mobile application. On the back office side we will connect various cloud providers, which offer applications of Copernicus and other EO data access. Additionally, these work packages 3, 4, and 5 include the interfacing with various EO data processes (as seen in ***Table 1***). They will gain input from the potential user community, which will be addressed and organized in *work package 2*. Several already existing workflows for EO data processing will be adapted towards the new openEO syntax in *work package 6* to ensure the usability of the new interface and deliver information about practical problems. The core and the front and back office parts of openEO will be evaluated and validated by project internal and external experts in *work package 7*. This also includes the validation of openEO's ability to switch easily between the different back offices, i.e. to carry out the same process with minimal adjustment on different back-ends. *Work package 1* contains the project management and technical coordination and will ensure the achievement of the project goals with guidance throughout the project duration.

[46] A scientific network of European remote sensing institutes, coming from both academia and the commercial/industrial sector

Figure 9 shall give an overview of the complete work plan by visualising the WPs and their dependencies.

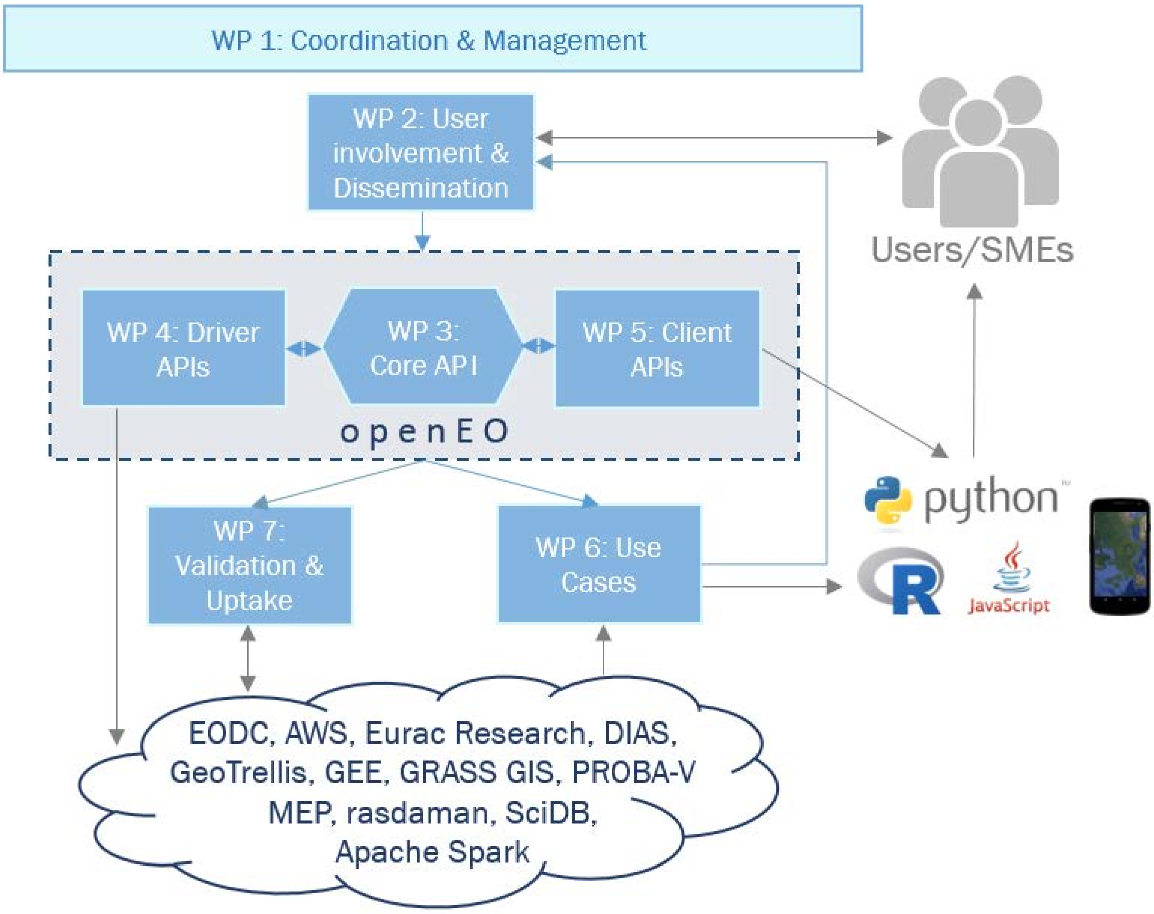

Figure 12: Overall project work plan

Following, the different work packages will be described in detail:

Table 4: Work Package Description

WP #	**1**
WP Title	**Coordination & Management**

Objectives

The overall scope of this WP involves the efficient coordination of all project-related consortium activities as well as the effective communication between all involved persons and parties. Organisation, monitoring and control of project related activities will be performed to guarantee the achievement of the project goals and includes administrative, financial and project reporting towards the EC.

Description of work

The overall project's coordination, communication and progress monitoring will be implemented in this WP to guarantee smooth realisation of the project's goals. The WP is divided into two tasks:

Task 1.1 Project management & EO Coordination

Project management (PM) of openEO will include coordination of the work, progress tracking and control, maintenance of project schedule and budget and timely delivery of reports and deliverables. A continuous monitoring of deliverables and milestones will ensure that deviations from the proposed work plan will be identified at an early stage. Change management will help to handle problems and risks according to the overall project plan, ensuring the achievement of the project goals and will furthermore include innovation management to address and embrace upcoming opportunities.

Face-to-face meetings (as described in the respective tasks) will be complemented by regular virtual meetings (videoconferences via GoToMeeting and Skype). If necessary, additional meetings with individual partners will be organised. Furthermore, quarterly videoconferences for the project board will be realised. While communication within the WPs will be organised independently by the WP leaders, interactions between WPs and other affected parties will be performed within this WP 1. This includes the organisation of several project meetings, which are relevant for the entire consortium. The Kick-Off meeting will be held in month 1, the midterm reviews in months 12 and 24, and the Final meeting at project end (month 36). To save travel costs, the midterm reviews will be accompanied by a selection of additional meetings. Documentation and minutes will be provided for all meetings.

Communication tools (e.g. a file sharing system like ownCloud) and techniques, templates and procedures will be defined at project start.

TU Wien will act as the main interface towards the EC and the consortium. Liaison with EC comprises meetings with the project officer and timely and clear communication regarding problems or delays.

Furthermore, this task encompasses administrative and financial management of openEO and providing guidance to partners on ethical, contractual, administrative, legal and financial matters. In terms of financial management, TU Wien as project lead will be responsible to provide detailed information on reporting to the consortium. Still, all beneficiaries are obliged to follow the rules as defined in the Grant Agreement. TU Wien will account for reviewing the financial statements of the beneficiaries before they are submitted to the EC.

openEO additionally proposes a Visiting Scientist activity, which has proven to be outstandingly valuable to several projects such as the ESA Climate Change Initiative (CCI) or the EUMETSAT Satellite Application Facility (SAF). For this activity, TU Wien as project coordinator will financially support external experts in joining one of the project partners for short visiting periods (e.g. 1-2 weeks), thereby giving valuable input to the project. Experience from the mentioned programmes showed that the external experts shall only be selected during the project lifetime, as new and worthwhile acquaintances of experts are often made during events, conferences, etc. Also the Hackathon (hosting potential users of openEO, see Task 2.2) may lead to a valuable collaboration with experts.

Task 1.2. IT Development Coordination

The openEO project involves a considerable amount of technical specification and implementation, the development of which by necessity is distributed over several partners. This imposes challenges in terms of coordinating and managing the quality of the technical work. This task coordinates the quality management of the technical developments in openEO.

This task will start with defining on how technical specifications shall be written, and how code development shall be quality controlled. In particular, a standard or template for documentation of software and system architecture such as arc42 will be adopted. For code development, this task will maintain documentation on which code is being developed by whom for which purpose. Also, the code development strategy, e.g. whether every code commit needs to be in the form of a pull request reviewed by another developer, and how code review is carried out (when, and by whom) will be defined. Finally, the unit testing frameworks adopted and code coverage reporting will be prescribed.

One of the challenges will be that several programming languages (at least 3: JavaScript, Python and R) are involved. But because each of them have already sufficiently evolved quality control mechanisms, only some practical guidance on programming practices and workflow management is needed to master this challenge.

WP #	2
WP Title	User Involvement & Dissemination

Objectives

The aim of the work package is to engage with multiple user communities for (1) the assessment of needs to define the system characteristics based on the user requirements, (2) to directly co-develop the system with users in the system development phase (hackathon), and (3) to facilitate user feedback systems for product improvement and refinement of requirements. The efforts include a comprehensive dissemination portfolio (i.e. website, social media, user forum). Towards the end of the project, the experiences of users will be evaluated and a gap analysis will be performed to address remaining issues and stimulate a sustainable uptake of the system; including the addition of new applications and workflows.

