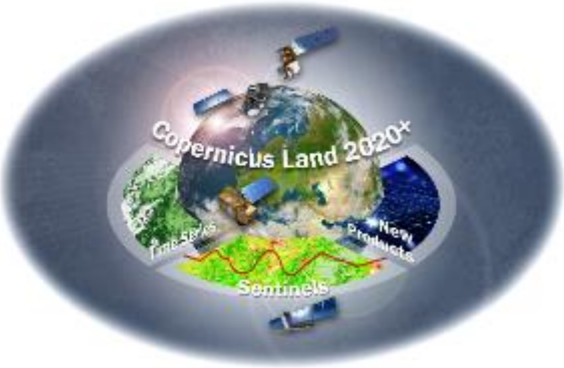
Horizon 2020

Space Call - Earth Observation: EO-3-2016: Evolution of Copernicus services

Grant Agreement No. 730008

ECoLaSS

Evolution of Copernicus Land Services based on Sentinel data



D5.2

"D23.1b: Service Infrastructure/Architecture Requirements Report"

Issue/Rev.: 2.0

Date Issued: 20.12.2019

submitted by:

DLR

in collaboration with the consortium partners:

SIRS, JOANNEUM RESEARCH DIGITAL, UCL Université catholique de Louvain, GAFAG an e-GEOS (ASI / Telespazio) Company

submitted to:

European Commission – Research Executive Agency

This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme, under Grant Agreement No. 730008.

CONSORTIUM PARTNERS

No.	PARTICIPANT ORGANISATION NAME	SHORT NAME	CITY, COUNTRY
1	GAF AG	GAF	Munich, Germany
2	Systèmes d'Information à Référence Spatiale SAS	SIRS	Villeneuve d'Ascq, France
3	JOANNEUM RESEARCH Forschungsgesellschaft mbH	JR	Graz, Austria
4	Université catholique de Louvain, Earth and Life Institute (ELI)	UCL	Louvain-la-Neuve, Belgium
5	German Aerospace Center (DLR), German Remote Sensing Data Center (DFD), Wessling	DLR	Wessling, Germany

CONTACT:

GAF AG
Arnulfstr. 199 – D-80634 München – Germany
Phone: ++49 (0)89 121528 0 – FAX: ++49 (0)89 121528 79
E-mail: copernicus@gaf.de – Internet: www.gaf.de

DISCLAIMER:

Document Release Sheet

	Name, Function	Date	Signature
Author(s):	Andreas Hirner (DLR) Gernot Ramminger (GAF) André Stumpf (GAF) Sophie Villerot (SIRS)	31.01.2019	
Review:	Annekatrin Metz-Marconcini (DLR) Katharina Schwab (GAF) Markus Probeck (GAF)	15.02.2019	A. Metz-Marconcini
Approval:	Eva Sevillano Marco (GAF)	20.12.2019	
Acceptance:	Massimo Ciscato (REA)		
Distribution:	Public		

Dissemination Level

Dissemination Level		
PU	Public	X
CO	Confidential: only for members of the consortium (including the Commission Services)	

Document Status Sheet

Issue/Rev	Date	Page(s)	Description / Changes
1.1	20.12.2017	24	First issue of the Service Infrastructure/Architecture Requirements Report
2.0	18.02.2019	35	Second issue of the Service Infrastructure/Architecture Requirements Report with new chapters 3.2 and 4.2, major updates of chapters 2.4, 2.4.1, 2.4.2, 2.6.1.3 and 5, and minor updates of chapters 1.1 and 2.6.1. Annex 1. ECoLaSS experiences and potential interaction with DIAS

Applicable Documents

ID	Document Name / Issue Date
AD01	Horizon 2020 Work Programme 2016 – 2017, 5 iii. Leadership in Enabling and Industrial Technologies – Space. Call: EO-3-2016: Evolution of Copernicus services. Issued: 13.10.2015
AD02	Guidance Document: Research Needs Of Copernicus Operational Services. Final Version issued: 30.10.2015

ID	DOCUMENT NAME / ISSUE DATE
AD03	Proposal: Evolution of Copernicus Land Services based on Sentinel data. Proposal acronym: ECoLaSS, Proposal number: 730008. Submitted: 03.03.2016
AD04	Grant Agreement – ECoLaSS. Grant Agreement number: 730008 – ECoLaSS – H2020-EO-2016/H2020-EO-2016, Issued: 18.10.2016
AD05	D5.2 – D23.1b: Service Infrastructure/Architecture Requirements Report (Issue 1.1), Issued: 18.02.2019

EXECUTIVE SUMMARY

The Horizon 2020 (H2020) project, "Evolution of Copernicus Land Services based on Sentinel data" (ECoLaSS) addresses the H2020 Work Programme 5 iii. Leadership in Enabling and Industrial technologies - Space, specifically the Topic EO-3-2016: Evolution of Copernicus services. ECoLaSS is being conducted from 2017–2019 and aims at developing and prototypically demonstrating selected innovative products and methods as candidates for future next-generation operational Copernicus Land Monitoring Service (CLMS) products of the pan-European and Global Components. ECoLaSS assesses the operational readiness of such candidate products and eventually suggests some of these for implementation. This shall enable the key CLMS stakeholders (i.e. mainly the Entrusted European Entities (EEE) EEA and JRC) to take informed decisions on potential procurement as (part of) the next generation of Copernicus Land services from 2020 onwards.

To achieve this goal, ECoLaSS makes full use of dense time series of High-Resolution (HR) Sentinel-2 optical and Sentinel-1 Synthetic Aperture Radar (SAR) data, complemented by Medium-Resolution (MR) Sentinel-3 optical data if needed and feasible. Rapidly evolving scientific developments as well as user requirements are continuously analysed in a close stakeholder interaction process, targeting a future pan-European roll-out of new/improved CLMS products, and assessing the potential transferability to global applications.

The ECoLaSS project comprises several EO institutions and companies with different IT infrastructures. These infrastructures developed in accordance to the processing requirements faced by each stakeholder and were increasingly influenced by three general trends that developed during the last 10 years. These trends are:

- Availability of medium to high resolution EO data with a high temporal resolution on a global scale
- the possibility to create cheap off-the-shelf computer clusters and (private) cloud environments that can be adapted for a wide range of storage and processing tasks
- a big set of (mostly Open Source) processing frameworks that monitor the hardware and help organize the processing tasks

In order to cope with different setups of single, independent IT structures several techniques show great promise for interaction:

- Virtualization allows decoupling of processing algorithms from the underlying hardware or OS and therefore, minimizes dependency management
- Processing services offer the possibility to run anybody's algorithm on huge data sets by exchanging code instead of data

Adoption of the last two strategies should enable increased collaboration of ECoLaSS project partners as well as of any operational Copernicus service providers and help making the transition from test to operational production processing environments much easier.

In order to be able to assess future processing environments an evaluation catalogue is presented that will help to analyse the capabilities of target processing platforms that will be accessible to the project. A result of this evaluation will constitute a major part of the second deliverable in this work package. However the catalogue will also be helpful in choosing an appropriate architecture of the tools provided by this project.

The second issue of this report used the evaluation catalogue to assess the state of all DIAS centres. As of this date the services offered look similar but differ in many details regarding spatial and temporal availability of EO data sets and ancillary data. A selection of a specific processing environment will therefore always benefit from a requirement analysis of a processing task.

Table of Contents

List of Figures and Tables

Abbreviations

AATRS	Advanced Along-Track Scanning Radiometer
Apache SF	Apache Software Foundation
API	Application Programming Interface
ARD	Analysis-ready Data
ASAR	Advanced Synthetic Aperture Radar
AWS	Amazon Web Services
C3S	Copernicus Climate Change Service
CAMS	Copernicus Atmosphere Monitoring Services
CCE	Cloud Container Engine
CEMS	Copernicus Emergency Management Service
CLMS	Copernicus Land Monitoring Service
CLS	Collecte Localisation Satellites
CMEMS	Copernicus Marine Environment Monitoring Service
CSW	Catalogue Search Web Service
CPU	Central Processing Unit
DAGs	Directed Acyclic Graphs
DEM	Digital Elevation Model
DIAS	(Copernicus) Data and Information Access Services
DLR	Deutsche Zentrum für Luft- und Raumfahrt
EC	European Commission
ECMWF	European Centre for Medium-Range Weather Forecasts
ECS	Elastic Cloud Server
EEA	European Environment Agency
EEEs	Entrusted European Entities
ENS	Elastic Node Server
ENVISAT	Environmental Satellite
EO	Earth Observations
EOC	Earth observation centre
ESA	European Space Agency
EVS	Elastic Volume Service
EW	Extra-Wide Swath Mode
FWaaS	FireWall as a Service
GB	GigaByte
GDAL	Geospatial Data Abstraction Library
GEE	Google Earth Engine
GFS	Google File System
GPFS	General Parallel File System
GPU	Graphical Processing Unit
GOME	Global Ozone Monitoring Experiment
GRD	Ground Range Detected
HDA	Harmonized Data Access
HDFS	Hadoop Distributed Filesystem
HPC	High Performance Computing
HR	High Resolution
HRL	High Resolution Layer
I/O	Input/Ouput
IaaS	Interface as a Service
ICT	Information and Communications Technology
IRS	Indian Remote Sensing Satellite
IT	Information Technology
IW	Interferometric Wide Swath Mode

LBaaS	Load Balancers as a Service
LUCAS	Land Use/Cover Area frame statistical Survey
MERIS	MEdium Range Imaging Spectrometer
MODIS	Moderate Resolution Imaging Spectroradiometer
MSI	Multi-Spectral Instrument
NAS	Network Attached Storage
NPP	National Polar-orbiting Partnership
OCN	Ocean
OGC	Open GIS Consortium
OLCI	Ocean and Land Color Instrument
OLDA	(EUMETSAT's) OnLine Data Access
OS	Operating System
OTB	Orféo ToolBox
OTC	Open Telekom Cloud
PM	Project Manager
PROBA-V	PRoject for On-Board Autonomy–Vegetation
RAM	Random Access Memory
RD	Research & Development
RDD	Resilient Distributed Dataset
RDS	Relational Database Service
REST	Representational state transfer
RUS	Research and User Support
S-1	Sentinel-1
S-2	Sentinel-2
S-3	Sentinel-3
SATA	Serial AT Attachment
SDN	Software Defined Networking
SF	Software Foundation
SLC	Single Look Complex
SLSTR	Sea and Land Surface Temperature Radiometer
SM	Strip Map Mode
SPARQL	SPARQL Protocol and RDF Query Language
SRAL	Sentinel-3 Ku/C Radar Altimeter
SSD	Solid State Discs
TB	TeraByte
TEP	Thematic Exploitation Platform
ToA	Top of Atmosphere
TROPOMI	TROPOspheric Monitoring Instrument
URL	Uniform Resource Locator
USGS	United States Geological Survey
VITO	Vlaamse Instelling voor Technologisch Onderzoek
VM	Virtual Machine
VPN	Virtual Private Network
VPNaaS	VPN as a Service
WBS	Work Breakdown Structure
WIZIPISI	Wrocławski Instytut Zastosowań Informacji Przestrzennej i Sztucznej Inteligencji
WMS	Web Mapping Service (OGC)
WMTS	Web Map Tile Service
WFS	Web Feature Service (OGC)
WP	Work Package
WPS	Web Processing Service (OGC)
WPD	Work Package Description
XML	Extensible Markup Language

YARN	Yet Another Resource Negotiator
ZFS	Zetta File System

For more specific terms see chapter 1.3 – Terminology.

1 Introduction

1.1 Current and Future Challenges

The launch and operation of the Sentinel 1, 2 and 3 Earth observation satellites has resulted in a sharp increase of remote sensing data (SERCO Spa, 2016) for global land and ocean monitoring services. Together with already existing data sets (e.g. Landsat) users suddenly see new possibilities and are willing to tackle questions with a continental perspective or even global scope, which is not limited to a single time slice but tries to reveal changes and developments in order to explain the dynamics of land and ocean processes.

Rapid developments in the IT industry epitomized by the buzz words of “big data” and “cloud” open new ways of dealing with large volumes of data. Large data volumes do not need to be downloaded and stored in local computing environments, now they can be processed on site. New processing paradigms change the way data can be processed and IT provider offer different environments, which are geared towards specific tasks.

Therefore, users face the problem of dealing with massive EO data sets in a new way. Approaches taken by different organizations are highly diverse and rapidly evolving, especially since the appearance of readily available commercial EO processing platforms (e.g. Google Earth Engine, Amazon Web Services).

This second issue of this report provides also an investigation of the five existing DIAS centres, which were evaluated according to the checklist outlined in the first issue. Therefore chapter 3 has a new subtopic (3.2, Comparison of DIAS Providers) that contains the aforementioned comparison in a tabular form and corresponding comments. Apart from that there are some updates to projects, where new information was available. Finally chapters 4 (requirements) and 5 (Outlook) reflect on some of the new findings.

1.2 Components

Hardware of a minimal processing system usually consists of at least the following components:

- Storage units
- Processing units

In its simplest form, this can be a single computer, but with large volumes of data and many processing tasks, storage is distributed over many hard drives and computing is done by multiple processors with many cores spread across many computers. Therefore

- Network infrastructure

becomes an important issue. There are many ways to organize computer infrastructure, which strongly depend on the main purpose of the processing system. A subsequent chapter will discuss the rationale and consequences of different hardware setups. In addition to that, fault tolerance requires dedicated nodes for monitoring and managing the hardware. The same is valid for scheduling and managing the processing tasks, resulting in hardware solely responsible for process execution, coordination and orchestration.

Minimal processing software might be just one or more executables that are triggered on demand, but growing hardware complexity also leads to specialised software components. Common software components are used for:

- Data transformation
- Data transfer
- (Scientific) computing
- Task scheduling
- Task monitoring
- Hardware resource monitoring

These functions are also dependent on the processing infrastructure and can get extremely complex, requiring dedicated software packages for each of them.

1.3 Terminology

Many terms in computing are ambiguous and may have a different meaning in a given IT subdomain. This chapter describes terms as they are considered in the frame of this document.

Table 1: Terminology of common but ambiguous IT phrases

Resource	a storage (usually measured in Bytes) or processing unit (usually measured in CPUs or cores)
Cloud	the sum of specific computing infrastructures with specific interfaces accessible in the internet (public cloud) or intranet (private cloud). Computing infrastructures usually offer specific services that can be categorized in different categories
IaaS (Infrastructure as a service)	offers computing resources in the form of operating systems, virtual machines, container or storage
PaaS (Platform as a service)	offers services to develop and host custom built applications
SaaS (Software as a service)	offers software like databases, servers, default applications, etc.
Orchestration	a procedure that accepts job requests, monitors available resources and triggers new jobs whenever there are sufficient resources
Virtual machine (VM)	a hardware system virtualization managed by a hypervisor software running on a host
Container	an operating system level virtualization managed by the kernel of a host OS
I/O operations	read and write operations
Node	one computer in a computer cluster

2 Analysis of service infrastructure and architecture

The following chapter has its focus on the analysis of service infrastructure and architecture. It is divided into several subsections describing the processing concepts and challenges (section 2.1), the analysis of the current situation and the limitations (section 2.2), and the background and implications of processing paradigms (section 2.3). Furthermore, section 2.4 is dedicated to DIASes, followed by section 2.5 on EO data initiatives in Europe. The last section in this chapter compiles the experiences regarding processing within the ECoLaSS consortium (section 2.6).

2.1 Processing concepts and challenges

Each setup of a processing system has advantages and disadvantages. This chapter will highlight common aspects regarding storage, networking and processing units.

Storage can be organized in several ways.

- A data storage server with many hard drives connected to a network (e.g. NAS – network attached storage)
- A group of ordinary computers with one more hard drives

In the first case, the storage server is the single point of entry and might become a bottleneck once the data flow through the network gets saturated. A big advantage of this setup is its ability to hide the single hard drives and offer the sum of their total volume as a single logical unit and provide redundancy.

The latter case allows linking cheap off the shelf computers in a network in such a way that each can be accessed via the network. This increases the potential data transfer volume, but has one big drawback: How are data distributed and accessed? In a primitive setup data must transferred manually and if storage of a given computer is full, data relocation and even modification will become a problem. The same applies for search and access functions. Each computer must be searched and indexed in order to find a data set. Despite that this setting offers a big advantage regarding processing: Data on a given computer can be processed with the CPUs of this machine without data transfer across the network. This opens the possibility for parallel processing. But if it is cumbersome to locate data, how are processing tasks assigned according to different search criteria? There are numerous strategies to solve these problems and developments in the last decade addressed many problems and provided technical solutions. They will be discussed in the following chapter.

2.2 Analysis of current situation and limitations

One development regards distributed file systems (also called network file systems). Here multiple hard drives across many computers are linked by a specific protocol and managed by dedicated software present on each node. There are numerous implementations with different design goals like transparency (clients reading data do not need to know where data are), scalability, redundancy and heterogeneity. Common examples are *General Parallel File System* (GPFS) by IBM (IBM, n.d.), *Google File System* (GFS) by Google (Ghemawat, Gobioff, & S.T., 2003), *Zetta File System* (ZFS) by Oracle (Oracle, 2012), *Hadoop Distributed Filesystem* (HDFS) by Apache SF (Apache, 2013) and many more.

