

## D2.1 - Research and Innovation Landscape analysis report

Work Package	WP 2, Requirements Elicitation and Conceptual Framework Specification
Lead Author (Org)	Stefano Natali (SISTEMA)
Contributing Author(s) (Org)	Clemens Rendl, Alexandra Bojor, Maximilien Houel (SISTEMA) George Domalis (NOVELCORE) Kakia Panagidi, Charalampos Andreou, Stathes Hadjiefthymiades (NKUA) Thomas Azrak, Marios Sophocleous (EBOS) Marcel Heckel (FRAUNHOFER) Lionel Blonde (HESSO) Armagan Karatosun, Vasileios Baousis, Mohanad Albughdadi (ECMWF) Giuseppe Trotta, Lucia Rodriguez Munoz, Francesco Maria Cultrera, Balasubramanian Chandramouli (CINECA)
Due Date	28.02.2023
Date	24.02.2023
Version	V0.4

### Dissemination Level

- PU: Public
- PP: Restricted to other programme participants (including the Commission)
- RE: Restricted to a group specified by the consortium (including the Commission)
- CO: Confidential, only for members of the consortium (including the Commission)

## **Disclaimer**

This document contains information which is proprietary to the EO4EU Consortium. Neither this document nor the information contained herein shall be used, duplicated or communicated by any means to a third party, in whole or parts, except with the prior consent of the EO4EU Consortium.

## Versioning and contribution history

Version	Date	Author	Notes
0.1	30.11.2022	Stefano Natali, Clemens Rendl (SISTEMA)	TOC and V0.1
0.2	31.01.2023	Stefano Natali, Clemens Rendl, Alexandra Bojor, Maximilien Houel (SISTEMA)	First complete version
0.3	10.02.2023	Stefano Natali, Clemens Rendl, Alexandra Bojor, Maximilien Houel (SISTEMA)	Updated version
0.4	24.02.2023	Stefano Natali, Clemens Rendl, Alexandra Bojor, Maximilien Houel (SISTEMA) George Domalis (NOVELCORE) Kakia Panagidi, Charalampos Andreou, Stathes Hadjiefthymiades (NKUA) Thomas Azrak, Marios Sophocleous (EBOS) Marcel Heckel (FRAUNHOFER) Lionel Blonde (HESSO) Armagan Karatosun, Vasileios Baousis, Mohanad Albughdadi (ECMWF) Giuseppe Trotta, Lucia Rodriguez Munoz, Francesco Maria Cultrera, Balasubramanian Chandramouli (CINECA)	Updated version with contributions from project partners  Updated Chapters: 4, 5, 6, 7  Added Chapters: 11, 12
0.5	dd.mm.yyyy	Name (Partner)	
1.0	dd.mm.yyyy	Name (Partner)	

## Terminology

Terminology/Acronym	Description
AI	Artificial Intelligence
ADAM	Advanced geospatial DATA Management
ADS	Copernicus Atmosphere Data Store
API	Application Programming Interfaces
AR	Augmented Reality
ASI	Italian Space Agency
CAM	Copernicus Atmosphere Monitoring Service
CC	Climate Change
CDAS	Copernicus Space Component Data Access
CDS	Copernicus Climate Data Store
CMA	China Meteorological Administration
CMCC	Euro-Mediterranean Center on Climate Change
CSA	Coordination and Support Action

Terminology/Acronym	Description
DDS	Data Delivery System
DestinE	Destination Earth
DIAS	Data and Information Access Services
DoA	Description of Action
EC	European Commission
ECMWF	European Centre for Medium-Range Weather Forecasts
EGNOS	European Geostationary Navigation Overlay Service
EO	Earth Observation
EO4EU	Horizon Europe project called: AI-augmented ecosystem for Earth Observation data accessibility with Extended reality User Interfaces for Service and data exploitation
EOP	Earth Observation Portal
ER	Extended Reality
EU	European Union
EUMETSAT	European Organisation for the Exploitation of Meteorological Satellites
FAO	Food and Agriculture Organization
FEMs	Forest Ecosystem Models
FMI	Finnish Meteorological Institute
FTP	File Transfer Protocol
GA	Grant Agreement to the project
GEOSS	Global Earth Observation System of Systems
GeoTIFF	Geographic Tagged Image File Format
GIS	Geographic Information System
GPU	Graphics processing unit
GRIB	GRIdded Binary or General Regularly-distributed Information in Binary form
HPC	High-performance computing
HTTP	Hypertext Transfer Protocol
IES	Intelligence for Environment & Security
INSPIRE	Infrastructure for spatial information in Europe

Terminology/Acronym	Description
ISTAT	Italian National Institute of Statistics
ISRIC	World Soil Information
IVI	Fraunhofer Institute for Transportation and Infrastructure Systems
KEMEA	Kentro Meleton Asfaleias
KPI	Key Performance Indicator
LMCS	Copernicus Land Monitoring Core Service
LU	University of Latvia
ML	Machine learning
NASA	National Aeronautics and Space Administration
NDVI	Normalized difference vegetation index
NetCDF	Network Common Data Form
NKUA	National and Kapodistrian University of Athens
NOAA	National Oceanic and Atmospheric Administration
PASYFO	Personal Allergy Symptom Forecasting System
PU	Public
UC	Use Case
UI	User Interface
URL	Uniform Resource Locator
US	United States
VM	Virtual Machines
VR	Virtual Reality
VU	Vilnius University
WP	Work Package

## Web Resources

#	REFERENCE ID / Link	REFERENCE DESCRIPTION
[URL 1]	<a href="https://cds.climate.copernicus.eu/#!/home">https://cds.climate.copernicus.eu/#!/home</a>	CDS website
[URL 2]	<a href="https://cds.climate.copernicus.eu/cdsapp#!/toolbox">https://cds.climate.copernicus.eu/cdsapp#!/toolbox</a>	CDS Toolbox
[URL 3]	<a href="https://cds.climate.copernicus.eu/api-how-to">https://cds.climate.copernicus.eu/api-how-to</a>	CDS API
[URL 4]	<a href="https://atmosphere.copernicus.eu/">https://atmosphere.copernicus.eu/</a>	ADS website
[URL 5]	<a href="https://land.copernicus.eu/global/">https://land.copernicus.eu/global/</a>	LMCS website
[URL 6]	<a href="https://scihub.copernicus.eu/">https://scihub.copernicus.eu/</a>	Copernicus Open Access Hub
[URL 7]	<a href="https://www.eumetsat.int/">https://www.eumetsat.int/</a>	EUMETSAT website
[URL 8]	<a href="https://view.eumetsat.int/productviewer?v=default">https://view.eumetsat.int/productviewer?v=default</a>	EUMETView
[URL 9]	<a href="https://www.ecmwf.int/">https://www.ecmwf.int/</a>	ECMWF website
[URL 10]	<a href="https://www.asi.it/en/">https://www.asi.it/en/</a>	ASI webpage
[URL 11]	<a href="https://www.asi.it/en/earth-science/prisma/">https://www.asi.it/en/earth-science/prisma/</a>	ASI PRISMA data
[URL 12]	<a href="https://www.istat.it/en/">https://www.istat.it/en/</a>	ISTAT website
[URL 13]	<a href="http://dati.istat.it/?lang=en">http://dati.istat.it/?lang=en</a>	IstatData database
[URL 14]	<a href="https://www.nasa.gov/">https://www.nasa.gov/</a>	NASA website
[URL 15]	<a href="https://firms.modaps.eosdis.nasa.gov/map/">https://firms.modaps.eosdis.nasa.gov/map/</a>	NASA FIRE hotspot platform
[URL 16]	<a href="https://www.fao.org/home/en">https://www.fao.org/home/en</a>	FAO website
[URL 17]	<a href="https://www.fao.org/faostat/en/#home">https://www.fao.org/faostat/en/#home</a>	FAOstat database
[URL 18]	<a href="https://www.isric.org/">https://www.isric.org/</a>	ISRIC website
[URL 19]	<a href="https://www.isric.org/explore/isric-soil-data-hub">https://www.isric.org/explore/isric-soil-data-hub</a>	ISRIC soil database
[URL 20]	<a href="https://adamplatform.eu/">https://adamplatform.eu/</a>	ADAM website
[URL 21]	<a href="https://eurodatacube.com/">https://eurodatacube.com/</a>	EuroDatacube website
[URL 22]	<a href="https://www.copernicus.eu/en/access-data/dias">https://www.copernicus.eu/en/access-data/dias</a>	DIAS
[URL 23]	<a href="https://sinergise.com/en/news/big-news-sinergise-and-earth-observation-community">https://sinergise.com/en/news/big-news-sinergise-and-earth-observation-community</a>	CDAS information page
[URL 24]	<a href="https://dds.cmcc.it/#/">https://dds.cmcc.it/#/</a>	CMCC-DDS
[URL 25]	<a href="https://www.wekeo.eu/">https://www.wekeo.eu/</a>	WEKEO
[URL 26]	<a href="https://www.noaa.gov/">https://www.noaa.gov/</a>	NOAA
[URL 27]	<a href="https://www.weather.gov/documentation/services-web-api#/">https://www.weather.gov/documentation/services-web-api#/</a>	NOAA National Weather Service (NWS) API

# Table of Contents

1	Introduction .....	10
1.1	Purpose of the document .....	10
1.2	Mapping EO4EU outputs.....	10
1.3	Deliverable overview and report structure .....	10
2	Use cases overview.....	11
2.1	Use cases summary.....	11
2.2	Requirements collection .....	12
3	Research Areas .....	15
3.1	Identified research areas .....	15
4	Data accessibility & exploitability .....	16
4.1	Introduction .....	16
4.2	Data access platforms state of the art.....	16
4.2.1	Copernicus data sources .....	17
4.2.2	Geospatial data platforms.....	19
4.2.3	EO data platform aggregators.....	20
4.2.4	Semantic annotation & Knowledge graph .....	21
4.3	Data accessibility.....	22
4.3.1	Project needs versus existing solutions .....	22
4.3.2	Identified gaps per use case.....	23
4.4	Data licensing.....	24
4.4.1	Project needs versus existing solutions .....	24
4.4.2	Identified gaps per use case.....	25
4.5	Direct data exploitability (processing close to data) .....	26
4.5.1	Project needs versus existing solutions .....	26
4.5.2	Identified gaps per use case.....	27
5	Processing capabilities and scale up.....	28
5.1	Introduction .....	28
5.2	Processing capabilities and scale up state of the art.....	28
5.3	Requirements for cloud infrastructure .....	30
5.3.1	Project needs versus existing solutions .....	31
5.3.2	Identified gaps per use case.....	34
5.4	Specific computational needs (HPC Infrastructure, GPUs and vGPUs).....	35
5.4.1	<i>Project needs versus existing solutions</i> .....	35
5.4.2	Identified gaps per use case.....	36
6	Algorithm capabilities.....	37
6.1	Introduction .....	37
6.2	Algorithms capabilities state of the art.....	37
6.2.1	Fusion .....	38
6.2.2	Waterfall algorithm development .....	<b>Error! Bookmark not defined.</b>
6.2.3	Self-supervised annotation-efficient learning.....	39
6.2.4	Learning-based compression .....	41
6.3	Traditional (waterfall) analysis.....	43
6.3.1	<i>Project needs versus existing solutions</i> .....	43
6.3.2	Identified gaps per use case.....	43
6.4	ML/AI based approaches .....	44
6.4.1	<i>Project needs versus existing solutions</i> .....	44
6.4.2	Identified gaps per use case.....	44

7	Results presentation / communication / delivery .....	46
7.1	Introduction .....	46
7.2	Results presentation / communication / delivery state of the art .....	46
7.2.1	GUI state-of-the-art.....	46
7.2.2	Workflow tools and Domain Specific Language (DSL) .....	50
7.2.3	User Management.....	52
7.3	Traditional vs. interactive data presentation tools.....	54
7.3.1	<i>Project needs versus existing solutions</i> .....	55
7.3.2	Identified gaps per use case.....	57
8	Intellectual properties management tools .....	58
8.1	Introduction .....	58
8.2	Intellectual properties management state of the art .....	58
8.2.1	<i>Proprietary and open-source software management</i> .....	58
8.2.2	<i>Protection of input data and workflows</i> .....	59
8.2.3	<i>Protection of results</i> .....	59
9	Concluding remarks .....	60
10	Annex A: Requirements collection form template .....	62
10.1	Input data specifications.....	62
10.2	Data preparation.....	63
10.3	Data Processing.....	63
10.4	Results analysis .....	63
11	Annex B: Collected use case requirements .....	65
12	Annex C: References.....	66

## List of Figures

---

Figure 1. EO4EU requested datasets sources .....	16
Figure 2. Open data access .....	23
Figure 3. Data licensing.....	25

## List of Tables

---

Table 1. Summary of collected requirements broken down by topic. ....	13
Table 2. Summary of collected requirements for data collection and preparation .....	13
Table 3. Summary of collected requirements for computational resources needs .....	13
Table 4. Summary of collected requirements for processing service workflows .....	14
Table 5. Summary of collected requirements for results presentation and provision.....	14
Table 6. Summary of data access platforms and features, relevant to the EO4EU UCs.....	21
Table 7. Mapping of open data versus existing solutions.....	23
Table 8. Summary of data processing platforms features.....	30
Table 9. Comparison Table for OpenStack vs VMWare. ....	32
Table 10. Summary of the performed gap analysis per research area per use case.....	61



## Executive Summary

---

A vast amount of Earth Observation data is produced daily and made available through online services and repositories. Contemporary and historical data can be retrieved and used to power existing applications, foster innovation, and improve EU citizens' lives. However, an undersized audience follows this activity, leaving huge volumes of valuable information unexploited.

The EO4EU project aims to provide innovative tools, methodologies and approaches that would assist a wide spectrum of users, from domain experts and professionals to simple citizens to benefit from accessing EO data. It strives to deliver dynamic data mapping and labelling based on AI augmented modules, adding Fairness to the data and introducing an ecosystem for holistic management of EO data. EO4EU envisages to bridge the gap among domain experts and end users, while aims to bring in the foreground technological advances to address the market straightness towards a wider usage of EO data.

The project will support the wider exploitation of EO data by delivering: (i) Machine Learning (ML) methodologies for Semantic Annotation of existing and growing data sources, (ii) semantically enhanced knowledge graphs that will enable structuring of content around diverse topic areas and building step by step journeys from different sources into a unified approach, (iii) data fusion techniques to extend the scalability of existing distributed systems, (iv) Augmented and Virtual Reality for interactive user experience, and (v) advanced data analytics visualizations for improved learning and evidence-based interpretations of environmental observations. Its operational and technical capacity will be demonstrated within seven distinct pilots that cover different thematic areas, such as personalized health care, sea route planning, ocean monitoring, food security, food ecosystems, soil erosion, environmental pest, and crisis management. These thematic areas will engage a wide spectrum of involved stakeholders, from EO providers, policy makers and actors, researchers and academics to citizen scientists and the general public to join efforts and provide their multidisciplinary expertise to support the Commission's strategic goals towards further exploitation of EO data.

The current document summarises a detailed work performed to identify the main driving lines on which the platform implementation work shall be organised around. Starting from the use cases to be implemented within the project, requirements in terms of data, processing tools, algorithms, results presentation and IPR have been collected and assessed. A detailed analysis on the gaps with respect to the state of the art has been performed and the results, per research area, have been aggregated based on the level of criticality.

The outcome showed that main criticalities remain on the access to relevant data and data sources, removing barriers to some datasets that prevent the development of services on them, and on the possibility to deploy a variety of computational resources close to the data. Besides traditional CPU-based processing technologies, GPU and HPC applications still remain not fully exploited and need some push also on the facilitation of the exploitation of these tools. The adoption of ML and AI tools is not yet widely diffused because these tools are not yet well known by the service developers than for technological gaps, thus there is a need of pushing awareness and knowledge of their functionalities and capabilities within the community.

# 1 Introduction

## 1.1 Purpose of the document

This document, the Research and Innovation Landscape Analysis Report, provides a GAP analysis for each of the seven use cases to support the definition of the final requirements and overall system architecture.

## 1.2 Mapping EO4EU outputs

EO4EU GA Component Title	EO4EU GA Component Outline	Respective Document Chapter(s)	Justification
T2.1	Conducts a detailed analysis of the various research areas in each of the fields in which EO4EU aims to base its innovation strategy. A state-of-the-art for each of the research areas relevant for the proposal will be examined, documented, and discussed with the Consortium in a truly multidisciplinary approach. A gap analysis for the selected use cases and technologies will be performed by mapping existing tools.	Section 3: research areas Section 4 to Section8: research areas analysis, state of the art and gap analysis per use case	The document is based on a complete survey performed on the use cases to collect their needs. On this basis, relevant research areas have been identified and, for each research area, state of the art and gap analysis per use case is performed.
D2.1	The deliverable will conduct a capability GAP analysis for each use case and help inform the final requirements, and overall system's architecture.		

## 1.3 Deliverable overview and report structure

The document is structured as follows:

- The first section represents the Executive Summary, that aims at summarizing the main document's contents and the conclusions;
- Section 1 (this section) introduces the document's scope and the main sections,
- Section 2 provides a summary of the different use cases,
- Section 3 contains the overview of the identified research areas,
- Section 4 to Section 8 provide the detailed description of the identified research areas, including the state of the art and the gap analysis. For each research area, the use cases for which gaps are identified are also provided,
- Section 9 summarizes the document's conclusions,
- Annex A provides the requirements collection form template.

## 2 Use cases overview

---

### 2.1 Use cases summary

The EO4EU project involves the implementation and execution of seven use cases to present application examples for efficient uptake of geospatial data that take advantage of state-of-the-art technologies, including machine learning approaches. The use cases have the purpose of demonstrating the advantages of unifying large data infrastructures to deliver services to the public.

The seven use cases cover the topics of personalized health care services (UC1), maritime supply chain (UC2), food security (UC3), forest ecosystems (UC4), soil erosion (UC5), environmental pests (UC6) and civil protection (UC7). Following, a summary of each of the use cases is provided.

**UC1**, led by the Finnish Meteorological Institute (FMI), with the participation of the Vilnius University (VU) and the University of Latvia (LU), focuses on further expanding the capacity of the PASYFO model. The Personal Allergy Symptom Forecasting System (PASYFO) model is an operational symptom forecasting model that includes a mobile application, which provides personal allergy symptoms forecasting. Currently available for Latvia and Lithuania, the scope of UC1 is to expand the coverage of the PASYFO model to the globe, making it universally available. This will significantly increase the geography features of users and thus contribute to raising public awareness and help preventing chronic diseases (such as allergies and consequent asthma).

**UC2**, led by DANAOS Shipping, with contribution from the National and Kapodistrian University of Athens (NKUA), aims to incorporate cutting-edge EO technologies and data to optimize ship routing in terms of fuel consumption, safety and arrival time precision. The scope of the use case is to integrate the EO4EU capability of handling extreme volumes of data by fusing the meteorological data collected from EO data sources with on-board vessel sensory information to perform route optimization during the voyage of the ship.

**UC3**, led by SISTEMA, with contribution from the Euro-Mediterranean Center on Climate Change (CMCC), will perform crop productivity estimations based on EO data and AI. The scope of the use case is the use of specific climate indicators to evaluate the impact of extreme climate events and other adverse phenomena on agricultural crops. The indicators are used to estimate risk, make forecasts, and issue alerts for potential production losses. This way, it is possible to investigate feedback loops of environmentally damaging food systems on the climate and food production. As a result, transformative adaptation is enabled, promoting long-term resilience by continually shifting the geographical locations where specific types of crops and livestock are produced, aligning agricultural production with changing landscapes and ecosystems, and/or introducing resilience-building production methods and technologies across value chains.