Description of work

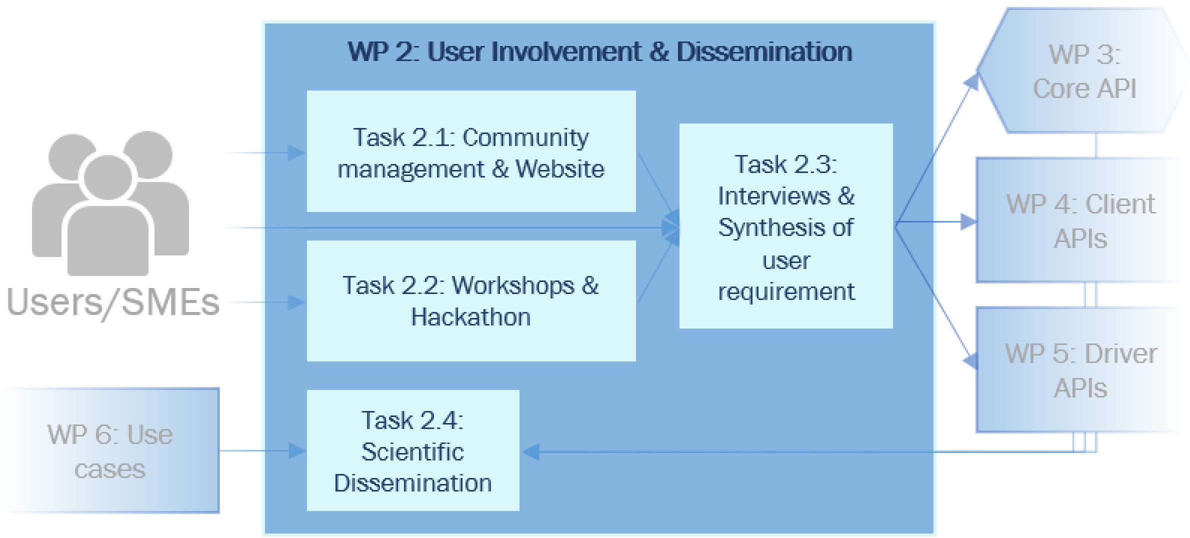

Figure 13: Work package 2

This work package treats the involvement of potential users into the openEO decision-making process and the strategy for the time after the project's funding. Therefore, we pursue several communication strategies to connect with the users in every stage of the project (see *Figure 3*). In detail, we will provide following tasks.

Task 2.1. Community management & Website

Within the first three months, we will release and maintain a project website (Deliverable 2.1), which will show the actual state of openEO, important progress and the achievement of milestones to a broad audience of researchers, policy-makers, end-users, industry and other interested individuals. Via a maintained forum, we will be able to encourage potential future users of openEO to be involved in decision-making processes. For the same purpose the project will also open and maintain accounts on several social media services as:

- Twitter
- Blog
- Others (e.g. Facebook, Google+)

In the same context, we will follow the Horizon 2020 programme Twitter account @EU_2020, to link our followers to an even broader audience. For the same reason and to pronounce EU-funded research we will add the H2020 hashtag #ResearchImpactEU while announcing main achievements on Twitter. Potential openEO users will be asked to voluntarily register so they can be informed directly about news and updates.

Additionally, a user forum will be established online for them to share their experiences and exchange with other partners, users and the developers. This forum will be part of an existing GitHub organisation[47], which will be extended and administered to guarantee transparency during the project and to assist the communication with the users. It will also include all available information about the project to guarantee timely user feedback.

We will create several tutorials containing guidance about the use and extension of openEO regarding e.g. the creation of new commands, additional back office drivers and chaining up new use cases. These tutorials will be published in the GitHub account as templates for the user community.

To guarantee the openEO functioning for the time after the project funding we will install a **post-project openEO Steering Committee** (see *Figure 17*), which will be responsible for the generation of a strategic plan about openEO. Among others, this will include (1) the management of expansions of openEO in GitHub and (2) uptake and integration of activities with key organisations like EARSEL, ISPRS, and OSGEO. The steering group will include their strategy into the final user workshop report (Deliverable 2.4). The GitHub account including the open source openEO API will stay accessible for the user community after the projects funding period (Deliverable 2.5).

Task 2.2. Workshops & Hackathon

In month 6 we will organize a Strategic & Use Case Workshop to discuss the already implemented openEO core API prototype with the results of the Proof of Concept (Deliverables 3.1, 4.2, 5.1) and the metadata standards, used by the inspected back offices (Deliverable 4.1). At this occasion, we will analyse the first user requirements synthesis (see Task 2.3). In addition, the needed processes for the pilot use cases will be debated here to progress towards a first overview, which is planned to be finished after 8 months (Deliverable 6.1). The results of the workshop will be captured in a report (Deliverable 2.2)

A Hackathon will take place on EODC/TU Wien premises in month 9. The aim of the hackathon is to test the basic functionality of openEO and get an early feedback directly from potential users, to identify potential strengths and weaknesses. EODC, VITO, Solenix, WUR, WWU and TU Wien will provide mentors who will guide the participants (who will work in teams of 2-5 people but depending on the total number of participants) and provide help if needed. This two-day event will kick off with an introduction to openEO and the back offices that are supported (at the date of the event). An additional benefit of the Hackathon may be the acquaintance of experts and talents who could be further supported and thereby could bring valuable input to the project as e.g. being included in the Visiting Scientist program (see Task 1.1). A report about the Hackathon (Deliverable 2.3) will include its results and will list its input for the openEO implementation.

[47] https://GitHub.com/open-EO

Around month 29 we plan to hold a Final User Workshop as an open session during or next to a relevant international meeting (e.g. EODC forum). There we will present the conclusions of our synthesis of user requirements, present the first iteration of the implemented use case process chains (Deliverable 6.2) to a broad audience, and engage in active discussions with the user community. The resulting report (Deliverable 2.4) and a personal guidance about the handling of the preliminary use case chains will be given to the pilot users for generating a first feedback about the openEO usability.

Task 2.3. Interviews & Synthesis of User Requirements

We will develop and implement a broad online user survey (using survey monkey). This process will include the identification of user communities and their invitation and stimulation to participate in the survey. Here we will build upon known user networks and communities that several of the project partners are active in and include EC bodies, governmental organizations, Copernicus, EIONET[48], EEA[49], GOFC-GOLD[50], GFOI[51], UN-Agencies, NGOs (Non-Governmental Organisation), Google-user groups, EARSeL, etc. to capture their needs and expectations. From the survey results a preliminary report about user requirements will be synthesized to a Preliminary User Requirements report (Deliverable 2.2) and presented at the Strategic & Use Case Workshop (see Task 2.2). In addition, we will conduct detailed interviews with the project's pilot users (FAO, BMLFUW, ACF, ICIMOD, Province of Bolzano). We will further engage active users and our project's pilot users in an online user feedback survey to systematically acquire the ways the system has been applied and how it can be further developed and improved. This will be done towards the end of the project. The results of the online survey, the key user interviews and from the User Workshop will be summarized in the report of the Final User Workshop (Deliverable 2.4) and provide the underpinnings for the system specifications. With this workshop we will bring developer and user communities together with the aim to address remaining gaps and outline strategies and stimuli for a sustainable uptake of the system. This includes the further development of the openEO system by different open-source communities.

Task 2.4. Scientific dissemination

All participants will disseminate the project results at conferences and meetings and in scientific publishing. Papers resulting from the project will be published using the "gold" open access model with the aim to have peer-reviewed papers freely available worldwide and thus more widely read. Next to GitHub (see Task 2.1) the papers, openEO software modules, and implemented use cases will be submitted to the open access repository Zenodo and relevant social networks (e.g. ResearchGate) for wider distribution within the scientific community.

It is furthermore envisaged to work towards a dedicated session within a suitable conference, e.g. the Conference on big data from space, yearly organised by ESA, to distribute the project results to an even wider audience.

Deliverables

D 2.1 Operative online dissemination portals (website, social media, GitHub)

D 2.2 Preliminary user Requirements report (incl. Use Case Workshop report 1)

D 2.3 Hackathon report

D 2.4 Final user workshop and report (incl. steering strategy)

D 2.5 Published GitHub account incl. openEO

[48] https://www.eionet.europa.eu/
[49] http://www.eea.europa.eu/de
[50] http://www.gofcgold.wur.nl/
[51] http://www.gfoi.org/

WP #	3
WP Title	Core API

Objectives

1. Design and implement the openEO Core Interface
2. Connect the commands and resource queries from the front offices (standardized in Client API - see WP 4) to commands from back offices (standardized in Driver API - see WP 3)