Another development is concerned with computing resource management. While distributed file systems allow users to manage disc space, the load of CPUs on each node has to be monitored in order to effectively distribute processing tasks. The main components of cluster management software are job scheduler and workload manager. The former accepts processing requests and stores them in a queue, the latter monitors the load of computer nodes. Whenever there are computing resources available on one of the nodes, the workload manager searches the schedulers for a matching task and assigns the task to that node for execution.

Processing requests often have to be registered beforehand with the job scheduler in order to link the processing software with required hardware resources like RAM and number of cores. This is important in order to optimize compute resource allocation. In addition to that cluster management software includes job monitors and might offer tools for deployment, scaling and other tasks. Popular examples are YARN (Yet Another Resource Negotiator) and Mesos (Apache, 2017), both by Apache SF, and Kubernetes (Linux Foundation, 2017) by Google.

An area linked to cluster management deals with the combination of single jobs into complex workflows. This is sometimes also part of orchestration. Dedicated software systems allow users to define tasks, construct workflows and register processing endpoints on local or public processing systems. Once triggered the orchestration software conditionally executes a workflow and logs its progress and status. Popular products are Ansible (Red Hat, Inc., 2017) developed by Red Hat, Jenkins (Jenkins Project, n.d.) by Open Project, Airflow (Apache Inc., n.d.) and Mesos (both developed by Apache SF).

Substantial improvements regarding the configuration and deployment of processing software was achieved with containers. Containers are an operating-system-level virtualization technique, which allows the execution of an encapsulated OS on a host computer. Since the technique only requires very limited resources with a minimal footprint it is possible to run many containers on a host system. The main strength of this technique is the fact that a user can construct a container image that contains all processing software needed for a specific task. Once the container image is constructed, it can be deployed on the nodes of a processing cluster. If deployment is complete a job request can create a container and execute the software provided by it. If the job is complete the container is destroyed and its resources are freed. Using containers removes the need to install specific software packages on the processing nodes and considerably simplifies the management dependencies. Well known examples are Docker (Docker Inc., 2017) and LXC (Canonical Ltd, n.d.) .

Last but not least there are many efforts to increase processing algorithms itself. A major processing bottleneck is I/O operations. The development of stream processing frameworks tries to limit read and write operations and provides software tools that allow feeding datasets through a system of pipes where the data stream can be modified or analysed on the fly before the final product is written to the disc. Other processing frameworks are adapted to the structure of a specific data type and provide optimised tools and procedures for their efficient handling. They will be introduced in the following chapter.

A final paragraph is concerned with hardware development. The power of graphical processing units (GPUs) has increased massively due to the ever-rising demands of the gaming industry. But this kind of hardware is also perfectly suited for massively parallel processing tasks. This capability is now exploited in numerous specific fields (e.g. machine learning, geometric calculations). Developments of solid-state drives (SSD) opened up new possibilities regarding storage. I/O operations on these types of disk are much quicker and can lead to significant performance boosts during processing. Creation of specific computer systems with large amounts of RAM is also geared towards the speed-up of I/O operations. All these techniques are often used together with conventional CPUs and discs in a hybrid setup.

It should be mentioned that large scale processing has traditionally been done on high performance computing (HPC) clusters with specific hardware geared towards optimal performance. These systems are usually very expensive and the access is commonly restricted. However, there are increasing numbers of HPC services from public and commercial providers, which can be utilized for specific tasks.

2.3 Processing paradigms: background and implications

The chapter above was concerned with developments in separate fields, but recent trends have resulted in processing paradigms that describe recipes for specific processing scenarios. These recipes were picked up by the Open Source community in concert with commercial actors, which implemented processing frameworks that can be adapted and utilized by anybody.

An influential development was the Map-Reduce concept published by Google in a scientific paper in 2003 (Ghemawat, Gobioff, & S.T., 2003). The concept describes how large data sets are broken up and distributed across processing nodes where each fragment is analysed separately (Map). The results from each node are then collected and aggregated (Reduce).

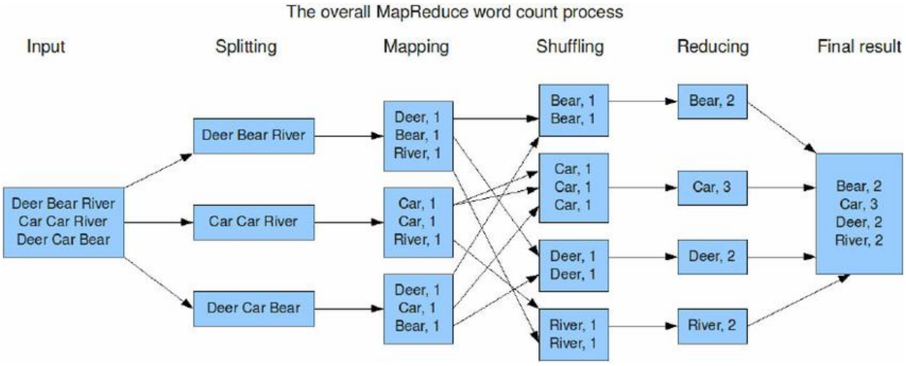


Figure 1: MapReduce data flow

The idea was immediately picked up by developers and resulted in the Apache-Hadoop project. The central components of Hadoop are the Hadoop Distributed File System (HDFS) and Hadoop YARN, a platform responsible for managing computing resources in clusters. A third component is the Hadoop MapReduce API, which defines an interface for developers to implement their own tasks. The Hadoop framework has since grown into large ecosystem, where HDFS and Yarn can be used to manage not only Map-Reduce tasks but also other processing paradigms.

One of those is implemented in the Apache Spark framework, which permits in-memory querying of data instead of disk I/O operations. Spark relies on its own data format called resilient distributed dataset (RDD) and is thus able to implement iterative algorithms that visit their dataset multiple times in a loop, and analysis like the repeated database-style querying of data. Therefore, it is better suited for tasks where data are related to each other. It also allows constructing workflows as multi-step directed acyclic graphs (DAGs) (Thulasiraman & Swamy, 1992) and executing those DAGs all at once, not step by step.

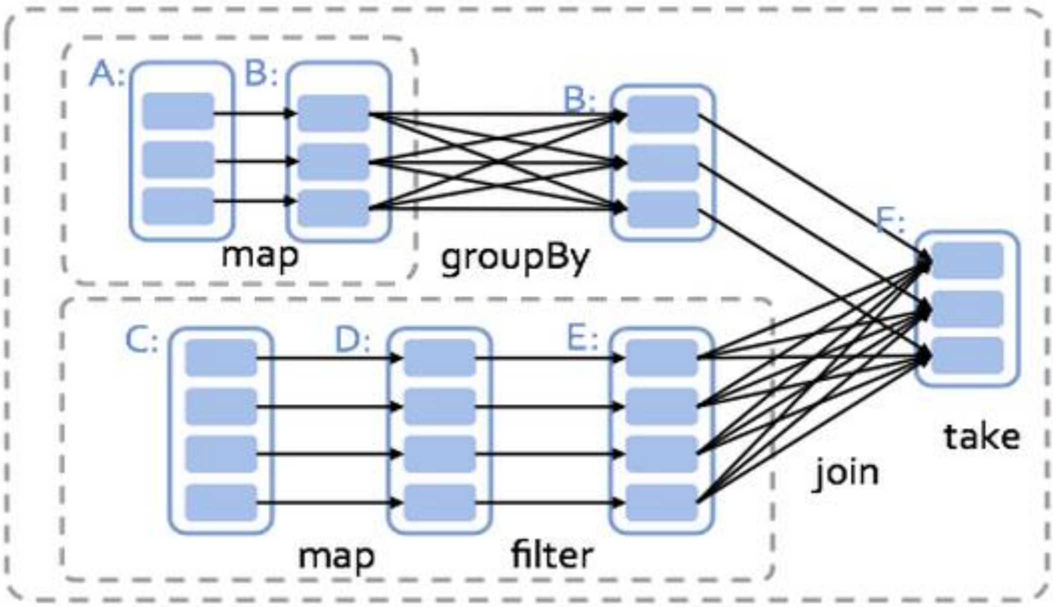


Figure 2: Data flow in Apache Spark

The concept of DAGs is also used by other frameworks that are used to construct complex workflows, e.g Ansible and Apache Airflow.

Other frameworks (Apache Storm, Apache Kafka, Apache Flink) read data and create a stream that can be manipulated in successive steps. These systems achieve an extremely high transfer volume and are able to react on events and define workflows in the form of DAGs. Some of the systems are able to split and distribute the streams to different nodes and therefore increase processing speed. Others read data from a range of producers and allow consumers to query and analyse one or more of them.

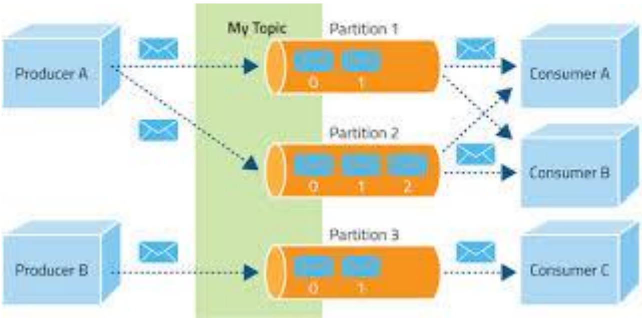


Figure 3: Data flow in Apache Kafka

A strictly organisational approach was pursued by the Apache Mesos framework. Mesos provides applications with APIs for resource management and scheduling across entire data centres and cloud environments. It can orchestrate many different processing frameworks (e.g., Hadoop, Spark, Kafka, Elasticsearch) and most importantly, supports launching Docker containers. It is highly fault tolerant and scalable. Main components are the Chronos scheduler and the Marathon orchestration system.

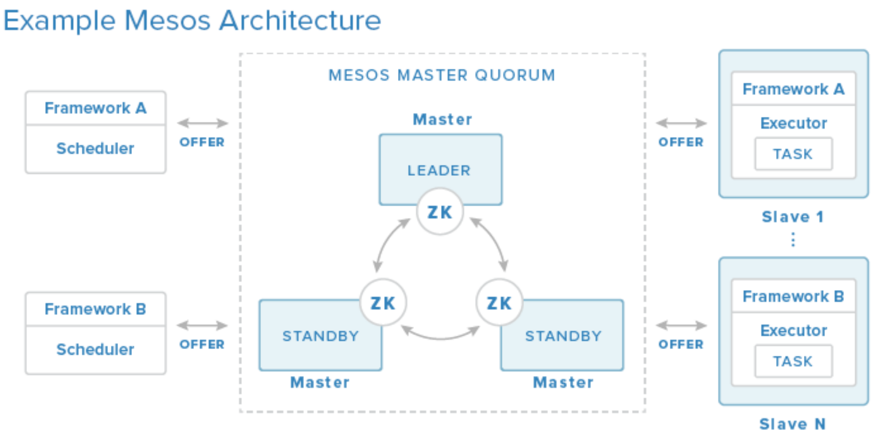


Figure 4: Mesos concept

Similar systems geared strictly towards containers are Kubernetes (Google) and Docker Swarm (Docker Inc.). They share many of the same goals of Mesos, but also manage the distribution and provisioning of containers across nodes.

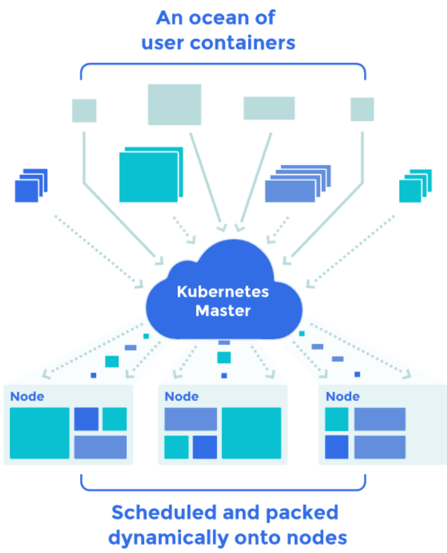


Figure 5: Kubernetes concept

The specific requirements of EO data processing have already attracted service providers to offer specific solutions by Google and Amazon. Both services run on the server farms of the two companies and are accessible in the cloud.

Google Earth Engine, abbreviated into GEE (Google, 2017), computing platform provides a large collection of satellite data (e.g. Landsat, MODIS, Sentinel-1 and -2, and DEMs) that are registered in a catalogue and a corresponding set of APIs for Python and JavaScript for statistical and geospatial analysis. Customers use the API to write their own code, which selects the desired data and algorithms and subsequently triggers processing. The size of the results usually is much smaller than the amount of processed data and can be downloaded by the user. In addition, a user can upload ancillary raster and vector data. Upon registration, the offer is free for research, education and non-profit usage.

Amazon also offers a range of geospatial and satellite data in their S3 storage engine (Amazon Web Services, Inc., 2017). These include Landsat, Sentinel 2, MODIS, Goes, Nexrad, DEM and OSM. A user can work with the tools of Amazon Web Services (AWS) in order to write code, which accesses and manipulates data in a similar way as GEE. The range of tools is extensive (including a range of languages, APIs and tools like databases, webservers, etc.) and there are more different ways to find a solution compared with GEE. The services are billed according to the AWS business model and include costs for data storage, processing and transfer.

SAP using its HANA in-memory database also offers services for processing geospatial data (SAP, 2017). The technique allows different modes of storing and processing data with a minimal amount of I/O operations. The technique shares many of the capabilities of Apache Spark.

2.4 DIAS

The Copernicus Data and Information Access Service (DIAS) aims to maximize the exploitation of Copernicus data by a broad user community adopting a dynamic approach using the latest ICT and EO technologies (European Commission, 2017). It focuses on increased industry responsibility in a federation of centres that consist of distributed and integrated ground segments. Section 2.4.1 explains the DIAS concept, and section 2.4.2 provides a comprehensive overview of the current DIAS providers and technical implementation.

2.4.1 Concept

Each of the centres offers so-called operations that fall into two categories, the back-office and front-office operations. A given DIAS provider is in charge of the entire storage and processing infrastructure, its services and corresponding interfaces, which make up a single back-office. These services enable efficient local processing of the complete local archive of Copernicus data and/or existing Copernicus distribution services. A back-office is designed to be scalable and includes a broad range of services, interfaces and tools to access and manipulate the data. In addition to data retrieval and manipulation modules, users can also expect discovery, catalogue and view services.

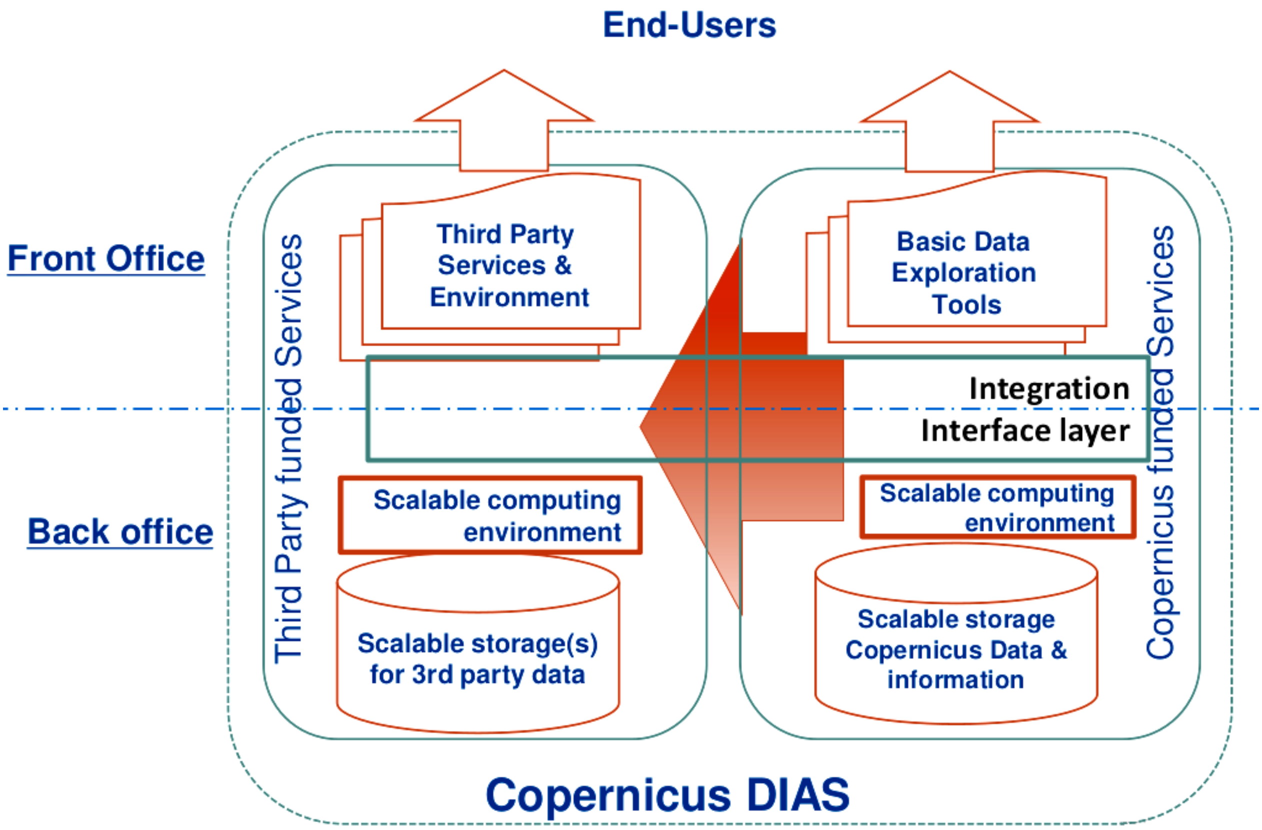


Figure 6: DIAS concept

Products generated and services provided by the back-office operations can be utilized by front-office operations. Any third party that wishes to exploit the data can use the back-office infrastructure to generate their own products and related services in a front-office entity. Therefore, the DIAS provider offers technical support for integration of third party data sources and software whenever necessary. The concept also envisions that services and products of any third party is also usable by other third parties, thus establishing a broad utilization of data while at the same time trying to minimize redundancies.

