EO4EU
Earth Observation Data & Service Accessibility

| | |
|---|---|
| Project Title | AI-augmented ecosystem for Earth Observation data accessibility with Extended reality User Interfaces for<br>Service and data exploitation |
| Project Acronym | EO4EU |
| Grant Agreement No. | 101060784 |
| Start Date of Project | 1.06.2022 |
| Duration of Project | 36 Months |
| Project Website | eo4eu.eu |
| GUI | |

# D2.2 – EO4EU End-user Requirements Analysis & Business Process Flows (a)

| | |
|---|---|
| Work Package | WP 2, Requirements Elicitation and Conceptual Framework Specification |
| Lead Author (Org) | Mohanad Albughdadi (ECMWF) |
| Contributing Author(s) (Org) | Lakis Christodoulou(EBOS), Kakia Panagidi(NKUA), Beatrice Chiavarini (CINECA), Roberta Padulano(CMCC), Christina Karafylli (KEMEA) |
| Due Date | 30.11.2023 |
| Date | DD.MM.YYY |
| Version | V1.0 |

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)

Funded by
the European Union

## Versioning and contribution history

| Version | Date | Author | Notes |
|---|---|---|---|
| 0.1 | 11.07.2023 | Mohanad Albughdadi (ECMWF) | TOC and V0.1 |
| 0.2 | 12.07.2023 | Roberta Padulano (CMCC) | Contribution to use case requirements |
| 0.3 | 29.08.2023 | Kakia Panagidi (NKUA) | Contribution to DoA requirements |
| 0.4 | 03.10.2023 | Lakis Christodoulou (EBOS) | Technical and Quality Review |
| 0.5 | 11.09.2023 | Mohanad Albughdadi (ECMWF) | Updates, review, and cleanup |
| 0.6 | 14.09.2023 | Mohanad Albughdadi (ECMWF) | Updates, review, formatting, and cleanup |
| 0.7 | 10.10.2023 | Babis Andreou (NKUA), Olga Sozinova (LU), Pablo Strasser (HESSO), Lionel Blonde(HESSO) | Review |
| 0.8 | 14.11.2023 | Mohanad Albughdadi (ECMWF) | Review and formatting |
| 0.9 | 20.11.2023 | Lakis Christodoulou (EBOS), Kakia Panagidi (NKUA), Pablo Strasser (HESSO), Mohanad Albughdadi (ECMWF) | Review and formatting |

## 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.

# Table of Contents

## List of Figures

## List of Tables

# TERMINOLOGY

| Terminology/Acronym | Description |
|---|---|
| API | Application Programming Interface |
| AR | Augmented Reality |
| AWS | Amazon Web Services |
| BPMN | Business Process Modeling and Notation |
| CLI | Command Line Interface |
| CORS | Cross-Origin Resource Sharing |
| DDAV | Dashboard Data Analytics Visualization |
| DoA | Description of Actions |
| DSL | Domain Specific Language |
| Dx.y | Deliverable x.y |
| EO | Earth Observation |
| EOSC | European Open Science Cloud |
| FaaS | Function as a Service |
| GUI | Graphical User Interface |
| IAM | Identity Access Management |
| JWT | JSON Web Tokens |
| ML | Machine Learning |
| NDI | Normalized Difference Index |
| NDVI | Normalized Difference Vegetation Index |
| SQL | Structured Query Language |
| SSL | Self-Supervised Learning |
| TLS | Trasport Layer Security |
| Tx.y | Task x.y |
| VHR | Very High Resolution |
| VQ-VAE | Vector-Quantized Variational AutoEncoder |
| VR | Virtual Reality |
| WP | Work Package |
| XR | Extended Reality |
| XSS | Cross-Site Scripting |

# Executive Summary

This document is dedicated to the requirements collection and analysis as well as the Business Process Modeling and Notation in the EO4EU project. In a first step, the document describes the approach and methods used to collect the requirements coming from three distinct sources, namely, the Description of Actions, the use cases established by the project's consortium and the external users. In a second step, the document summarizes these requirements. This allows designing the platform to meet the needs of a wide range of users and to be single point of access to Earth Observation data and processing tools.

The document then proceeds by describing the user journey in using the platform for multiple scenarios using Business Process Modeling and Notation (BPMN) concepts. These BPMNs describe the user and the platform interaction from a high-level perspective.