Description of work

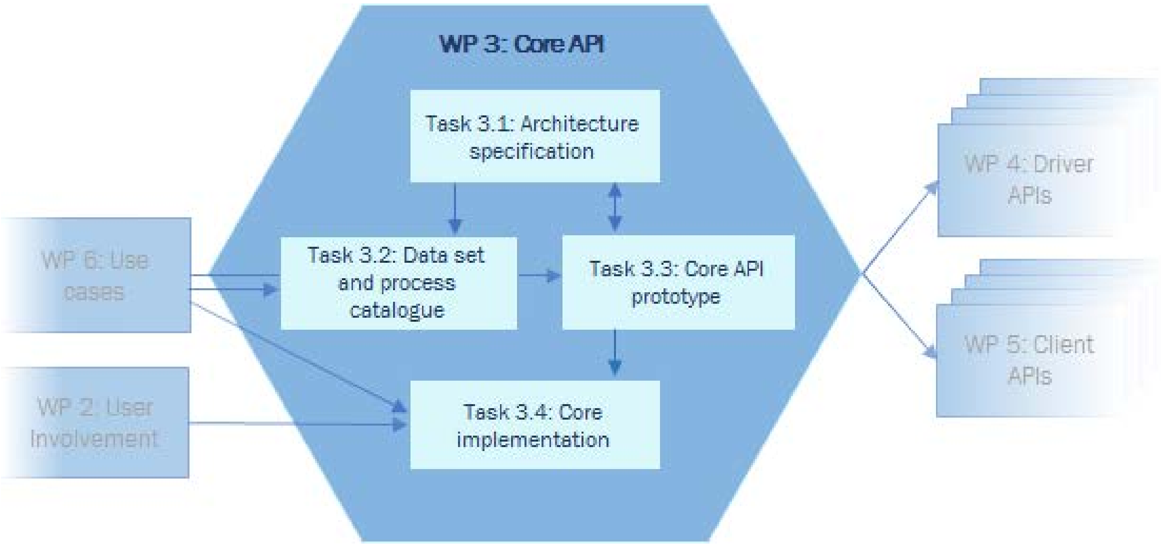

Figure 14: Work package 3

This work package designs and implements the core interface of openEO, which prescribes and constraints what front office clients can communicate with back office drivers, and how this is done. The core interface will appear as a single service with two interfaces: to back office drivers and front office clients. Internally, it will be built of microservices, small components that solve a relatively isolated task, such as user authentication or data discovery. The functional description will define how back office drivers interact with the core API, and how front office clients interact with the core API. The core API will be the interface for all functionality mentioned in *Table 1*. Aspects of this interface involve the functions that can be carried out, but also the degree of usability that will be realised. All REST-based micro-services and the openEO REST[52] API will expose their capabilities using the OpenAPI interface description (former swagger). This assures well-defined and documented service interfaces and allows the auto-generation of clients for each service.

Task 3.1 Architecture specification

This task will develop the specifications for the openEO architecture, in the form of an architecture document (Deliverables 3.2, 3.5), following an established template such as arc42[53]. The core specification shall be concise, and refer to other documents for longer and/or more detailed listings. A Proof of Concept will be developed together with Tasks 4.2 and 5.1 (Deliverable 3.1) by implementing first coarse processes for a

52 Representational State Transfer
53 http://arc42.org/

single back office. The specification will be developed in an iterative approach, using feedback from practical flaws, which might be revealed during the implementation process of the openEO interface.

Task 3.2 Data set and process catalogue

This task will choose the standards for describing data sets and computational processes (which will result in Deliverable 3.3), and define how the client and back-end APIs will handle them. Therefore, we will use the results of WP 4 here (Deliverable 4.1).

Task 3.3 Core API prototype

Based on the preliminary results of the Architecture Specification (see Task 3.1) this task will implement, test, deploy and document a first prototype of the openEO core interface (Deliverable 3.1), along with prototype implementations of a subset of driver (WP 4) and client (WP 5) interfaces; the goal of this task is to get all developers work together and understand the different challenges, and better assess the critical challenges of the openEO API and to deliver information about practical problems regarding the specifications of the planned openEO architecture.

Task 3.4 Core implementation

Based on the experiences of the prototype (Task 3.3) and input from internal and external users the openEO core interface will be implemented in this task in an iterative approach in cooperation with WPs 4 and 5 (Deliverables 3.4, 3.6).

Deliverables

D 3.1 openEO core API prototype including Proof of Concept

D 3.2 openEO architecture specification, first version

D 3.3 openEO data set and process descriptions

D 3.4 openEO full API implementation, first version

D 3.5 openEO architecture specification, final version

D 3.6 openEO final API implementation

WP #	**4**
WP Title	**Driver APIs**

Objectives

1. Coordinate back-end driver development with WP 3, Core API development
2. Develop the individual drivers for the different back offices

Description of work

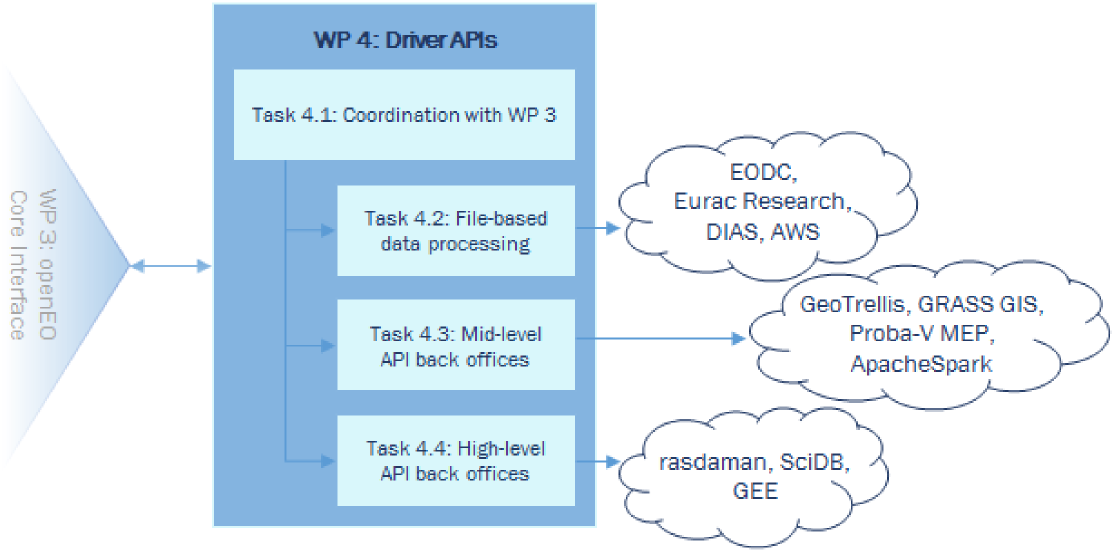

Figure 15: Work package 4

This work package coordinates common objectives with WP 3 (Task 4.1), and implements drivers that connect the openEO core interface (deliverable of WP 3) to one of the back offices (Task 4.2-4.4). For this, the back offices have been loosely divided into three categories: file-based, mid-level API, and high-level API, in order to avoid a task for each back-end. This categorisation is merely meant for grouping similar activities.

Firstly, we will investigate the metadata standards and interfaces of the back offices, which are involved by this project (Deliverables 4.1, 4.3). Based on that the development of the various back office drivers will take place in different iteration steps (Deliverables 4.2, 4.4, 4.5). Depending on the progress of the launch of DIAS, we will include them into the developing process. Result will be a status report on DIAS as well as on the GEE back-end driver (Deliverable 4.6).

Task 4.1. Coordination with Core API

Common elements of the core openEO API (WP 3) will be categorized, and compared across the different back offices. These elements include authentication, data discovery, process discovery, resource management, process management and accounting. Interaction between the core API and the drivers has to abide – where possible – already existing (metadata) standards and protocols. Thus, an agreement on specific standards supported by both the core API and the driver needs to be found.

Two early prototypes of back office drivers will be implemented (Deliverable 4.2) in close coordination with WPs 3 and 5. Therefore, two weeks of intense collaboration will be organised between the developers of WPs 3, 4, and 5 during months 3 and 4. Within the first six months, additional bi-weekly development sprints and regular video conferences will be installed to guarantee a close collaboration of the virtual development teams of the three openEO layers.

During the development of the three layers for the first openEO version we will organize in month 12 an openEO API merging workshop between the participants, responsible for WPs 3, 4, and 5. Goal is to strengthen the interaction and compatibility of the different layers to guarantee the development of openEO as one consistent API. Also aside from this workshop these participants will develop the different layers very closely. This includes also the implementation of the use case chains (Deliverables 6.2, 6.3) and the user inputs and validation results from WP 2 and WP 7.

Task 4.2. File-based data processing

Interfaces to raw Copernicus (and other) data accessible via POSIX or REST (NFS, Object Storage) with back office providers. Back offices involved are: EODC, EURAC, DIAS, AWS.

Task 4.3. Mid-level API back offices

This task implements the back offices with mid-level APIs, where APIs to software systems can be combined with low-level, file-based access. The back-ends involved are GRASS GIS, Spark, and GeoTrellis.

Task 4.4 High-level API back offices

This task implements the high-level APIs where data and processes are only accessible through API calls. The back-ends involved are SciDB, rasdaman, and Google Earth Engine.

Deliverables

D 4.1 Overview document about back offices metadata standards and interfaces

D 4.2 Two early prototype back-ends (file-based, SciDB)

D 4.3 Report assessing back-end driver variability

D 4.4 First versions of back-end drivers

D 4.5 Final back-end drivers

D 4.6 Status report on GEE and DIAS back-end drivers

WP #	5
WP Title	Client APIs

Objectives

The objectives of this work package are to create Client API's for R, Python and JavaScript that can be imported as libraries and used in web interfaces. Python and R are the two programming environments which are extensively used within the EO community. JavaScript is supported for web browser integration and mobile applications via webviews.