Due to this construct a vital part of DIAS is the quality, stability but also the flexibility of interfaces and services. If they are too rigid, users will find the system inconvenient to utilize and move to other service providers. In case they are too instable, a third-party front-office user will consider the constant effort for maintenance too tiresome and also look for another provider. It therefore aims to closely resemble established and thus stable industry and consortia standards, which will help users to quickly adapt to the platform without major adjustments. It is also necessary to offer not just one processing paradigm, but maybe several different ones in order to offer the right tools for a given processing task.

Apart from the software capabilities the underlying hardware platform needs to be powerful, scalable and adaptable while still being cost efficient. Particularly the cost of the services has meanwhile established as a main discriminating criterion, also in comparison with other commercial cloud service providers like

AWS. Even small unit costs do have significant impacts when large quantities of data are to be processed. A considerable advantage of the European-industry-led DIASses may be the data protection and security levels offered. It is expected that the DIASses may be primary target platforms for the next-generation CLMS services as assessed in the frame of this project.

2.4.2 Current DIAS implementation

On 14th of December 2017, DG GROW had announced consortia to implement and run five DIAS centres (European Commision, 2017). During 2018 all of them went online. The current platforms and corresponding providers are as follows:

Table 2: Overview of DIAS consortia

Dias	URL	Consortium
CREODIAS	https://creodias.eu https://cf2.cloudferro.com/project/	CreoTech (PM) Sinergise (EO data Manager) Cloudferro (Main Technical Substractor) Geomatys (Data Service Setup Participant) Eversis (Web design/ website maintenance) WIZIPISI (EO RD Manager)
ONDA	https://www.onda-dias.eu	Serco Italia (PM) OVH (Cloud provider) Gael Systems (data access) Sinergise (EO data Manager)
Mundi	https://mundiwebservices.com https://myworkplace.t-systems.com/MyWorkplace/Login.aspx	Atos (PM) T-Systems (Cloud) e-Geos Sinergise GAF EOX Spacemetric Thales Alenia Space DLR (Demand and Data Management)
Sobloo	https://sobloo.eu	Airbus (PM) Orange CapGemini CLS VITO
WekEO	https://www.wekeo.eu	EUMETSAT (PM) ECMWF Mercator Ocean International

The products offered by the DIAS centres will be discussed in the following paragraphs, whereas the evaluation guideline established in Chapter 3 will be used to evaluate the technical capabilities of each of them.

The table below lists the Sentinel products and corresponding temporal availability of all DIAS providers.

Table 3: Sentinel data availability on DIAS centres

Available data	CREODIAS	ONDA	Mundi	Sobloo	WekEO
S-1 Availability	Full archive	Full archive	48 months global	Full archive	Full archive
S-1 RAW Level 0	x	x	x	x	-
S-1 GRD Level 1	x	x	x	x	x
S-1 SLC Level 1	x	x	x	x	x
S-1 OCN Level 2	x	x	x	x	-
S-2 Availability	Global from 2016/10/18	Full archive	48 months global	Full archive	Full archive
S-2 MSI Level 1C	x	x	x	x	x
S-2 MSI Level 2A	x	x	x	x	-
S-3 Availability	Full archive	Full archive	36 months for EU, 12 months global	Full archive	Full archive
S-3 OLCI Level 1	x	-	x	x	x
S-3 OLCI Level 2	x	x	missing info	-	x
S-3 SLSTR Level 1	x	-	x	x	x
S-3 SLSTR Level 2	x	x	missing info	x	x
S-3 SRAL Level 1	x	x	x	-	x
S-3 SRAL Level 2	x	x	x	x	x
S-3 SYN Level 2	x	-	-	x	-
S-5 Availability	Full archive		36 months for EU, 12 months global		
S-5 TROPOMI L1	x	-	x	missing info	missing info
S-5 TROPOMI L2	x	-	x	missing info	missing info

All DIAS centres also host additional datasets, including other satellite products, climate data, derivatives (e.g. Copernicus Services) and auxiliary data useful for data verification and control. The most important ones are listed in the table below. Please check the glossary for abbreviations.

Table 4: Additional remote sensing data availability on DIAS centres

Available data	CREODIAS	ONDA	Mundi	Sobloo	WekEO
Landsat 5	x	-	-	-	-
Landsat 7	x	-	x	-	-
Landsat 8	x	x	x	-	-
ENVISAT/MERIS	x	x	-	-	-
Jason 3	x	-	-	-	-
MetOp GOME	-	x	-	-	-
Suomi NPP	-	x	-	-	-
IRS-P5	-	-	x	-	-
COSMO Skymed	-	-	x	-	-
Proba V	-	-	x	-	-
CAMS	x	x	-	x	-
CEMS	x	x	x	x	-
CLMS	x	x	x	x	-
CMEMS	x	x	-	x	-
C3S	x	x	-	x	-
LUCAS	-	x	-	-	-
ECMWF data	-	-	-	-	x
CMEMS data	-	-	-	-	x

It is apparent that most DIAS providers host similar products. However, there are some differences regarding the temporal and spatial availability of data sets. Customers will have to evaluate which of the centres provides the data needed for a particular task.

Table 5: Prices for common services on DIAS centres

	CREODIAS	**ONDA**	**Mundi**	**Sobloo**	**WekEO**
Price service VM (1 cpu/4ram/8disk) + external IP	€34/month (2c/4r/16d)	€22/month (2c/7r/50disk)	€34/month €3/month	€35/month (s1.medium)	announced for Q2/2019
Price compu. VM (2cpu/16ram/40d.)	€142/month (4c/16r/64d)	€62/month (4c/15r/100d)	€87/month	€142/month (s1.xlarge)	announced for Q2/2019
Price for storage (1 TB)	€31/month	€40/month	€25/month	€20/month	announced for Q2/2019
Price for network (100 GB)	€0.7/100 GB	missing info	€7/100 GB	€7/100GB	announced for Q2/2019

Prices are the second most important topic, when it comes to choosing the right provider. In case all data are available, costs on platforms to be considered are determined by the amount of processing time, storage and network traffic. These parameters can be widely different for any given processing task.

2.5 EO data initiatives in Europe

In Europe, the Copernicus initiative of EU and ESA has resulted in many national and regional initiatives to provide and handle Sentinel data sets. Collaborations between commercial service providers, universities and research institutes are currently being planned and implemented. Examples (ESA, 2017) include, among others:

- CODE-DE project in Germany (Infrastruktur, BVDI - Bundesministerium für Verkehr und digitale Infrastruktur, 2017),
- EODC in Austria (EODC Earth Observation Data Centre for Water Resources Monitoring GmbH, 2017),
- PEPS in France (CNES - Centre National d'Études Spatiales, 2017),
- Terrascope for Belgium (VITO - Vlaamse Instelling voor Technologisch Onderzoek, 2017),
- Hellenic National Sentinel Data Hub for Greece (NOA - National Observatory of Athens, 2016)
- upcoming DIAS by the EC.

They are all designed for processing EO data and adhere, with some deviations, to one of the designs mentioned above.

2.5.1 TEP Urban H2020

There are various programs by EC and ESA to explore the potential utilization of EO data by a large community of users.

An example is the Thematic Exploitation Platform (TEP) initiative by ESA (ESA, 2017, a). Here users of various topics (e.g. Coastal, Forestry, Hydrology, Geohazards, Polar, Urban themes, and Food Security) are encouraged to use an on-line processing platform that provides tools and relevant data to evaluate and analyse certain aspects of their research field. This is in stark contrast to the previous practice of downloading tools and data to a local processing destination, which can be quite time consuming and ineffective. In addition, this approach allows the provision of restricted data that can be visualized and even used in calculations. In the long run, each platform is seen as a common play ground that fosters rapid development of ideas, products, tools and even commercial activities.

Each exploitation platform has to solve similar problems regarding storage, processing and user interaction. The solutions developed in each project will come with specific conventions, services and interfaces bundled into a web application and corresponding back-end systems. The lessons learned from these systems can betray potential preferences and concepts, which might lead to general recommendations and therefore improve the design of other projects.

2.5.2 Research and User Support (RUS) Service portal

The RUS service (ESA, 2017, b) is an expert service for Sentinel users funded, managed and operated by the European Commission, European Space Agency and CS SI respectively. Here registered users can obtain a personal computing environment, which includes a virtual desktop environment based on Linux and connected to various cloud services to provide the user with the required resources (CPU, memory, storage, etc.). The user is able to utilize and enhance his computing environment like an ordinary Linux system, by adding the necessary libraries for his development tasks. In addition to that the user can connect to his environment with other means like ssh/pssh (interaction) and ftp (data exchange). Tools like scihub_download enable quick access to Sentinel data. It is therefore possible to rapidly setup a processing environment and work with Sentinel data. Platforms like that might serve as a test ground for products developed within the consortium.

2.6 Processing in the consortium

Within the consortium there are different approaches regarding the processing chains. In order to describe these experiences appropriately, each subsection focusses on the specific experiences and workflows from one partner (c.f. sections 2.6.1-2.6.4).

2.6.1 High Resolution Layer production - Experiences from GAF

The following sections will focus on the experiences regarding processing gathered throughout the production of both the HRLs 2015 and 2018. Besides the requirements on the selected Service Infrastructure and Architecture the selected setups will be described as well as lessons learnt and an outlook (if possible).

2.6.1.1 High Resolution Layers 2015

In June 2016, GAF together with its main partners SIRS and GeoVille were awarded by EEA with the production of the five CLMS continental component's High Resolution Layers 2015. The following section provides an overview about the chosen Infrastructure and Architecture, required for the production of the HRLs 2015, which GAF was responsible for. The HRLs have been produced mainly on 20m spatial resolution Sentinel data for an area of approx. 5.850.000 km² (EEA39) and covering multiple points in time:

- Imperviousness Layer: Built-up area & imperviousness degrees 2015 and change from 2012, re-processing of 2006-2009-2012 time series, imperviousness change and reference database,
- Forest Layer: Dominant leaf type 2015 and change from 2012, tree cover density 2015 and change from 2012, forest reference database,
- Permanent grassland mask 2015, grass vegetation probability index 2015, ploughing indicator (back to 2008),
- Water & wetness 2015, water wetness probability Index 2009 to 2015,
- Small woody features 2015 based on VHR/HR data 2015 (+/- 1 year)

2.6.1.1.1 Requirements on the selected Service Infrastructure and Architecture

For the production of the HR Layers 2015, GAF was mainly (beside other) responsible for the following tasks also providing the main requirements on the service infrastructure and architecture:

- pre-processing of all optical data (timeframe from 2005 to 2016, covering 8 different sensors from HR IMAGE2006 and HR IMAGE2009 (IRS-P6, Spot-4, Spot-5), HR IMAGE2012 (IRS-P6, Resorcesat-2, Spot-4, Spot-5) HR IMAGE2015 (Resourcesat-2, Spot-5, Sentinel-2) as well as additional Sentinel-2 and Landsat-5/-7/-8 (2005 to 2016)
  - data selection and ordering system
  - centralised storing/archiving system (for ~ 70 TB data or ~ 65.000 datasets)
  - automated, centralised, pre-processing system with parallel processing capabilities to handle data orders of up to 2000 scenes in parallel (cloud-/cloud-shadow masking, TOA correction, geometric validation, geometric correction (to a Sentinel-2 Level 1C geometric reference mosaic), topographic normalisation and indices calculation,)
  - dissemination system to provide pre-processed data to all partners
- production of the forest (~5.850.000 km²) and part of the grassland layers (~1.930.500 km²)
  - mainly automated centralised and database driven thematic processing system (multi-temporal satellite data)
  - parallel processing capabilities (multiple production units per week)
  - high performance computing power for image segmentation

2.6.1.1.2 Selected Service Infrastructure and Architecture Setup for HRL 2015 production

For the HRL 2015 production, GAF selected a hybrid setup using:

- capabilities of its own private cloud for parallel processing, storing, archiving together with the
- public cloud (AWS) especially for storage and data dissemination.

A process to the data approach, using e.g. one public cloud in a more centralised way was not appropriate, as required EO data were only available on different platforms/systems (e.g. ESA Data Warehouse, Sentinel-2 Copernicus Hub, Amazon, Google and USGS). An overview about the selected approach is given in the following Figure 7.

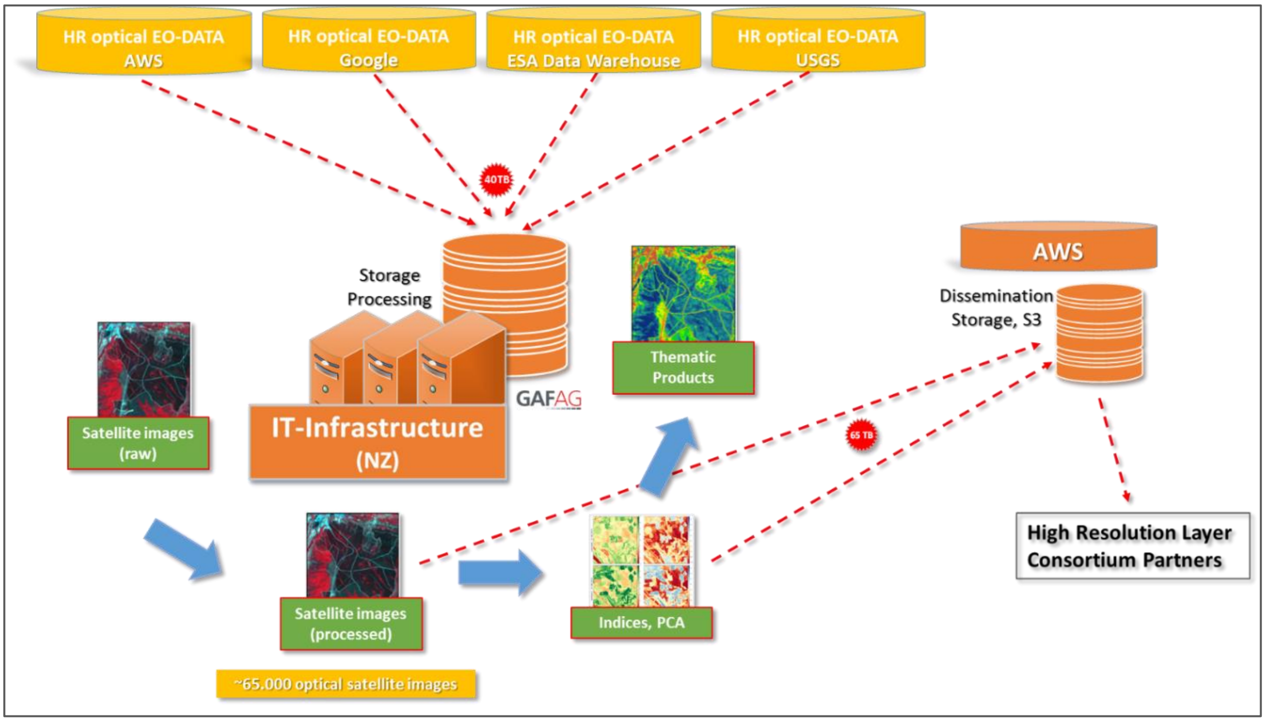


Figure 7: Overview of Service Infrastructure and Architecture Setup of GAF for the production of the HR Layers

A private cloud infrastructure, already available at GAF premise, was selected as central processing hub for GAFs activities with respect to pre-processing of optical EO data and thematic processing of the HR

Forest as well as Grassland (part, covered by GAF) layers. All EO optical data selected by GAF and consortium partners, were acquired from various sources (ESA Data Warehouse, ESA Copernicus Hub, AWS, Google and USGS) through automated download procedures and stored in a central data hub in the private cloud (80 TB on StrongBox reserved for the project, accessibility fast and medium). After metadata extraction and storage into a central database, an orchestrated (GAF intern development in C++) automated workflow to pre-process all required EO data was implemented using a scalable (up to 30 virtual processing nodes, each 16 to 32 GB RAM and 4 CPUs) processing environment based on internal GAF tools implemented in C++. Based on the set-up described before, around ~65.000 optical satellite scenes were downloaded and pre-processed during the project lifetime. Most of the pre-processed data (some data was only used for the production of the HR Forest Layer) was uploaded afterwards to Amazon Web Services (AWS), where a centralised data hub was established. This hub on AWS served as big data dissemination hub, accessible for relevant Consortium partners with regular status updates and automated download capabilities on partner request. Most of thematic processing of the HR Forest Layers, which followed the pre-processing, was also implemented in Python using a centralised approach on the private cloud and based on up to 5 Linux server with 32 to 64 GB RAM each for automated processes as well as 9 virtual desktops for semi-automatic and/or manual processes. Some post-processing procedures were done on local workstations/desktop systems.