**UC4**, led by CMCC, with the support of SISTEMA, will evaluate the impacts of current and future climates on forests using Forest Ecosystem Models (FEMs). These models simulate the fluxes of water, energy, and carbon, and can be used to project the effects of modified climates on forest growth and carbon sequestration. This information can be used to enhance sustainable forest management in the face of climate change and the demand for forest ecosystem services. Climate change may cause forest species to adapt to new conditions or migrate to areas with more suitable conditions for survival. Changes in climate dynamics can impact the quantity and quality of goods and functions provided by forests, known as forest ecosystem services. The scope of this use case is to generate information which can be used to identify and validate a business model for forestry companies that exchange ecosystem services produced by sustainably managed forests.

**UC5**, led by CMCC, with contribution from SISTEMA, aims to integrate datasets on rainfall and soil susceptibility to assess the risk of water erosion. Soil erosion occurs when detached soil is transported and deposited away due to rainfall, runoff, snow melting, or irrigation. If the soil erosion rate is higher

than the soil formation rate, the soil becomes depleted and the land's productivity is reduced. The economic costs of soil erosion are high and affect many sectors. The developed service will provide information on water-induced soil erosion to a variety of end-users, considering the potential impacts of climate change on erosion. This information can help land management actors and territorial planners make informed decisions on farming practices, forest management, and soil recovery after disturbances, and can also be used to design more resilient infrastructure. Investments can also be more appropriate if a range of future outlooks is considered when evaluating modified risks for infrastructure.

**UC6**, led by SISTEMA, with contribution from CMCC, aims at providing information services for assessing and predicting the impact of locust plagues. The service combines Earth Observation and climate data using AI and machine learning techniques to improve the reliability and effectiveness of the monitoring and prediction service. The use case aims to improve the information service by introducing new climate models for prediction and using high-performance computing infrastructure for computational aspects. Desert locust plagues are strongly influenced by climate conditions and vegetation status. Previous studies have separately analysed Earth Observation techniques and climate predictions, but this service combines both to provide a more comprehensive approach.

**UC7** is led by the Intelligence for Environment & Security (IES), with the support of the Center for Security Studies (Kentro Meleton Asfaleias, KEMEA), the Fraunhofer Institute for Transportation and Infrastructure Systems (IVI) and NKUA. This use case will improve the use and exploitation of European Union-observed datasets in Civil Protection operations. The goal is to improve Civil Protection activities by providing timely, targeted reactions to dynamic events such as earthquakes, forest fires, landslides, and floods. The use case aims to deliver updated imagery with automatically detected changes caused by these events, using Earth Observation for Europe services to discover and present relevant satellite-derived information to end users.

## 2.2 Requirements collection

The basis for the gap analysis presented in this document are requirements collected from each of the use case conductors (see Section 2.1). The requirements were elaborated based on a questionnaire, which was drafted by EBOS and SISTEMA. Annex A (Chapter 10) contains the questionnaire template. It contains four sections, which covers input data specification needs, data preparation needs, data processing needs and means of presenting and visualizing results.

To fill the individual questionnaires, it was first provided online to the use case leaders (seven teams involved), enabling them to familiarise with the presented sections. In a second round, one-to-one meetings/ telcos were held by SISTEMA with the corresponding use case conductors, to complete the questionnaires and eventually get a complete overview of the workflow conducted within each use case. Table 1 summarises the collected requirements per topic. All collected requirements per use case can be accessed through the links provided in Annex 2.

Table 2 summarises the requirements for data collection and preparation. At the current stage, all use cases foresee the local download of data to be processed, but six out of seven would prefer to execute pre-processing close to the data; five out of seven use remote data sub-setting functionalities while requesting the data, five out of seven apply locally further spatial sub-setting and interpolation operations, while only three of them perform temporal sub-setting and aggregation.

Domain	#Req.
Data	108
Data preparation / pre-processing	101
processing / algorithms	47
results presentation	36
<b>TOTAL</b>	<b>292</b>

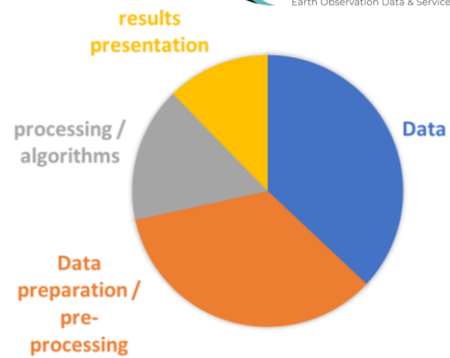


Table 1. Summary of collected requirements broken down by topic.

	UC1	UC2	UC3	UC4	UC5	UC6	UC7	# UCs
Data download	X	X	X	X	X	X	X	7
Remote data sub-setting		X	X	X	X	X		5
Spatial sub-setting / interpolation			X	X	X	X	X	5
Temporal sub-setting aggregation				X		X	X	3
Pre-processing close to data	D		D	D	D	D	D	0

Table 2. Summary of collected requirements for data collection and preparation

Table 3 lists the computational needs: all use cases require traditional CPU resources, while five of them (will) require GPU resources. Elastic processing is intended as on-the-fly improvement of computational resources to sustain e.g., variable number of users (three services), while the scale-up represents the possibility to extend the service capabilities e.g. a larger spatial or temporal domain (six services). High computational resources (e.g., HPC) are requested, at this stage, by two of use cases that foresee the use of numerical models but can be envisaged to be used by other use cases with adequate support.

	UC1	UC2	UC3	UC4	UC5	UC6	UC7	# UCs
CPU	X	X	X	X	X	X	X	7
GPU		X	X	X	X	X		5
Elastic processing	X	X				X		3
High computational resources			X		X			2
Scale up possibility		X	X	X	X	X	X	6

Table 3. Summary of collected requirements for computational resources needs

Table 4 provides information about the current and expected status of the services. There is a variety of maturity stages among the services: two of them already feature a high level of operability (Technology Readiness Level - TRL=9), two of them are pre-operational, and three of them are still at prototype level. Most of them (six out of seven) feature a traditional waterfall workflow, while only one already includes the application of ML/AI, even though at least other four plan to include ML/AI based steps within their pipelines. Full processing automation is expected by five use cases.

	UC1	UC2	UC3	UC4	UC5	UC6	UC7	# UCs
<b>Traditional waterfall workflow</b>	X	X	X	X	X		X	6
<b>ML/AI processing</b>		D	D	D	D	X		5
<b>Automatic processing</b>	X	X		X		X	X	5
<b>Maturity level (TRL)</b>	7	9	4	7	4	4	9	6,286
<b>4. prototype</b>			X		X	X		3
<b>7. pre-operational</b>	X			X				2
<b>9. Operational</b>		X					X	2

**Table 4. Summary of collected requirements for processing service workflows**

Table 5 summarises data presentation requirements: the majority of the use cases require the provision of the results via web application, while other results provision modalities (APIs, mobile application, data download and result push) are only requested by few services. At the current stage, no advanced data exploration/exploitation functionalities (e.g., immersive technologies, virtual/augmented/mixed reality) are requested, but they will possibly be implemented by some use cases once available through the platform.

	UC1	UC2	UC3	UC4	UC5	UC6	UC7	# UCs
<b>Web app</b>	X	X	X	X	X	X		6
<b>API</b>		X				X		2
<b>Mobile App</b>	X	X						2
<b>Data download</b>				X	X	X		3
<b>Results push</b>							X	1

**Table 5. Summary of collected requirements for results presentation and provision**

## 3 Research Areas

---

### 3.1 Identified research areas

The current document aims at describing in detail the research areas on which the EO4EU project aims to base its innovation strategy. The relevant research areas are identified by analysing the use cases being implemented in the framework of the project. It represents a meaningful sample of existing services based on Earth Observation data.

Based on the elaborated requirements (see Section 2.2), the research areas (and sub-areas) have been identified by investigating the different requirements, identifying commonalities and differences, and highlighting the critical topics. These findings will need to be addressed in the framework of the project as goals of the innovation strategy.

The following list breaks down the identified research areas and sub-areas:

- Data accessibility & exploitability
  - Data accessibility
  - Data licensing
  - Direct exploitability (processing close to data)
- Processing capabilities and scale-up
  - Computational resource needs
  - Specific computational needs (HPC, GPUs, ...)
- Algorithm capabilities
  - Traditional (waterfall) analysis
  - ML/AI-based approaches
- Data presentation / communication / delivery
  - Traditional vs. interactive data presentation tools
- Intellectual properties management tools
  - Open-source software management
  - Infrastructure security
    - Data Protection: secure results transmission/delivery
    - for publicly available results, how to avoid copying/cloning/reselling.

The following sections describe the research areas, the current state of the art and the gap versus the collected requirements, broken down per use case.

## 4 Data accessibility & exploitability

### 4.1 Introduction

The research topic of data accessibility and exploitability reviews the required means of accessing and leveraging datasets requested by the seven use cases. The outcome, which is extracted from the previously mentioned questionnaire (see Annex A: Requirements collection form template), is compared to current standards and possibilities of data access and usage. Based on this comparison, an analysis of the gap is performed, elaborating on the shortcomings of current approaches, and highlighting the innovation potential.

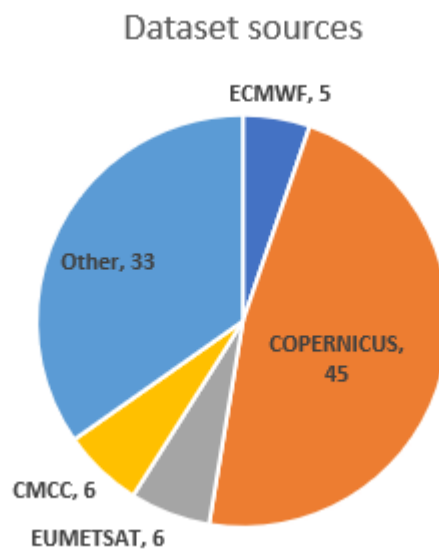


Figure 1. EO4EU requested datasets sources

Figure 1 provides an overview of the data sources of the 95 individual datasets requested by the service providers for the conduction of the seven use cases within EO4EU. 45 of the datasets, which correspond to 48%, are requested from the various Copernicus initiatives, including Copernicus Land, Climate and Atmosphere. Six datasets each are requested from EUMETSAT and CMCC. Five datasets from ECMWF are high-resolution weather forecast data, which are proprietary. This topic is discussed in Section 4.4. The 33 datasets summarised as Other have various sources. They include datasets provided by UC stakeholders (UC3, UC4, UC7) and datasets which are generated by the service provider (UC2). Moreover, datasets requested from the following sources are placed in the Other category: ASI, ISTAT, NASA, ISRIC, EC, NOAA, and FAO.

### 4.2 State of the art on data access platforms

In EO4EU, massive batches of historic, current and predictive EO-based observational and model datasets, along with open-access data from national entities and pre-identified databases are used. Additionally, open-source data collections from numerous sources, private and/or public entities, EU-funded initiatives and projects are utilized (see Figure 1). This chapter lists the means of accessing the requested datasets relevant to the use cases and elaborates the features of the data sources (see Figure 1). It is structured around sources of the requested datasets. First, the various means of accessing Copernicus data are discussed. Subsequently, the platforms and sources to access data relevant to the use cases other than Copernicus data are explained. The chapter closes with a brief mention of two EO data platform aggregators and an overview table of the discussed data platform sources and their features.



### 4.2.1 Copernicus data sources

The gross of the requested data for the conduction of the seven use cases are related to the Copernicus programme; 45 of the datasets, which correspond to 48%, are requested from the various Copernicus initiatives. For the Copernicus programme, a free, full, and open data policy has been adopted by the EU. Outlined below are some of the Copernicus program data access portals that are relevant to this project.

European Commission has funded the deployment of five cloud-based platforms, that provide centralized access to Copernicus data and information, as well as to processing tools. These platforms are known as the **Data and Information Access Services (DIAS)** [URL 22]. The five DIAS online platforms allow users to discover, manipulate, process and download Copernicus data and information. All DIAS platforms provide access to Copernicus Sentinel data and to the six operational services of Copernicus, together with cloud-based tools (open source and/or on a pay-per-use basis). Moreover, each of the five platforms provides access to additional commercial satellite or non-space data sets.

The **Copernicus Space Component Data Access (CDAS)** [URL 23] is a system that will enable efficient processing of satellite EO data. It represents a central point for the entire archive of collected data and is thus a system that will be used by research institutions, companies and also public institutions for the development of applications that use satellite data. Data will be available through various interfaces, such as direct access with STAC items and cloud-optimized formats to streamlined access APIs. There will be a web-based application built on top of a very popular EO Browser technology as well as a JupyterLab and several on-demand processors capable of building non-default formats and derived product.

**WEkEO** [URL 25] is the EU Copernicus DIAS reference service for environmental data, virtual processing environments and skilled user support. It provides users with a single distributed tool for accessing, visualising and analysing all Copernicus data and services, including big-data analysis tools, to develop applications tailored to their specific needs, providing direct, up-to-date and non-replicated access to the Copernicus portfolio (including the Copernicus Atmosphere, Marine, Climate and Land Monitoring services) and satellite data from the Copernicus Sentinel missions. **WEkEO** offers the following two plans to access and process the data: The Essential plan, free and open, provides access to all Copernicus data, Jupyter Notebooks and access to WEkEO user support; the Advanced plans, under a variety of pricing schemes and offers, include, on top of the features of the Essential plan, cloud-based virtual computing environments and tools and free networking.

The **Copernicus Climate Data Store (CDS)** [URL 1] is a component of the Copernicus program that provides access to a wide range of climate-related data, including observations, forecasts, reanalyses and climate model data. The CDS is designed to support the use of this data in applications such as climate research, weather forecasting and climate risk assessment. CDS contains a toolbox and an API to access the data and its documentation materials. The **CDS Toolbox** [URL 2] is the place where one has access to a full set of different documentation materials and tutorials, to an editor where one can edit and run applications, to API references library and to an application gallery to gain inspiration for future applications. The **CDS API** [URL 3] is a service that provides programmatic access to the data for different operating systems (e.g., Windows, macOS, Linux) to all these features can be done by registering for free.

Similarly, the **Copernicus Atmosphere Data Store (ADS)** [URL 4] provides access to observations, forecasts, reanalyses and model data specifically from the atmospheric domain. Its purpose is to support the use of this data in applications such as air quality monitoring, climate research, and weather forecasting. Data from ADS can be requested through different methods: The **ADS web interface**, which is an interactive system where the user fills a web form to construct their query. The user can then currently choose between two options, submit the form, and download the result of

the query or show the query as an API request. The **CDS API** is a service providing programmatic access in Python to ADS data. Users need to have a CDS account to use it with their related CDS API credentials.

It is worth mentioning that both ADS and CDS store data in a standardized format. Its datasets can be discovered via a search tool, while the textual description content can be extracted from the provided HTML code. More specifically, ADS and CDS are not providing direct access to data, but allow users to generate the API requests towards accessing/downloading the datasets needed. API requests are essentially json file types that use key-value pairs to encode the query parameters. Each dataset contains various products, each containing data for a set of variables that can be provided using different data formats. Complementary to the API generator, a constrained json field directly displays how the various products and their corresponding features are grouped together. Below an example of API request from ADS is presented:

```
import cdsapi
c = cdsapi.Client()
c.retrieve(
    'cams-global-emission-inventories',
    {
        'version': 'latest',
        'format': 'zip',
        'variable': [
            'acids', 'carbon_dioxide',
        ],
        'source': 'anthropogenic',
        'year': '2000',
    },
    'download.zip')
```

The Copernicus Open Access Hub offers access to raw EO data from a system of satellites that generate substantial amounts of high-res images/observations of Earth. Provides complete, free and open access to Sentinel-1, Sentinel-2, Sentinel-3 and Sentinel-5P raw data. There are multiple avenues to gain access to the data: through the use of APIs, FTP servers and specialized services. These enable users to run queries through web GUIs. Potential access points:

- <https://scihub.copernicus.eu/>
- <https://scihub.copernicus.eu/dhus/#/home>
- <https://scihub.copernicus.eu/userguide/WebHome>

Another data provision platform of the Copernicus program is the **Copernicus Land Monitoring Core Service (LMCS)** [URL 5] which provides a series of qualified bio-geophysical products on the status and evolution of the land surface, information on the land cover, land use and its changes. Also, information on the different hotspots that are prone to specific environmental challenges and problems is provided. To access the data, an account must be registered. LMCS offers various data access methods, including interactive **notebooks**, to create and share documents that contain live code, equations, visualizations and narrative text; web services or web-based **APIs**, which assist with executing standard functionality over the web; Virtual Machines (**VMs**), providing remote desktop environments; **Manifest files**, which are text files containing web links (URLs) to the main data files, in their native format (netCDF4 or Cloud-Optimized GeoTIFF), per product collection (e.g., NDVI 300m version 1); regular **FTP** access, suitable for downloading large portions (or even entire) collections of products, in their native format; the **legacy portal**, allowing to interactively search for and order, free-of-charge, individual or sets of products; GEONETCast which is an initiative of the global Group on Earth Observations which inter-connects the systems for the broadcast of data via telecom satellites

operated by: US-NOAA over Americas, EUMETSAT over EU-Africa and CMA over China/Asia. The Copernicus Global Land Service products are available on the EUMETSAT system, called **EUMETCast** and thus broadcast over EU, Africa and surrounding countries

Also, a part of the Copernicus program is the **Copernicus Open Access Hub** [URL 6] (previously known as Sentinels Scientific Data Hub) which offers complete, free and open access to Sentinel-1, Sentinel-2, Sentinel-3 and Sentinel-5P user products and a platform where the users can visualize and download the needed products through a free account registration. It is also possible to access the Sentinel products through a dedicated **API Hub**.

#### 4.2.2 Geospatial data platforms

As it can be seen from Figure 1, besides Copernicus data, additional geospatial datasets are requested for the implementation of the use cases. This chapter lists the platforms and sources for the requested remaining datasets and elaborates means of accessing the data.

**EUMETSAT** [URL 7] is an intergovernmental organisation based in Germany that provides satellite data and products that are vital to weather forecasting, making a significant contribution to the monitoring of the environment and climate change. EUMETSAT data can be visualized on their platform EUMETView [URL 8], which is accessible via a **WebUI** and **API**. The API allows users to integrate EUMETView into other applications and systematically download images and products without accessing the web user interface. This service and other data can be accessed through a dedicated portal namely the Earth Observation Portal (EOP), where users need to create an account.

**ECMWF** [URL 9] is the European Centre for Medium-Range Weather Forecasts and it produces global numerical weather predictions by offering quality-assured information on climate change, atmospheric composition, flooding and fire danger. Some data is made available under licence, some are publicly available. Depending on the data needs, ECMWF data can be accessed through different methods: **Computer access** provides vital resources for computer modelling of the global atmosphere and ocean and weather forecasting research; **API** access enables one to programmatically request and retrieve data via HTTP from the ECMWF data archive for use in your web, mobile, or desktop applications. The data request is made using the ECMWF MARS scripting language and the data is received as NetCDF, GRIB or JSON. Depending on the API service used; real-time data, archive data, or charts can be ordered. Real-time data can be delivered to authorised or licensed users directly by ECMWF, while open real-time data can be accessed free of charge via the public **FTP**.

The **CMCC Data Delivery System (DDS)** [URL 24] provides a unique, consistent and seamless access point for all data produced and used by CMCC through a unified **API** interface. The user can browse the catalogue and the available datasets through the **DDS Web Portal** and access and download data through the **DDS API** Python client.

The data listed as Other in Figure 1 includes 33 datasets which have sources other than the previously mentioned platforms. These sources are described in the following paragraphs.

The **Italian Space Agency (ASI)** [URL 10] provides a portal where users can request PRISMA data [URL 11], through free registration and use it in different applications, e.g. vegetation analysis.

The **Italian National Institute of Statistics (ISTAT)** [URL 12] is a public research organization and the main producer of official statistics. It provides data on different domains that can be used to fill the gap in the satellite data. In this project information on administrative boundaries and agriculture will be used. The datasets are available on a spreadsheet and have an introductory and methodological notes which can be downloaded for free. The data can be accessed through the Istat Data [URL 13] database with or without registration.