Each Client API will support all back-end methods, and include:

- Integration with the client environment package ecosystems, e.g. in the case of Python with SciPy, NumPy, pandas; in case of R with the R spatial packages including sp, sf, raster; and in addition potentially more dedicated packages
- In case user-defined functions are supported by the back-end, integration with the client environment
- In the case of the mobile application, cross-platform support to different mobile operating systems will be provided. The approach will take into account the use of native features, in order to increase efficiency.
- data discovery
- querying back-end capabilities
- asynchronous operations/batch jobs
- user authentication

Description of work

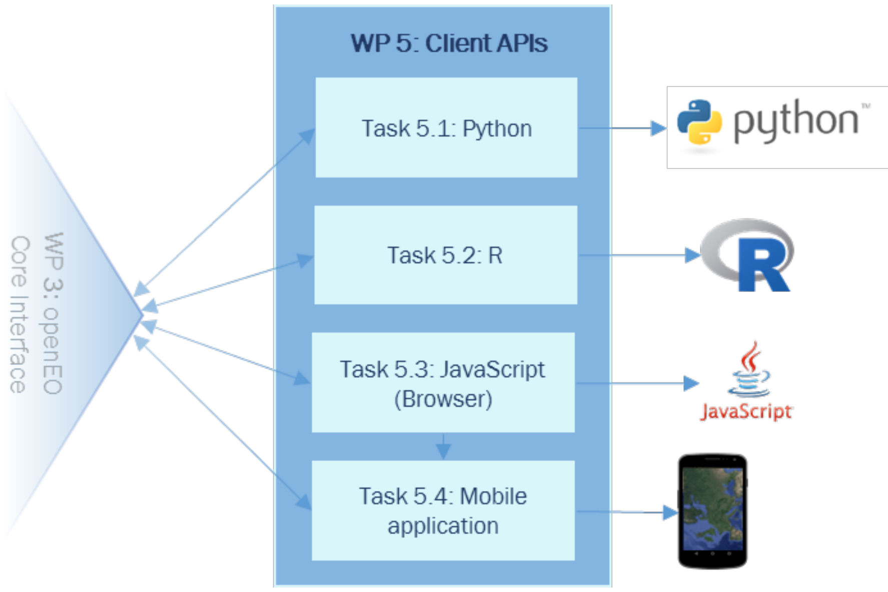

Figure 16: Work package 5

In this work package various Client APIs are implemented to connect the Core API of the openEO interface (delivery WP 3) to the corresponding front office environment in an iterative way accordingly to the development of WPs 3 and 4 (Deliverables 5.1, 5.2, 5.3). Additionally, this work package covers the development of a mobile application (Deliverable 5.4). Therefore, we will provide four different ways for the interaction with users and SMEs (see *Figure 13*).

Task 5.1. Python

This task includes the creation of the Python package which exposes all of the functionality offered by the common back-end API as Python methods and contains additional modules for standardized authentication and data processing and analytics. This can be imported into the source code by users or SMEs for chaining up commands.

Task 5.2. R

This task includes the creation of an R package with functions and methods for standardized authentication and data processing and analytics. This can be imported into the source code by users or SMEs for chaining up commands.

Task 5.3. JavaScript

This task includes next to a process library also the creation of a Graphical User Interface for various web browsers, which will allow the users and SMEs to chain up commands. The JavaScript client will include a self-contained library that allows access to the back-end services, as needed by task 5.4. This client will be HTTP-based and RESTful where possible.

For the client interfaces developed in Tasks 5.1, 5.2 and 5.3, libraries of standardised data processing commands will be implemented according to the abilities of the back offices. This also includes the standardised versions of the possible results (deliveries WP 4). These libraries may be imported manually into source code by the users and SMEs (Python, R) or are already included automatically in the GUIs (Web browser/mobile application). All client environments will include options for product visualization.

Task 5.4. Mobile application

The goal of this task is to demonstrate that openEO is capable of exposing the services offered by the different back-ends to mobile devices. Mobile devices are widely used as clients nowadays and it is thus important to demonstrate that openEO works seamlessly with them as well. In order to do so, we propose to implement a mobile application available for both tablets and smartphones across different platforms. This task builds on the JavaScript library, which will be developed in the scope of task 5.3 to access the different back-ends via the Core API of openEO.

The proposed cross-platform mobile application will use Apache Cordova and WebWorldWind for the visualisation. In order to make the app accessible to the general public, the app will show indicators and trends rather than raw data. There will be a list of known indicators that can be retrieved from the different back-ends. This list will include e.g. vegetation quality, vegetation type, air quality, average snow height, average cloud cover, etc. As an exemplary workflow:

- The app obtains the location of interest (location of the device, manually set by the user, etc.)
- The app queries openEO for which indicators from the list can be computed for this location
- The app gets the data from openEO, i.e. from pre-computed indicators and without authentication i.e. open data.
- The app shows the data as a map for the region, trends for a particular location or average, possibly aggregating multiple indicators for this location

Deliverables

D 5.1 Proof of Concept (Python)

D 5.2 First versions of three front-end drivers

D 5.3 Three front-end drivers (R, Python, JavaScript)

D 5.4 Mobile application

WP #	6
WP Title	**Use Cases**

Objectives

1. testing the automated process of several use cases with a focus on Copernicus data
2. testing portability of use cases

Description of work

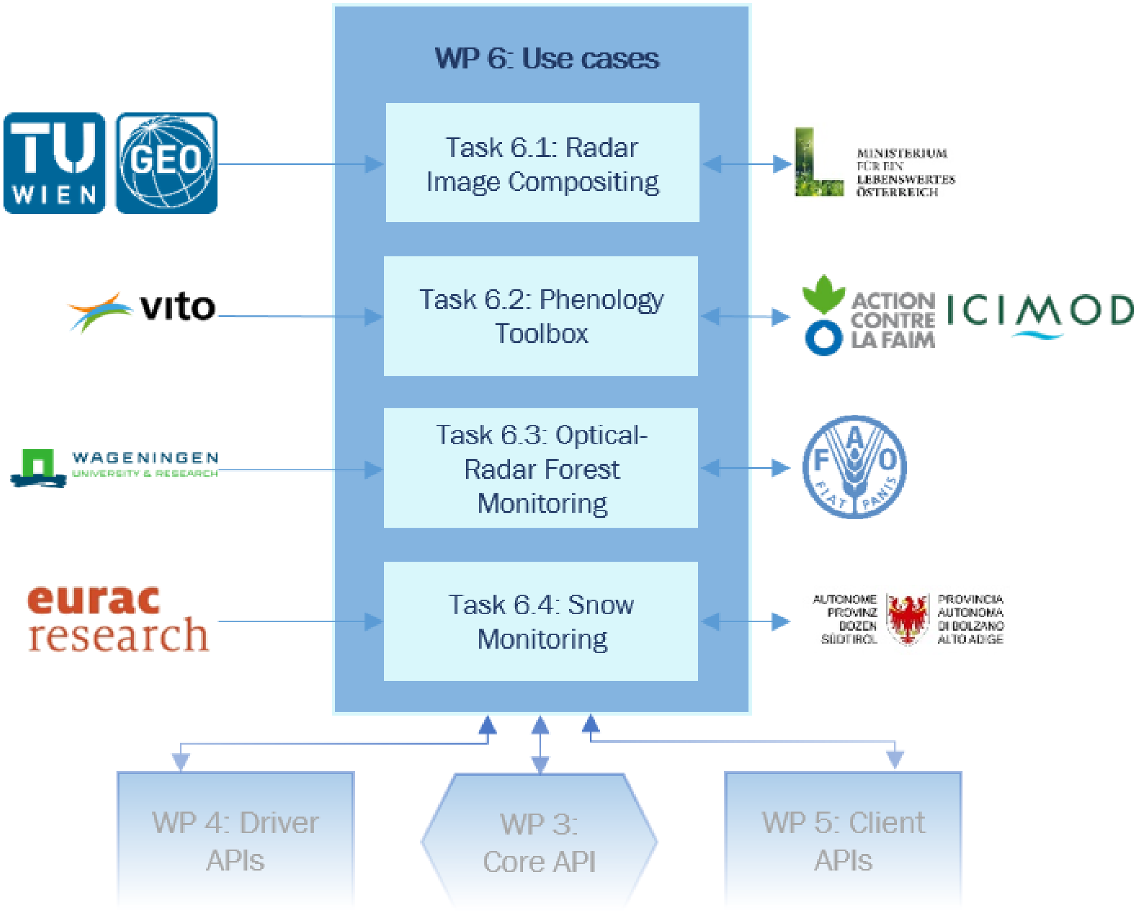

Figure 17: Work Package 6

This work package serves the proof of usability of the openEO API. Therefore, several use cases will be implemented for pilot users, based on the implementation of the various openEO layers (Deliverables within WPs 3, 4 and 5). These use cases will also work as a control of the functioning connection between front office users and back office services as well as of the generalisability of openEO for additional future use cases.