2.6.1.1.3 Lessons learnt and outlook

The development of a more centralised approach (than described before), where consequently processes (or Software environment) are moved towards the data using the same infrastructure, was not implemented as almost all of the required EO data was only available on different cloud infrastructures or different data hubs. Using a more centralised approach based on GAFs private cloud allowed scalability with respect to processing and storage capability. Nevertheless, especially data transfer (download/upload) and data storage were identified as potential bottlenecks at the beginning of the project and therefore upgraded during project lifetime. All activities described above were implemented using mainly GAFs private cloud environment (with the bottlenecks identified before) and AWS for data dissemination. The whole setup could also be transferred completely into a public cloud environment (e.g. potential cost reduction, if any). Implementation based only on traditional infrastructure e.g. single desktop systems/work stations would not be recommended anymore.

2.6.1.2 High Resolution Layers 2018

For the High Resolution Layers 2018 GAF is leading two consortia entrusted with the production of two Lots being:

- Lot 2: Forest products (together with SIRS): Tree Cover Density & Dominant Leaf Type and changes, as well as Forest Type products
- Lot 3 (together with GeoVille and e-Geos): Grassland and Grassland change

This comprises the generation of several status and change products, which are either newly introduced or characterised by refined product specifications when compared to the reference year 2015. New challenges arise with the refined product specifications (status layers at 10m, newly introduced change products, additional key-intermediate products, a slightly extended AOI of approx. 6 million km²) but in particular due to the availability of dense time series of Sentinel-1 and Sentinel-2 data. The following section provides an overview of the Infrastructure and Architecture, which was selected to address those challenges.

2.6.1.2.1 Requirements on the selected Service Infrastructure and Architecture

The main computational load for the HRL 2018 production can be grouped into two main categories being pre-processing of satellite imagery to analysis-ready data (ARD) and higher-level analytics for the production of status layers and change products:

- Processing of ARD is focused mainly on Sentinel-1 and Sentinel-2 data (Landsat-8 as a backup solution) imposes the following requirements:
  - Central storage of approximately 800 TB for the reference years 2015 and 2018
  - Flexible storage policies to quickly archive readily processed data
  - Automatic pre-processing of optical data including atmospheric correction, topographic normalization and the generation of cloud masks
  - Automatic pre-processing of SAR data including the generation of orthorectified GRD amplitudes and interferometric coherences
  - Central data handling system to allow easy access to all partners
- The production of status and change layers furthermore implies the following needs:
  - Central storage to host auxiliary in situ and image datasets (e.g. VHR IMAGE 2012 -2018, existing land cover products)
  - Central database to host all relevant metadata, training and test data and information on the production status
  - Customizable development environments to allow all partners to roll-out their processing services
  - On-demand computing resources which can be deployed close to the data to scale up production to the Pan-European scale
  - Workspaces services to allow interactive processing and the use of desktop GIS software

2.6.1.2.2 Selected Service Infrastructure and Architecture Setup for HRL 2018 production

The HRL 2018 production relies for the first time fully on a public cloud infrastructure. The cloud infrastructure provides the central storage and access point for all relevant satellite and ancillary datasets as well as the processing environment to deploy and scale-up the analytical services for the production. This choice is in particular motivated by the increased data volume resulting from dense time-series of Sentinel-1 and Sentinel-2 data which implies the need to minimize data transfers and bring the analytical services close to the data. Considering relevant technological and economic aspects the consortia selected the Mundi Web Services, one of the new available Copernicus DIAS, as central processing framework for the 2018 production. An overview of the infrastructure architecture is provided in Figure 8.

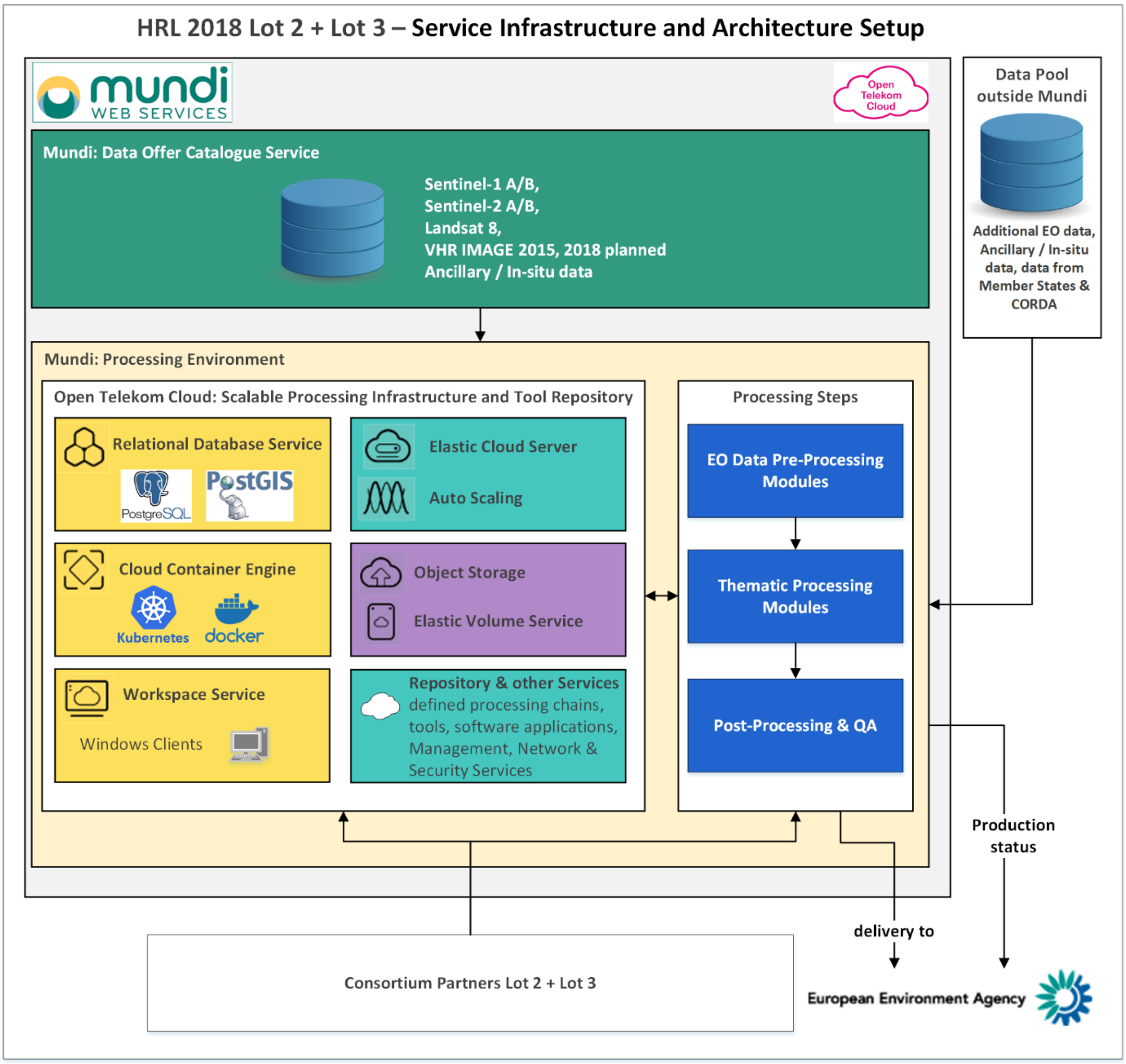


Figure 8: Overview of Service Infrastructure and Architecture Setup for the production of the HR Layers 2018 in Lot 2 and Lot 3.

All relevant EO data is made available on Mundi and can be directly accessed via the available Mundi Catalogue Data Service by all consortium partners. This includes not only pre-processed HR satellite data (approximately 800TB) but also auxiliary datasets such as VHR image collections for the reference years 2012, 2015 and 2018 as well as existing land cover and in situ datasets. A central *Relational Database Service (RDS)* contains a) all relevant metadata, b) production status, c) reference and training samples and d) other data management & production relevant information. Furthermore, Mundi features several technological solutions from the Open Telekom Cloud (OTC). The use of Mundi´s *Cloud Container Engine (CCE)* allows to support the different processing frameworks of the consortium partners, proper workflow orchestration and scalability, and modular workflows. The consortia will make use of OTC *Elastic Cloud Server (ECS)* environment, which allows to scale up the processing to the necessary numbers of virtual machines. The derived clusters are predominately based on machines with 8 VCPUs and 64GB RAM to accommodate in particular memory intensive computing. Intermediate processing steps and final products are stored in the *Object Storage* available in OTC. Datasets which require quick access during the processing are additionally stored in the *Elastic Volume Service (EVS)* for shorter time periods. Programming and workflow interfaces as well as classical desktop GIS applications are made available through dedicated *Workspace Service* which resemble a remote desktop application. Relevant software, scripts and container services are hosted in a central repository accessible to all partners.

2.6.2 EO data processing at SIRS

SIRS mainly uses two different processing platforms. A private cloud serves as a data warehouse and provides storage for vector and alphanumeric data in a Database (PostgreSQL) and raster data on a NAS system.

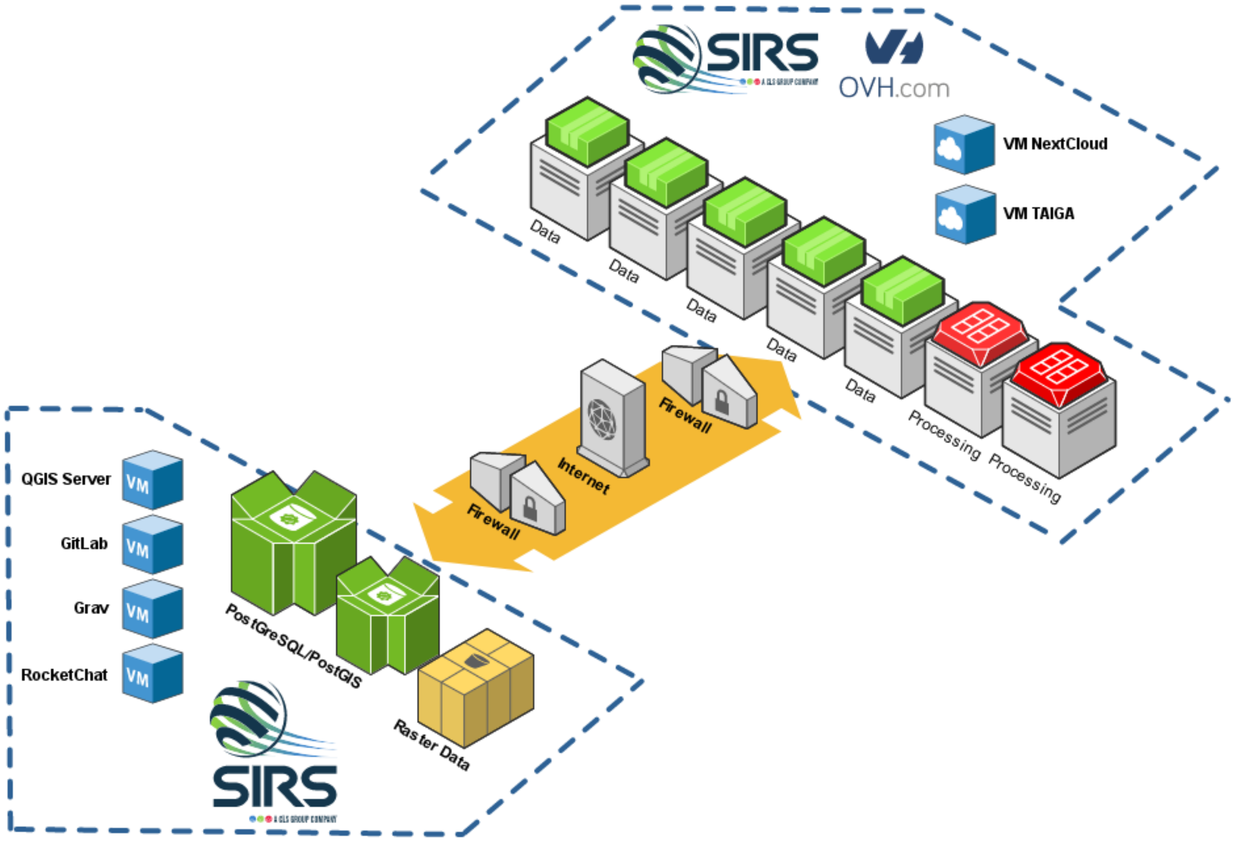


Figure 9: Overview of Service Infrastructure and Architecture Setup at SIRS

A second processing platform utilizing hardware and services of OVH (a French cloud computing company) offers extended storage and processing capabilities.

In general SIRS offers processing tools for raster and vector data. Both contain automated processing modules as well as applications for workflows with user interaction

Raster tools are based on several Open Source applications and libraries:

- QGIS processing linked to GRASS, GDAL and OTB
- SNAP (ESA software)
- GDAL with Linux Bash automation

The vector tools are based on popular Open Source applications and the corresponding interfaces:

- QGIS/GRASS algorithms
- PostGIS functions

2.6.3 EO data processing - Experiences from DLR

The Earth observation centre (EOC) at DLR has two large-scale processing environments, the Geofarm and Calvalus system.

The Geofarm in its current configuration consists of 2 Blade centres hosting 16 servers with a total of 672 Opteron cores, 3.3TB RAM, 488TB SATA and 50TB SAS storage interconnected with 10Gb/s Ethernet. In addition to that the setup also includes human resources for coordination, integration, configuration and virtualization of the environment. Within the next two years the system will encompass 4300 cores, 33TB

RAM, 1.9PB SATA, 100TB SAS and 8TB SSD. The standard service footprint allocated for users at the Geofarm includes 2 virtual machines with 4 cores, but can be expanded to the full capacity on-demand.

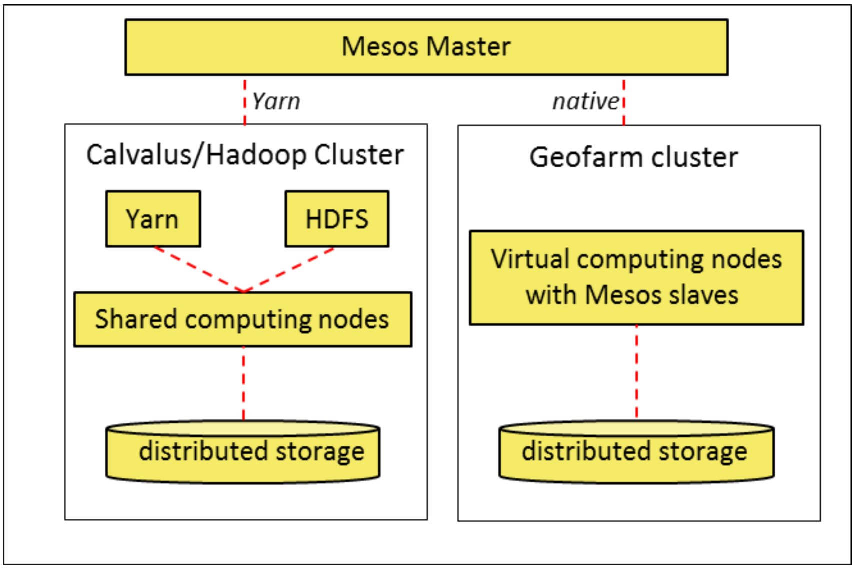


Figure 10: Overview of the two processing centres in the EOC at DLR

The DLR Calvalus Cluster was acquired in the frame of the OPUS project and delivered by Brockman Consult. It is a combined hard- and software stack based on Hadoop MapReduce. It consists of 50 compute nodes with 32 GB RAM each and 1 quad-core Intel Xeon 3.4 GHz CPU, and a distributed file system with 1PB. The Calvalus system extends and adapts the classical MapReduce paradigm with tools that are geared for EO data processing. The nodes are controlled by master running Hadoop Yarn. Data ingestion is managed by single point of entry called the feeder node that distributes data into the HDFS file system of the cluster. Calvalus integrates the SNAP and Sentinel toolboxes and allows developing own modules using Java, Python and Bash. A major capability of Calvalus is the integration of Docker containers. There are a set of tools for deploying and managing Docker container, which in turn are orchestrated via the central YARN master. This flexibility is extremely important in a research environment where people use a large set of programming languages and libraries and like to test prototypes and run immature code. Ordinary dependency management would quickly smudge and clog the single processing nodes with unnecessary and conflicting tools and libraries.

In order to trigger request for processing users can use any Web processing service (WPS). The WPS OGC standard allows submission of requests as XML and REST calls. The WPS requests are then serialized and ingested by a Mesos master, an Open Source resource management and scheduling system. The Mesos master knows the state and controls all available computing resources; in case of DLR the Geofarm and Calvalus clusters. The master checks availability of suitable processing resources (Mesos slaves) and triggers the calculations. In case of the Geofarm cluster each virtual machine is managed as one Mesos slave, whereas the Calvalus cluster is treated as one giant resource that is scheduled via the Mesos-YARN interface. Once the job is finished and the results are transferred to the distribution nodes, the master is informed by the corresponding slave that processing is finished and resources are ready for new tasks. This system ensures flexible and transparent management of a heterogeneous processing infrastructure. It also allows the flexible integration of additional processing nodes.