The **National Aeronautics and Space Administration (NASA)** [URL 14] is America's civil space program and the global leader in space exploration. It offers a wide range of data and products, such as fire

hotspots, through a dedicated dashboard. NASA has several data archives providing the public with datasets from a particular domain, field of science, or mission. For example, for the fire hotspot [URL 15] there is a dedicated platform where one can download data that are not older than seven days. For downloading the product files, one can use a web browser or the command-line, FTP and LFTP utilities. The platform also provides the possibility to visualize current and historic data. Similarly, the **National Oceanic and Atmospheric Administration (NOAA)** [URL 26] is also an US government agency. While NASA focuses on space exploration and aeronautics research, NOAA focuses on weather forecasting, climate monitoring and scientific studies of the oceans and coasts. NOAA’s free and open National Weather Service (NWS) API allows developers access to critical forecasts, alerts and observations along with other weather data [URL 27].

The **Food and Agriculture Organization (FAO)** [URL 16] is a specialized agency of the United Nations that leads international efforts to defeat hunger. FAO also provides different data products, like in this project in-situ data on the geolocation of the locust outbreaks. They provide a webpage, called FAOstat [URL 17], where data can be visualized and downloaded without registration.

**ISRIC – World Soil Information** [URL 18] is an independent foundation that provides datasets on soil moisture through a dedicated data hub. The soil information products are available to data users through ISRIC Soil Data Hub [URL 19] and can be downloaded without registration.

### 4.2.3 EO data platform aggregators

Not just the data providers themselves, but also other sources can be used to obtain geospatial data and associated products. Such platforms are called aggregators and might offer data from e.g., remotely sensed satellites from various sensors and model data. In addition to data provision, aggregators often provide possibilities for data visualization and exploitation, e.g. the generation of geographic data subsets or point-location timelines for certain parameters. Lastly, other means of data access are provided, such as APIs. Two examples of such aggregator-platforms are the following:

The **Advanced geospatial DATA Management (ADAM)** [URL 20] platform is an efficient and robust system, that manages the full data cycle: discovery, access, exploration, processing and visualization services are made available on top of the 3D virtual globe powered by ESA-NASA Web World Wind. This is the natural environment where users find easy-to-use service functionalities to dynamically interact with EO products. It implements the Digital Earth concept allowing accessing a large variety of multi-year global geospatial collections enabling data discovery, visualization, combination, processing and download. Furthermore, the exploitation of different data types from a global to local scale is permitted.

**EuroDatacube** [URL 21] is a platform that provides different services and makes available a massive amount of continuously updated datasets from a wide range of imagery providers and different data sources in one place. It offers users data access through a unified API, batch processing, xcube, (a fully customizable data pipeline to generate tailored cubes), geoDB (a fully-featured PostgreSQL database) and also EOxHub Workspace (a Kubernetes-powered execution of dockerised applications). To be able to order the Euro Data Cube services, one will first need to create a user account on the EUDC platform.

Table 6 summarises the data access platforms, listing their main data access and exploitation features.

Portal / data source	Access mode (open / registration)	Access via web UI	Access via APIs	Direct data access (ftp / http / openDAP)	Exploitability (processing close to data)
<b>Copernicus data sources</b>					
<b>DIASes / C-DAS</b>	Registration	Yes	Yes	No	Yes
<b>WeKEO</b>	Registration	Yes	Yes	No	Yes

Portal / data source	Access mode (open / registration)	Access via web UI	Access via APIs	Direct data access (ftp / http / openDAP)	Exploitability (processing close to data)
<b>Copernicus data sources</b>					
<b>CDS</b>	Registration	Yes	Yes	No	No
<b>ADS</b>	Registration	Yes	Yes	No	No
<b>CLMS portal</b>	Registration	Yes	Yes	Yes	Yes
<b>Copernicus Open Access Hub</b>	Registration	Yes	Yes	Yes	No
<b>Geospatial data platforms</b>					
<b>CMCC DDS</b>	Registration	Yes	Yes	No	No
<b>ECMWF</b>	Registration	Yes	Yes	No	No
<b>EUMETSAT</b>	Registration	Yes	Yes	Yes	No
<b>ASI PRISMA portal</b>	Registration	Yes	No	No	No
<b>FAO</b>	Open	Yes	No	Yes	No
<b>ISRIC</b>	Open	Yes	No	Yes	No
<b>ISTAT</b>	Open	Yes	No	Yes	No
<b>NASA/NOAA</b>	Open	Yes	Yes	Yes	No
<b>EO data platform aggregators</b>					
<b>ADAM</b>	Registration	Yes	Yes	No	Yes
<b>EUDC</b>	Registration	Yes	Yes	No	Yes

**Table 6. Summary of data access platforms and features, relevant to the EO4EU UCs**

#### 4.2.4 Semantic annotation & knowledge graph

Various of services across EU provide access to both processed datasets as well as raw EO data. Such an enormous volume of information requires an intelligent approach to identify which datasets should be used.

EO data require interpretation and rely on a comprehensive knowledge base of value-chain analysis (Matevosyan et al., 2017). Therefore, knowledge representation and inference of Big Data are crucial issues in the research field and a promising solution for Big Data analysis.

Towards enabling knowledge acquisition and reasoning from data, the concepts of Semantic Annotation and Knowledge Graph (KG) have been introduced. The term “Semantic Annotation”, closely related to the term of “Semantic Web”, refers to the automatic process of adding formal structure and semantics (metadata and knowledge) to web content for the purpose of more efficient management and access (Kiryakov et al., 2005). On the other hand, “Knowledge Graph” does not have a clear definition, although many exist in the literature (Zou, 2020). A domain agnostic approach gives an informal definition based on characteristics that KGs should possess (Paulheim, 2017), namely:

- Models real world entities along with their relationships and provides them as a graph.
- Defines classes and relationships between entities in a schema.
- 
- Allows the potential interconnection between arbitrary entities.

- Consists of data that covers several topics and domains.

According to the literature, the extraction of semantic annotations for EO Data, as well as the creation of a knowledge graph that models various disparate data sources is a major technical challenge (Wu et al., 2021) directly interconnected with the respective domain. Therefore, in each domain different approaches of semantic annotation and knowledge graph are being introduced. Regarding land and geological applications an approach called KnowWhereGraph (Janowicz et al., 2022) present an approach that (i) comprises of a large spectrum of integrated datasets at the human–environment interface; (ii) introduces their application areas; and (iii) discusses geospatial enrichment services on top of their KG. With regards to atmosphere and climate analysis, (Yacoubi Ayadi et al., 2022) discuss the semantic modeling issues related to spatio-temporal data in the context of meteorological EO data. They support the reuse of a KG of existing ontologies to define a model that semantically defines and integrates meteorological parameters. In terms of marine and maritime applications, (Gan et al., 2022) build an application to facilitate flag state control (FSC) inspection that ensures maritime safety. In this approach KGs are used to integrate heterogeneous knowledge sources. Firstly, an ontology model is built to systematically describe the knowledge and guide the construction of the KG. Then, the BERT-BiGRU-CRF model is used to extract entities from unstructured FSC inspection data.

### 4.3 Data accessibility

Free and open data is data that is freely available to the public for any purpose, without requiring permission from the owner. It is often licensed to allow for reuse, to promote transparency, collaboration, and innovation. This section reviews the accessibility of data in terms of openness. Open data accessibility can be limited through e.g., an access fee or the non-availability of individual datasets on the internet, i.e., non-publicly available data. Another form of restriction could exist through the necessity to submit a request for data before accessing the data. Contrarily, an indication of openness is the availability of direct data download capabilities, such as the provision of an application programming interface .

#### 4.3.1 Project needs versus existing solutions

Considering the potential restrictions and limitations to open data access, 70 individual data products (74%) requested for the use cases of EO4EU are open data. 25 of the overall requested 95 individual datasets (26%) have some type of restriction when accessing the data.

Non-open datasets include:

- Data from stakeholders (UC3, UC4, UC7)
- Restricted data (ECMWF high resolution forecast)
- Data which needs to be requested (PRISMA; CFRS Sicily Forest Corps)
- Data not freely and openly available on the internet (UC5)

### Data accessibility

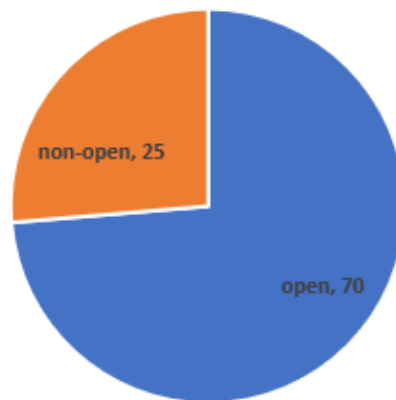


Figure 2. Open data access

Following, Table 7 gives an overview of the sources of the open datasets requested for EO4EU. The table hence includes only portals, which include open data requested by the use case conductors. Note: the total number of datasets does not correspond to the 70 individual datasets, as some individual datasets are requested by multiple use cases. Summarised as Other portals are open data retrieved from DANAOS (UC2), ESA (UC3), JRC (UC4), PROFOUND (UC4), ISPRA (UC5) and OSM (UC5).

Portal / data source	UC1 (Health)	UC2 (Ocean)	UC3 (Crop)	UC4 (Forest)	UC5 (Soil)	UC6 (Food)	UC7 (Fires)
CDS	12		10	14	6	8	
CLMS portal			1	1	3		1
Copernicus Open Access Hub			2				
EUMETSAT							6
FAO						2	
ISRIC				1	1		
CMCC			5		1		
ISTAT			2				
NASA / NOAA		1					1
Other		1	1	1	2		
<b>Total</b>	<b>12</b>	<b>2</b>	<b>21</b>	<b>17</b>	<b>13</b>	<b>10</b>	<b>8</b>

Table 7. Mapping of open data versus existing solutions

#### 4.3.2 Identified gaps per use case

The following section is an overview of the identified gaps in each of the seven use cases concerning data accessibility, structured as bullet points:

- UC1 (Health)

- Request for ECMWF high resolution forecast data
- UC2 (Ocean)
  - No gap identified concerning data accessibility
- UC3 (Crop)
  - Request for PRISMA data
  - Request for stakeholder input data (crop yield and crop field polygons)
- UC4 (Forest)
  - Request for stakeholder input data (various site-specific forest model input data (e.g., available soil water, soil type))
- UC5 (Soil)
  - Request for observational rainfall and soil erosivity data
- UC6 (Food)
  - Request for ECMWF high resolution forecast data
- UC7 (Fires)
  - Request for stakeholder input data (static regional vegetation fire danger)

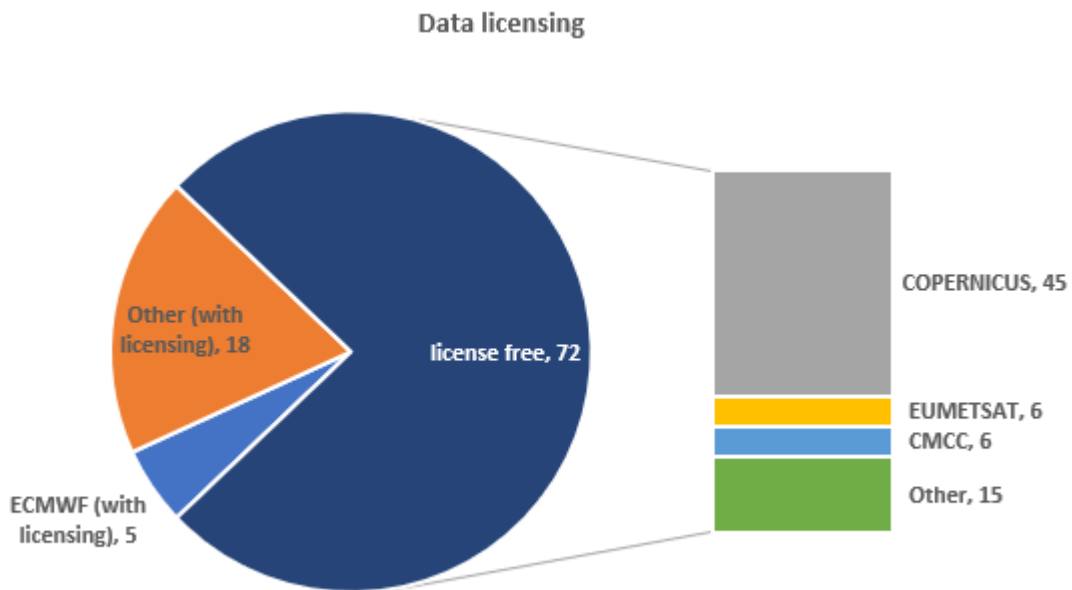
The existing gaps regarding data accessibility (openness) fall into three categories. The first gap is the request for ECMWF high resolution forecast data: access to these data is limited by an access fee. These data are requested by UC1 and UC6. The second type of gap exists with respect to the nature of the demand for specific data. To obtain PRISMA data, a formal written request must be made, which can potentially be denied. UC3 is requesting PRISMA data. The third type of gap consists of data from stakeholders relevant to the conduction of the use cases. The gap in accessibility includes data generated during the use case (UC2) or data generated as input to the implementation of the use cases (UC3, UC4, UC5, UC7).

## 4.4 Data licensing

This section elaborates on the type of data licensing. The characteristic of data being free and open, as opposed to proprietary, is often referred to as its licensing. Free and open data is data that is freely available to the public, without any restrictions on its use. This means that anyone can access, use, and distribute free and open data without obtaining permission from the owner of the data. In contrast, proprietary data is data that is owned by a specific individual or organization and is not freely available to the public. This means that if someone wants to use proprietary data, they must obtain permission from the owner of the data and may have to pay a fee to access it.

### 4.4.1 Project needs versus existing solutions





**Figure 3. Data licensing**

Out of the 95 individual datasets requested, 23 datasets are proprietary. When comparing accessibility (openness) with potential licensing restrictions, license-free data does not automatically correspond to open data. As elaborated previously (see Figure 2 in Chapter 4.3), 25 of the 95 requested datasets are non-open, whereas 23 datasets have some type of licensing. The difference in two dataset is due to the characteristics of the requested PRISMA dataset and a dataset requested from CMCC. The PRISMA dataset is license-free, i.e., PRISMA data can be processed and used for commercial purposes, but it is not considered open, because of the requirement to submit a formal request, when retrieving PRISMA data. In case of the CMCC dataset, it is also license-free, but not openly accessible on the internet; it is available only on demand. The 23 proprietary datasets, i.e., data with license, include high resolution weather forecast data from ECMWF, which are not freely available, as well as data used to implement some of the use cases. Figure 3 gives an overview of the data licensing characteristics of the data requested within the seven use cases. To the left, a pie chart displays the data licensing characteristics, whether the requested data are license-free or have some kind of licensing. The 23 datasets with licenses include the previously mentioned ECMWF data (light blue) and other data (orange). To the right, the data characterised as license-free are displayed as a bar chart. Out of the 72 available license-free datasets, the majority of data is requested from the Copernicus initiative, corresponding to 45 datasets, which represents 48%. Six individual datasets are each requested from the sources CMCC and EUMETSAT. The remaining 15 requested ones have various sources and include the following sources: DANAOS, NOAA, FAO, ASI, NASA, ISRIC and ISTAT.

#### 4.4.2 Identified gaps per use case

This section elaborates on the topic of data licensing on the level of the seven use cases. In total, 23 of the 95 requested datasets for the implementation of the seven use cases have some kind of licensing or are proprietary. For five of the 23 datasets to be used, a fee needs to be paid. These five datasets are data requested from ECMWF and are used by both UC1 and UC6. UC3 uses licensed input data provided by the stakeholders, including field polygons and data on crop production. Likewise, UC4, UC5 and UC7 use proprietary data provided by stakeholders to implement their use cases.

- UC1 (Health)
  - ECMWF high resolution forecast (i.e., dew point, air temperature)

- UC2 (Ocean)
  - No gap identified concerning data licensing
- UC3 (Crop)
  - Stakeholder provided input data (i.e., data from FoodChain)
- UC4 (Forest)
  - Stakeholder provided input data (i.e., various site input data for forest modelling)
- UC5 (Soil)
  - Stakeholder provided input data (i.e., soil modelling input data)
- UC6 (Food)
  - ECMWF high resolution forecast (i.e., air temperature, total precipitation, solar radiation, soil water content)
- UC7 (Fires)
  - Stakeholder provided input data (i.e., static regional vegetation fire danger)

## 4.5 Direct data exploitability

The research area of data exploitability refers to the ability of processing and analysing the data close to the source. Processing data at the source has several key advantages over other approaches to data analysis and management. One of the main benefits of accessing the data as it is being generated, without having to wait for it to be transferred, is the ability to analyse data in real-time or near real-time, which can be useful for monitoring ongoing processes or making immediate decisions.

Another advantage of direct access to data is avoiding duplication of the data. Aside from the fact that such a direct approach saves time and resources, it also reduces the risk of errors or inconsistencies in the datasets. By processing the data directly, you can ensure that you are working with the most up-to-date and accurate information, without having to create and maintain separate versions of the data. This can be especially useful for organizations and domains that generate or collect large amounts of data, as it can reduce the complexity and overhead of managing multiple copies of the data. Additionally, following such an approach can reduce the amount of data that needs to be transferred or stored. This can save time and resources, as well as reduce the risk of data loss or corruption during the transfer process, which can be especially useful for large or complex datasets.

Finally, accessing and processing data at the source could have a positive effect on the integrity and accuracy of the data. Transfer latencies, lossy compressions and decompressions during transfers are amongst the possible causes of data inconsistencies. By analysing the data as it is being collected, one can ensure that the raw data is complete, accurate, and consistent, without the need for additional processing or cleaning. This of course depends on the further usage and steps to be performed with the collected data. This can be especially important for data that is sensitive or critical to an organization, as it can help to prevent errors or inconsistencies that could impact the accuracy of the analysis or decision-making.

### 4.5.1 Project needs versus existing solutions

The analysis of the seven use cases reported that none of the use cases is currently featuring the so called “processing close to the data” paradigm.

All analysed use cases follow the same data preparation and processing workflow:

- Data download from the original data source(s)
- Data preparation (subsetting/regridding)

- Data processing (single dataset analysis or synergistic exploitability of two or more datasets)

One main limitation is the fact that most of the data access facilities do not allow data preparation and processing close to the data: taking Table 6 as reference, only five out of 13 platform allow deploying processors close to the data. Among them, the CDS and ADS do not allow this operation.

This issue is partially mitigated by the availability of spatial and temporal data subsetting APIs, that allow reducing download time and volume once requesting the data: five use cases out of seven exploit this capability once downloading the data.

#### *4.5.2 Identified gaps per use case*

The main gap identified that is common amongst six out of seven use cases (all but UC2) is the **possibility to perform data processing close to the data**. The availability of this capability would not only avoid massive data transfer and need of local infrastructures for data hosting and processing but will also simplify the business workflow. For what regards UC2, a specific requirement related to **secure data access** has been collected: data access regulation and restriction enforcement is provided in Section 8.2.3 thus it has been mapped a gap to be filled.

## 5 Processing capabilities and scale-up

---

### 5.1 Introduction

Among the different features needed by an integrated data exploitation platform, processing capabilities and scale-up represent the core of the services development (first) and service operational provision (later) workflow.

The two phases (development and operational provision) might need different types of IT resources:

- Development phase shall require a mix of traditional and cloud computing infrastructure, combined with elastic resources (e.g., vGPUs), to support activities ranging from prototyping, training models and validation models and operational tests.
- The Operational phase shall require scalable cloud computing infrastructure, exploited with HPC resources to cope up with the requirements to run both traditional and demanding workloads, such as training of AI-based modules, where the execution of the inference step fits more with CPUs/HPC than with GPU, or model refinement and re-training (systematic or on-demand). In the operational mode, the workflows shall be designed in a way to allow “seamless” scale-out at larger domains, e.g., geographic areas or time ranges.