The use cases, which already exist as operational local workflows (including all the aspects mentioned below), will be newly implemented in an iterative approach to use now the openEO syntax. After the completion of the Proof of Concept and the core API prototype (Deliverables 3.1, 4.2, 5.1) the pilot use cases will be used to identify a first overview of the various processes, (see *Table 1,* Deliverable 6.1). These identified processes will be part of the openEO process catalogue, which will be defined later within WP 3 (Deliverable 3.3). Parallel to the development of the first version of the openEO development (resulting in Deliverable 3.4) process chains will be implemented for each of the project's use cases, controlling the usability of the various openEO front office interfaces in all aspects of the data lifecycle (Deliverable 6.2). They will then be validated by the pilot users for helping the develop of further iterations and to extend openEO towards a usable API. Accompanying the openEO development iterations, the use case chains will be upgraded until they serve the purposes of the pilot users (Deliverable 6.3).

Task 6.1. Radar Image Compositing

This use case will be implemented for the Austrian Federal Ministry for Agriculture and Forestry, the Environment and Water Resources (BMLFUW). It will content following aspects:

- Gather all Sentinel-1 observations for different polarizations (e.g. VV, VH) in the period of interest e.g. of one month,
- Perform quality checking and masking of NaN values in the input data,
- Calculate statistics like mean or median on the resulting stack of images, and
- Store the products as RGB composites. There are several options to assign data to colour channels, one possibility is storing subsequent months in time as different colours e.g. R=June, G=July and B=August. Another is to make composite images for the same time span but using different polarizations e.g. R=VV-July, G=VH-July, B=VV-July/VH-July

For further information see *Pilot use case 1:* Radar Image Compositing

Task 6.2. Multi-source Phenology toolbox

This use case will be implemented for Action Contre la Faim (ACF) and tested on Western Africa region. This use case includes following subtasks:

- Port the SPIRITS and/or TIMESAT phenology tool (time series) using the openEO Python client API and test with moderate (Sentinel-3 and Proba-V) and high (Landsat-8, Sentinel-2) optical resolution datasets,
- Port data fuse tools (e.g. Kalman filtering, Neural Networks) to Python using the openEO client API,
- Generate multi-source phenology metrics,
- Ingest ancillary datasets in SciDB through the client API, and
- Develop different correlation metrics to link phenological information with ancillary information

For further information see *Pilot use case 2:* Multi-source Phenology toolbox.

Task 6.3. Optical-Radar Forest Monitoring

This use case will be implemented for the Food and Agriculture Organization of the United Nations as a partner of the Peruvian Ministry of the Environment (MINAM). It includes following issues:

- Port the “Bayesian approach” using the openEO R and Python client APIs and test with Sentinel-1 and Sentinel-2 data,
- Implement and test near real-time deforestation system using the openEO functionalities, and
- Test and compare the access of Sentinel-1 and Sentinel-2 data from different back-ends

For further Information see *Pilot use case 3*: Optical-Radar Forest Monitoring.

Task 6.4. Snow Monitoring

This use case will be implemented for the Italian Province of Bolzano. Following aspects will be covered in this task:

- Data access, data discovery, data management within one single interface will be enable to the Hydrological Office via the openEO.
- EURAC research will be able to run their algorithms on any back-ends correctly interfaced with the OpenEO and where the input data are stored.

For further information see *Pilot use case 4:* Snow monitoring with Sentinel- 1 and Sentinel -2.

Deliverables

D 6.1 First overview of needed processes from use cases

D 6.2 First iteration of use case chains

D 6.3 Final use case process chains

WP #	7
WP Title	Validation & Uptake

Objectives

The objective of this work package is to validate the full openEO API, and facilitate its uptake by SMEs, pilot users and the EO data community. We will do that by (1) in depth software validation and review by all involved SMEs, (2) testing and evaluation of the use cases by the involved pilot users, and (3) testing the openEO on other back offices.

Description of work

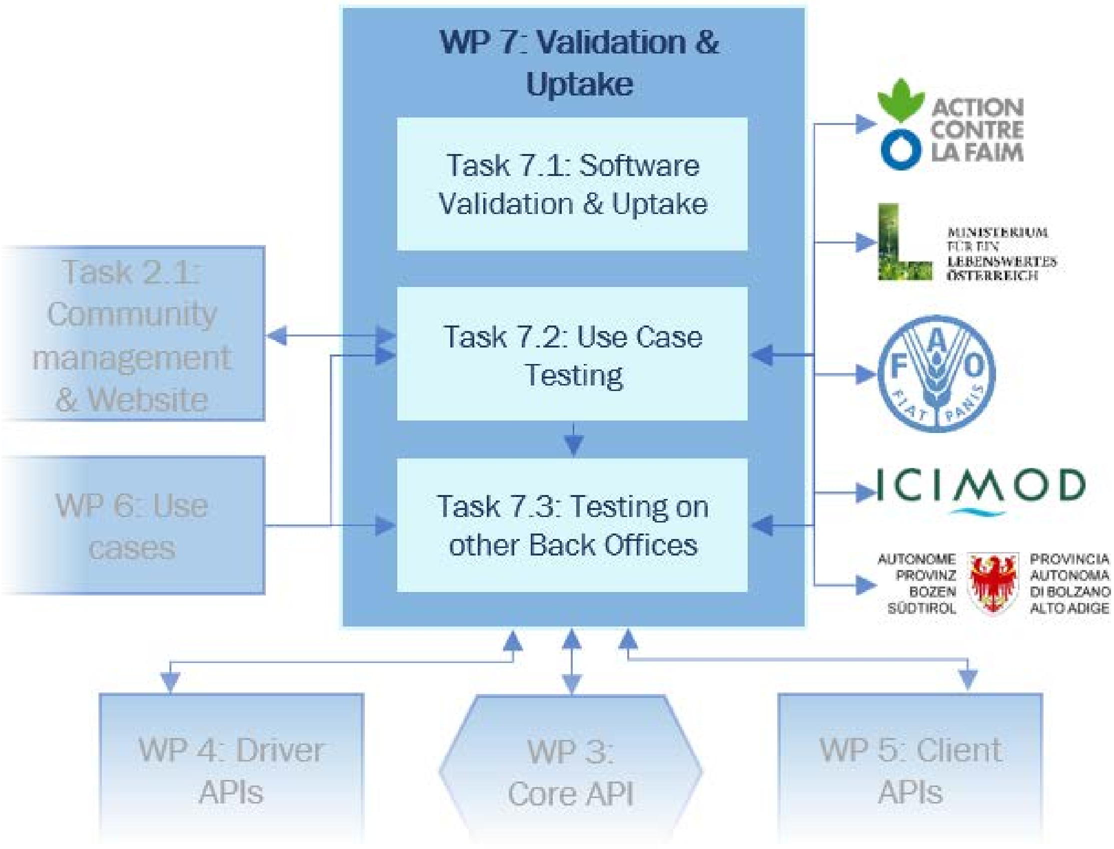

Figure 18: Work Package 7

Validation and uptake of the full openEO API will be executed in three key steps descripted in detail below. Key is that here all involved parties, i.e. the users involved within the consortium, SME's, and pilot users will focus on the testing the software and interface stability, flexibility and user-friendliness.

Task 7.1. Software Validation & Review

The aim of this task is to validate the requests from the different front offices for the selected back offices by checking the processes stability (Deliverable 7.1). This will be done by executing the same process call from different front offices (Python, R, JavaScript), and comparison of the respective results.

The validation process shall consider reviews from the technical Advisory Board, which consists of external experts of EGI, ESA and JRC. Therefore, a first Advisory Board meeting will be held at M12 to present the implementation of the first iteration of openEO API (Deliverable 3.2) and to discuss the conclusions of the preliminary user requirements report (Deliverable 2.2). A second Advisory Board meeting will be held at M24 to present the preliminary final version of openEO and available commands / command chains. Overall goal of the Advisory Board meetings is to ensure reviews of external technical experts to guarantee scientific acceptance of the openEO interface.

Task 7.2. Use Case testing

The involved pilot users (i.e. BMLFUW, ACF, ICIMOD, FAO, Prov. Bolzano) will evaluate and test the different use cases, assess the user-friendliness, stability, and flexibility and will compare it with the use of their already existing process chains. (Deliverable 7.2). The preliminary results will be considered to optimize the openEO interface and the process chains. We will also test in this context the developed tutorials and software, published via GitHub (see Task 2.1) to improve them and to facilitate the uptake.

Task 7.3. Testing on other back offices

The users' ability of interchangeability between the back office providers by using openEO will be tested here (Deliverable 7.3). The use case testing of Task 7.2 will be broadened up to test them also on the other back office, which we are interfacing for the project. For example, within the context of WP 6.3 we will test and compare the access of Sentinel-1 and Sentinel-2 data from different back offices (e.g. Google vs. Amazon vs. EODC, see *Pilot use case 3*). The preliminary test performance will allow us conclusions for further developments of the WPs 3, 4, and 5.