2.6.4 EO data processing at Joanneum

The IT infrastructure at Joanneum comprises a NAS system with a capacity of 50TB and a cluster of five Linux nodes for processing. In addition to that there is further disk space on a second workstation and the possibility to integrate local, idle desktop PCs for processing.

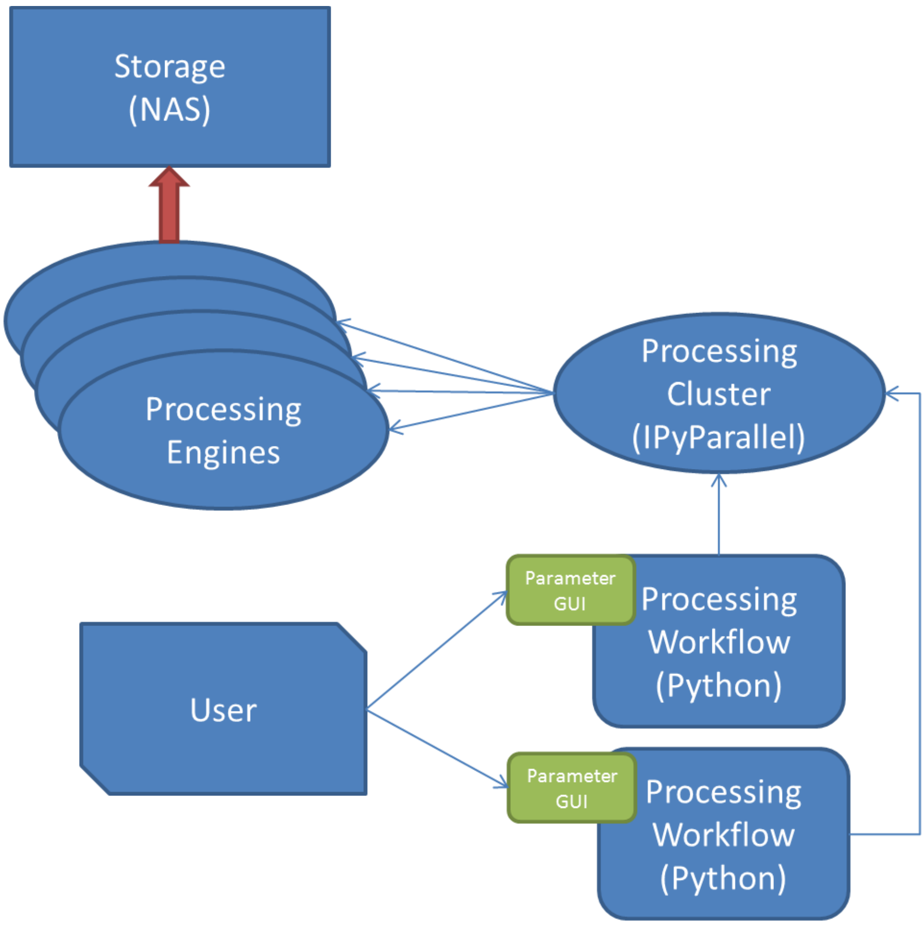


Figure 11: Overview of a processing work flow at Joanneum

Software development is tightly linked to two commercial processing suites, one for optical and one for radar data. The processing suites offer a wide range of algorithms and tools and can be combined in workflows, which are executed in parallel on the processing cluster.

3 Assessment of processing platforms regarding future developments in the EO context

The approach of the assessment of processing platforms regarding developments in the EO context is as follows: in a first step the platforms will be analysed (section 3.1); in a second step the different DIAS providers will be compared to each other (section 3.2).

3.1 Analysis of processing platforms

The WP 23 description called for an analysis of available and upcoming IT infrastructures in the frame of the Copernicus program, especially in regard to:

- network/data transfer capabilities
- storage architecture
- processing capacities, scalability and interfaces

In order to do that this document outlines a guide for a system evaluation that will be followed whenever a target platform is evaluated. The evaluation results will be matched with specifications of processing tools created to produce the data sets specified in ECoLaSS tasks 3 and 4. The purpose of the second analysis is supposed to influence the design of the processing tools in order to effectively run on the target platforms. The result of system evaluations and analysis is presented in chapter 3.2.

3.1.1 Initial estimates

In order to obtain some ideas regarding data volumes during processing, some simple estimates might shed some light on basic processing constraints.

Data transfer is usually expressed in Mbits/s, meaning that a 1 Gigabit/s connection is able to transfer 125MB/s. Tests using Linux command line tools like the ones below

network only, 1Gbit line

```
dd bs=1024 count=10000000 if=/dev/zero | ssh $targetIp dd of=/dev/null
→ 10240000000 bytes (10 GB, 9,5 GiB) copied, 98,8508 s, 104 MB/s
```

network and writing, 1Gbit line

```
dd bs=1024 count=10000000 if=/dev/zero | ssh $targetIp dd of=outfile
→ 10240000000 bytes (10 GB, 9.5 GiB) copied, 123.495 s, 82.9 MB/s
```

to measure the real transfer volume between two given nodes show that the theoretical limit is almost completely utilized for one or few connections. In our example case it is safe to assume that the theoretical transfer rate is equivalent the real one.

During 2017 an average S2 granule in Central Europe contained 80 scenes with approximately 50 GB of data. That means that each granule grows by 1 scene of 0.5GB every 3 days, clearly indicating that processing single scenes poses no problems. The complexity grows whenever processing deals with time series or larger regions, since a granule covers only 100x100km.

Now let's consider processing a time series of one year for a single S2 granule: Transferring a single S2 granule for a given year worth of 50GB over a 1Gbit line takes about 7 minutes, excluding reading and writing, which strongly depends on the corresponding storage medium. Atmospheric correction averages 55 minutes [1] per scene for reading, calculating and writing, or 3 days for the entire granule. During that

[1] This is the average processing time of approximately 600 scenes from several Central European granules using sen2cor (latest version 2.4.0) on a Hadoop cluster, where transfer is not relevant. Processing times range from a few minutes to almost 3 hours.

operation, the data volume grows by 40 percent to 70GB, which require another 9 minutes of transfer to the output destination. It seems that data transfer is a minor issue (15 minutes transfer vs. 3 days of processing), especially since this case only considered pre-processing and neglected all the subsequent processing steps necessary to create any product.

However, this deliberation has a major flaw: It assumes that band-width and storage are uncontested whereas processing is. That changes quickly if multiple parallel processes simultaneously try to access storage via a network connection with a fixed maximum value regarding data volume or number of file handles. This can virtually bring processing to a standstill. Therefore, it is of vital importance that processing systems are designed in way that parallel, concurrent processing does not lead to network starvation.

Any processing system suitable for EO calculations on a continental or global scale must be able to offer at least several dozens to hundreds computing nodes that can read and write concurrently from/to a storage area without overwhelming the network or associated storage devices. This can be achieved with many different hardware architectures, as was already outlined in chapter 2.

3.1.2 Evaluation of a processing system

The purpose of this chapter is to give a detailed understanding and definition of the system evaluation for a processing framework.

In order to have a common understanding the following terms will be used accordingly:

Table 6: Terminology of terms in the context of processing environments

Processor:	A Processor is an application startable in the course of a data processing Workflow. This could be an operational implementation of a scientific algorithm.
Request:	A Request is the unit of processing of data. It is a set of Steps which form a complete workflow which creates output products from input Products.
Step:	A Step is the smallest unit of a processing request. It can be e.g. the startup of a Processor as well as the allocation of cache space or the transfer of a product between systems.
Workflow:	A Workflow describes the sequence of events needed to fulfil a Request. It can be depicted in the form of a tree diagram, which includes Steps, decision-Steps and possible interconnections between these.
Resource:	A Resource is an allocateable object needed to fulfil the requirements of a Request.
Schedule:	A Schedule is a plan that includes a sequence of Steps and allocated Resources for these Steps.
Processing Power:	Processing Power is the number of Threads supported by a Resource/required by a Step.
Operator:	An Operator is a user of the processing system, which handles off-nominal situations as well as unavailability of Resources etc. but also has control of the nominal task fulfilment.
Ingestion:	Ingestion is the process of registering and archiving a Product.
Administrator:	An administrator is an actor responsible for the hardware and software platform.
Integrator:	An Integrator is a user of the processing system framework implementing Workflows.
Instance:	An Instance is a concrete incarnation of an abstract definition.
Module/Modular:	A Module is a separately runnable element of the System. Modular design facilitates exchange of module-implementations by defining appropriate interfaces.
Product:	A Product is the result of a Step.

It is assumed that a processing centre consists of some of the hard and software resources shown in the image below:

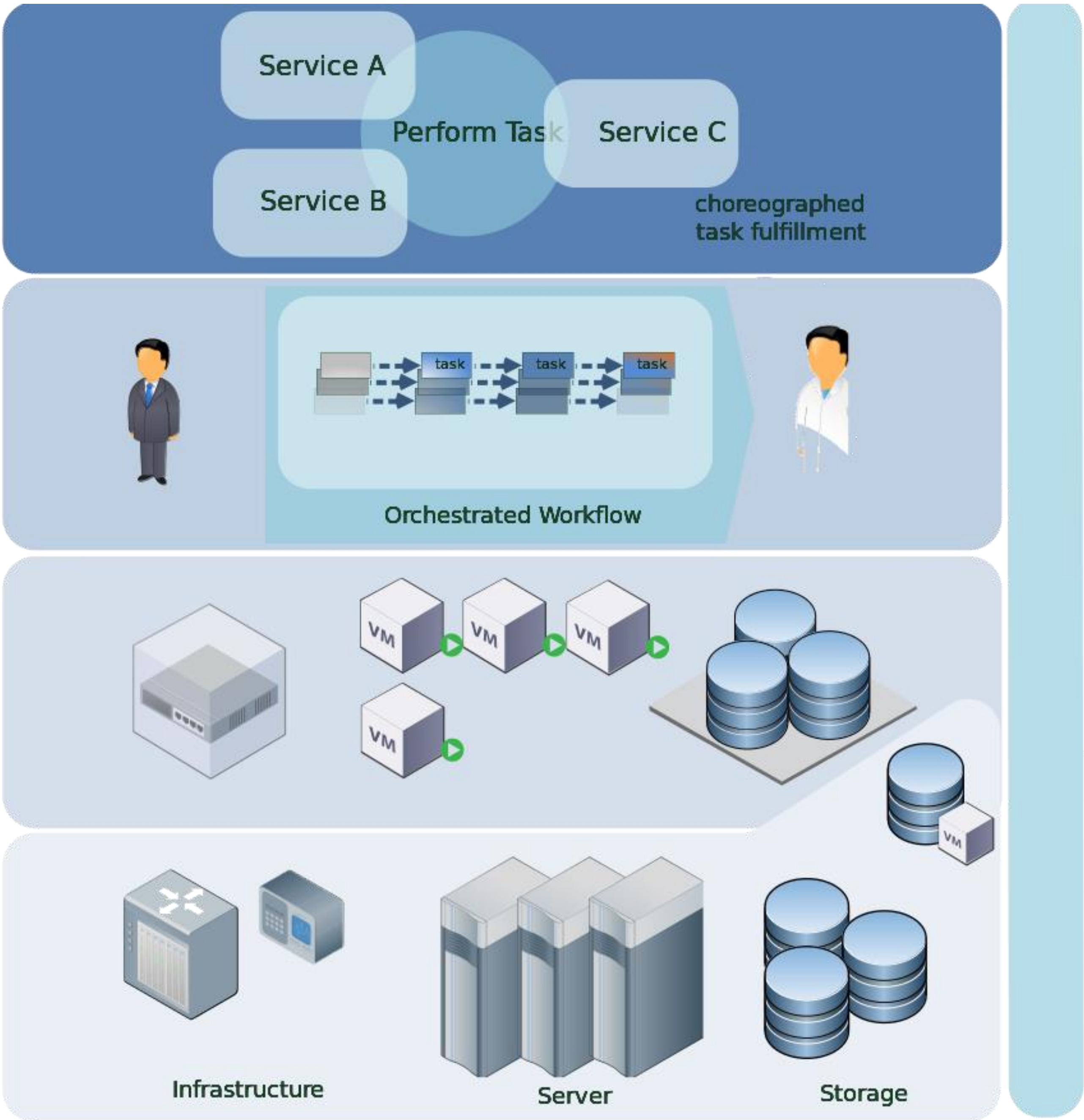


Figure 12: Dynamic approach for hardware and software resource allocation

Today most processing environments are based on a virtualized infrastructure. This facilitates a disconnection of hardware acquisition and hardware allocation for any project. The concept of separated resources can be transferred to the application layer as well. In Figure 12 the introduction of an abstraction layer between hardware and application – a dynamic infrastructure – is shown. With respect to the application layer an introduction of a dynamic layer separates project specific workflows from non project specific resource allocation tasks.

Processing management in this context has an essential role. It needs to provide a degree of freedom to facilitate inter-project resource usage as well as the means to easily create new processing workflow implementations.

Typical use cases of a processing environment are:

a) Ingestion of single products
b) Full automatic systematic data driven processing
c) Request driven processing
d) Iterative workflow handling
e) Mass-data processing
f) Validating parallel processing
g) Ad-hoc processing
h) Processing with operator interaction
i) Near real time processing

In our case we do not need to be concerned with point a) since we assume that all the required data all already present and accessible.

Typical, closely related use cases are b), c) and d), which describe processing tasks beginning with the availability of data to a defined end product (b), a specific request triggered to obtain a defined end product (c) and the repeated processing of a data set until a certain degree of maturity is achieved (d).

Topic e), request driven processing of a large number of products assumes that data stored in an archive and need to be transferred in smaller parts to the processing cache. After processing, the output could be transferred to the archiving system or to other places.

A problem is the possible overload of the archiving system (read/write), network overload and the cache- and resource- management of the processing system. The cache and the processing resources are normally not powerful enough to handle all productions at the same time. These cases should be managed by a single mass processing request, which handles the whole processing request of the entire data set. In contrast to request driven processing, mass processing manages all elements of the workflow depending on the available resources.

Cases involving parallel requests to validate and compare a new version with an older one (f), ad-hoc requests of processors for test purposes with only limited configuration (g) and automatic processing until a step requires interaction of the operator (h) will not be considered in detail since these cases are only important for the management of the system.

Finally, in case i) - near real time processing (NRT) - the processing time window is the most important condition. The problem is the mix of NRT and non NRT tasks in processing facility, which shares the resources of the system. Here the processing facility must be able to prioritize the usage of the resource. In critical NRT situations the processing facility must even be able to "steal" the resources from low priority processes.

3.1.3 Checklist for evaluation

With these use cases in mind, the processing environments will be evaluated according to the following criteria falling into several groups:

Design Constraints

This assumes that the processing environment is a modular architecture including interfaces to internal and external systems.

- Are interfaces present
- Are interfaces open

- Are there interfaces for processor integration
- Are there interfaces that provide a complete set of information about a processing run
- Are there interfaces for workflow construction
- Are there interfaces for setting breakpoints in workflows
- Are there interfaces for local, distributed and remote storage
- Are there interfaces for version control
- Are there restrictions/quotas/limits regarding processors and resources

Resource Handling

A core functionality of a processing framework is the handling of resources like disc space, CPUs and memory.

- Is storage space handled as a resource
- Can a resource like storage be associated to workflow or single workflow steps
- Can memory be allocated beforehand
- Are products handled as a resource
- Are processors handled as a resource
- Can processors be transferred to data
- How are product resources assigned processor resources
- How are CPUs allocated to processes
- How is memory allocated to processes

Scheduling

To be able to handle more than one request a time a processing environment provides automatic scheduling.

- Can product ingestion be scheduled
- What type of use cases (see above) are scheduled
- Can workflows be scheduled
- Can jobs be scheduled in parallel
- Can workflow steps be parallelized
- How are queues handled
- How are tasks selected and prioritized

Processing constraints

- Which of functions are supported by the processor interface:
  start/stop/status/restart/suspend/resume
- Can a processor trigger another request
- Does the processor interface return status reports, log messages or return values
- Is there a mechanism for debugging
- How are failures treated by the system
- Are processing task logged and archived
- Are there quotas for memory, CPU, storage
- Are there user right management constraints

3.2 Comparison of DIAS Providers

The chapters above established a list of criteria for the evaluation of any processing system. Since the last issue of this report (AD05), each of the five DIAS centres is in operation and offers a specific set of data sets and services. The data sets were already discussed in chapter 2.4.2, whereas the services will be investigated in the following paragraphs utilizing tables for ease of comparison.

Design Constraints

This section shows the most important properties regarding interfaces and processing environments.