One element to be taken into account is the availability of involved data access and pre-processing tools (see Section 4.2) that, in most of the cases, permit to move some of the processing loads within the data preparation step, namely as close as possible to the data: this allows optimising data extraction and preparation steps, lightening at the same time the needs of the computational modules. It shall be noticed that, in operational phase, the preparation capabilities shall ensure adequate performance to support the scale-up of the processing modules. For this reason, an adequate data flux design shall be implemented throughout the entire processing pipeline.

This section has been filled considering the processing requirements collected from the use cases operators through the form provided in Annex A (tab “Data Processing”), thus the focus has been put on the topics of interest to the project and some arguments have been only mentioned.

### 5.2 Processing capabilities and scale up state of the art

As mentioned in the previous chapter, the EO4EU platform shall require flexible computing platform with modern features to support all the requirements from the use-cases. Distributed compute capabilities combined with the scale-out architecture shall allow EO4EU to position workloads strategically, utilizing computing resources in close-proximity to the data.

By adopting an event-driven and distributed nature, EO4EU platform shall require dynamic scheduling of the infrastructure resources in combination with utilizing the rare resources such as vGPUs or even HPC workloads, allowing efficient use of the platform, model trainings and scheduling AI/ML workflows.

In line with the requirements, several infrastructure capabilities are required for the EO4EU platform, allowing dynamic scheduling of the infrastructure capabilities:

- Allowing self-service resource provisioning via APIs
- Offering higher level services (e.g., Kubernetes as a Service, Platform as a Service, Functions as a Service) in combination with traditional cloud infrastructure (Infrastructure as a Service)
- Elastic resource scheduling (horizontal scaling)
- Close proximity to the EO data sources
- Access to the rare compute resources (e.g., HPC infrastructure, GPUs)

- Modern ways to expose storage resources (e.g., object storage - S3 compatible)
- Providing Open-source tools compatible within the community (upstream compatibility).

Based on the features mentioned above, several cloud infrastructure/environments have been evaluated.

Public Cloud Providers:

- Amazon Web Services (AWS)
- Microsoft Azure
- Google Cloud Platform
- Scaleway

Community Cloud Providers and DIASes:

- WeKEO
- DIASes/C-DAS
- European Weather Cloud/Provided by ECMWF & EUMETSAT
- ADA Cloud/CINECA

Table 8 summarizes the cloud infrastructure platforms based on their capabilities with respect to data processing and proximity, ability to host different types of workloads (e.g., HPC, GPU, traditional) and resource elasticity.

Provider	Self-service provisioning	Higher level services	Elastic Resource	Close-proximity to any EO Data	Access to HPC and GPUs	Object storage	Upstream Compatibility	Public vs Community
<b>Amazon Web Services (AWS)</b>	Offers huge range of services via marketplace and APIs	Offers managed services ranging from IaaS to FaaS	Infinite resources over multiple geo-locations	Hosts some EO data.	HPC: No GPU: Yes	Yes	Mostly proprietary software, risk of vendor lock-in	Public cloud
<b>Microsoft Azure</b>	Offers good range of services via marketplace and APIs	Offers managed services ranging from IaaS to FaaS	Infinite resources over multiple geo-locations	No	HPC: No GPU: Yes	Yes (S3 compatible)	Mostly proprietary software, risk of vendor lock-in	Public cloud
<b>Google Cloud Platform</b>	Offers huge range of services via marketplace and APIs	Offers managed services ranging from IaaS to FaaS	Almost infinite resources over multiple geo-locations	No	HPC: No GPU: Yes	Yes (S3 compatible)	Proprietary software with some open-source offerings, risk of vendor lock-in	Public cloud

<b>Scaleway</b>	Offers highly specialized selected services and APIs	Available with limitations	Mediocre	No	HPC: No GPU: Yes	Yes (S3 compatible)	Proprietary software stack, risk of vendor lock-in	Public cloud
<b>WeKEO</b>	Offers IaaS and some PaaS capabilities and APIs	Extendable via self-managed tools.	Limited within the user quota	Yes	HPC: No GPU: Yes	Yes (S3 compatible)	Open source	Community Cloud
<b>DIASes/C-DAS</b>	IaaS with APIs	Limited	Limited within the user quota	Yes	No	Yes (S3 compatible)	Open source	Community Cloud
<b>European Weather Cloud</b>	IaaS with APIs	Extendable via self-managed tools.	Limited within the user quota	Yes	HPC: No GPU: Yes	Yes (S3 compatible)	Open source	Community Cloud
<b>ADA Cloud - CINECA</b>	IaaS with APIs	Extendable via self-managed tools.	Limited within the user quota	No	Yes	Yes (S3 compatible)	Open source	Community Cloud

**Table 8. Summary of data processing platforms features.**

### 5.3 Requirements for cloud infrastructure

The analysis of the use cases shows the usage of a traditional computational infrastructure is essential, both the development and operational phase. Three out of seven use-cases reported that the given workloads are mostly CPU-driven for the entire development and operational lifecycle. This points out the fact that the bare minimum requirement for the EO4EU platform to provide infrastructure as a service in order to ensure the service continuity.

Another important factor to consider is elastic resource scheduling (horizontal scaling), allowing the EO4EU platform to dynamically allocate resources based on demand. A difference between resource allocations shall be expected between the service development and operational phases, such as a large number of requests to a mobile app (e.g., use-case 1: health), as three out of seven use cases require this capability.

Possibility to allocate computational resources constantly during the operational phase can be easily identified as another requirement requested by all the use cases for the following reasons:

- Larger geographic domain with respect to the development phase
- Larger temporal range with respect to the development phase

The ability to provision and distribute storage resources on-demand is another important factor to consider, especially for the operational phase. Such functionality would allow the efficient and programmable distribution of the data to the end-user, preferably via well-established S3 standard, instead of traditional and insecure, inefficient, and non-programmable ways to distribute data over the network (e.g., FTP).

Chapter 5.3.1 compares the requirements discussed within this chapter with the existing solutions mentioned in the Table 3. Requirements and the comparison for the rare compute resources such as HPC infrastructure, GPU/vGPUs shall be discussed in the Chapter 5.4 and 5.4.1, respectively.

### 5.3.1 Project needs versus existing solutions

Based on the identified requirements and the analysed features of the cloud infrastructure platforms in Table 3, it has been identified that the basic CPU-driven needs of the EO4EU platform, at least for the development phase, can be satisfied by any cloud infrastructure provider. Nevertheless, it shall be noted that "close-proximity" computation to any major EO data holding is a major preference. In this case, the DIASes, WEkEO, EWC, and Amazon Web Services (AWS) have the advantage of being close-proximity to EO data. It shall also be noted that, since none of the platforms allows access to all needed data (see Table 2), there is still a partial gap in the identification of a platform that can optimally satisfy the use case needs (processing close to the data).

For what regards elastic processing, Table 8 shows that, with different modalities, all platforms allow deploying pipelines that can be up-sized and down-sized on the fly once the service requests these functionalities. This applies also to upscaling, but with different cost models, namely:

- Elastic processing is billed per time unit, usually hours
- Upscale processing is billed with long contracts, namely yearly or multi-yearly

Table 9 provides a comparison between OpenStack, VMWare and public cloud providers as existing solutions for processing distribution technologies.

	OpenStack (WeKEO, ADA, EWC and DIASes)	VMWare	Public Cloud Providers (Amazon, Google, Azure)
<b>Scalability</b>	Highly scalable and can manage large complex environments	Primarily designed for on-premises data-center virtualization	Infinite scalability over multiple geo-locations
<b>Hypervisor Support</b>	KVM, Xen, VMware and others	VMWare ESXi	Depends on the provider, mostly in-house developed versions of KVM or Xen
<b>License</b>	Open Source	Proprietary	Proprietary
<b>Deployment Models</b>	Supports private, public and hybrid cloud deployments including edge computing environments	Primarily used for on-premises data center virtualization	Public cloud with some on-premises options. Primarily used to balance capex vs opex.
<b>Community</b>	Large and active community of developers and users. Adopted by NASA, CERN, CINECA and EWC.	Large and active community, but primarily focused on VMware products	Depends on the provider, each provider has a separate community.

<b>Technology Focus</b>	Focuses on open-source cloud computing and managing multiple types of cloud infrastructure resources, including VMs, containers, bare metal servers, network and storage resources.	Primarily focuses on virtualization and managing VMs. Offers some storage virtualization capabilities with separate product (VSAN)	Focuses on proprietary and managed cloud computing resources, including VMs, containers, bare metal servers, network and storage resources as well as several additional auxiliary services (DNS, GIT etc.).
<b>Higher Service Levels (PaaS – FaaS)</b>	Possible with the extensions.	Limited extension possibilities. Mostly relies on other products.	Offers managed services on all service layers.
<b>APIs</b>	Unified and well-established Open-Source REST API(s) and CLI	Offers proprietary REST API(s) and some extensions via PowerShell	Each provider offers separate REST API(s). Interoperability is not an option.
<b>Pricing Model</b>	Depends on the implementation can be billed with long contracts or pay-as-you-go.	License fee per CPU or socket. Each product might have separate prices.	Pay-as-you-go.

**Table 9. Comparison Table for OpenStack vs VMWare.**

From the scalability, operational costs, and managed services perspective, the public cloud providers have the advantage. Amazon Web Services (AWS), a major public cloud provider, also hosts some EO data and offers it as a service to its users. However, it should be noted that from the pricing model, licensing, and interoperability perspectives, utilizing the public clouds for the EO4EU platform has a clear risk of vendor lock-in. Almost all the public cloud providers have some sort of proprietary software backed up by their services and they lack interoperability in between. Therefore, any software developed, for example, with using Amazon Elastic Kubernetes Service (EKS), will work best on the EKS and might face problems on the upstream (open-source) Kubernetes platform.

All the available platforms do not provide access to all needed data (Table 7) and therefore, interoperability between cloud environments is crucial for the EO4EU platform. Interoperability would allow the deployment and operation of the platform in any of the underline infrastructures and would ensure the interconnection and data flow amongst them. Another limitation of the public cloud providers is the lack of traditional HPC Infrastructure. Both Google and Amazon offer HPC-driven flavours to tackle this issue, but this is not very-well adopted yet by the HPC community because of the lack of features. Therefore, an access to the traditional HPC Infrastructure is still needed.

The usage of ADA Cloud from CINECA and WEkEO from EUMETSAT/ECMWF is considered as the most appropriate approach. Both offer OpenStack as the cloud infrastructure technology which ensure the interoperability between the two clouds. ADA Cloud environment also provides direct access to the CINECA HPC resources which is another requirement for the EO4EU platform. They also offer scalability, although within the quota limits, and access to Copernicus data. As a downside, some of the higher-level functionalities (e.g., Kubernetes, Functions) shall be deployed and managed separately, which will increase the support/maintenance effort.

In this respect, the combination of the WEkEO and ADA Cloud platform sets the most convenient cost model for all the EO4EU use cases.

EO4EU platform shall adopt Kubernetes as the building block for providing higher-level services on the PaaS and FaaS layer (e.g., containerized workload orchestration). Kubernetes is the de-facto standard to deploy and operate containerized applications. Although there are several other projects (e.g., Docker Swarm, Apache Mesos, and Hasicorp Nomad) to manage containerized workloads, Kubernetes offers the full realization of the underlying cloud capabilities with the Openstack Cloud Controller



Manager (CCM) and Cluster API plug-in, effectively utilize all the resources (networking, storage and compute) based on events. This allows the EO4EU platform to adopt the event-driven architecture, which can self-heal and adaptively auto-scale via Horizontal Node and Pod Autoscaling.

Via adopting distributed cloud approach, the EO4EU platform shall utilize at least two different Kubernetes clusters on both cloud infrastructure providers, WEkEO and ADA Cloud, to schedule workloads based on the specific needs (e.g., close-proximity to the data). This can introduce complexities in the deployment and the management of the platform, reducing the visibility over the overall architecture and troubleshooting.

There are several "Kubernetes Cluster Managers" available in the open-source community to tackle this issue. A review of the main "Kubernetes Cluster Managers" presented below (Table 10). The review compares Kubernetes Cluster Managers in terms of upstream compatibility, main focus and platform, community support, added value, complexity and maturity level.

Characteristic/Functionality	Rancher Cluster Manager	Red Hat OpenShift Cluster Manager / Open Cluster Manager	Kubespray	Gardener	Kubeadm
<b>Upstream Compatibility / Licensing</b>	Compatible with upstream Kubernetes.  Opensource.	Partial compatibility with upstream Kubernetes.  Offers opensource alternative to commercial product.	Compatible with upstream Kubernetes.  Opensource.	Compatible with upstream Kubernetes.  Opensource.	Developed by Kubernetes community.  Opensource.
<b>Main Focus</b>	Focuses to provide full life-cycle management of Kubernetes, to deploy and run clusters anywhere and on any provider	Focuses on management of the downstream clusters, lacking the cluster deployment functionality.	Focuses on deployment of the downstream clusters, lacking management capabilities.	Focuses on the delivery of Kubernetes, providing abstraction over multiple Kubernetes clusters.	Focuses on deployment Kubernetes clusters, lacking multi-cluster deployment and management capabilities.
<b>Platform</b>	Rancher can be deployed either as a standalone container in a VM or in any Kubernetes via Helm Charts.	Uses Openshift/OKD as the main platform. Can be deployed on upstream Kubernetes with customizations.	Can be deployed on AWS, GCE, Azure, OpenStack, vSphere, Baremetal	Gardener requires running Kubernetes cluster. Can be deployed via Helm Charts.	CLI tool. Can be used to deploy clusters from anywhere.
<b>Community Support</b>	Large and active community of developers and users.	Active community, but primarily focused on Redhat products.	Active community.	Small but active community. Loses attention because of the complexity.	Large and active community of developers and users.
<b>Added Value</b>	Provides centralized cluster deployments, management, logging, monitoring, and alerts.  Provides Rancher GUI and abstract	Provides centralized cluster management, logging, monitoring, and alerts.  Provides Cluster Management GUI.  Provides cloud-specific addons	Provides centralized cluster deployments, management, logging, monitoring, and alerts.  Provides Cluster Management GUI.	Effective way to provide managed Kubernetes Clusters and Kubernetes Engines (e.g., GKE or EKS)  Provides Gardener Dashboard GUI.	Provides building blocks to deploy minimum viable Kubernetes cluster.  By design, it focuses on bootstrapping, not about

	<p>REST APIs to expose resources.</p> <p>Provides cloud-specific addons (both public and private)</p> <p>Provides advanced Role-based access controls, allowing proper isolation between clusters.</p> <p>Provides GitOps tool to standardize the downstream Kubernetes clusters.</p> <p>Provides native Terraform plugins to manage full lifecycle.</p>	<p>(both public and private)</p> <p>Provides Role-based access controls, allowing isolation between clusters.</p> <p>Provides GitOps tool to standardize the downstream Kubernetes clusters.</p>	<p>Provides cloud-specific addons (both public and private)</p> <p>Provides limited Role-based access controls.</p>	<p>Provides cloud-specific addons (both public and private)</p> <p>Provides Gardener CLI</p>	<p>provisioning machines.</p> <p>Likewise, installing various addons like the Kubernetes Dashboard, monitoring solutions, and cloud-specific addons, is not in scope.</p>
<b>Maturity</b>	Mature	Under active development. Lacks proper documentation.	Mature	Mature	Very mature
<b>Complexity</b>	Low	High	Medium	Very High	High

**Table 10. Comparison Table for Kubernetes Cluster Managers.**

Based on the comparison matrix, the usage of Rancher Cluster Manager is considered as the most appropriate approach for the deployment and management of the Kubernetes Clusters over the cloud infrastructure. Rancher Cluster Manager is offering the most mature platform in combination with significant added value (e.g., unified REST API, fine-grained RBAC, and cloud plug-ins) while not introducing considerable complexity. Deploying Rancher Cluster Manager is also significantly easier than its counterparts, making it an ideal platform for EO4EU.

### 5.3.2 Identified gaps per use case

- UC1 (Health)
  - Need of traditional processing
  - Need of elastic processing
  - Need of scale up capabilities
- UC2 (Ocean)
  - Need of traditional processing
  - Need of elastic processing
  - Need of scale up capabilities
- UC3 (Crop)
  - Need of traditional processing
  - Need of scale up capabilities

- UC4 (Forest)
  - Need of traditional processing
  - Need of scale up capabilities
- UC5 (Soil)
  - Need of traditional processing
  - Need of scale up capabilities
- UC6 (Food)
  - Need of traditional processing
  - Need of elastic processing
  - Need of scale up capabilities
- UC7 (Fires)
  - Need of traditional processing
  - Need of scale up capabilities

## 5.4 Specific computational needs (HPC Infrastructure, GPUs and vGPUs)

Among the analysed services, only one (UC6 – Food) already exploits ML tools to correlate and predict the presence of locusts based on weather-climate and surface conditions, while four other expect to include ML/AI modules within their workflow. To this end, one may opt for using traditional CPUs resources possibly with some scalability mechanism. However, ML/AI as well as modelling-based services can be better scaled by using Graphic Processing Units (GPUs) and High-Performance Computing (HPC). For UC6, using GPUs will have a high impact during the model training phase as compared to the service operation (inference). Furthermore, availability of GPU resource might be necessary, on a periodic basis (e.g., yearly) to retrain the ML models with additional volumes of data to improve the model accuracy and performance. Besides UC6, two other use cases (UC2- Ocean and UC3 – crop) will examine whether their workflow is suitable for integrating ML/AI tools. Thus, increasing the demand for using GPU resources.

The need for HPCs is in general related to the use-cases which depend on intensive numerical modelling; two use cases (UC3 – crop and UC5 – soil) will attempt to integrate HPC resources in the service operational scenarios. It is worth nothing that the need to compute and elaborate high volumes of data, sometimes with almost real-time constraints, requires an HPC infrastructure that can support such scenarios. To address the high-speed accessibility of voluminous data, the best solution could be to utilize data storage technologies, such as S3 object storage. Such a facility can also act as a bridge between the operational services on cloud and HPC jobs that rely on common input data.

### 5.4.1 Project needs versus existing solutions

While most of the cloud providers make available GPUs resources on demand/on the fly, the cost models make a big difference. Amongst the analysed platforms (see Table 8), scaleway offers the best conditions for both on-demand and continuous GPU resources provision. As an alternative, one may opt for using HPC infrastructure integrated with cutting edge GPU accelerators. However, this doesn't come easy and utilising HPC for such applications brings with it other challenges: adapting algorithms to efficiently run on HPC requires strong design and implementation efforts and, in most of the cases, the adaptation of the code is not platform-agnostic. However, the issues related to the portability of the ad-hoc codes can be addressed by using container technology that can be deployed by tools like Singularity.

### 5.4.2 Identified gaps per use case

- UC2 (Ocean)
  - Need of GPU resources
- UC3 (Crop)
  - Need of GPU resources
  - Need of HPC resources
- UC4 (Forest)
  - Need of GPU resources
- UC5 (Soil)
  - Need of GPU resources
- UC6 (Food)
  - Need of GPU resources

## 6 Algorithm capabilities

---

### 6.1 Introduction

Traditionally processors devoted to analysis of satellite-based observations have been developed mainly following the physical principles of the interaction between radiation and matter. This heritage comes from the scarce availability of satellite data, that forced scientists and software developers to extract as much information as possible in most of the cases from single time multispectral satellite data. A first change has been introduced in the analysis of hyperspectral sensors, that lead researchers to start using dimensional-reduction methods, among which the Principal Component Analysis (PCA) had the largest impact in EO. Once satellite platforms have started growing in number, multi-sensor and multi-temporal data techniques became more and more popular, but still based on physically-based approaches and with a limited number and volume of images. In the last ten years the availability of satellite data and complementary data sources, together with the contemporary evolution expansion of machine learning and artificial intelligence applications and hardware resources, has introduced a new line of processors that have now a strong importance in the EO data processing pipeline

The EO4EU platform, aiming and becoming a reference environment for multi-source data exploitation, shall consider both traditional as well as new data processing techniques, providing to its users' libraries and software management tools that will permit not only the development of one or the other approaches, but that will enable exploiting the most from each methodology.