Deliverables

D 7.1 Report on Software validation and review

D 7.2 Report on Use case testing by pilot users

D 7.3 Report on Use case back-end cross tests

Finally, a list of deliverables is presented below:

Table 5: List of Deliverables

Deliv. #	Deliverable name	WP	Type[54]	Diss. level[55]
D 2.1	Operative online dissemination portals (website, social media, GitHub)	2	DEC	PU
D 3.1	openEO core API prototype including Proof of Concept	3	DEM	PU
D 4.1	Overview document about back offices metadata standards and interfaces	4	R	PU
D 4.2	Two early prototype back-ends (file-based, SciDB)	4	DEM	PU
D 5.1	Proof of Concept (Python)	5	DEM	PU
D 2.2	Preliminary User Requirements report (incl. Use Case Workshop report 1)	2	R	PU
D 6.1	First overview of needed processes from use cases	6	R	PU
D 2.3	Hackathon report	2	R	PU
D 3.2	openEO architecture specification, first version	3	R	PU
D 4.3	Report assessing back-end driver variability	4	R	PU
D 3.3	openEO data set and process descriptions	3	R	PU
D 3.4	openEO full API implementation, first version	3	OTH	PU
D 4.4	First versions of back-end drivers	4	OTH	PU
D 5.2	First versions of three front-end drivers	5	OTH	PU
D 7.1	Report on Software validation and review	7	R	PU
D 3.5	openEO architecture specification, final version	3	R	PU
D 6.2	First iteration of use case chains	6	OTH	PU
D 2.4	Final user workshop and report (incl. steering strategy)	2	R	PU
D 6.3	Final use case process chains	6	OTH	PU
D 3.6	openEO final API implementation	3	OTH	PU

[54] R: Document, report (excluding the periodic and final reports); DEM: Demonstrator, pilot, prototype, plan designs; DEC: Websites, patents filing, press & media actions, videos, etc.; OTHER: Software, technical diagram, etc.
[55] PU = Public, fully open, e.g. web; CO = Confidential, restricted under conditions set out in Model Grant Agreement; CI = Classified, information as referred to in Commission Decision 2001/844/EC.

D 4.5	Final back-end drivers	4	OTH	PU
D 5.3	Three front-end drivers	5	OTH	PU
D 5.4	Mobile application	5	OTH	PU
D 2.5	Published GitHub account incl. openEO	2	DEC	PU
D 4.6	Status report on GEE and DIAS back-end drivers	4	R	PU
D 7.2	Report on Use case testing by pilot users	7	R	PU
D 7.3	Report on Use case back-end cross tests	7	R	PU

3.2 MANAGEMENT STRUCTURE, MILESTONES and PROCEDURES

(a) Management structure

A well-considered management structure of openEO will ensure a reliable environment for the whole consortium to fully concentrate on the goals of openEO. The management will be implemented within WP 1 and encompasses project coordination, tracking and risk management. Furthermore, it will involve proactive communication with project partners as well as the EC to enable the smooth realisation of openEO. For a clear understanding of the goals, openEO has been structured to seven well-balanced work packages, which are again split into several tasks (see section 3.1).

The management structure of openEO is visualised in *Figure 17*:

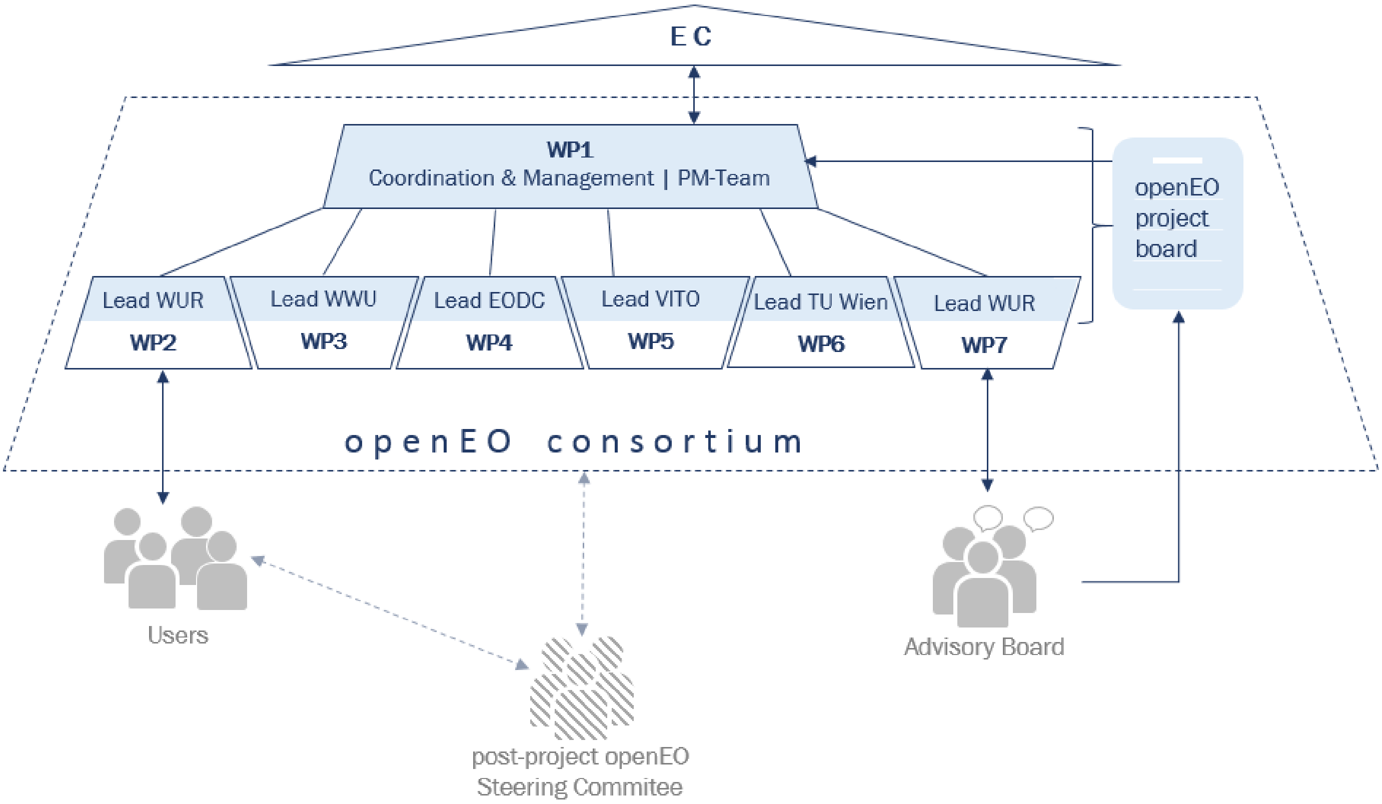

Figure 19: openEO management structure

The **openEO consortium** consists of all involved project partners, containing ten institutions from seven different countries (further details on the consortium composition in section 3.3). Most of the partners have extensive and longstanding experience in large research projects funded by various European sources, including FP6, FP7, H2020, ESA, ECMWF and EUMETSAT. The openEO coordinator TU Wien has already successfully led large national and international projects, including the ESA Climate Change Initiative Phase 1 (for Phase 2 the lead has been handed over to EODC) and the national Global Monitoring of Soil Moisture for Water Hazards Assessment project[56]. WWU also coordinated and managed several software development projects funded under FP6 (INTAMAP) and FP7 (UncertWeb), which predestines them for the coordination of approaching the openEO development from the IT-side and to lead the development of the core API (WP 3) as the central element of openEO. WUR is hosting the GOFC-GOLD land cover office[57], and coordinating the R&D component of the Global Forest Observation Initiative (GFOI), which can facilitate a key role in involving users, dissemination, and uptake of openEO. Their former activities gives them the expertise to lead the User Involvement and dissemination (WP 2) and coordinate openEO validation and uptake (WP 7) to maximise the openEO impact (see section 2.2). VITO is hosting currently the user segment of the PROBA-V MEP, as they did for the SPOT-VEGETATION 1 and 2 missions. They have many activities for ESA, EC, FAO and several commercial companies. With their strong expertise in software and hardware related to EO processing chains, distribution and exploitation platforms VITO fulfils the needed requirements to lead the development of the various client APIs (WP 5). EODC is leading the ESA Climate Change Initiative Phase 2 for soil moisture and leads the ECMWF Copernicus Climate Change Service (C3S) Lot 7 – soil moisture initiative. The openEO consortium will meet in yearly meetings.