Table 7: Interfaces and protocols on DIAS centres

	CREODIAS	ONDA	Mundi	Sobloo	WekEO
Access interface	• EO Hub: WMS/WMTS - REST API • Filesystem Interface (NFS) • Object Data Access API: • SWIFT/S3 API	• EO Hub: WMS	• EO Hub: WMS – REST API • Object Data Access API: • SWIFT/S3 API • Linux Fuse Clients	• WMTS – REST API	• REST API
Search Interface	• SPARQL • Semantic search metadata similar to OpenSearch	• OpenSearch • Elastic Search • ENS	• OpenSearch API • CSW • OGC	• OpenSearch API • CSW • OGC	• OGC • HDA API • SPARQL
Internal data protocol	• OpenStack S3 bucket	• https	• OpenStack S3 bucket	missing info	• https
external data protocol	missing info	• https	• https	• https	• https

Table 8: Additional information on standards and capabilities on DIAS centres

	CREODIAS	ONDA	Mundi	Sobloo	WekEO
Third Party Applications	x	x	x	x	x
Certificates	ISO 9001, 27001	ISO 27001, CSA	ISO 20000, 27001, 27017, 27018 CSA Star Level 2 TISAX/ENX, Trusted Cloud (EU-GDPR) TÜV Trusted Cloud	ISO 27001, ISO 9001, ISO 20000, ISAE 3402 type II, MTCS level 2	ISO/IEC 27001-2013

All of the platforms offer a range of multiple search and access services that allow potential users to find and read data sets and finally retrieve any processing results from the cloud provider. Standards differ but at least cover some well-established standards. Users are free to use any of them for online and offline processing. The section “Processing Constraints” further below will discuss dedicated geo-data processing APIs.

Resource Handling

This category shows available options regarding resources. Available information covering data sets to be operated on are listed in
, information regarding allocation of CPU or RAM can be gleaned from **Error! Reference source not found.**

Table 9: Capabilities regarding storage on DIAS centres

	CREODIAS	ONDA	Mundi	Sobloo	WekEO
Custom data	By URL linking By upload in storage	By upload in storage	Users can upload additional data or ask Mundi for managed hosting. For attractive use cases additional data may be hosted free of charge.	By upload in storage	By upload in storage
Ordering from archive	x	missing info	Yes, automated.	Yes, automated	Managed by user (e.g. Landsat, Envisat)
Restricted data	missing info	On demand	Includes functionality and is compliant to manage restricted data	Yes, under development	Managed by user (upload + restriction)
Data repository	x	x	x	x	x
Privacy	Public cloud Private cloud Hybrid cloud	Public cloud Private cloud	Public cloud Private cloud Hybrid cloud	Public cloud Private cloud Hybrid cloud	Public cloud Private cloud

Table 10: Hardware resources on DIAS centres

	CREODIAS	ONDA	Mundi	Sobloo	WekEO
GPUs	x	On demand	missing info	On demand	missing info
Processors	3500 vCores (reading cloud) 1500 vCores (processing cloud)	For one pre-configured instance: 16 or 32VCores Can be customized	Intel, Unlimited	Intel, Unlimited by design (Public Cloud) with scalable capacity	missing info
Overall RAM	7 TBytes (reading cloud) 7 TBytes (processing cloud)	For one pre-configured instance: 120Gb or 240Gb Can be customized	Unlimited, up to 1T per server	Unlimited at the Cloud platform layer (Public Cloud) Size per VM depends on flavor	missing info
Overall storage	7.5 PBytes, 150 PBytes on tapes (reading cloud) 500 Tbytes (processing cloud)	Unlimited	Unlimited	Unlimited by design (Public Cloud) with scalable capacity on Block Storage and Object Storage	missing info

Stream speed	100 GBits/s	250Mbps or 500Mbps for one instance Can be customized	InfiniBand, 25G, 10G and 1G Ethernet options	1 Gbps throughput, up to InfiniBand with 25 Gbps	2 * 10 GBits/s
Network	SDN network switching	Serco Italia Spa in collaboration with OVH	Internet MPLS (L3) Private Link (L2) all based on Deutsche Telekom Network	Internet (multi-homing including Orange Telecom) Private Direct Connect (L3) based on Orange MPLS or Equinix Cloud Exchange offer	Deutsche Telekom Network

The capabilities of the DIAS centres are difficult to compare, but for small and medium scale projects (short time series, limited geographical extent) all of them potentially offer a suitable environment. Tests might reveal the true performance of the processing system in question.

Scheduling

This topic is difficult to ascertain. Since all centres offer VMs and containers (see next section) it is certainly possible to fall back on custom options. Little information is available regarding the nature of any options provided by the platform.

Processing constraints

Processing involves the analysis and manipulation of data with different algorithms. Any user can chose to work with commonly available standard tools (proprietary or open source) or his own implementation. Requirements of users are difficult to ascertain for platform providers, because of the heterogeneity of tasks and processing philosophy. The tables below reflect this conundrum, because the providers clearly opted to offer solutions that offer maximum flexibility.

Table 11: Capabilities related to virtualization and container deployment

	CREODIAS	ONDA	Mundi	Sobloo	WekEO
Processing platform	x (OpenStack)	x (OpenStack)	x (OpenStack)	x	x (Morpheus API)
Containers	• Docker Swarm • Kubernetes	• Docker	• Docker, Kubernetes as integrated services. Users can deploy their own container types and frameworks.	• Docker • Kubernetes • Monocular	• Docker • Kubernetes
VMs	• CentOS • Ubuntu • Debian • Scientific Linux • Red hat • SUSE • Microsoft Server	• CentOS • Ubuntu • Debian • ISO image mounting possible	• CentOS • Ubuntu • Debian • Scientific Linux Oracle Linux • Red Hat • SUSE • Microsoft Server Microsoft Windows • Users can also upload customized images	CentOS • Ubuntu • Debian • OpenSuse • Fedora • CoreOS • Suse Enterprise • RedHat Microsoft Windows	• CentOS • Ubuntu • On demand
GUI provisioning VM	x	-	x	-	-

Table 12: Additional processing services offered at DIAS centres

	CREODIAS	ONDA	Mundi	Sobloo	WekEO
Service offers	• Single server VMs • Bare Metal Dedicated Servers • Physical media Data Import/Export • Load Balancers (LBaaS) • IaaS, VPNaaS, FWaaS • Direct data Connections	• Single server VMs • IaaS, LBaaS, VPNaaS, FWaaS, PaaS, SaaS on demand • Dedicated Servers	• Single server VMs • Dedicated Hosts • Bare Metal Databases • Load Balancers • Workspaces • Network Security • PaaS e.g., Message Services • SaaS e.g., Data Analytics	• Single server VMs • Physical servers (Bare Metal) • Load Balancers (LBaaS) • IaaS, VPNaaS, FWaaS • Direct data connections	• Single server VMs • LBaaS • FWaaS • VPNaaS (limited scope) • IaaS • PaaS (to come)

Therefore, it is no surprise that all providers offer the two main virtualization techniques, namely VMs and (Docker) container. This allows users to replicate and encapsulate their processing infrastructure and move them to different processing environments. It provides anybody with the maximum level of flexibility. On the other hand, this freedom regarding processing techniques demands expert knowledge in the fields of software development and remote sensing data management. Low level users might be better served with other options listed in chapter 2.5.

4 Requirements

In section 4.1 initial requirements made in context to the submission of the first issue of this deliverable (AD05) can be found. Since this document is the last issue of the WP23 deliverable, final considerations are given in section 4.2.

4.1 Initial requirements

As mentioned above the main conclusions will be presented in deliverable WP23.1b where the evaluation scheme above will be matched against potential processing environments. The result of each evaluation will be used to formulate recommendations regarding the design of any processors developed in this project. But it is possible to come up with some preliminary recommendations.

The previous chapters show that processing infrastructures within the consortium are extremely heterogeneous. They reflect the historical development and current emphasis of each entity.

SIRS and DLR have access to large connected hardware clusters that allows them to store and find data across multiple nodes as well as process data on some or all of the nodes using dedicated process management systems. Joanneum is strongly focused on software development and maintains several smaller and isolated processing clusters, which are orchestrated using a series of scripts developed in house. They can afford this approach because they have a suite of software tools, that operates in a specific manner and thus need less flexibility, whereas software tools present at e.g. DLR are from multiple development teams. Thus, they are very heterogeneous and must be used in a different manner requiring a processing management tool with a large degree of flexibility.

In the course of the project the system design approach outline before will need to identify the key resources of each processing platform and evaluate them in respect to a general set of requirements derived from potential target platform like DIAS.

As mentioned above IT infrastructures of the consortium members are diverse, resulting from different needs and requirements. The same is valid for the software used for analysing data and managing infrastructure. But the ECoLaSS project could benefit from a few design considerations and maybe some activities related to collaboration in the field of prototype development and processing exercises. Here are some considerations:

HARDWARE INFRASTRUCTURE

The advent of virtualization techniques and cluster management tools allows users to avoid buying powerful, but expensive hardware geared for high level processing and obtain ordinary, run of the mill computers. New techniques allow combining single resources into larger processing entities. At the same time, one has to be aware that certain processing tasks might require the procurement of special hardware (e.g. machine learning favours GPU processing units). However, specialized systems might be replaced by utilizing services from third parties like commercial service providers and public computing centres.

It is even possible to utilize single, ordinary desktop computers connected via intranet. Projects like the Berkeley Open Infrastructure for Network Computing (BOINC) try to employ unused and idle computers for scientific processing. Therefore, a client on each computer registers the activity and notifies a master if it is ready to perform additional jobs. The master then assigns data and software to the node and thus distributes processing tasks. Distributed computing projects like that are similar to some of the systems described above. Unfortunately, they are quite rigid regarding the choice of software and data structures.

Since hardware setup at each consortium member is fixed no recommendations can be given for any collaboration activities.

SOFTWARE INFRASTRUCTURE

Software infrastructure is probably an area where differences across and maybe even within institutions are extremely large. The choice of programming languages and tools, the level of operationalization, skill and number of programmers and many more influence the functionality and capability of software on numerous levels.

It is therefore really difficult to give recommendations for collaboration. Some ideas might include:

- Definition of common product specifications
- Usage of a set of common libraries
- Sharing of code for common tasks (import, export, transformation, algorithms)
- Definition of common APIs
- Provision of web services for data (WMS, WFS, and other OGC services) and functions
- Exchange of virtual images or containers (see below)

PLATFORM INFRASTRUCTURE

In contrast to the subject covered above platform infrastructure holds big promises for exchange of software. Central to this is the last topic in the list mentioned above: Virtualization techniques enable stakeholders to exchange data, software and services without dependencies regarding applications, libraries and languages except for the virtualization system itself. Thus, they offer an isolation of the underlying hardware or operating system and permit encapsulation of entire applications in a virtual or container image.

Establishing processing platforms that can handle virtual images or containers allows platform providers to invite users to work with their preferred collection of tools without the necessity to change the setup of the host system. Images can even be replicated multiple times for parallel processing since the footprint usually is negligible. It is thus possible to distribute software to one (or multiple) processing centre, where it can efficiently analyse data sets present on the host platform. Testing is also simplified because new software can be incorporated in a container and subsequently deployed on the host system. Different versions of the software can be tested next to each other without interference or changes of the underlying system.

A less flexible way is the usage of platforms where the user has to adhere to the dependencies of the host system. This might be an API for services, a set of languages and libraries, or even a series of applications. In this case the user has to use the technique provided by the host provider, but is at least able to write his own software. During the last years this is becoming easier, because of the development of some de facto standard libraries (e.g. GDAL for I/O operations) and languages like Python or R that are used by many programmers in the field of data sciences.

A combination of containers and standards and conventions regarding their execution (e.g. API regarding "docker run" calls) provides maximum flexibility. In this case different parties can contribute data manipulation tools to platform service providers who can rely on the conventions to automatically deploy and run them on the host platform.

Exchanging images for certain processing tasks (e.g. pre-processing) is considered to be one of the least intrusive and therefore most promising forms of collaboration within the consortium.

OVERARCHING ORCHESTRATION

In simple cases orchestration can be achieved by the command line. But once processing becomes more complex users will quickly look for ways to automate monitoring resources, conditionally manage job requests (e.g. construct workflows) and check the status of completed processing tasks. An even more complex scenario involves coordination of one or more remote processing services. In order to be able to orchestrate a group of tasks it becomes imperative to define an API or at least provide conventions for calling these processing tasks.

It has become very popular to furnish single tasks with web services based on protocols like REST or WPS. Orchestration tools usually allow the construction of adapters that interact with established services and some popular Open source tools already offer many built-in connectors for the most common protocols.

It would be interesting to investigate if orchestration across services provided by some project members could speed up certain processing tasks, especially if one could render transfer of large data sets obsolete.

4.2 Final considerations

At first glance all DIAS centres offer similar services. A closer look reveals various differences regarding completeness of functionalities, services and data availability.

Longer time series stretching more than two years cannot be run on every platform due to the data eviction policy of some centres. Any user requiring any such data will have a limited choice of providers.

Different service interfaces have similar consequences. For example, a report by (Böttcher, 2018) tested the search capabilities on all DIAS centres and made the following observations.

- ONDA catalogue GUI temporal search: useless because it always starts 1970 and more than 500 klicks required for something useful; no search by insertion date; OLCI and SLSTR not easily visible in catalogue GUI
- MUNDI provides S1 data as zip files while S2 data is unpacked SAFE. S2 has to be copied recursively with s3cmd, cannot be easily downloaded with HTTP, unique ID from search cannot be used.
- CREODIAS has all data unpacked.
- SOBLOO does not seem to be ready yet. Several tests were not possible.
- WekEO search results list unique IDs only. There is no way yet to determine the file name for download from the search result. Wekeo search supports bounding box only. Wekeo search does not support insertion date.

Despite it is known that the ambition of the EC and ESA is clearly that users should be able to flexibly switch between one DIAS and another as desired case by case, the consequences of this analysis show that the portability of applications designed and implemented on a given centre seems not yet such that the applications could be easily moved to another DIAS without a certain effort and a bunch of modifications. This may change in the near future with the further maturing of the five DIASses and will closely be observed by ECoLaSS and the whole community.

If reusability is of importance, applications need to be generally designed in a modular way, so that data search, access and processing are decoupled or highly configurable. A user has to weigh options if he wants to be mobile across DIAS centres and how to achieve this. He has to design his processors in a way that some modules for a proprietary or specific function can be changed or swapped. He can also design generic functions that can be configured for some specificities. All of this will result in a sharp increase of complexity.
This may sound trivial, but it should be considered that even the path name of data on a given DIAS centre will differ from a local application. There are enough examples of processing applications that will need a certain fixed input pattern, which might not be present in a different environment like the DIAS centres.

5 Outlook

As mentioned throughout this report developments in the IT domain during the last 10 years have significantly shifted our perception of the way we use large data sets. Data volumes produced by companies like Google, Twitter, Amazon, Facebook and others are huge and not only constitute business core data, but also protocols and logs of people using these services. It should be noted that the tremendous (daily) volumes of EO data generated by space agencies and commercial providers are still somewhat minor compared to real-time data of the large IT corporations. These companies not only want to store but also evaluate and analyse real-time data. The resulting processing concepts and accompanying collaborative, Open Source implementations also benefit other fields in the IT community.

It is also important to realize that the expertise and infrastructure of IT corporations is centred on hard and software techniques related to large scale computing. This is usually not (yet) the case with classical EO data users comprising universities, research institutes, government agencies and small to medium sized companies. They typically developed their IT knowledge from bottom up to match the tasks at hand. For a long time, these users did not have a global perspective. But it is imperative to look at the latest developments and try to benefit from them.

In particular users will have to decide whether to process data locally or remotely using services in the clouds. In order to stay flexible and avoid unnecessary and costly data transfer traffic, processing techniques will favour solutions that move software to data. This opens the way for collaborations with partners and keeps stakeholders independent of hardware and service providers. Other considerations (security, confidentiality, technical requirements, etc.) might lead to in-house solutions, but new technical concepts already offer many building blocks to setup a maintainable, inexpensive and flexible processing environment. Preliminary experience of consortium members and chapter 3.1.1 show that the EO community does not necessarily need a high performance infrastructure but something known as “large scale analytical platform” that enables users to process multiple data sets in parallel without running into IO squeezes.

DIAS has the potential to provide users with data and hardware resources that are superior to many local processing platforms. Users can now evaluate these offers and compare them to the ones offered by global IT companies like Google and Amazon, which had entered the market a few years earlier and offered compelling data handling and processing approaches that have quickly become accepted in the geo data community. In order to attract new customers, some of these approaches or soon-to-be standards might be well worthwhile inclusions in the portfolio of some DIAS providers.

Besides that, it can be stated, that for the time being, the DIASes are still striving towards full operational data availability and maturity. The DIAS providers are evidently working continuously on improving their offers and fixing known issues, so that it is impossible to capture in one report “the status” of the DIASses. This puts current and future users in a slightly tricky position as they will hardly have the time, resources and patience to continuously monitor and test the status and ongoing development of the DIASses. It is therefore necessary that the DIASses will reach operational maturity in short time, since otherwise there is a serious threat that other, already more established platforms like AWS and Google, will in the end profit from the situation and in the end bypass or, in worst case, marginalise the European-led DIASses in the long run.

Furthermore it has been observed that, despite the EC’s and ESA’s ambition to enable users to flexibly switch between one DIAS and another as desired, the consequences of this analysis show that the portability of applications designed and implemented on a given centre seems not yet such that the applications could be easily moved to another DIAS without a certain effort and modifications. This may change in the near future with the further maturing of the five DIASses and will closely be observed by ECoLaSS and the whole community.