### 6.2 State of the art on algorithms capabilities

The following chapter elaborates on the development approaches of algorithms relevant for the domain of Earth Observation.

#### 6.2.1 *Waterfall algorithm development*

Algorithms have come a long way in the domain of Earth Observation data processing. They are capable of processing vast amounts of data from various sources such as satellite imagery, aerial photography and ground-based observations to extract meaningful information and insights. Algorithms can perform image classification, object detection, change detection and various other analyses to support decision making in various fields including agriculture, forestry, urban planning and environmental monitoring. They can accurately identify patterns, classify land cover and use, detect and monitor natural disasters, and even track and predict the spread of invasive species. With the advancements in machine learning and deep learning, algorithms are becoming more sophisticated, providing more accurate results with lower processing times. Overall, algorithms have become an essential tool in the processing and analysis of Earth Observation data, providing valuable insights into our planet and its changes.

Waterfall algorithm development is a linear sequential model for software development. This model involves dividing the software development process into several distinct phases, including requirements gathering, design, implementation, testing, and deployment. Each phase is completed before moving on to the next phase, hence the name "Waterfall".

In the domain of Earth Observation, the Waterfall model is relevant as it provides a structured approach to developing algorithms for processing large amounts of complex and diverse data. For example, in the requirements gathering phase, the project team would gather and define the requirements for the algorithm, such as the desired outcomes and the type of data to be processed. In the design phase, the team would design the architecture of the algorithm and the steps to be taken to achieve the desired outcomes. In the implementation phase, the algorithm would be coded, tested,

and validated to ensure its accuracy and effectiveness. The testing phase would involve various types of tests, such as unit testing, integration testing, and system testing, to ensure the algorithm meets the requirements. Finally, in the deployment phase, the algorithm would be integrated into the Earth Observation system, ready for use. Overall, the Waterfall model provides a systematic approach to developing algorithms for the domain of Earth Observation, helping to ensure the accuracy, effectiveness, and reliability of the algorithms.

There are several tools that can be used to manage the Waterfall algorithm development in the Earth Observation domain. These tools can help to streamline the Waterfall algorithm development process, ensuring that the algorithms are developed efficiently and effectively in the Earth Observation domain. Some of these include:

- **Project Management Software:** Common project management software can be used to keep track of project tasks, deadlines, and progress, ensuring that each phase of the Waterfall model is completed in a timely manner.
- **Source Control Management Systems:** Source control management systems, such as Git and SVN, can be used to manage and track the changes made to the codebase, making it easier to revert to previous versions if necessary.
- **Integrated Development Environments (IDEs):** IDEs, such as PyCharm, Visual Studio, and Eclipse, provide a comprehensive environment for coding, testing, and debugging algorithms.
- **Cloud Computing Platforms:** Cloud computing platforms, such as Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform, can be used to store, process, and analyse large amounts of Earth Observation data (see Section 5 above).
- **Automated Testing Tools:** Automated testing tools, such as Selenium, Appium, and TestNG, can be used to perform various types of tests on the algorithms, helping to identify and fix any issues before deployment.

### 6.2.2 Fusion

In the last decades, Earth Observation (EO) data has offered the opportunity to monitor and model the processes on the surface and their interaction with atmosphere. These processes were further enhanced with the use of new observations and data measured coming from a plethora of different sources measuring states and physical variables at unprecedented spatial, spectral, and temporal resolutions. These sources can be mounted on satellites, airplanes and drones but also in-situ observations (increasingly from autonomous sensors) at, and below, the surface and in the atmosphere.

Therefore, the use of information fusion of heterogeneous data, both EO and sensors 'measurements coming from different sources, is essential to obtain robust models with physical coherence and high accuracy able to describe different ecosystem' processes. A variety of information fusion schemes have been proposed in the context of EO. Broadly speaking, information fusion is concerned on the multisource data combination and support decision making. Each fusion method is designed for a specific problem, so it is challenging, if not impossible, to define a full taxonomical overview of the field. The main building blocks, however, have to do with the exploitation of i) disparate inputs, ii) data (pre)processing approach, iii) fusion mechanism, and iv) outputs post-processing. This research was based on the spatio-temporal functionality of fusion applied on EO data.

A generic presentation of fusion focusing on information of EO applied in coarse grain and broad description of computational models applied on EO domain can be found in (Wald, 2000), (Cherkassky et al., 2006) and (Kibirige et al., 2020). (Torabzadeh et al., 2014) present a fusion on spectroscopy

images and laser systems for forecast ecosystem characterization. In (Chen et al., 2015) an analysis of state-of-the art spatio-temporal fusion methods based on remote sensors is presents. Multi-model classification techniques for remote sensing fusion are shown in (Gómez-Chova et al., 2015) and (Schmitt et al., 2016). Image fusion algorithms are described in (Ghassemian, 2016) and (Garzelli, 2016) for remote sensing. In (Yokoya et al., 2016) and (Ghamisi et al., 2017), the main methods and algorithms for hyperspectral and multi- spectral data fusion in remote sensing have been reviewed, whereas in (Ghamisi et al., 2019) the basis, state-of-the-art and challenges of multisource and multi-temporal data fusion algorithms are discussed. At classification algorithmic area (Alajlan et al., 2012) presents a decision level method which combines SVM and fuzzy C-means clustering for fusing hyperspectral images information is proposed. In this approach, the SVM is used to generate a spectral- based classification map, whereas the fuzzy C-means is used to provide an ensemble of clustering maps. A feature extraction fusion of hyperspectral, light detection and Lidar images is presented in (Ghamisi, Höfle, Zhu, 2017) in order to enhance the classification process of hyperspectral data processing. The features derived from two sources are fused via either feature stacking or graph-based feature fusion, and the fused features are fed to a deep CNN with logistic regression to produce the final classification map. The fusion of hyperspectral data, LiDAR and other data sources with CNN models is studied in (Xu et al., 2018).

Domain area	Machine Learning Algorithms
<b>General Review on EO data</b>	<b>Neural, Fuzzy and Evolutionary Computation</b> (Wald, 2000), (Cherkassky et al., 2006) [16]
<b>Reviews on applications of remote sensing that focus on different data/level fusion</b>	<b>Neural Networks</b> (Gómez-Chova et al., 2015), (Ghassemian, 2016), (Ghamisi, Höfle, Zhu, 2017) <b>Extreme Learning Machines</b> (Ghamisi et al., 2017) <b>Support Vector Machines</b> (Gómez-Chova et al., 2015), (Ghassemian, 2016), (Yokoya et al., 2016), (Ghamisi et al., 2019) <b>Deep learning</b> (Gómez-Chova et al., 2015), (Ghassemian, 2016), (Ghamisi et al., 2017), (Ghamisi et al., 2019) <b>Fuzzy C means Clustering</b> (Alajlan et al., 2012) <b>Convolutional Neural Networks</b> (Ghamisi et al., 2017), (Ghamisi, Höfle, Zhu, 2017) [47] <b>Unsupervised cooperative sparse auto-encoder method</b> (Xu et al., 2018)

### 6.2.3 Self-supervised annotation-efficient learning

In recent years, contrastive learning has rapidly emerged as an essential element of the machine learning practitioner’s toolkit, mainly due to annotation-efficiency benefits in low data regimes. In essence, contrastive learning is a machine learning technique that can learn the general features of a collection of data points without any annotations or labels. It works by telling the model about similarities and differences between points in the dataset, while not giving any hint to the model as to what characterizes each point in isolation (e.g., to what category such a point might belong). The model learns features about a data point only relatively to the other points (and augmentations thereof, as we will see below).

Similarly, to how the human brain can learn high-level, general features simply by trying to recognize what makes two scenes different and similar, an AI model trained on earth observations by contrastive learning can extract general features if it is trained to tell how different or similar earth scenes are. For example, by “contrasting” earth observations containing healthy forests with ones containing burned areas, the model has the means to extract and learn what features characterize either types

of scenes, while also learning (providing enough data points are fed to the model) their common features.

Importantly, such a contrastive learner extracts general features simply by contrasting datum pairs; it does not see any annotations or labels like a model trained in a supervised way (e.g., to solve a classification task) would. Since the contrastive model is trained in a completely unsupervised manner, one could say that it learns “by itself” without help from the practitioner. As such, it is “self”-supervised.

The learning methods leveraging contrastive learning that have emerged as the go-to staples in recent years have been SimCLR from the Google Brain team —the original version is SimCLRv1 (Chen et al., 2020-1), followed by a subsequent version, SimCLRv2, whose pipeline has been significantly scaled up, described in (Chen et al., 2020-2)— and Momentum Contrast (MoCo) (He et al., 2020) from Meta (still called Facebook at the time of release). Following the steps of the Google Brain team, the authors of MoCo also proposed an improved version of their technique which they analogously named MoCo-v2 (Chen et al., 2020-3). The latter, among other minor additions, notably improves the data augmentation, and adds a projection head similarly to SimCLR’s second iteration.

Standing as a state-of-the-art contrastive learning technique, the data processing and learning mechanism pipeline of SimCLRv2 can be described in three stages through which every input image is piped:

1. Transform each input image in the dataset into a pair of augmented images. Each augmented positive pair is to be understood by the model as depicting similar scenes and objects, since the two items of the augmented positive pair are in effect different versions (alternative views) of the same image. To augment an image, the latter is fed through a stack of image transformations carefully designed by the practitioner to allow the contrastive model to grasp the relevant features for the task as hand while remaining invariant to features that are irrelevant for the said task. For example, to allow the learner to understand the concept of burned area, allowing the stack of transforms to involve a grayscale filter is probably something to avoid as the model would otherwise be encouraged (by the practitioner) to think that colours are an irrelevant feature for the unsupervised task of contrastive learning. As such, designing stacks of stochastic image transforms must carefully tuned for each type of data and task. In terms of inputs for the unsupervised contrastive task, the augmented pair for each input image (likely different augmentations since at least one transform in the stack is stochastic in nature) form a positive pair, while negative pairs are obtained by creating pairings between either of the two augmented images and every other image in our mini-batch. The mini-batch size (set as a hyper-parameter of the machine learning task) therefore plays a significantly greater role: it determines the ratio of positive and negative pairs that are available to the contrastive model at a given step. Imbalances in positive-negative pairings naturally cause numerous issues in training dynamics.
2. Feed the pairs to a deep neural network backbone (each image of a pair goes into the same feature extractor modeled with the said neural network) and obtain a (feature) vector representation of each image as a result of the extraction. The “ResNet” architecture (He et al., 2015) is a prime candidate.
3. Minimize the contrastive loss. It involves both positive and negative pairs, and its optimization pushes the contrastive model towards returning similar representations for similar images (positive pairs) and different representation for different images (negative pairs). Through the lens of metric learning, Sohn introduced the “NT-Xent” loss, or Normalized Temperature-scaled Cross Entropy Loss, which is a prime candidate for the loss function used to solve the contrastive learning task (Sohn, 2016). In addition, it is possible to fend off and counteract the



imbalances in positive-negative pairings by applying a data-dependent scaling rule at the level of the loss function as it is being optimized.

By being trained in such a way, a contrastive model learns vector representations that enable it to distinguish between images without ever being told what those images are or contain. This ability is particularly appealing when labels are scarce and data points are bare, without any annotation, which is a situation often met when labels are expensive or technologically difficult to obtain for the data points. When facing such a scarcity, the practitioner might want to turn to semi-supervised learning, where one uses both labeled and unlabeled data, in a setting sitting right in-between supervised and unsupervised learning.

Due to its ability to capture vector-embedded semantics from unannotated data, self-supervised learning (and a fortiori contrastive learning) naturally shines in semi-supervised learning: the label-scarcity-facing practitioner can use a self-supervised learning approach as a pre-processing step. First, the model learns general features from (a priori abundant) unlabeled data by attempting to solve a self-supervised task. Then, the practitioner fine-tunes the parameters of the “pre-trained” neural network model by subsequently trying to solve a (supervised) image classification task using the limited labeled data that is available. It has been demonstrated in the literature that such a semi-supervised approach is very label-efficient. Self-supervised learning enables the practitioner to leverage large amounts of unlabeled data to learn general features that can then be refined for a supervised task for which labels are scarce. It is also not uncommon for this two-stage label-efficient approach to surpass purely supervised methods.

Despite the fact that the appellation of self-supervised is recent, the use of contrastive learning in remote sensing is neither new nor has seen limited adoption. On the contrary, it has been widely used throughout the field (Zhang et al., 2021; Yang et al., 2020; Rao et al., 2022; Jing et al., 2021; Zhang et al., 2020; Geng et al., 2022), and its origins can be traced back to (Cheng et al., 2018), as outlined in the recent review paper on self-supervised learning in remote sensing from Wang et al. (Wang et al., 2022). In this original source (Cheng et al., 2018), the authors regularized the features of a ConvNet trained for remote sensing scene classification, where the regularization term constraining the features was based on the contrastive loss discussed earlier in this section. Albeit technically using a contrastive cost as a regularization term to shape the intermediary representation vectors towards more general yet salient features, (Cheng et al., 2018) is not per se proposing a self-supervised learning technique: it proposes a supervised learning technique assisted by an unsupervised, contrastive constraint. By contrast, Tile2Vec (Jean et al., 2019) seems to be the first pure self-supervised work and leverages contrastive learning to learn a salient image representation for remote sensing image. As the name hints at, Tile2Vec is inspired by Word2Vec.

#### 6.2.4 Learning-based compression

On a typical day, the European Centre for Medium-Range Weather Forecasts (ECMWF) produces approximately 230 TB of data (Klöwer et al., 2021) and most of the data are archived on magnetic tapes in its cold storage facilities. One can naturally expect that most other weather-focused HPC centers across the planet operate by following similar guidelines, as they perform new climate simulations and accumulate new data day after day. In effect, the daily operational cost of such facilities is strongly tied to how much is stored every time a new simulation is carried out. It has been estimated that the data production would quadruple within the next decade, as the forecast models become increasingly able to handle greater and greater spatial resolutions. In order to cut down those operational storage costs (but also transmission and sharing costs), one can turn towards data compression. Codecs (compression algorithms) have indeed become an essential part of the digital era, as the number of data sources continue to creep up over the years.

When it comes to climate data (or more generally data that can be categorized under the umbrella of remote sensing), go-to compressors (high performance, high efficiency, low complexity, easy to

implement for a given piece of hardware) are only seldomly exploiting the high correlation that exist between the various geophysical and geochemical variables. The difficulty lies in the fact that these correlations appear in all of the dimensions of these variables, and current multi-dimensional compressors are not yet up to the task of exploiting these efficiently. (Klöwer et al., 2021) notably tackles the exploitation of the multidimensional correlation of climate data in order for the resulting compression to exhibit higher levels of compression efficiency. Such ingenuity and engineering feats aim to strike the best balance of between compression size reduction, execution speed, and reconstruction precision.

Lossy compression (e.g. JPEG) can reach higher compression factors (better file size reduction) for smaller files but at the cost of introducing rounding errors. In some critical low-latency scenarios (e.g. transmission of earth observation), the trade-off might be such that the practitioner opt for the use of lossy compression to ensure low latency of the transmission. Besides, since the data being compressed also comes with a certain uncertainty (e.g. sensor), the rounding errors that are caused by a lossy codec do not necessarily cause a loss of real information; the rounding errors might also get rid of information that the practitioner is happy to see removed from the pre-compression image. In short, a reduction in compression precision does not always mean a loss of real (here to be understood as “salient”, “useful”) information. Finally, achieving higher rates comes at the cost of having to use codecs that are far more complex, and this complexity very often translates to decreased efficiency (slower speeds).

To explore the space of trade-off there is to strike between compression rates and speeds and increase the flexibility of the transformations performed by the various codecs in existence, one can turn to the relatively new field of “learning to compress”. The advent of deep learning has given rise to the emergence of the new field of “neural compression”, that tackles the design of neural-networks-based codecs. The field of codec design has arrived to a point where neural compression occupies a central place in the compression landscape, and neural networks have become an essential piece of the codec developer’s toolkit. The role played by the neural networks in these newly developed codecs can vary widely from one neural compression approach to another. Neural networks can be involved as a substitute to one or several pieces of pre-existing legacy codecs (Gueguen et al., 2018), often because these pieces are complex and therefore tedious and expensive to implement for the hardware at hand. In these scenarios, using a neural network as a function approximator for the pieces difficult to deal with can cut down those operational compression costs overall. (Li et al., 2021) notably uses a neural network to transform the wavelet domain from the large scale to the small scale. This substitution allows the other components of their post-transform compression pipeline to remain intact while the neural network replaces the inefficient piece, reportedly making the resulting neural-augmented compression pipeline efficient for remote sensing compression.

Some authors like (Theis et al., 2017) have taken this approach to the extreme by leaving behind the traditional way to build codecs and improving them piece by piece with neural-based sub-components. Instead, they have created codecs modeled and trained entirely as one neural network architecture, even learning a neural model for the standard compression sub-routines, such as quantization first in (Agustsson et al., 2017), and later entropy coding in (Balle et al., 2018; Habibian et al., 2019; Mentzer et al., 2018; Minnen et al., 2018). Entropy coders are an essential piece in codec design because, as Claude Shannon wrote in his seminal paper of information theory, “A Mathematical Theory of Communication” in 1948, the length of a message representing a given piece of datum is proportional to the entropy of this piece of datum. Being able to lay a hand of the said entropy would allow an entropy coder, in theory, to design the shortest possible message to represent the information to encode. This desideratum is however unattainable a priori because we do not know the entropy of the data, since it is by definition a direct function of the probability distribution of the data, which we do not know. This realization is the crux of why deep generative modeling has played such a prominent role in neural compression in recent year: by leveraging learning techniques able to model the distribution of the data such as the ones in the deep generative modeling landscape, is

effectively modeling the entropy of the data by proxy, which is extremely valuable for efficient entropy coding (Balle et al., 2018; Habibian et al., 2019; Mentzer et al., 2018; Minnen et al., 2018). Modeling the data distribution with deep generative models have also enabled meaningful progress in reconstruction (Mentzer et al., 2020) and inference (Yang et al., 2020). Another method that has seen widespread adoption in the domain of neural adoption is the vector-quantized variational auto-encoder method, or VQ-VAE (van den Oord et al., 2017). In essence, the latter learns a codebook of vector embeddings, and each vector in the latent space is mapped to its closest code in the learned dictionary of codes before being fed to the decoder. Importantly, even though neural networks can provide a convenient alternative to complex components in a compression pipeline, they still need to be trained. As an auto-encoder by nature, the VQ-VAE technique trains its model with a reconstruction loss. The method has been recently improved to work on with even larger models and volumes of data.

### 6.3 Traditional (waterfall) analysis

The performed analysis reports a traditional waterfall scenario for six of the seven use cases: this includes both service provision logic (waterfall analysis) and data processing techniques. Moreover, five out of seven use cases already foresee a fully automatic processing workflow, that makes them at an advanced stage for their operationalization and operation, but that brings also some difficulty in introducing innovations in the data processing pipeline.

#### 6.3.1 Project needs versus existing solutions

The analysed processing pipelines show mature algorithmic approaches and well-established workflows. The tools identified in the SOTA (see Section 6.2) shall be used to move from the prototype/ re-operational status of the current services to the operational status. At the time of the requirements collection, two services already feature operational pipelines deployed on the final environment (TRL=9), three of them can be considered pre-operational featuring consolidated pipeline not yet operated (TRL=7), while three services are still at the prototype development phase (TRL=4) and require more work on the consolidation (first) and on the final deployment and operation (second). The tools identified in the SOTA will be applied to speed up and homogenize the technology readiness level of the services bringing them to (or close to) the operation by the end of the project, and establishing a baseline that can be adopted by the upcoming platform users.