[56] Wagner, W.; Hahn, S.; Kidd, R.; Melzer, T.; Bartalis, Z.; Hasenauer, S.; Figa-Saldanã, J.; de Rosnay, P.; Jann, A.; Schneider, S.; Komma, J.; Kubu, G.; Brugger, K.; Aubrecht, C.; Züger, C.; Gangkofer, U.; Kienberger, S.; Brocca, L.; Wang, Y.; Blöschl, G.; Eitzinger, J.; Steinnocher, K.; Zeil, P.; Rubel F. (2013) *The ASCAT soil moisture product: A review of its specifications, validation results, and emerging applications*, **Meteorologische Zeitschrift** 22(1), pp. 5-33.
[57] http://www.gofcgold.wur.nl/

The project will be guided by the Project Management team (**PM-Team**), which is composed of the project coordinator TU Wien and the IT development coordinator WWU. Prof. Wolfgang Wagner from TU Wien will act as overall project and EO coordinator. As the head of one of Europe's leading university research groups in remote sensing he has already coordinated a number of comprehensive projects. As co-founder and Head of Science of the EODC, his cooperation network was important for the foundation and start-up phase of the EODC. Dr. Matthias Schramm, an expert in EO data analysis and responsible for the day-to-day coordination of the openEO project, will support Prof. Wagner. Furthermore, DI Alexandra von Beringe, a certified IPMA Level C project manager, will take care of the administrative, legal and financial coordination of the project. Prof. Edzer Pebesma (WWU) will complete the PM-Team. He will oversee and manage the IT development of the openEO APIs. Prof. Pebesma is deputy head of the Institute for Geoinformatics of the University of Muenster, and is not only a long-term open source software developer, but also one of the founding leaders of the community who use R for analysing spatial and spatiotemporal data. The PM-Team will meet within regular videoconferences to ensure a profound communication and project guidance.

TU Wien, WUR, WWU, EODC and VITO will lead all WPs of the project. Together with the PM-Team, the WP-leads will form the **openEO Project Board**. Led by Prof. Wagner, the project board will provide scientific and technical guidance to the course of openEO. The board will be responsible for decisions relating the project and solving any occurring disputes. It is the goal to come to democratic decisions, the decision process as well as regulations on dispute resolution will be defined within the CA. The board will meet quarterly via videoconferences to review the project's progress and provide improvements to the PM-Team. In addition, the board will assist the PM-Team in preparing reports towards the EC and will prepare decisions to be made within the yearly consortium meetings.

(b) Innovation management

A specific objective of this project is the involvement of **users** and developers in the decision-making process. In WP 2 several platforms (e.g. website, social media, source code repositories) will be implemented to establish a direct link between user community and project participants from the first day on. This thereby created constant feedback loop throughout the project will help us understanding the needs and constraints of potential users. The final goal is the creation of an openEO API, which answers the actual needs of the community. Thus, very diverse users will be able to benefit from the project. This is also guaranteed by the selection of five very diverse pilot users – covering governmental, non-governmental, scientific, and environmental sectors in different regions – which will help in lowering barriers for potential users to enter the community.

The pilot users will in addition validate the usefulness of the test use cases and the interchangeability of openEO among various back offices (see WP 7 in section 3.1). This will serve help to control whether openEO represents an interface to facilitate standardised interchange between users and applications (see section 1.1).

An important role is given to the **Advisory Board**, which consists of three key experts, who will provide reviews of the project results based on their state-of-the-art knowledge to ensure the acceptance of the openEO interface. Diego Scardaci (EGI), Peter Strobl (JRC) and a representative from ESA (probably Sveinung Loekken or Günther Landgraf) will form the Advisory Board and will meet in two dedicated Advisory Board meetings within the yearly consortium meetings in months 12 and 24. All Advisory Board members will also aim at integrating additional experts within their field of action to further increase the extent of their input. The feedback from the experts will be discussed in the openEO project board, guiding the further project course. WP 7 will be responsible to communicate with the Advisory Board to regularly receiving input on the project development.

openEO is designed as an **open source** initiative. We will publish all our results on short-term nature in an existing GitHub organisation[58] and will actively announce new deliveries via various channels to interact with the user community. This will help future users and SMEs to include openEO into their workflows. In addition,

[58] https://GitHub.com/open-EO

the project participants are planning to include this new interface in their active as well as future programmes (see section 2.2). This and the open source nature of the new API will guarantee the usefulness for additional Copernicus data using enterprises.

The project partners will publish several scientific papers under "gold" open access conditions (see section 2.2). In addition, we will present openEO at various scientific conferences to obtain additional feedback and critical review by the scientific and user community. This will also help to further increase the impact of and knowledge on openEO.

To ensure the acceptance and sustainability of the project achievements, a **post-project openEO Steering Committee** will be established over the course of the project to steer a stable and self-organising user and developer community after project end. This committee shall ensure the sustainability of the project outcomes by creating a strategic plan including follow-up projects and implementation possibilities (see Task 2.1).

Table 6: *List of milestones*

Milestone number	Milestone name	Related WP(s)	Due date (in month)	Means of verification
M1	Kick-Off meeting	WP 1	1	Minutes of meeting
M2	Proof of Concept	WPs 3-5	6	All core features of openEO as defined in section 1.3 (b) are up and running
M3	Midterm Review 1	WPs 1-7	12	Minutes of meeting including Advisory Board review
M5	openEO first version	WPs 3-5	20	Microservices and their tasks for the openEO API core layer as defined in ***Table 1*** are up and running on all front offices and minimum two back offices
M6	Midterm Review 2	WPs 1-7	24	Minutes of meeting including Advisory Board review
M7	Use cases implemented	WPs 3-6	26	Use case processing chains released for use case validation
M8	User workshop	WPs 2-6	29	Workshop accomplished; post-project openEO Steering Committee is established
M9	openEO API	WPs 3-5	34	openEO API functionality is validated by pilot users in terms of stability and flexibility between front and back offices

(c) Risk Management

Risk management is a process to be performed throughout the project duration. A Risk analysis has to be accomplished at the beginning of the project (i.e. at the Kick-Off (KO) meeting) with the consortium. This is necessary as only the complete project team is able to identify possible risks and classify them regarding probability of occurrence and extent of damage. In addition, the mitigation measures have to be defined. Risks will be regularly reviewed within WP 1. A selection of possible risks the project may have to face is:

Table 7: Critical risks for implementation

Description of risk	Level of likelihood[59]	Work package(s) involved	Proposed risk-mitigation measures
Time Slippage	M	WP 1	Regular Progress Monitoring and Review by PM-Team will highlight any problems with timeliness of delivery of project. Delays can be noted and will immediately communicated with EC.
Unavailability of Copernicus data for Use Cases	M	WPs 3-7	EO data can be obtained from various sources so a high level of redundancy is already considered. In case that data, which are crucial for the use cases, are not available on the relevant back offices, these data can be readily obtained from ESA and various other sources (e.g. the project consortium has strong links to the Austrian and German and Belgian collaborative ground segments)
Too low momentum to ensure openEO sustainability	M	WPs 1-7	Consortium has broad network within respective community, including links to EC Copernicus, ESA, Proba-V MEP, which will be engaged to take up openEO.
Loss of key staff	L	WPs 1-7	Re-distribution of tasks within organization and/or consortium to ensure continuity. If possible, replacement with qualified staff within resp. organization or engagement of new staff.
Loss or non-performance of consortium members	L	WPs 1-7	The consortium members have a longstanding cooperation, having worked together in various projects and partner combinations over several years. Thus, a non-performance is not expected. Procedures to handle opt-outs or similar events will be set within the CA.
Conflicts of interest between partners	L	WPs 1-7	Measures at the KO meeting: Creating the big picture of the project to ensure a common understanding of the project goals, clear definition of partner roles, communication rules and approaches to solve disputes. Dispute regulations to be defined within the CA.
Overspending	L	WP 1	Clear communication of financial details from TU Wien to partners, financial monitoring on partner level.

[59] L: Low, M: Medium, H: High

Unavailability of processing infrastructure	L	WPs 3-7	Project is not reliant on one processing environment (i.e. EODC, VITO, AWS). In case of *force majeure* events (fire, earthquake, strike, etc.) back office services may be unavailable for prolonged periods. In this case, the openEO implementation and use cases will simply be tested at other back offices participating in the project.
Disputes on licenses of source code used within openEO	L	WPs 1, 3-6	openEO will solely use open source code based on the Apache license 2.0. Non-open source code will not be accepted as parts of the project deliverables.
Lack of user feedback	L	WPs 2-5	WP 2 involves a variety of user involvement measures, i.e. social media accounts, webpage, user workshops, hackathon and interviews and will actively seek for constant user feedback. The EO data network the consortium is already part of will be actively engaged to give feedback.

(d) Decision finding

Decisions will be made throughout the project on all levels. If a decision affects the project course and/or more than one partner, an agreement between the affected parties has to be found. The first person to bring about an agreement is the respective WP-lead. If more than one WP is involved, the PM-Team has to be engaged. In case this still does not lead to a solution, the openEO Project Board will be included. There, a solution via democratic vote will be made. In addition, dispute resolution and the design of decision findings will be defined within the CA.

3.3 CONSORTIUM as a WHOLE

The openEO consortium is composed of 10 organisations in total, including 4 SMEs (**EODC, Mundialis, Synergise, Solenix**), 3 universities (**TU Wien, WWU, WUR**), 2 private research centres (**VITO, EURAC**) and the **JRC**. The consortium covers seven European countries: Austria, Germany, The Netherlands, Belgium, Slovenia, Italy and Switzerland.

Figure 20: Countries participating in openEO

It is worth mentioning here that the consortium will be supported by external organisations committed to the development of the openEO interface. This includes the Google Earth Engine team who will actively participate in open source community activities and develop a driver API for GEE.