6 References

Amazon Web Services, Inc. (2017). Retrieved 12 18, 2017, from Earth on AWS: https://aws.amazon.com/earth/

Apache. (2013, 04 08). Retrieved 12 18, 2017, from HDFS Architecture Guide: https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html

Apache. (2017). Retrieved 12 18, 2017, from Apache Mesos: http://mesos.apache.org/

Apache Inc. (n.d.). Retrieved 12 18, 2017, from Project - Airflow Documentation: https://airflow.apache.org/

Böttcher, M. (2018). *DIAS for Data Processing, internal report in the TEP Urban iniciative.* Internal Report: Brockman Consult.

Canonical Ltd. (n.d.). *LXC Introduction*. Retrieved 12 18, 2017, from Linux Containers: https://linuxcontainers.org/lxc/introduction/

CNES - Centre National d'Études Spatiales. (2017). Consulté le 12 18, 2017, sur PEPS - Plateforme d'Exploitation des Produits Sentinel (CNES): https://peps.cnes.fr/rocket/#/home

Docker Inc. (2017). Retrieved 12 18, 2017, from Docker - Build, Ship, and Run Any App, Anywhere: https://www.docker.com/

EODC Earth Observation Data Centre for Water Resources Monitoring GmbH. (2017). Retrieved 12 18, 2017, from We bring the Software to the Data and Information Products to the Users: https://www.eodc.eu

ESA. (2017). *Existing/Planned collaborative ground segment.* Retrieved 12 18, 2017, from Sentinel Online: https://sentinel.esa.int/web/sentinel/missions/collaborative/existing-planned

ESA. (2017, a). Retrieved 12 18, 2017, from TEP - Thematic Exploitation Platform: https://tep.eo.esa.int/

ESA. (2017, b). Retrieved 12 18, 2017, from Research and User Support: https://rus-copernicus.eu/portal/

European Commision. (2017, 12 14). *Copernicus DIAS contracts signed*. Retrieved from Copernicus: http://copernicus.eu/news/copernicus-dias-contracts-signed

European Commission. (2017). *The upcoming Copernicus Data and Information Access Services (DIAS)*. Retrieved 12 18, 2017, from Copernicus: http://copernicus.eu/news/upcoming-copernicus-data-and-information-access-services-dias

Ghemawat, S., Gobioff, H., & S.T., L. (2003). The Google File System. *Proceedings of the nineteenth ACM Symposium on Operating Systems Principles - SOSP '03*.

Google. (2017). Retrieved 12 18, 2017, from Google Earth Engine: https://earthengine.google.com/

IBM. (n.d.). *IBM Spectrum Scale - Overview*. Retrieved 12 18, 2017, from https://www.ibm.com/us-en/marketplace/scale-out-file-and-object-storage

Infrastruktur, BVDI - Bundesministerium für Verkehr und digitale Infrastruktur. (2017). Retrieved 12 18, 2017, from CODE-DE: https://code-de.org/

Jenkins Project. (n.d.). Retrieved 12 18, 2017, from Jenkins: https://jenkins-ci.org/

Linux Foundation. (2017). Retrieved 12 18, 2017, from Kubernetes | Production-Grade Container Orchestration: https://kubernetes.io/

NOA - National Observatory of Athens. (2016). Retrieved 12 18, 2017, from Hellenic National Sentinel Data Mirror Site - Overview: https://sentinels.space.noa.gr/

Oracle. (2012). *What is ZFS?* Retrieved 12 18, 2017, from https://docs.oracle.com/cd/E23823_01/html/819-5461/zfsover-2.html#gayou

Red Hat, Inc. (2017). Retrieved 12 18, 2017, from Ansible Is Simple IT Automation: https://www.ansible.com/

SAP. (2017, 12 18). *Intro to SAP HANA Geospatial - Spatial Reference Systems | SAP*. Retrieved 12 18, 2017, from https://www.sap.com/developer/tutorials/hana-spatial-intro6-srs.html

SERCO Spa. (2016). *Sentinels Data Access Annual Report 2016.* ESA. Retrieved 12 18, 2017, from https://scihub.copernicus.eu/twiki/pub/SciHubWebPortal/AnnualReport2016/COPE-SERCO-RP-17-0071_-_Sentinel_Data_Access_Annual_Report_2016_v1.2.pdf

Thulasiraman, K., & Swamy, M. (1992). 5.7 Acyclic Directed Graphs. In *Graphs: Theory and Algorithms* (p. 118). John Wiley and Sons, Inc.

VITO - Vlaamse Instelling voor Technologisch Onderzoek. (2017). Retrieved 12 18, 2017, from Terrascope | EO needs in Belgium: https://www.terrascope.be/

Annexe 1 – ECoLaSS experiences with the DIAS

Document Release Sheet – Annex 1

	Name, Function	Date	Signature
Author(s):	André Stumpf (GAF), Christophe Sannier (SIRS), Sophie Villerot (SIRS) Heinz Gallaun (JR) Benjamin Leutner (DLR)	20.09.2019	
Review:	Christophe Sannier (SIRS)	30.09.2019	
Approval:	Eva Sevillano Marco (GAF)	20.12.2019	
Acceptance:	Massimo Ciscato (REA)		
Distribution:	ECoLaSS project consortium; Project Officer and European Commission Services		

Dissemination Level

Dissemination Level		
PU	Public	X
CO	Confidential: only for members of the consortium (including the Commission Services)	

Document Status Sheet

Issue/Rev	Date	Page(s)	Description / Changes
2.0	20.12.2019	9	Second issue of Service Infrastructure/Architecture Requirements Report (Annex 1)

A. Introduction

In addition to the technical and financial descriptions of cloud processing platforms in general and the DIASes in particular, which was delivered with the "Second issue of the Service Infrastructure and Architecture Requirements", this appendix summarizes the concrete experiences of the ECoLaSS consortium partner's with implementing prototype processing chains on DIAS platforms. Three DIAS services were tested in phase two of the ECoLaSS project, namely SOBLOO, MUNDI and CREODIAS, respectively summarized in sections 2, 3 and 4 of this Annex.

B. SOBLOO

SIRS has built partnership in order to test SoBloo provided by the Orange Flexible Engine Technology to offer private and secured cloud. Contract start from 24th of September 2019 to 31th of December 2019. A wide range of computing services was available: from bare metal to autoscaling cloud computing instance and GPU instances for deep learning efficiency. A SoBloo Linux desktop was also subscribe (SoBloo reference: cc3.18xlarge.4) and offer 76 vCPU, 304GB RAM and 2 Tb disk space. This virtual machine allowed production team to check easily results and launch some process which required some manual phases. It comes with various software preinstalled such as ESA Snap, BRAT, OTB, Sen2Cor, QGIS and GDAL and some development platform and framework like Eclipse Neon, C++, Java, Python, Sciab and R. If some libraries are missing, it's possible to install it as root user based on an Ubuntu Xenial version. In addition, the offer adds a large storage object of 20 TB as a S3 bucket to store external input, intermediate and results. SoBloo also propose to test Muscate, a toolbox for multitemporal data, but it hasn't been tested as the ECOLASS budget will be overspending. Complete SoBloo pricelist is detailed on https://sobloo.eu/sites/default/files/Cloud%20Services%20-%20sobloo%28Rev.%200226%29.pdf
Sobloo works with private tenant and common tenant where spatial data as Sentinel2 L2A and free processing chain are available as following on Figure 13.

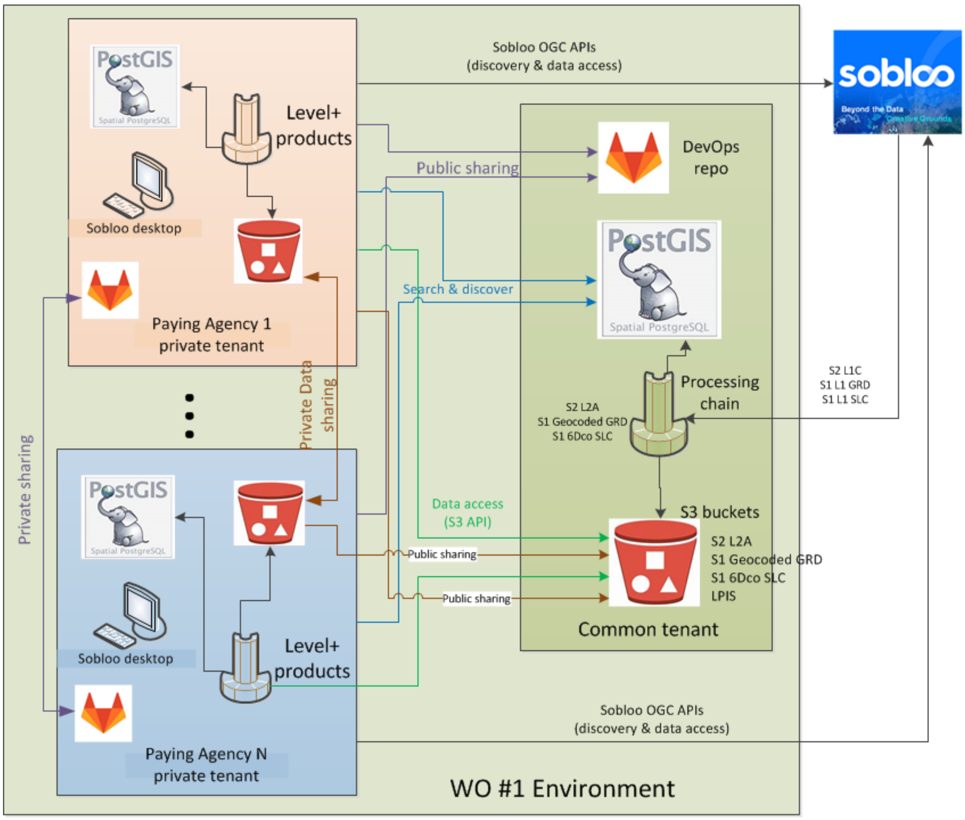


Figure 13: SoBloo environment architecture.

Virtual Machine are managed by OpenStack in backend and guaranty high availability and quick deployment for virtual instances. Kubernetes is also present in backend for container engine but it's not possible use it directly with kubectl in command line, user must use custom API provided by SoBloo.
One of the strengths of this DIAS is the DataCatalog GUI available at https://sobloo.eu/wui/ allowing users to explore data by filtering level, cloud mask, timeseries, and extra metadata. Without registration, production team can evaluate if data is available and relevant for use before subscribing some paid services. See screenshot in Figure 14.

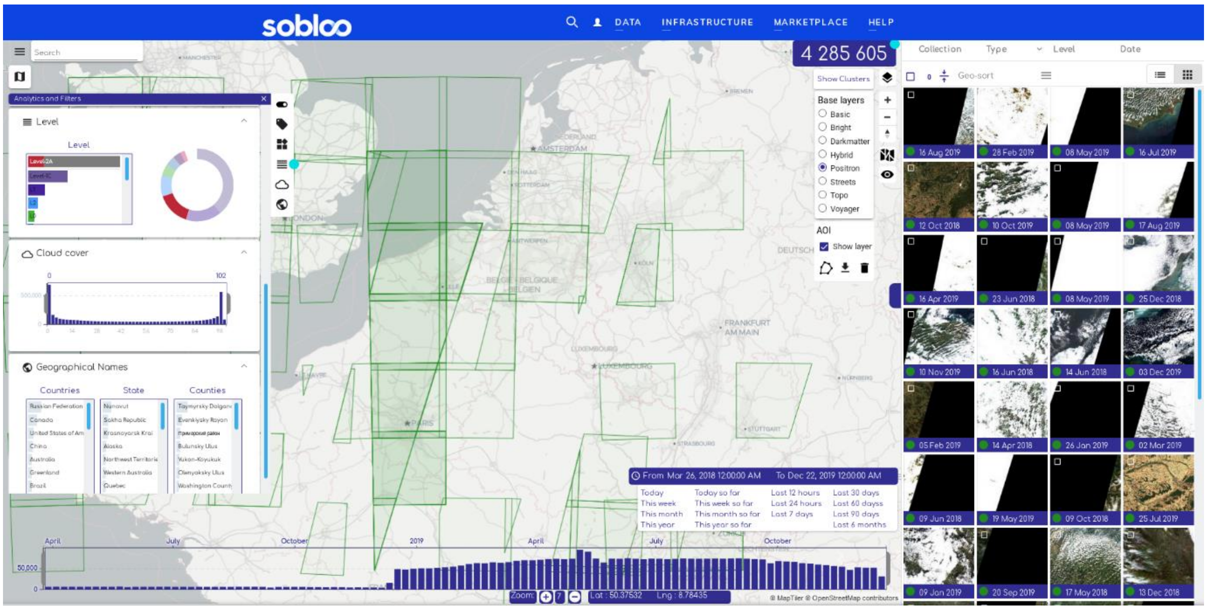


Figure 14: SoBloo Data Catalogue Web GUI.

As main of remote sensing toolbox are already "dockerized", a private Docker registry has been installed in order to deploy easily container engine processing with our custom docker images. We also tested some custom Python scripts from the virtual machine.
Our experiment on DIAS was about pre-processing, the computation of time series metrics to produce the phenological layer as well as some test on different segmentation algorithms.
SoBloo is not only focused on computing services. A large scale of other features is proposing on the technical console (see Figure 15). It looks similar to smaller AWS interface with data analysis, distributed cache database for web application, security and management services in order to optimize costs.

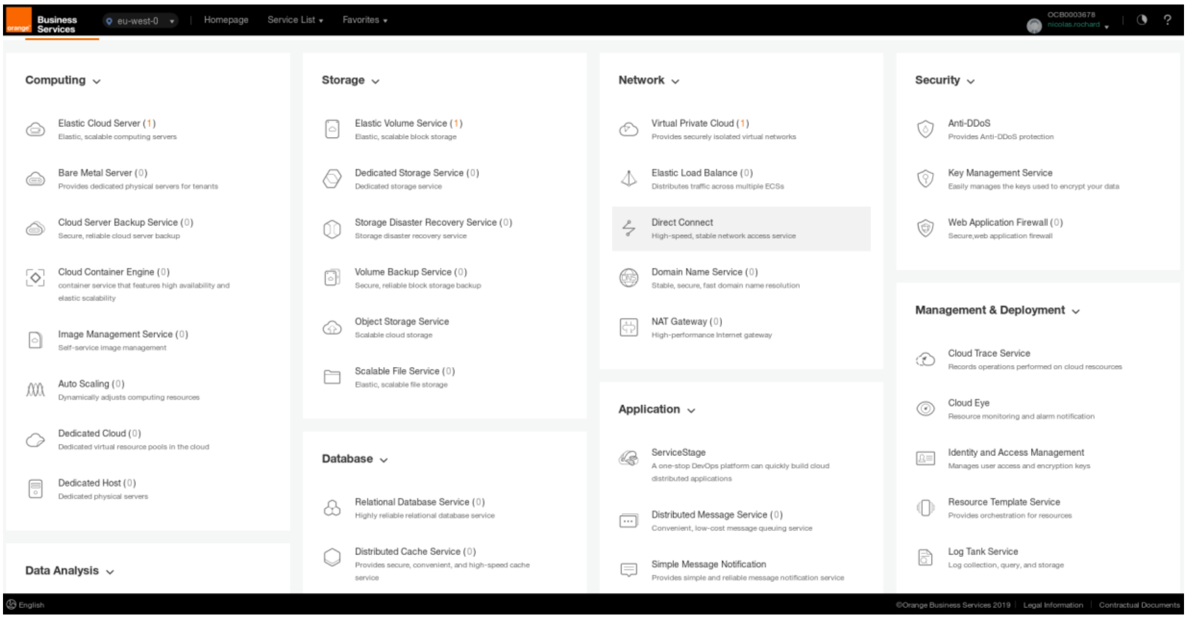


Figure 15: SoBloo Technical Console.

Lessons learned:

- Sentinel 2 data is corrected by Sen2Cor. If user preferred MAJA, S2 data already on DIAS provided will need to be reprocessed.
- Dockerize process is not enough, user / developer needs to adapt process to S3 bucket API. Some analysis can take longer due to bandwidth limit.
- Some of the software applications provided in SoBloo Desktop are outdated (QGIS 2.18, GDAL or OTB 6.0 for example).
- Raster data has to be provided by an WM(T)S server to be add and visualize on SoBloo DataCatalog WebView. However, the integration of private S3 bucket should be possible as common S3 buckets are displayable.
- Documentation writing in progress, still incomplete. (but support is very reactive, wiki is growing fast).
- Only WMS and WMTS API, no WCS service to adjust image stretching.
- Not specific to DIAS but all cloud provider: a significative amount of time is necessary to understand how to optimize cost and avoid useless spending like forgetting to stop an instance, etc. SoBloo provide some template to merge in user code for shutdowning an instance when the process is over or failed.