#### 6.3.2 Identified gaps per use case

- UC1 (Health)
  - Need of operationalization
- UC3 (Crop)
  - Need of workflow implementation / finalization
  - Need of operationalization
- UC4 (Forest)
  - Need of operationalization
- UC5 (Soil)
  - Need of workflow implementation / finalization
  - Need of operationalization
- UC6 (Food)

- Need of workflow implementation / finalization
- Need of operationalization

## 6.4 ML/AI based approaches

The analysis of the use cases reported a scenario not yet mature in the use of ML/AI tools within data processing workflows. Only one use case (UC6, Food) already makes use of AI-based solutions to predict the presence of locusts based on meteo-climate and soil conditions information. Four other use cases (UC2, Ocean and UC3, Crop) envisage potentialities in the integration of these methodologies in the current workflow, while the other four use cases do not see the need.

Considering that this status can be generalised, the gap in the integration of ML/AI techniques within EO-based processing workflows is not only technological but mainly a knowledge gap: there is a need to show/demonstrate and, in general teach how ML/AI based modules work in a way that the algorithm/pipeline/workflow developers can take informed decision on which approach to follow to implement a specific step or a full process.

### 6.4.1 Project needs versus existing solutions

Besides UC6, the existing technological approaches need to be analysed against the use cases needs to check whether there are advantages for the specific use cases in adopting ML/AI techniques within their business logic. UC2 and UC3 already started the process, while the other use cases need to go through it to identify whether ML/AI based implementations can improve in quality/performance/stability of the existing business process.

Section 6.2.2 demonstrates that different fusion techniques (needed by most of the use cases but UC2) are currently available, thus it is necessary to analyse, on a case by case basis, which one is the most adapted for each service. Self-supervised annotation efficient Learning techniques (Section 6.2.3) can support classification as well as learning modules, most likely needed by UC1, UC2 and possibility UC4 and UC6, while learning-base compression (Section 6.2.4) can support focusing and optimizing existing pipelines e.g. for UC3 and UC7. The process of identifying the most adequate ML/AI technique, its adoption, implementation and effectiveness evaluation can become a common practice within the Earth Observation based business workflows.

### 6.4.2 Identified gaps per use case

- UC1 (Health)
  - Workflow analysis
  - Integration (evaluation, implementation, assessment) of ML/AI steps
- UC2 (Ocean)
  - Integration (evaluation, implementation, assessment) of ML/AI steps
- UC3 (Crop)
  - Integration (evaluation, implementation, assessment) of ML/AI steps
- UC4 (Forest)
  - Workflow analysis
  - Integration (evaluation, implementation, assessment) of ML/AI steps
- UC5 (Soil)

- Workflow analysis
- Integration (evaluation, implementation, assessment) of ML/AI steps
- UC6 (Food)
  - Critical revision of the current methodology
- UC7 (Fires)
  - Workflow analysis
  - Integration (evaluation, implementation, assessment) of ML/AI steps

## 7 Results presentation / communication / delivery

---

### 7.1 Introduction

Exploring, visualizing, and analysing data is a core task for experts in numerous applications. Data visualization provides intuitive ways for the user to interactively explore and analyse data, enabling them to effectively identify interesting patterns, infer correlations and causalities, and support sense-making activities. Based on all the findings described in the above chapters, the presentation of the different data is a very important aspect in every project that is dealing with different data types and sources.

In the next sections a short description of the state of the art on data visualization tools, types of visualization tools of the geospatial data and findings on the needs and gaps on this topic will be presented.

### 7.2 Results presentation/communication/delivery state of the art

According to the definition, data visualization is the presentation of data in a pictorial or graphical format, and a data visualization tool is the software that generates this presentation. Data visualization provides users with intuitive means to interactively explore and analyse data, enabling them to effectively identify interesting patterns, infer correlations and causalities and supports sense-making activities.

Exploring, visualizing, and analysing data is a core task for data scientists and analysts in numerous applications, especially when they are dealing with a great number of datasets that are dynamic, noisy and heterogeneous.

The easiest and straightforward way to visualize and present geospatial data is through maps. Static maps are traditional maps that are created using GIS software or web mapping platforms and are displayed in a non-interactive manner. Static maps can be used to visualize geospatial data in a basic way and can be useful for communication or education purposes. Web maps including interactive maps are a popular way to present and communicate geospatial data, as they allow users to interact with and explore the data through panning, zooming, and querying. Web maps can be created using a variety of software tools and frameworks and usually allow users to visualize multiple datasets. Furthermore, interactive web maps offer the possibility to incorporate also non-map-based elements of data presentation.

Non-map-based ways of presenting geospatial data including charts and tables.

- Charts and diagrams can be used to represent geospatial data in a way that is easy to understand and interpret. Common types of charts and diagrams used for geospatial data include scatterplots, line graphs, bar charts, and pie charts.
- Geospatial data can also be presented in tabular form, using tables or spreadsheets to display the data in rows and columns. This can be useful for presenting detailed data or for comparing values across different locations or periods.

#### 7.2.1 GUI state-of-the-art

Graphical user interfaces are constantly evolving, new development and functionalities are being introduced, following is a general overview of the current advancements:

1. **Design Trends:** Design is never static. It is constantly evolving, some changes could be drastic, others can result from slight modification. Lately the design has shifted from the use of skeuomorphic elements to minimalism and flat design. Skeuomorphic elements were needed before to familiarize users with graphical elements and make it look more familiar and closer to real components. Today design trends put more emphasis on clean and simple interfaces, more visually appealing and easy to use, especially with the ever-increasing penetration of smart phone with smaller screens.
2. **Interactivity and animations:** the focus on transforming the user interfaces into interactive one has increased in the last few years, this is due to the need to engage users, and respond to their needs. Interactivity include hover effects and animations that provide feedback based on the action done by the user.
3. **Natural language processing:** Natural language processing (NLP) is increasingly being used to enhance how people interact with GUIs. For instance, some user interfaces let users enter requests using natural language rather than a series of menu options and clicks. Interfaces may be simpler to use as a result, especially for non-technical users.
4. **Voice interfaces:** Voice interfaces, which let users communicate with their gadgets through voice commands, are becoming more widespread. This can make it simpler to carry out specific operations without having to touch the device, such as making a phone call or playing a music.
5. **Virtual and augmented reality:** With the increase in the number of apps and devices utilizing virtual and augmented reality, GUI design is becoming increasingly common. This can make it possible for experiences to be more immersive and engaging and can also make it simpler to engage with digital data in novel and creative ways.
6. **Accessibility:** Making GUI useable for people with disabilities is becoming a bigger priority in GUI design as a result of accessibility. In addition to an emphasis on utilizing clear and concise language, high-contrast color schemes, and other design aspects that make it simpler for people with disabilities to utilize the interface, this can include support for screen readers, keyboard navigation, and other accessibility features.

In general, new developments and trends are continually arising, changing the state of the art of graphical user interfaces. While using emerging technologies like NLP and VR/AR to produce creative and immersive experiences, the emphasis is on developing interfaces that are aesthetically pleasing, simple to use, and available to a variety of people. A graphical user interface for earth observation data, shall enable users to interact with and evaluate substantial volumes of satellite data. With the use of such a GUI, a variety of users, including scientists, decision-makers, and the general public, would be able to access, view, and analyze earth observation data in an easy and straightforward manner.

Typical components and features of a GUI for earth observation data include the following:

1. **Data visualization:** GUI for earth observation data must include data visualization since it enables users to spot patterns, trends, and anomalies in the data and acquire understanding of the planet's ecosystem. The following are some essential attributes and capabilities for data visualization:
  - a. **Maps:** Interactive maps that let users visualize satellite data on a geographic scale are often part of the GUI. It may also contain tools for displaying data in various ways, such as heat maps, contour maps, and shaded relief maps, as well as for studying data at various sizes, from global views to more granular regional or local views.

- b. Graphs and charts: a tool for displaying data in the form of graphs and charts, such as line graphs, bar graphs, and scatter plots, so users can see patterns and trends in the data over time.
- c. Image visualizations: a tool for visualizing satellite data that is presented as images, such as false color composites, histograms, and color balance adjustments. These tools would enable users to see details and patterns in the data that might not be apparent in a typical grayscale or RGB image.
- d. Interactive visuals: The GUI should have interactive visualizations that let users manipulate how the data is shown and how it is explored. These interactive visualizations can take the form of sliders, buttons, and drop-down menus.
- e. Data layering: To obtain a more comprehensive image of the earth's environment, the GUI should also include capabilities for superimposing other statistics sources, such as adding weather predictions, population data, and land use data.
- f. Export and sharing: Users should be able to export and share the visualizations they produce using the GUI, whether by storing them as photos or interactive web pages or by sending them to others through email or social media.

In conclusion, data visualization is an essential component of any GUI for earth observation data, enabling users to spot patterns, trends, and anomalies in the data and acquire understanding of the earth's ecosystem. In addition to providing collaboration, data integration, and advanced data analysis, the objective is to provide a user-friendly and straightforward approach to access, view, and analyze satellite data.

2. Data Exploration: it enables users to interact with and evaluate huge volumes of satellite data, data exploration is a crucial component of any GUI for earth observation data. The following are some essential attributes and capabilities for data exploration in a GUI for earth observation data:
  - a. Data filtering: The GUI should include options for data filtering, such as choosing data based on date range, location of the world, or data type, allowing users to concentrate on certain areas of interest and minimizing the quantity of data they need to work with.
  - b. Data selection: To enable users to concentrate on certain areas of interest and conduct in-depth analysis of the data, the GUI should also include tools for selecting particular data points or regions, such as utilizing a brush or lasso tool to choose a particular area on a map.
  - c. Data comparison: The GUI should include capabilities for contrasting various datasets, like side-by-side comparisons of various photos or time series data, enabling users to detect differences and similarities in the data as well as to spot patterns and trends.
  - d. Data aggregation: To help users comprehend the data and spot outliers and anomalies, the GUI should include capabilities for aggregating the data, such as calculating statistics like the mean, median, and standard deviation, or constructing histograms or frequency distributions.
  - e. Data exploration history: The GUI should include a feature that allows users to keep track of their data exploration history, such as a list of the data filtering, selection, and comparison operations they've carried out. This will make it simple for them to go back and contrast their results with earlier steps.

In conclusion, data exploration, which enables users to interact with and understand huge volumes of satellite data, is a crucial component of any GUI for earth observation data. The objective is to provide robust and adaptable tools for data exploration, enabling users to filter, select, compare and



aggregate the data as well as maintain account of the history of the data exploration, making it simpler to go back and compare findings from earlier phases.

3. Data analysis: Any GUI for earth observation data must include data analysis since it enables users to understand the planet's environment and to make wise judgments. Here are some essential attributes and capabilities for data processing in a graphical user interface for earth observation data:
  - a. Statistical analysis: To enable users to find connections and patterns in the data, the GUI should include capabilities for doing statistical analysis on the data, such as regression analysis, correlation analysis, and hypothesis testing.
  - b. Data modelling: To enable users to create predictions and projections based on the data, the GUI should also include tools for creating models utilizing the data, such as machine learning methods.
  - c. Spatial analysis: The GUI should include tools for performing spatial analysis on the data, including building and analyzing spatial data layers, building and visualizing spatial distributions, and building and analyzing spatial models. This will enable users to learn more about the environment of the earth.
  - d. Image analysis: The GUI should offer tools for carrying out image analysis, such as object detection, image segmentation, and image classification, for satellite data that is in the form of images. This will enable users to gain insights into the earth's environment and to spot patterns and trends in the data.

To sum up, data analysis is an essential component of any GUI for earth observation data, enabling users to understand the planet's environment and to take wise decisions. With the use of robust and adaptable tools, users will be able to undertake statistical analysis, data modeling, geographical analysis, picture analysis, data integration, data export, and data sharing, which will make it simpler to obtain insights into the earth's environment and to work with others.

4. Data integration: it enables users to aggregate and evaluate data from many sources to get a more comprehensive view of the earth's environment. The following are some essential attributes and capabilities for data integration in a GUI for earth observation data:
  - a. Data import: To enable users to integrate and evaluate the data, the GUI should include capabilities for importing data from various sources, including satellite data, weather predictions, demographic data, and land use data.
  - b. Data fusion: In order to provide users a more comprehensive understanding of the earth's ecology, the GUI should include capabilities for merging various data sources, such as satellite data with weather predictions, demographic data, and land use data.
  - c. Data normalization: The GUI should include capabilities for normalizing data, such as translating data from many scales or units to a single scale or unit, to make it easier for users to compare and interpret the data.
  - d. Data cleaning: To enable users to work with high-quality data, the GUI should include capabilities for cleaning the data, such as deleting missing or incorrect data, and correcting mistakes and outliers.
  - e. Data enrichment: To provide consumers a more comprehensive understanding of the earth's ecosystem, the GUI should include options for enhancing the data, such as adding derived data or other data sources.

Thus, data integration is a crucial component of any GUI for earth observation data, enabling users to aggregate and evaluate various data sources to obtain a more comprehensive view of the earth's

ecosystem. To make it simpler for users to get insights about the earth's environment, it is intended to provide robust and adaptable tools for data integration. These tools will enable users to import, fuse, normalize, clean, and enhance the data.





5. Accessibility: Any GUI, including one for earth observation data, must be accessible in order for individuals with impairments to access and utilize the information. Here are some essential attributes and capabilities for an accessible user interface for earth observation data:
  - a. Screen reader support: enabling users who are blind or visually challenged to access and utilize the data through voice output.
  - b. Keyboard navigation: Users who are unable to use a mouse should still be able to access and utilize the data since the GUI should be completely keyboard navigable.
  - c. High contrast mode: The GUI should have a high contrast option so that those with limited vision or color blindness may access and utilize the data.
  - d. Text resizing: The GUI should enable text scaling so that those with limited eyesight may access and utilize the information.
  - e. Alternative text for images: The GUI needs to have alternative text for images so that visually challenged users may grasp the images' information.
  - f. Video captions: The GUI should include video captions so that users who are hard of hearing or deaf may understand the videos' content.

## 7.2.2 Workflow tools and Domain Specific Language (DSL)

One aspect of user interface is the possibility of interacting with processing tools and workflows. Following subsections provide a state of the art of the available workflow tools and Domain Specific Language (DSL) technologies.

### 7.2.2.1 Workflow tools

The field of data engineering and machine learning is constantly evolving, and new open-source tools for managing and orchestrating data workflows are emerging all the time. Here are some state-of-the-art open-source tools that are popular in the industry:

-  Apache Airflow  
 Airflow is a popular platform for programmatically authoring, scheduling, and monitoring data workflows. It provides a web-based user interface that allows to monitoring and troubleshoot workflows, and a set of operators for common data processing tasks.
-  Kubeflow  
 Kubeflow is a cloud-native platform for running and managing machine learning workflows on Kubernetes. It provides tools for training, deploying and serving models, and monitoring and debugging workflows.
-  Luigi  
 Luigi is a Python-based workflow management system that provides a high-level API for defining tasks and dependencies. It supports both batch and streaming data processing workflows.
-  Prefect  
 Prefect is a Python-based workflow management system that allows you to define and orchestrate complex data workflows. It provides a modern, user-friendly interface, and supports both local and distributed execution.

These open-source tools are widely used in the industry and are actively maintained and developed by their respective communities. They offer a range of features for managing and orchestrating data workflows, and can be tailored to meet the specific needs of your data.

#### 7.2.2.2 DSL tools

Apart from the data flow and orchestration tools, Domain Specific Language technologies were also evaluated. Compared to the previously mentioned tools a DSL being developed specifically for the needs of EO4EU would offer extended capabilities and features. A DSL would be developed to overtake the role of a validation and orchestration tool to handle the incoming data, giving users a better experience allowing them to directly interact with the metadata and their flow, but also allowing for greater flexibility.

For the implementation of DSL several platforms and frameworks are available providing various utilities and features. The three more prominent platforms will be analysed below:

- **Eclipse Xtext** is an open-source framework for developing DSLs and their associated tooling. It provides a set of tools and libraries for generating language-specific editors, parsers, compilers, and code generators, as well as a rich set of features for managing DSL projects and codebases. One of the main advantages of Xtext is that it provides a high level of abstraction for DSL development. This allows developers to focus on the semantics of the language they are creating, rather than the implementation details of the tools that support the language. This makes it easier to develop high-quality, error-free code in less time, while reducing the risk of errors and inconsistencies.
- **textX** is an open-source meta-language tool, inspired by Eclipse Xtext that allows developers to define and use domain-specific languages (DSLs) using a simple, intuitive syntax. It provides a set of tools and libraries for generating language-specific parsers, compilers, and code generators, as well as a rich set of features for managing DSL projects and codebases.
- **JetBrains MPS (Meta Programming System)** is a language workbench that allows developers to create and use DSLs. It provides a powerful set of tools for designing and building custom languages, as well as generating code for those languages. MPS is unique in that it uses a projectional editor, which allows developers to work with code using a visual, tree-like representation. This makes it easier to work with complex codebases and ensures that the code is always well-formed and consistent.

Evaluating the different available DSL platforms, the Eclipse Xtext was selected as the more suitable DSL Platform. JetBrains MPS while offering a plethora of features and solutions such as the “Projectional Editor” was not deemed as flexible enough compared to Xtext due to its closely tied and integrated development and operational environment and due to its restricted capabilities in integration with other components and technologies. Xtext allows for a simplified approach in the development of language servers that comply to Language Server Protocol (LSP) meant that integration with other components would be easier and more efficient. “textX” was also evaluated but was not selected because it has not matured enough as a platform (compared to Xtext, from which textX was originally inspired), and also because its programming environment is too architecturally different compared to the rest of the back-end system.

#### 7.2.2.3 Frontend

Mainly two libraries can be used in the development of the front-end DSL: React Flow and Monaco Editor.

- React Flow is an open-source library for building web-based flowcharts and diagrams using "React.js". It provides a set of components and utilities that you can use to create custom diagrams and visualizations that are interactive and responsive. React Flow is built on top of the HTML5 canvas element, which provides a high-performance drawing surface for rendering complex diagrams and visualizations. It includes a wide range of features, including support for drag-and-drop interaction, customizable styles and themes, zooming and panning, etc. React Flow is a popular choice for building web-based workflow management systems, data visualization tools, and other applications that require complex diagrams and visualizations. It is actively maintained by a community of developers and is available under the MIT license.
- The Monaco Editor is a web-based code editor that was originally developed by Microsoft for the Visual Studio Code (VS Code) editor. It can also be used as a standalone component in other web applications. The Monaco Editor is open-source software. It is highly customizable and provides a rich set of features for editing code, including syntax highlighting, code completion, code folding, error checking, and more. It supports a wide range of programming languages and file types, and it is designed to be fast and responsive, even when working with large files.

#### 7.2.2.4 Communications KAFKA

Apache Kafka is the backbone for many of the modern real-time streaming data architectures. It depends on an event mesh capable to power real-time applications and analytics while being able to scale its performance. Kafka is a fast, scalable, durable and fault-tolerant publish-subscribe messaging system. The core components of Apache Kafka are the brokers and the clients that can be either producers or consumers. Kafka brokers receive, store and transmit the log messages while the clients "produce" or "consume" them. The data inside Apache Kafka is organized in logical groups called topics and each client must be aware of the topics that transmit or receive messages. For better scalability, topics are divided into partitions that can serve multiple clients in parallel. These partitions can reside in one or more brokers, where these brokers form a Kafka cluster.

Although Kafka is not designed for multitenancy, several architectural decisions can help to achieve it. In modern systems, Apache Kafka is frequently deployed on the Kubernetes container management system, which is used to automate the deployment, scaling, and operation of containers across clusters of hosts. A key benefit of running Apache Kafka on Kubernetes is infrastructure abstraction. A Kubernetes pod can be configured once and run everywhere. Kubernetes allows applications to scale resources up and down with a simple command, or scale automatically based on usage, to make the most economical use of computing, networking, and storage resources. Kubernetes also offers Apache Kafka the portability to span across on-premises and public, private, or hybrid clouds, and use different operating systems, achieving low latency for network and storage and high availability.