Some project participants have their focus more in the EO side and infrastructure provision (**EODC, TU Wien, VITO, EURAC**), some more as developers of software packages which open the data to a wide range of users i.e. they bring new technologies and concepts into the world of EO processing (**Mundialis, Synergise, Solenix, WWU, WUR, JRC**). However, irrespective of the fact whether they have entered the field of Big EO Data from the EO- or IT-side, all participants of the openEO consortium have already experience in carrying out practical studies with very large amounts of diverse EO data. They all know how to set up cloud environments to access, process and analyse Big EO Data. This guarantees that we can start from day one with our own experience.

The consortium partners cover a large spectrum of EO data processing and handling expertise:

- ranging from thematic EO know-how to IT-centred expertise
- ranging from radar to optical sensors
- ranging from medium resolution at global scale to (very) high resolution at local scale
- ranging from operating cloud based infrastructures to using these platforms
- ranging from operational services/solutions to software packages which are used in 3rd-party solutions

The project coordinator **TU Wien** brings in its thematic expertise in continental to global-scale EO data processing and – through the EODC – will provide some of the back office functions needed in this project (Petabyte storage, supercomputing). TU Wien is one of the founding organisations of the **EODC** which is a public-private partnership founded in response to the growing need for cooperation in EO due to the significant scientific, technical and organisational challenges when moving EO data processing into the cloud[60]. The mission of EODC is to work together with its partners from science, the public and the private sectors in order to foster the use of earth observation data for global monitoring of land and water by[61]:

(i) setting up, managing and operating a virtualised, distributed EO data centre
(ii) providing collaborative IT infrastructure for archiving, processing, and distributing EO data
(iii) organising collaborative software development processes for establishing fully automatic end-to-end EO data processing chains
(iv) building up comprehensive competence in processing large quantities of EO data

The **EODC** cooperates closely with its so-called Principal- and Associated Cooperation Partners. At the time of writing this proposal, the EODC cooperation network includes nine Principal- and seven Associate Cooperation Partners from nine Countries, including the openEO consortium partners TU Wien, VITO, and EURAC.

The Institute of Geoinformatics of **WWU** has a central role in shaping the specifications of the openEO API (specific objective 2 to define and develop the Core API; see section 1.1) from their broad experience in handling and analysing spatial and spatio-temporal data. WWU does this in close collaboration with back-end owners (VITO, EODC/TU Wien, EURAC, GEE, …; see specific objective 3 to develop driver APIs in section 1.1) and partners, who either have interest in a specific front-end technology for addressing the needs of their users (e.g. VITO and EODC for Python) or because they take the lead in e.g. R packages (WWU with R-Spatial), which relates to the expertise necessary for objective number 4 to develop client APIs.

With **WUR** and **EURAC** we have top European players on board who have a clear need to process and analyse large amounts of heterogeneous EO data at different platforms in environmental research, i.e. ideal consumers of the openEO interface. In addition, other project participants as **VITO** and **TU Wien** do have significant activities in developing and operating EO applications and will contribute here to the consortium as a whole. With **JRC** we have a key European player on environmental research on board with a solid experience and need (to shape the European research agenda) in moving massive EO processing to cloud based resources.

Many participants have very good relations with the EC Copernicus units and ESA (ESRIN). These will be used to establish an intense dialogue with them. **EODC** and **TU Wien** for example may use their involvement in ESA's Climate Change Initiative (CCI) and the Copernicus Global Land (C-GLOPS) and Climate Change Services (C3S) as a platform to introduce openEO to a broader audience and to discuss related issues. The same applies to

---

[60] Wagner, W. (2015): *Big Data infrastructures for processing Sentinel data*, **Photogrammetric Week 2015**, Dieter Fritsch (Ed.), Wichmann/VDE, Berlin Offenbach, pp. 93-104.

[61] Wagner, W.; Fröhlich, J.; Wotawa, G.; Stowasser, R.; Staudinger, M.; Hoffmann, C.; Walli, A.; Federspiel, C.; Aspetsberger, M.; Atzberger, C.; Briese, C.; Notarnicola, C.; Zebisch, M.; Boresch, A.; Enenkel, M.; Kidd, R.; von Beringe, A.; Hasenauer, S.; Naeimi, V.; Mücke, W. (2014): *Addressing grand challenges in earth observation science: The Earth Observation Data Centre for Water Resources Monitoring*, **ISPRS Commission VII Symposium**, Istanbul, Turkey, 29 September-2 October 2014, ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences (ISPRS Annals), Volume II-7, pp. 81-88.

VITO with their involvement in the Proba-V Mission Exploitation Platform (MEP) and Copernicus Global Land Services. Several of us, including as well Sinergise, WUR and Solenix are involved via several R&D projects with ESA and the EC in shaping the landscape of Copernicus related services and IT infrastructures. Also on the upcoming DIAS platform, most of the partners will have a role at least as a third party. All partners are used to work with large EO data (TB to PB range) and the consortium as a whole has both the IT and EO knowledge needed for this project. Thanks to the different important roles in operational platforms and services, we have a solid base to exploit the results of openEO after the project end. We have taken care that the persons, who do have the contacts with important stakeholders and users, are on board in the openEO project team.

All of the partners are active users of open source software, many of them have actively contributed to open source projects, and some of them (WWU, Mundialis) successfully are leaders of large community projects (R-spatial, GRASS). Having this experience in the consortium is necessary for the first objective of the project to establish openEO as an open source initiative and will significantly decrease the risk of the project not gaining enough momentum during the project's lifetime to survive, and reach its ambition to become "the GDAL of cloud-based EO processing" beyond the lifetime of the project. A small amount (€ 20.000) is reserved for subcontracting by Spatialys, a small company founded by Even Rouault, which is dedicated to providing expertise around Open Source geospatial software and standards. Even Rouault has been the main developer and project leader of GDAL over the last 5 years.

A potential risk of IT oriented projects is not so much the lack of technical skills, but rather the lack of clear user requirements and user involvement. This risk is kept to a minimum with the openEO project thanks to numerous project partners already strongly involved with specific user communities by operating services or providing dedicated software packages. We will further address the user community in the user workshops and the dissemination activities. As stakeholders in the infrastructures and software package, we will be encouraged by several of these users to move forward. In fact, the reason why we collaborate is that we need to realise the foreseen progress beyond state of the art as being pushed by our users who actively ask for easy-to-use API's on our data and/or complementary data. As an example Solenix, WUR and Sinergise build applications for end-users who need both access to Proba-V data (at VITO) and Sentinel-1/2 data (at EODC) via their preferred user oriented interface.

The four use cases (WP 6 – specific objective 5 to develop and publish use cases; see section 1.1) have been carefully chosen to cover a wide range of existing user communities in various domains on which we are convinced to have impact with this project. Key users who will support the evaluation of the project results have been selected; they have close working relationships with consortium partners and have an interest to serve as a champion user on the different platforms of the openEO interface. Letters of Support of these key users are available in annex of this proposal.

Specific objective 6 to validate the openEO interface (see section 1.1) is covered by all project participants since we all have a clear interest in the results of the project both in offering openEO as part of our back-end or in using the front-end in our projects and operational services. Of similar importance are the links to other back-ends offered by Google and Amazon, where we will test our openEO API (Task 6.3 in WP 6). The participants JRC, WWU, EODC and WUR already have large experiences here.

Partners TU Wien and WWU together lead the dissemination of openEO. The reason for this is that TU Wien has an excellent position in the EO data analysis community, and WWU has an excellent position in the geoinformatics and open source community. The WWU Institute of Geoinformatics' first mission statement is "to conduct high-quality research in geoinformatics and communicate it effectively to science and society, e.g. via open access publications and open source software". Hence, it considers projects like openEO as its most important activity. Jointly leading the dissemination work package guarantees leverage of the EO community as well as the geoinformatics and open source community.

The different partners of openEO do not only have the necessary expertise to realise the objectives of the project and ensure impact and further exploitation of the results as illustrated above. All partners have also a deep experience in research and development acquired in many cooperative projects within the previous EC Framework and ESA Programmes. Moreover, all partners have already cooperated fruitfully within past successful projects. Some examples: Solenix and Sinergise do work with services offered by VITO on the Proba-V MEP platform. EURAC, TU Wien and VITO are partners in EODC. TU Wien, VITO, JRC and WUR collaborate in various Copernicus services, e.g. Copernicus Land. Mundialis has working relationships via their GRASSGIS package with most of the partners. WWU and WUR have further close working relationships in joint research on optimizing statistical methods for remote sensing time series analysis; this has led to joint publications and joint supervision of PhD students. Mundialis and WWU have a long-term relationship in jointly organizing summer schools on analysing spatio-temporal data with open source software.

From past experience and common activities they built a common vision on how to evolve the European Copernicus ecosystem, reflected in this proposal.

Finally, all participants are strongly committed to the project objectives. These objectives are in fact in line with each Partner's strategic objectives and/or institutional mandate and evolution. All Partners see in the project a strong and sustainable development opportunity. The consortium for this proposal was organically shaped since we all see the need to realise the same objectives, fully in line with the agenda of ESA and EC on Copernicus and the upcoming DIAS platform. As explained above the partners are fully complementary since we cover platforms, user-oriented applications, different R&D areas and software packages.

TABLE of FIGURES