C. MUNDI

Several of the tools and methods which have been developed and tested within the framework of ECoLaSS have been re-implemented for further optimization and ported to Mundi for operational land cover classification at continental scale. This comprises for example land cover mapping within the framework of HRL 2018 (Forest and Grassland) which is currently in operational production and hosted on a scalable OpenShift cluster on Mundi. The application comprises several services for retrieving EO data and feature calculation (calculate-feature-srvc), training and testing of machine learning models (build-model-srvc), the roll out of such models over large areas (apply-model-srvc, batch-classification-srvc), mosaicking of classified patches (mosaicking-service), as well as the caching of readily retrieved data in an intermediate storage (Persistent Volume Claim, pvc-manager). The cluster comprises several flavours of virtual machines which are listed in the Table 13. The number of instances of each VM flavour can be increased / decreased according to the operational needs of the project (e.g. size of the production team).

Table 13: Overview of the different VM flavours that serve as building blocks for the OpenShift cluster on Mundi.

VM Flavour	CPUs	Memory (GB)	Purpose
s2.large.1	2	2	Internet Gateway for the cluster
c3.2xlarge.4	8	32	OpenShift Master
s2.large.1	2	2	Management of Persistent Volume Claim
s2.4xlarge.8	16	128	Apply model nodes for roll out over large areas
s2.large.4	2	8	Database host
s2.2xlarge.4	8	32	Data retrieval and feature calculation service

OpenShift provides an abstraction layer through which the services can access specific parts of the infrastructure. Thereby each service can access a limited number of so called Pods which are essentially containers for the application with a fixed IP address and defined amounts of allocated resources. The graphical frontend through which the OpenShift cluster and application logs can be accessed on Mundi is depicted in Figure 16

Analysis-ready time series (2016-2018) of Sentinel-2 and Sentinel-1 data are retrieved via the API of the Sentinel-Hub at native resolution and resampled to 10m by the application on Mundi. In terms of storage the application relies on a fixed Persistent Volume Storage (currently 2TB) for caching readily retrieved EO data and unlimited S3 compatible Object Storage Service (OBS) where computed features, models and classification results are stored. The caching mechanism proved as an important feature of the application and allows the synergistic reuse of readily retrieved EO data for different classification tasks. The fact that readily retrieved EO-data can be reused for different thematic classifications and iterative improvements of the same machine learning models significantly reduces costs for computation, networking and third-party services such as the Sentinel-Hub.
For further details on data and services available on Mundi we refer to the webpage under https://mundiwebservices.com. For an overview of the price model of the VMs and storage the price calculator of the underlying Open Telekom Cloud is a useful resource: https://open-telekom-cloud.com/en/prices/price-calculator. The pricing model for the Sentinel-Hub is described here https://docs.sentinel-hub.com/api/latest/#/API/processing_unit and here https://www.sentinel-hub.com/pricing-plans.

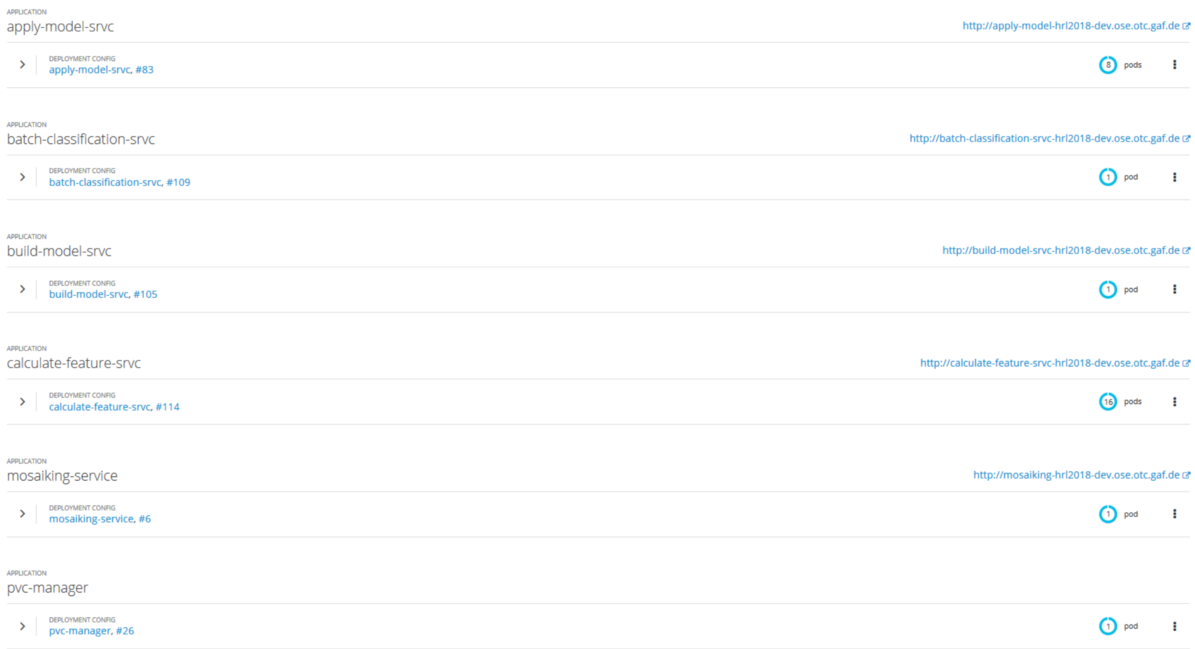


Figure 16: Screenshot of the graphical interface to the operational OpenShift cluster on Mundi.

The production team operates the application through a user friendly QGIS plugin which was developed in-house at GAF in the framework of several recent projects and is continuously improved to for performance, user-friendliness and new thematic applications. The QGIS runs on local machines which avoids additional costs and latency of remote desktop and removes the need to transfer locally available ancillary datasets to the cloud storage. An infographic of the general workflow as well as a screenshot of the CloudMapper (version 4.1) are provided in Figure 17. Each step in the workflow corresponds to one tab in the GUI including Workflow Management, Sample Selection, Feature Selection, Build Model, Map Preview and Batch Execution. The samples are hosted in a central database on Mundi which can be accessed and complemented by all authorized users. The users can select among a comprehensive set of time features and select and parameterize different machine learning models. Each functionality which triggers computation on the cloud (e.g. Build Model) is highlighted with a Mundi button. Once such a functionality is used a progress bar informs the user about the status of the computation on the cloud.

Lessons learned

- The infrastructure and services provided on Mundi / OTC are an excellent base for the implementation of cost-efficient scalable services for large scale land cover classification.
- Third-party services that provide well-documented and performant APIs for the retrieval of ARD can significantly lower the bar to implement new services for different users and thematic applications. Unfortunately the Sentinel-Hub is has not yet reached full performance on Mundi so that the EO-data is currently retrieved from the AWS instance of the service instead from Mundi itself. Solutions have been discussed and are currently being implemented.
- Related to the previous point it is important to note that available services and access to datasets constitute decision criteria (for or against a given cloud) which will in many cases overwrite slight differences in the pricing models.
- Compared to lower level frameworks (e.g. Kubernetes) OpenShift provides an excellent framework for managing and up-scaling computational resources. Downscaling is typically more difficult and requires respective mechanisms and communication from the application side.
- As many other cloud providers the OTC establishes quota which are essentially a backstop to avoid application errors that lead to undesired up scaling of the computation and potential financial damages. Increasing such quota requires written requests which needs to be kept in mind when planning the scalability of the application.
- In terms of cost factors it is important to note that porting and implementing applications (e.g. from a private to a public cloud environment) can easily exceed the costs for computation and storage. It is therefore crucial that the DIASes provide a long-term perspective and further establish common standards to avoid lock-in effects.
- Regarding the infrastructure costs the VMs are typically the main cost factors for applications with operator interaction (i.e. it is easier to fully exploit available resources with fully automated processed). Even with large amounts of data storage is very affordable and the costs for traffic and requests from/to the S3 storage is often above the costs of the storage itself.

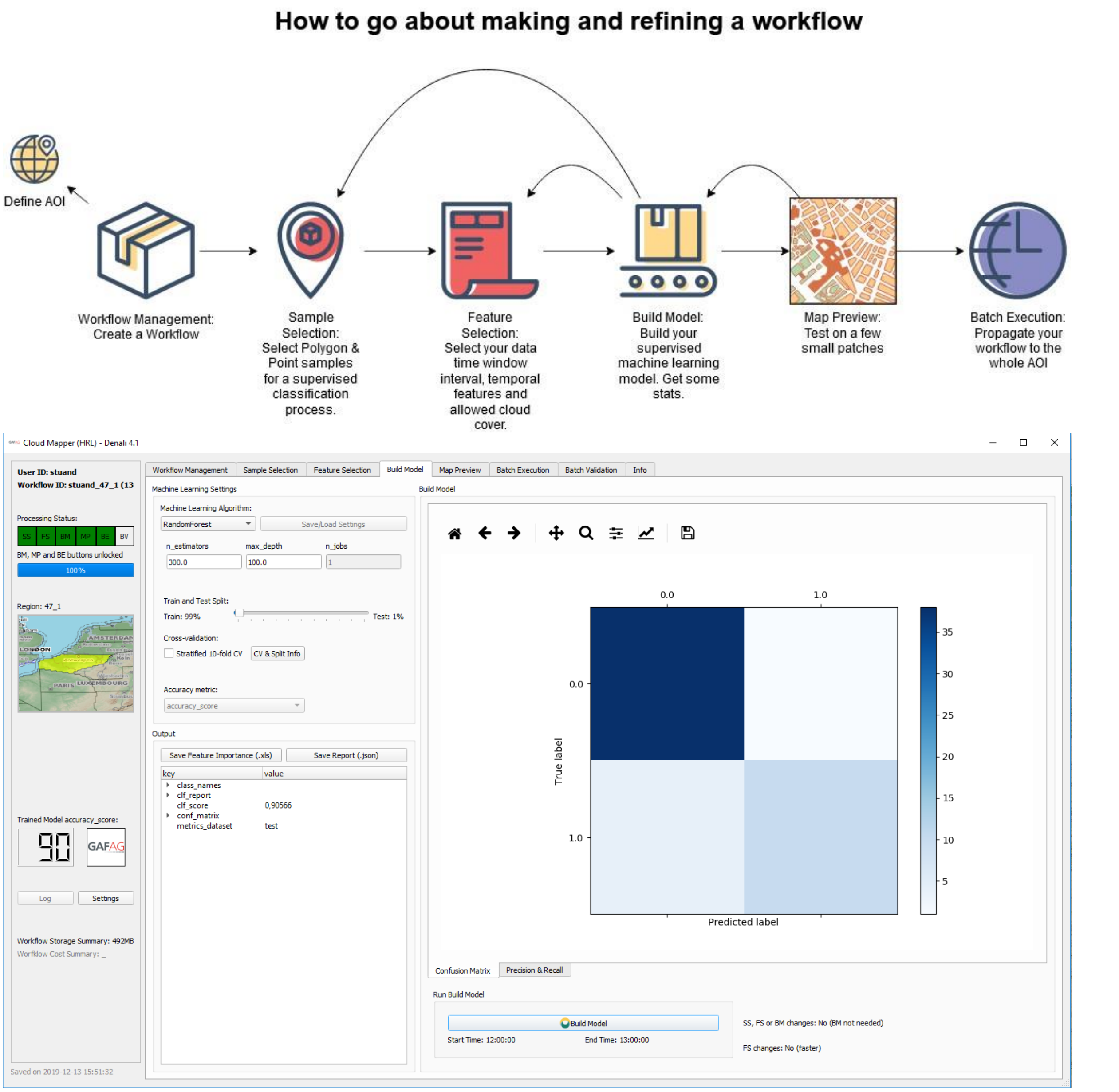


Figure 17: Screenshot of the CloudMapper 4.1 plugin which allows to access and steer the application on Mundi.

D. CREODIAS

Joanneum Research ordered access to the CREODIAS services for the period: 23rd May 2019 to 23rd December 2019 for following items:

Nr	VM's configuration	Quantity
1	VM hm.xlarge vcores 8 RAM 64 local disc SSD 256 GB	1 VM
2	Storage SSD	400 GB
3	Storage HDD	2000 GB
4	Internet Access	1000 GB

For detailed info on the cloud service CREODIAS which is offered by CloudFerro please see their www-page https://creodias.eu/ .

Joanneum Research implemented various tools at the platform with main focus on pre-processing of large amounts of Sentinel-2 Level 1C data to Level 2A as well as for generating improved cloudmasks, generation of Analysis Ready Data (Sentinel2 – Level 3 spatio-temporal data cubes) and for automated derivation of statistical time-features from time-series imagery. Implementation of the software has been performed based on "docker" images. The service proved useful especially in case that a high number of scenes has to be processed. As the software environment is comparable to other DIAS servers, the developed tools can be deployed to any other DIAS service provider which supports software deployments via docker images.

CREODIAS provides a modern State of the Art approach to cloud processing services. The provided OpenStack account enables the customer to define the needed environments in a simple and effective way. Predefined images can be used to speed up the deployment of customer software. Those images come with predefined ssh or rdp configurations, and after a few clicks it is possible to work via Remote Desktop on the newly defined cloud environment.
Access to the large Sentinel 2 data archive comes preconfigured via NFS or S3FS connections which proved to be reliable and with good performance. The data-sets are provided as unzipped S2 directory archive and therefore the possibly time consuming unzipping steps could be avoided.
By choosing a pre-configured "CentOS" image it was possible to have a familiar environment on the processing facility and hence the initial learning phase could be reduced to a very short time.
After pulling the latest software from the JR docker hub, processing could start immediately.

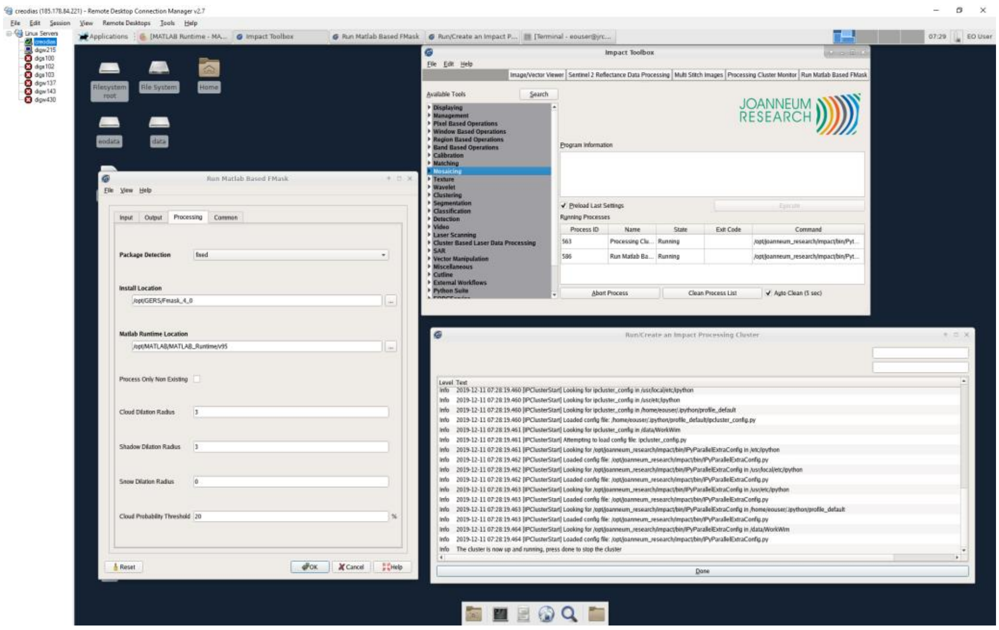


Figure 18: Graphical user interface of tools implemented at CREODIAS, example for cloud mask calculation from S2 time series

The main processing tasks performed on the DIAS environment was the generation of ARD Data (Analyses Ready Data) by an elaborate temporal mosaicking algorithm as well as derivation of statistical time-features from the Sentinel 2 time series.
The important cloud and cloud shadow detection task was done by deploying the FMask algorithm in a "dockerized" environment.
In general the CREODIAS service provided a valuable piece of infrastructure to get fast and easy access to a large scale processing facility of Sentinel 2 data-sets.

E. Conclusion

The use of DIAS services has become much more streamlined since their introduction. Large scale processing is possible on all of the three tested DIASes and has been successfully implemented by ECoLaSS consortium partners.

Nevertheless, each DIAS still requires a non-negligible amount of customization. Processing chains developed on one platform are not transferrable without further adaptations to other DIAS platforms. From a user's perspective increased efforts of the DIAS service providers for standardized APIs and processing environments would be desirable.

A comparatively easy tool to achieve such transferability is the availability of a controlled docker environments, as utilized by JR on CREODIAS, which allows end-users to bring with them their own operating system, without restrictions on software setups and installation as well as the ability for testing and developing locally before shipping a processing chain to DIAS. On the other hand there may be users requiring a higher level of abstraction, which be better served by workflows as tested by GAF on MUNDI with the cloud mapper QGIS plugin, which abstracts away the details of system management with high-level workflow builder. Hence in general, it is desirable to provide both, low-level access for expert users, or users building novel algorithms and processing chains and high-level workflow builders providing easy access to established processing workflows.

Besides the potential to scale up analyses to larger regions, the biggest advantage of the DIASEs is access to large parts of the Sentinel archives. While storage within the DIASes is available at competitive prices, importing and exporting for example from Amazon Web Services is a major cost factor. When evaluating the suitability of a DIAS, it is therefore important to know the available data archive in advance in order to consider potential external costs of data ingestion if further data are needed.