### 7.2.3 User Management

The EO4EU project will consist of several applications that need a user access management. For this a centralized user management system for all applications is needed, where the user needs only one account (federated identity) to access all authorized application of EO4EU. Additionally, the definition of general authorization roles should be possible in the user management system, so that the EO4EU application do not have to deal with them.

Also, the user does not want to login to each EO4EU application separately, therefor as a single-sign-on mechanism is needed.

For a low barrier to entry, the user should not have to create a new account first. The user should have the possibility to use an account from a public identity provider like Microsoft, Google, or Facebook

to register for the EU4EO services. Of course, it should also be possible to create a user account directly in the EO4EU platform.

For later integration of the EO4EU platform into other already existing environments, it would also be preferable, that the user management system can connect to existing user data bases (like LDAP) to integrate this external defined user data (user federation).

State-of-the-art protocols, to implement the project needs:

- LDAP: LDAP stands for “Lightweight Directory Access Protocol” and is defined in RFC 4510 to RFC 4532. It provides a tree base directory structure where personal data or right configurations can be stored. Using LDAP different application can access the same user database in a standardized way.
- OAuth 2.0: Using OAuth 2.0 a user (resource owner) can give an application (client) the right to access the API of a resource server using an access token that was exchanged via an authorization server. Resource server and an authorization server are off the same server. OAuth 2.0 is defined in RFC 6749 and RFC 6750. Also, server extensions are developed by the IETF-OAuth-working group 16.
- OpenID Connect (OIDC): OIDC is an authentication layer on top of OAuth 2.0. While OAuth 2.0 is designed only for authorization, for granting access to data and features from one application to another. OIDC adds login and profile information about the person who is logged in using an ID token (a JWT / JSON Web Token). OIDC was developed by the OpenID Foundation<sup>17</sup>
- SAML 2.0: SAML 2.0 is feature rich standard for exchanging authentication and authorization data between identity providers and service providers (applications). SAML transfers messages, called assertions, which contain the relevant security requirements for authenticating, authorizing, and determining the level of permissions a client will receive. SAML 2.0 was developed by the Organization for the Advancement of Structured Information Standards (OASIS)<sup>18</sup>
- OIDC vs SAML 2.0: Both can be used to implement secure identity providers and SSO mechanisms. While OIDC only transmits general user data, SAML 2.0 can contain direct authorization data (such as permissions). As OIDC is based on OAuth, API calls can also be authorized with it .

To make these needs clearer, some of the terms used are explained in the following:

- Storage of user data: The user data needs to be stored persistently. Many applications use non standardized schemas in relation databases to store user credentials and further user properties. But a more appropriate and standardized way of storing user credentials are LDAP based systems (like OpenLDAP or Active Directory).
- User Federation: A user federation is established when a user management system connects to several user databases (relational database, LDAP, etc.), where users are managed independently. This has the advantage, that several already existing user databases, can be integrated into to a new system.
- Identity Providers and Identity Brokering: Using an identity broker is an alternative of directly accessing a user database. Instead, an identity broke can securely connect to several external identity providers. So, a user does not have to create a new account in a new system but can reuse an existing account in another system (an identity provider). In general, the user can also decide witch account information he wants to share with the new system. Commonly used protocols are SAML 2.0, OAuth 2.0 and OpenID Connect (see below).

- Federated Identity: A federated identity is a user identity (account) stored in one place (like LDAP or OAuth identity provider) used in several independent applications via a user federation or identity brokering.
- Single-sign-on (SSO): Using single-sign-on technologies, a user must only authenticate (log in) once on a device and can access all authorized applications without the need to authenticate again on platform's different applications. Commonly used protocols are SAML 2.0 and OpenID Connect.

Existing software solutions:

The defined needs are similar in many large composite systems. Therefore, there are already various software solutions that cover the requirements. Some examples are listed below:

- Okta Single Sign-On / Auth0
- PingOne Cloud Platform
- Oracle Access Management Suite
- WSO2 Identity Server
- Keycloak

While all of them fulfil the project needs, only Keycloak is a free and open-source solution.

### 7.3 Traditional vs. interactive data presentation tools

Data visualization tools can be divided into two groups, traditional and modern or interactive visualization tools.

Most traditional exploration and visualization systems cannot handle the size of many contemporary datasets. They restrict themselves to dealing with small dataset sizes, which can be easily handled and analysed with conventional data management and visual exploration techniques. Further, they operate in an offline way, limited to accessing pre-processed sets of static data.

Nowadays, the Big Data era has made available large numbers of very big datasets, that are often dynamic and characterized by high variety and volatility. For example, in several cases, new data constantly arrive (e.g., on a daily/hourly basis); in other cases, data sources offer query or API endpoints for online access and updating. Further, an increasingly large number of diverse users (i.e., users with different preferences or skills) explore and analyse data in many different scenarios.

A solution to dealing with this kind of data is to use modern systems, which should be able to efficiently handle big dynamic datasets, operating on machines with limited computational and memory resources. The dynamic nature of nowadays data (e.g., stream data), hinders the application of a pre-processing phase, such as traditional database loading and indexing. Hence, systems should provide on-the-fly processing over large sets of raw data. Further, in conjunction with performance issues, modern systems must address challenges related to visual presentation, because visualizing a large number of data objects is a challenging task and modern systems have to “squeeze a billion records into a million pixels”<sup>1</sup>. Even in small datasets, offering a dataset overview may be extremely difficult, in both cases, information overloading is a common issue. Consequently, a basic requirement of modern systems is to effectively support data abstraction over enormous numbers of data objects. Apart from the aforementioned requirements, modern systems must also satisfy the diversity of preferences and requirements posed by different users and tasks. Modern systems should provide the user with the ability to customize the exploration experience based on her preferences and the

---

<sup>1</sup> Shneiderman, Ben. (2008). Extreme Visualization: Squeezing a Billion Records into a Million Pixels. 3-12. 10.1145/1376616.1376618.

individual requirements of each examined task. Additionally, systems should automatically adjust their parameters by considering the environment setting and available resources.

All these characteristics make the modern system more interactive by using of modern data analysis software and enabling users to directly manipulate and explore graphical representations of data. Data visualization uses visual aids to help analysts efficiently and effectively understand the significance of data. Interactive data visualization software improves upon this concept by incorporating interaction tools that facilitate the modification of the parameters of data visualization, enabling the user to see more detail, create new insights, generate compelling questions, and capture the full value of the data.

In short, a traditional visualization tool is more of a static data visualization, one that does not incorporate any interaction capabilities and does not change with time. As there are no tools to adjust the results of static visualizations, such as filtering and zooming tools in interactive designs, it is essential to consider which data to display and how the data is being displayed. Moreover, the interactive tools are providing an intuitive environment in which users can easily identify and explore trends across specific time frames and areas.

### 7.3.1 *Project needs versus existing solutions*

From the collected requirements for each of the use cases presented in this project, it can be concluded that almost all of them rely on interactive web maps when it comes to presenting and communicating geospatial data results.

The scope of the **UC1** is to raise public awareness and help prevent chronic diseases (i.e., allergies and asthma as a potential consequence), by using the operation forecasting model PASYFO. The daily model results will be displayed using online web maps showing different air quality indices, different types of pollen (e.g., olive, grass, birch, etc.), including the humidex index, an index used by Canadian meteorologists to describe how hot the weather feels to the average person, by combining the effect of heat and humidity. The extension of the existing mobile app is tentative and needs to be discussed.

**UC2** aims to provide a weather routing Decision Support Systems (DSS) for ships. The results will show alternative, fuel-efficient routes for ships, taking into consideration various vessel-based sensory information as well as weather forecast data. The preferred visualization modality is an online web map, visualizing current/planned routes and optimized routes, including additional information such as estimated time of arrival, fuel consumption and overall emissions for all available routes.

**UC3** aims to provide several services in support of food security:

- Crop impact analysis
- Damage estimation
- New areas with favourable conditions for specific crops
- Suitable crops for new climate conditions

The results from each service will be displayed on the online maps at a time range frequency that will be decided by the stakeholders. The services will provide the following maps results:

- Crop impact analysis service results will provide online maps showing future changes in crop production in correlation with relevant climate indicators.
- Damage estimation service results will show the vegetation indexes in correlation with weather parameter analysis at a high resolution.
- New areas with favourable conditions for specific crops service results will be maps showing areas with the best climate conditions for specific crops and cultivation potential values (different classes, from Low=0 to High=100).

- Suitable crops for new climate conditions service results will display maps showing climate projection with crops classified differently depending on their suitability in each specific area.

The results analysis of the **UC4**, which aims at simulating water, energy, and carbon fluxes in forestry towards enhancing forest management under climate change, towards sustainability and maintaining forest ecosystem services by using the Forest Ecosystem Model (FEM), will be delivered through web GIS with direct download and on demand. The maps will show the following:

- forest growth and carbon dynamics: 1) without CC nor forest management 2) under different CC scenarios 3) under alternative management methods 4) under different CC scenarios and considering alternative management methods.
- forest water dynamics: 1) without CC nor forest management 2) under different CC scenarios 3) under alternative management methods 4) under different CC scenarios and considering alternative management methods.

**UC5** aims to integrate climate and non-climate datasets to assess soil susceptibility to water erosion in Italy under climate change. The analysis will be delivered through web GIS with direct download and on-demand for the following current and future results:

- rainfall erosivity
- soil susceptibility
- soil loss
- rainfalls erosivity anomalies
- soil loss anomalies

The **UC6** objective is to provide information service for the locust plague impact assessment. Therefore, the analysis will be delivered through web GIS with a direct download of the following results:

- density map of the estimated appearance of locusts
- risk map, layer/mask showing estimation of the probability of appearance; binary output
- information about the reliability of the model output;

The frequency of the analysis delivery will be based on the model retraining frequency (e.g. (bi)-monthly) and on the weather forecast data.

The last use case, namely **UC7**, aims to ease the adoption of local, short-time prevention measure addressed to improve effectiveness and timeliness of firefighting actions on active fires. The delivery of the results is estimated to be daily or a 24h fire risk map, through direct delivery to recipient administrations by data push. The time-range frequency of the results delivery can be done daily or on demand.

As stated above and as can be seen from the description provided above and taken from the questionnaires, most of the use cases rely on the interactive web maps to display the results of each service that will be provided.

The advantages of delivering the results using web mapping are that you can access the service independent of user equipment and hardware location, possible to access resources independent of their location, allows user interaction with maps and production of tailored maps for particular purposes, are so widespread that use is intuitive, uses open standards, offers the possibility of grant access or publish and the most Web GIS application are offering user-friendly interface to view the results. However, there are also some small possible problems that can arise such as a lack of some useful features, slowness in data handling, unsupported data types, or licensing issues.



### 7.3.2 Identified gaps per use case

As can be seen from the description above all the use case owners want their results to be delivered as a web map.

According to the definition, web mapping is the process of using the internet to view, analyse, or share a visual representation of geospatial data in map form. It provides the ability to access geospatial mapping on the internet through a web browser interface and is sometimes referenced as web GIS (Geographic Information System). A primary aspect of its request is the ability for mapping to be accessed in a flexible manner, while providing an intuitive way to interact with location data, selecting different map data layers or features to view, zooming into a particular part of the map that you are interested in, inspecting feature properties, editing existing content, or submitting new content. Also, the tools for web mapping play a critical role in making location intelligence accessible to different people, such as senior management, business analysts, and customers. The intuitive functionality often associated with web mapping brings analysis to any desktop, website, tablet, or mobile device. Therefore, an effective web mapping means users can easily access, update, and visualize data anywhere and anytime, while being able to easily share mapping with co-workers, citizens, customers, and partners.

From all that has previously been mentioned , including the questionnaires that were filled out by each use case owners, it can be stated that there is no gap to report regarding the visualization part.

## 8 Intellectual properties management tools

---

### 8.1 Introduction

The topic of preservation within any computational business process is a key factor to ensure that every step is computed in the correct (expected) way and that the results conform with the original pipeline / workflow. This includes preservation of the intellectual property of the modules that compose the processing workflow, as well as the security (in terms of accessibility or no-contamination) of both the input and output data.

During the collection of the use case requirement, no needs have emerged with respect to the enforcement of IPR, since all tools and services that will be generated will be released as open source thus no property must be protected / preserved. Nevertheless, the topic can be critical for platform users, thus it deserved to be faced.

Differently from the previous research areas analyses, this section provides only an overview on the state of the art on IPR management tools: since to requirements have been expressed, no gaps can be identified.

### 8.2 Intellectual properties management state of the art

The following chapter elaborates the state of the art of licensing intellectual property in software management. The state of the art of licensing intellectual property in software management involves various mechanisms for managing the use and distribution of software and its associated intellectual property rights. The purpose of licensing is to protect the software from unauthorized use, ensure that it is used in accordance with the terms and conditions of the license, and to ensure that the software is used in a manner that benefits the owner and the users of the software.

There are several types of licenses used to manage intellectual property in software management, including proprietary licenses, open-source licenses, and dual-licensing models. Proprietary licenses are the most common form of software license and they typically provide exclusive rights to the software owner to use, distribute, and modify the software. Proprietary licenses are typically used by software vendors to protect their investment and generate revenue from their software. Open-source licenses, on the other hand, are licenses that allow anyone to use, modify, and distribute the software freely as long as they comply with the terms and conditions of the license. Open-source licenses are typically used to promote collaboration, encourage innovation, and to make software freely available to users. Dual-licensing models are licenses that allow the software owner to use a proprietary license for some applications and an open-source license for others. Dual-licensing models are used to balance the interests of the software owner and the users of the software.

#### 8.2.1 *Proprietary and open-source software management*

There are several mechanisms to manage licensing in software management. One of the most common approaches is using software license management tools, which help organizations to manage and track software licenses, automate license allocation and usage, and enforce compliance with licensing agreements. Another approach is through the use of software license agreements, which define the terms and conditions under which the software may be used, including restrictions on use, distribution, and the number of licenses required. Additionally, organizations may implement policies and procedures to monitor and manage software usage, such as regularly reviewing software inventory and usage reports, or conducting regular audits of software installations.

Open-source software management refers to the development and distribution of software where the source code is freely available for anyone to use, modify, and distribute. This type of software is built and maintained by a community of developers and users, who collaborate and contribute to its

evolution. Open-source software is often managed through a variety of collaborative tools, such as version control systems (e.g., Git), bug trackers, and project management tools. Additionally, many open-source projects have a governance model that outlines the roles and responsibilities of different contributors and provides a mechanism for decision-making and dispute resolution. An important aspect of open-source software management is ensuring that contributions are licensed in a way that is consistent with the open-source license of the project. This typically involves using a well-established open-source license, such as the MIT License or the GPL, and requiring contributors to sign a Contributor License Agreement.

### *8.2.2 Protection of input data and workflows*

When it comes to software management, protecting input data is a key consideration. There are several mechanisms that can be used to protect input data in software management. One common approach is to use digital rights management (DRM) techniques to control access to the data and prevent unauthorized use. This can be done through encryption, password protection, or other forms of access control. Another approach is to use licensing agreements, which specify the terms under which the data can be used, including who can access it, how it can be used, and what restrictions may be placed on its use. In some cases, data may also be protected by patents, copyrights, or other forms of legal protection.

Similarly, to protect data processing steps and workflows in software management, there are several solutions. One approach is to use a blockchain-based system to secure the flow of data from its source to its final destination, ensuring that the data is tamper-proof and that the processing steps are verifiable. This can be done through the use of digital signatures, which are cryptographically secure digital representations of the data and the processing steps. Additionally, blockchain technology can be used to create a tamper-evident record of the data and its processing, which can provide a permanent and secure way to track changes to the data and the processing steps. Another approach is to use a certification mechanism, such as a secure digital certificate, to verify the authenticity of the data and its processing steps. This can be done by using digital certificates issued by a trusted third-party, such as a certification authority (CA), which can confirm the authenticity of the data and the processing steps and help ensure that the data is protected from manipulating and unauthorized access.

### *8.2.3 Protection of results*

To avoid copying, cloning, or reselling of publicly generated insights and results, it is important to have strong intellectual property rights protection in place. One way to protect IPR is through copyright and trademark laws, which give the owner exclusive rights to their work and prohibit others from using it without permission. Another way to protect IPR is through patents, which provide the inventor with the exclusive right to make, use, and sell their invention for a certain period of time.

In addition to legal protection, there are technical measures that can be taken to prevent copying or unauthorized use of results. For example, digital watermarking can be used to embed information about the owner of the results into the data itself, making it easier to trace and enforce IPR in case of unauthorized use. Encryption can also be used to secure results and prevent unauthorized access.

Another approach is to use licenses to clearly specify the terms and conditions of use for the results. For example, a Creative Commons license can be used to specify that the results can be freely used, but only if they are attributed to the original creator and are not used for commercial purposes. This can help to deter unauthorized copying and reuse of the results.

## 9 Concluding remarks

---

This document has the main scope of analysing the use cases to be implemented in the framework of the EO4EU activity, to drive the development of the project platform, identifying existing gaps that will support the upcoming design and implementation activities. This includes the identification of a set of topics (research areas, see Section 3) on which the gaps have to be assessed that that will be considered during the implementation phases.

The original deliverable's scope was to analyse and report gaps per use case. While performing the activities (see Section 2 for an overview of the use cases and a description of the methodology) it turned out that an aggregated analysis per research area would had been more effective to identify communalities among the use cases both in terms of needs (= gaps) or methodologies that can be improved. Section 4 to Section 8 detail the outcome of the analysis per research area.

Considering the nature of the services to be implemented within the project and, in general, the users and use cases that the platform will be requested to support, the following summary conclusions can be drawn for the five main research areas:

- **Data accessibility & exploitability.** An extremely large variety and volume of geospatial / environmental data exist, and there is not a single data access and exploitation platform that can manage all of them. Most of these data are free and open, while some (critical ones such as high-resolution weather forecast) cannot be publicly accessed, thus there is a need to make these data open and free. There is a need to develop / deploy advanced data access and exploitation tools that can speed up and simplify the discovery and access to the data, with the possibility to perform pre-processing activities close to the data.
- **Processing capabilities and scale-up.** The current scenario of data processing platforms and cloud resources offer a wide range of services (CPUs, GPUs, HPC) systematically and on demand. The existing offer can more or less satisfy the needs of the services thus not critical gaps have been identified. On the other side, the use of rare computational capabilities within the services is still relatively limited, thus an information campaign/awareness transfer of the potentialities and available tools to include effective computational resources within the business workflow shall be performed, to allow platform users/service providers to associate each processing step with the most adequate computational resource.
- **Algorithm capabilities.** Despite the fact that the analysed services feature different levels of maturity, the main identified gap resides on the lack of usage of ML/AI based tools within the workflows. The gap is not only technological but mainly a knowledge gap: there is a need to show/demonstrate and, in general teach how ML/AI based modules work in a way that the algorithm/pipeline/workflow developers can take informed decision on which approach to follow to implement a specific step or a full process.
- **Data presentation/communication/delivery.** The use cases foresee the use of consolidated methodologies for what regards data provision to the users. No critical gaps have been identified. There is a need to implement an integrated scenario based on a common data pool. This will allow direct access to the results of the services as well as the use of the services via a web-based user interface and via mobile applications.
- **Intellectual properties management tools.** No specific needs have been identified for what regards the enforcement of IPR, since all tools and services that will be generated will be released as open source. Nevertheless, the topic can be critical for platform users, thus a short state of the at has been provided within the document.

To summarise the overall outcome of the analyses, Table 11 provides a one-look overview of the criticalities identified per use case per research area. Four different colours are used to assess the identified gap level with respect to the state of the art. The table shall allow driving and prioritising the developments to be carried on during the project.

	Data			Processing		Algorithms		visualization
	access	License	exploitability	Computation	GPUs/HPCs	traditional	AI based	
UC1	Critical	Critical	Relevant	Relevant	No gap	Relevant	Relevant	Relevant
UC2	No gap	No gap	Critical	Relevant	Relevant	No gap	Relevant	Relevant
UC3	Relevant	Relevant	Relevant	Relevant	Relevant	Relevant	Relevant	Relevant
UC4	Relevant	Relevant	Relevant	Relevant	No gap	Relevant	Relevant	Relevant
UC5	Relevant	Relevant	Relevant	Relevant	Relevant	Relevant	Relevant	Relevant
UC6	Critical	Critical	Relevant	Relevant	Relevant	Relevant	Relevant	Relevant
UC7	Relevant	Relevant	Relevant	Relevant	No gap	No gap	Relevant	Relevant

No gap / need identified
limited gap / need identified
Relevant gap / need identified
Critical gap / need identified

**Table 11. Summary of the performed gap analysis per research area per use case.**

## 10 Annex A: Requirements collection form template

### 10.1 Input data specifications

<b>Data ID</b>	<b>Dataset description</b> (satellite, sensor, parameter, shapefile, ...)	<b>Application Field</b>	<b>Unit &amp; Dimension</b> (main unit and space-time dimension of the dataset)	<b>Data format / Syntax</b> (original format; nc, gtiff, ...)	<b>Data spatial structure</b> (raster, points, lines, polygons, ...)	<b>Frequency of collection</b> (temporal resolution; e.g., acquisition or model interval)	<b>Time range</b> (time span)	<b>Spatial resolution</b>	<b>Spatial Coordinate System</b> (geographic (GCS) or projected (PCS) coordinate system)	<b>Geographic extent</b> (global, Europe, ...)
<b>DX</b>										

<b>Data ID</b>	<b>Vertical coverage/range</b> (vertical extent, e.g., surface, 2m)	<b>Vertical resolution</b> (vertical spacing, e.g., number of pressure levels)	<b>Source</b> (where is the data source)	<b>Upload</b> (preferred upload modality; for local / owned data)	<b>Access modality</b> (API, ...)	<b>Access link</b>	<b>Availability</b> (online, offline / to be ordered)	<b>Data Owner / License</b> (ESA, EC, public, ...)	<b>Dataset rationale</b> (Why did you decide for the specific dataset? Reasoning for the dataset)	<b>Notes</b> (Further information)
<b>DX</b>										

## 10.2 Data preparation

<b>Data ID</b>	<b>Dataset description</b>	<b>Data preparation interface</b> (GUI, CLI, other; API)	<b>manual / automatic</b> (how is the data preparation performed?)	<b>Data coverage</b> Spatial (domain definition) Temporal (period definition)	<b>Size</b> (estimated product size)	<b>Facility</b> (local or remote preparation?)	<b>Regridding / Interpolation</b> (Target grid / projection)	<b>Operations</b> (Data preparation operations, e.g., format conversion)	<b>Notes</b> (Further information)
<b>DX</b>									

## 10.3 Data Processing

<b>Processing step</b>	<b>UC development / processing step</b>	<b>Step description</b>	<b>Modeling details / requirements</b> (if applicable)	<b>Included datasets</b>	<b>Outcome</b> (Processing step outcome / product; if applicable)	<b>Outcome preservation</b> (keep / discard dataset after production; y/n)	<b>Commercial / proprietary tools</b> (used to perform the step)
<b>SX</b>							

<b>Processing step</b>	<b>Open-source tools</b> (used to perform the step)	<b>User-provided algorithms</b> (user-owned software tools)	<b>User algorithm provision</b> (source code, executable)	<b>Data processing interface</b> (algorithm / process execution)	<b>Facility</b> (local or remote processing?)	<b>Infrastructure needs</b> (CPU, HDD, ...)	<b>Notes / Wishlist</b> (Further information)
<b>SX</b>							

## 10.4 Results analysis

Result	Result	Result description	Result delivery (online, offline, direct delivery to user, e.g., data push/pull)	Result delivery frequency	Delivery infrastructure requirements (CPUs, HDD, ...)	Notes (Further information)
RX						



## 11 Annex B: Collected use case requirements

---

Annex B provides links to the spreadsheets containing the collected UC requirements for each of the seven use cases.

- UC1 (Health)  
[https://docs.google.com/spreadsheets/d/1tcV\\_nhp7FD8RIBt52mYI\\_COtOIBtL1Qo/edit?usp=sharing&oid=111887094069838382657&rtpof=true&sd=true](https://docs.google.com/spreadsheets/d/1tcV_nhp7FD8RIBt52mYI_COtOIBtL1Qo/edit?usp=sharing&oid=111887094069838382657&rtpof=true&sd=true)
- UC2 (Ocean)  
[https://docs.google.com/spreadsheets/d/19I9zsMJdf5fx9na40B4cSq\\_KL\\_8g77L4/edit?usp=share\\_link&oid=111887094069838382657&rtpof=true&sd=true](https://docs.google.com/spreadsheets/d/19I9zsMJdf5fx9na40B4cSq_KL_8g77L4/edit?usp=share_link&oid=111887094069838382657&rtpof=true&sd=true)
- UC3 (Crop)  
[https://docs.google.com/spreadsheets/d/1B0vd9pX8ANgphYeulX2-d2t-HXxFXIR/edit?usp=share\\_link&oid=111887094069838382657&rtpof=true&sd=true](https://docs.google.com/spreadsheets/d/1B0vd9pX8ANgphYeulX2-d2t-HXxFXIR/edit?usp=share_link&oid=111887094069838382657&rtpof=true&sd=true)
- UC4 (Forest)  
[https://docs.google.com/spreadsheets/d/11Cdr3rjSHojvquUBC1OP32C\\_Tt5EKcg/edit?usp=share\\_link&oid=111887094069838382657&rtpof=true&sd=true](https://docs.google.com/spreadsheets/d/11Cdr3rjSHojvquUBC1OP32C_Tt5EKcg/edit?usp=share_link&oid=111887094069838382657&rtpof=true&sd=true)
- UC5 (Soil)  
[https://docs.google.com/spreadsheets/d/1pr8bW8Z49X2Qx3G-xM\\_8NukjH4aEmaGS/edit?usp=sharing&oid=111887094069838382657&rtpof=true&sd=true](https://docs.google.com/spreadsheets/d/1pr8bW8Z49X2Qx3G-xM_8NukjH4aEmaGS/edit?usp=sharing&oid=111887094069838382657&rtpof=true&sd=true)
- UC6 (Food)  
[https://docs.google.com/spreadsheets/d/1wb7mkKH91xnMDjOJ1BT0VAtVkCML\\_8e/edit?usp=share\\_link&oid=111887094069838382657&rtpof=true&sd=true](https://docs.google.com/spreadsheets/d/1wb7mkKH91xnMDjOJ1BT0VAtVkCML_8e/edit?usp=share_link&oid=111887094069838382657&rtpof=true&sd=true)
- UC7 (Fires)  
[https://docs.google.com/spreadsheets/d/126EIJf-Da8wm6hanDCsl4UH6EWf4LIG/edit?usp=share\\_link&oid=111887094069838382657&rtpof=true&sd=true](https://docs.google.com/spreadsheets/d/126EIJf-Da8wm6hanDCsl4UH6EWf4LIG/edit?usp=share_link&oid=111887094069838382657&rtpof=true&sd=true)

## 12 Annex C: References

---

- (Agustsson et al., 2017) Eirikur Agustsson, Fabian Mentzer, Michael Tschannen, Lukas Cavigelli, Radu Timofte, Luca Benini, Luc Van Gool. Soft-to-Hard Vector Quantization for End-to-End Learning Compressible Representations. *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.
- (Alailan et al., 2012) N. Alailan , Y. Bazi , F. Melgani , R.R. Yager , Fusion of supervised and unsupervised learning for improved classification of hyperspectral images, *Inf. Sci.* 217 (2012) 39–55.
- (Balle et al., 2018) Johannes Balle, David Minnen, Saurabh Singh, Sung Jin Hwang, Nick Johnston. Variational image compression with a scale hyperprior. *International Conference on Learning Representations (ICLR)*, 2018.
- (Chen et al., 2020-1) Ting Chen, Simon Kornblith, Mohammad Norouzi, Geoffrey Hinton. A Simple Framework for Contrastive Learning of Visual Representations. *International Conference on Machine Learning (ICML)*, 2020.
- (Chen et al., 2020-2) Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, Geoffrey Hinton. Big Self-Supervised Models are Strong Semi-Supervised Learners. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.
- (Chen et al., 2020-3) Xinlei Chen, Haoqi Fan, Ross Girshick, Kaiming He. Improved Baselines with Momentum Contrastive Learning. *Tech Report*. <https://arxiv.org/abs/2003.04297>, 2020.
- (Chen et al., 2015) B. Chen , B. Huang , B. Xu , Comparison of spatiotemporal fusion models: a review, *Remote Sens.* 7 (2) (2015) 1798–1835.
- (Cherkassky et al., 2006) V. Cherkassky, V. Krasnopolsky, D.P. Solomatine, J. Valdes, Computational intelligence in earth sciences and environmental applications: issues and challenges, *Neural Net.* 19 (2), 113–121.
- (Geng et al., 2022) Wanxuan Geng, Weixun Zhou, Shuanggen Jin. Multi-View Urban Scene Classification with a Complementary-Information Learning Model. *Photogrammetric Engineering & Remote Sensing*, 2022.
- (Cheng et al., 2018) Gong Cheng, Ceyuan Yang, Xiwen Yao, Lei Guo, Junwei Han. When Deep Learning Meets Metric Learning: Remote Sensing Image Scene Classification via Learning Discriminative CNNs. *IEEE Transactions on Geoscience and Remote Sensing*, 2018.
- (Gan et al., 2022) Gan, L., Chen, Q., Zhang, D., Zhang, X., Zhang, L., Liu, C., & Shu, Y. (2022). Construction of Knowledge Graph for Flag State Control (FSC) Inspection for Ships: A Case Study from China. *Journal of Marine Science and Engineering*, 10(10), 1352. <https://doi.org/10.3390/jmse10101352>
- (Garzelli, 2016) A. Garzelli, A review of image fusion algorithms based on the super-resolution paradigm, *Remote Sens.* 8 (10) (2016) 797.
- (Ghamisi et al., 2017) P. Ghamisi , J. Plaza , Y. Chen , J. Li , A. Plaza , Advanced supervised classifiers for hyperspectral images: a review, *IEEE Geosci. Remote Sens. Mag.* 5 (1) (2017) 1–7.

(Ghamisi et al., 2019) P. Ghamisi , B. Rasti , N. Yokoya , Q. Wang , B. Hofle , L. Bruzzone , et al. , Multisource and multitemporal data fusion in remote sensing: a comprehensive review of the state of the art, *IEEE Geosci. Remote Sens. Mag.* 7 (1) (2019).

(Ghamisi, Höfle, Zhu, 2017) P. Ghamisi , B. Höfle , X.X. Zhu , Hyperspectral and LiDAR data fusion using extinction profiles and deep convolutional neural network, *IEEE J. Sel. Top. App. Earth Obs. Remote Sens.* 10 (2017) 3011–3024.

(Ghassemian, 2016) H. Ghassemian, A review of remote sensing image fusion methods, *Inf. Fusion* 32 (2016) 75–89.

(Gómez-Chova et al., 2015) L. Gómez-Chova , D. Tuia , G. Moser , G. Camps-Valls , Multimodal classification of remote sensing images: a review and future directions, *Proc. IEEE* 103 (9) (2015) 1560–1584.

(Gueguen et al., 2018) Lionel Gueguen, Alex Sergeev, Ben Kadlec, Rosanne Liu, Jason Yosinski. Faster Neural Networks Straight from JPEG. *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

(Habibian et al., 2019) Amirhossein Habibian, Ties van Rozendaal, Jakub M. Tomczak, Taco S. Cohen. Video Compression With Rate-Distortion Autoencoders. *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

(He et al., 2020) Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, Ross Girshick. Momentum Contrast for Unsupervised Visual Representation Learning. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

(He et al., 2015) Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. Deep Residual Learning for Image Recognition. Tech Report. <https://arxiv.org/abs/1512.03385>, 2015.

(Janowicz et al., 2022) Janowicz, K., Hitzler, P., Li, W., Rehberger, D., Schildhauer, M., Zhu, R., Shimizu, C., Fisher, C. K., Cai, L., Mai, G., Zalewski, J., Zhou, L., Stephen, S., Gonzalez, S., Mecum, B., Lopez-Carr, A., Schroeder, A., Smith, D., Wright, D., ... Currier, K. (2022). Know, Know Where, KnowWhereGraph: A densely connected, cross-domain knowledge graph and geo-enrichment service stack for applications in environmental intelligence. *AI Magazine*, 43(1), 30–39. <https://doi.org/10.1002/aaai.12043>

(Jean et al., 2019) Neal Jean, Sherrie Wang, Anshul Samar, George Azzari, David Lobell, Stefano Ermon. Tile2Vec: Unsupervised Representation Learning for Spatially Distributed Data. *AAAI Conference on Artificial Intelligence*, 2019.

(Jing et al., 2021) He Jing, Yongqiang Cheng, Hao Wu, Hongqiang Wang. Radar Target Detection With Multi-Task Learning in Heterogeneous Environment. *Conference on Robots and Vision*, 2021.

(Kibirige et al., 2020) Kibirige, D., Dobos, E., Soil moisture estimation using citizen observatory data, microwave satellite imagery, and environmental covariates. *Water* 2020, 12, 2160.

(Kiryakov et al., 2004) Kiryakov, A., Popov, B., Terziev, I., Manov, D., & Ognyanoff, D. (2004). Semantic annotation, indexing, and retrieval. *Journal of Web Semantics*, 2(1), 49–79. <https://doi.org/10.1016/j.websem.2004.07.005>

(Klöwer et al., 2021) Milan Klöwer, Miha Razinger, Juan J Dominguez, Peter D Düben, Tim N Palmer. Compressing atmospheric data into its real information content. *Nature Computational Science*, 2021.

(Li et al., 2021) Jin Li, Zilong Liu. Efficient compression algorithm using learning networks for remote sensing images. *Applied Soft Computing*, 2021.

(Matevosyan et al., 2017) Matevosyan, H., Lluch, I., Poghosyan, A., & Golkar, A. (2017). A Value-Chain Analysis for the Copernicus Earth Observation Infrastructure Evolution: A Knowledgebase of Users, Needs, Services, and Products. *IEEE Geoscience and Remote Sensing Magazine*, 5(3), 19–35. <https://doi.org/10.1109/MGRS.2017.2720263>

(Mentzer et al., 2018) Fabian Mentzer, Eirikur Agustsson, Michael Tschannen, Radu Timofte, Luc Van Gool. Conditional Probability Models for Deep Image Compression. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

(Mentzer et al., 2020) Fabian Mentzer, George D Toderici, Michael Tschannen, Eirikur Agustsson. High-Fidelity Generative Image Compression. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

(Minnen et al., 2018) David Minnen, Johannes Balle, George Toderici. Joint Autoregressive and Hierarchical Priors for Learned Image Compression. *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

(Paulheim, 2017) Paulheim, H. (2017). Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic Web*, 8(3), 489–508. <https://doi.org/10.3233/SW-160218>

(Rao et al., 2022) Weiqiang Rao, Ying Qu, Lianru Gao, Xu Sun, Yuanfeng Wu, Bing Zhang. Transferable network with Siamese architecture for anomaly detection in hyperspectral images. *International Journal of Applied Earth Observation and Geoinformation*, 2022.

(Schmitt et al., 2016) M. Schmitt, X.X. Zhu, Data fusion and remote sensing: an ever-growing relationship, *IEEE Geosci. Remote Sens. Mag.* 4 (4) (2016) 6–23.

(Shannon, 1948) Claude E Shannon. A Mathematical Theory of Communication. *The Bell System Technical Journal*, 1948.

(Sohn, 2016) Kihyuk Sohn. Improved Deep Metric Learning with Multi-class N-pair Loss Objective. *Advances in Neural Information Processing Systems 29 (NeurIPS)*, 2016.

(Theis et al., 2017) Lucas Theis, Wenzhe Shi, Andrew Cunningham, Ferenc Huszar. Lossy Image Compression with Compressive Autoencoders. <https://arxiv.org/abs/1703.00395>, 2017.

(Torabzadeh et al., 2014) H. Torabzadeh, F. Morsdorf, M.E. Schaepman, Fusion of imaging spectroscopy and airborne laser scanning data for characterization of forest ecosystems – a review, *ISPRS J. Photogram. Remote Sens.* 97 (2014) 25–35

(van den Oord et al., 2017) Aaron van den Oord, Oriol Vinyals, Koray Kavukcuoglu. Neural Discrete Representation Learning. <https://arxiv.org/abs/1711.00937>, 2017.

(Wald, 2000) L. Wald, A conceptual approach to the fusion of Earth observation data, *Surv. Geophys.* 21. 177–186.

(Wang et al., 2022) Yi Wang, Conrad M Albrecht, Nassim Ait Ali Braham, Lichao Mou, Xiao Xiang Zhu. Self-supervised Learning in Remote Sensing: A Review. Accepted by IEEE Geoscience and Remote Sensing Magazine, 2022.

(Wu et al., 2021) Wu, J., Orlandi, F., Alskaf, T., O’Sullivan, D., & Dev, S. (2021, September 27). Ontology modeling for decentralized household energy systems. SEST 2021 - 4th International Conference on Smart Energy Systems and Technologies. 4th International Conference on Smart Energy Systems and Technologies, SEST 2021. <https://doi.org/10.1109/SEST50973.2021.9543327>

(Xu et al., 2018) X. Xu , W. Li , Q. Ran , Q. Du , L. Gao , B. Zhang , Multisource remote sensing data classification based on convolutional neural network, *IEEE Trans. Geosci. Remote Sens.* 56 (2018) 937–949.

(Yacoubi Ayadi et al., 2022) Yacoubi Ayadi, N., Faron, C., Michel, F., Gandon, F., & Corby, O. (2022). A Model for Meteorological Knowledge Graphs: Application to Météo-France Data. In T. Di Noia, I.-Y. Ko, M. Schedl, & C. Ardito (Eds.), *Web Engineering* (pp. 283–299). Springer International Publishing. [https://doi.org/10.1007/978-3-031-09917-5\\_19](https://doi.org/10.1007/978-3-031-09917-5_19)

(Yang et al., 2020) Juntao Yang, Zhizhong Kang, Ze Yang, Juan Xie, Bin Xue, Jianfeng Yang, Jinyou Tao. A Laboratory Open-Set Martian Rock Classification Method Based on Spectral Signatures. *IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium*, 2020.

(Yang et al., 2020) Yibo Yang, Robert Bamler, Stephan Mandt. Improving Inference for Neural Image Compression. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

(Zhang et al., 2021) Lili Zhang, Wanxuan Lu, Jinming Zhang, Hongqi Wang. A Semisupervised Convolution Neural Network for Partial Unlabeled Remote-Sensing Image Segmentation. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2021.

(Yokoya et al. 2017) N. Yokoya, C. Grohnfeldt, J. Chanussot, Hyperspectral and multispectral data fusion: a comparative review of the recent literature, *IEEE Geosci. Remote Sens. Mag.* 5 (2) (2017) 29–56.

(Zhang et al., 2020) Lamei Zhang, Siyu Zhang, Bin Zou, Hongwei Dong. Unsupervised Deep Representation Learning and Few-Shot Classification of PolSAR Images. *IEEE Transactions on Neural Networks and Learning Systems*, 2018.

(Zou, 2020) Zou, X. (2020). A Survey on Application of Knowledge Graph. *Journal of Physics: Conference Series*, 1487(1), 012016. <https://doi.org/10.1088/1742-6596/1487/1/012016>