# 1 Introduction

T2.2 is a key task in the EO4EU project, which aims at making Earth Observation (EO) data more accessible, usable and allows creating added value from this data using the available analytics tools and cloud computing technology provided in the platform. The platform will provide a single point of access to EO data and services and is being designed to meet the needs of a wide range of users. This task is designed to extract user requirements, including use cases established by the EO4EU consortium, requirements exhibited in the Description of Actions (DoA) as well as external user requirements.

Two main phases are identified in this task. In the first phase, end-user requirements and operational procedures will be collected and analyzed using a variety of methods, including interviews and surveys. In the second phase, the planned research and implementation from end to end will be modeled using Business Process Modeling and Notation (BPMN). This will help to ensure the efficiency, effectiveness, and scalability of these processes.

The collected requirements and the BPMNs drawn from this task will be used to inform the development of the EO4EU platform in order to consider the various needs of the different user profiles that have shown interest in using the platform. The collaboration between T2.2, T2.1, T2.3 and T5.1 ensures a comprehensive understanding of both external and internal user requirements, facilitating the effective implementation of the EO4EU platform.

This deliverable is one of two deliverables in T2.2, which commenced in month 1 of the project (June 2022), and is expected to conclude in May 2024, spanning a duration of 24 months. The first deliverable of this task, D2.2 "EO4EU End-user Requirements Analysis and Business Process Flows," is scheduled for completion in month 18 of the project. The final deliverable, D2.3 "EO4EU End-user Requirements Analysis & Business Process Flows (b-final)," will mark the conclusion of Task 2.2.

The following sections of this deliverable will provide more details on the methods used, analyses, results, and conclusions of this task.

The rest of this document is organized as follows. Section 2 provides a detailed description of the adopted approaches to collect and analyze requirements coming from the three sources, namely, use case requirements, DoA requirements and external user requirements. Section 3 analyzes the collected requirements. BPMNs are depicted in Section 4. Finally, some conclusions are drawn in Section 5.

# 2 Requirements Collection

## 2.1 Use Case Requirements

The demands arising from the use case coordinators' requirements were assembled by SISTEMA and outlined in the deliverable D2.1 - Report on Research and Innovation Landscape Analysis. These requirements were developed using a questionnaire jointly created by EBOS and SISTEMA. Annex A (Chapter 10) of D2.1 includes the questionnaire template, consisting of four distinct sections addressing requirements for input data specification, data preparation, data processing, and methods for presenting and visualizing outcomes. Initially, the individual questionnaires were made available online to the use case leaders (comprising seven teams), enabling them to familiarize themselves with the provided sections. Subsequently, SISTEMA conducted personalized one-on-one meetings or teleconferences with the respective use case coordinators in a subsequent stage. This approach ensured the completion of the questionnaires, providing a comprehensive overview of the operational workflows within each unique use case scenario. In total, 292 specific requirements were gathered, covering the domains of Data, Data Preparation, Data Processing, and Results Presentation.

**Table 1. Overview of the requirements gathered from the use case conductors broken down by domain.**

| Domain | #Req. |
|---|---|
| **Data** | **108** |
| **Data preparation / pre-processing** | **101** |
| **Processing / algorithms** | **47** |
| **Results presentation** | **36** |
| **IPR-licensing** | **0** |
| **TOTAL** | **292** |

Concerning the Use Cases, additional requirements started to be collected outside of the Use Cases, although this process is still in progress. At the beginning of the Project an end-user map was created including the contacts of those institutions (academic, enterprises or single users) which had previously manifested their interest towards the Project by support letters, or additional users that were reached thanks to the extended network of the Use Case partners. Task 5.2 was particularly devoted to the collection of specific requirements for the Use Cases by interacting with such end-users by means of a questionnaire. In perspective, such requirements will drive some choices about how to best visualize and interact with the Use Cases of the Project. At this initial stage, “ice-breaking” questionnaire was conceived aiming at understanding some generical features useful as a first contact between the users and the project. The total number of participants is 21, and they are anonymously listed in Table 1 to show the repartition per Use Case.

**Table 1: Number of participants to the ice-breaking questionnaire.**

| Use Case ID | Use Case description | No. end-users |
|---|---|---|
| 1 | EO for Personalized Health Care Services | 4 |
| 2 | Ocean Monitoring | 1 |
| 3 | Food Security | 2 |
| 4 | Forest Ecosystems | 3 |
| 5 | Soil Erosion | 2 |
| 6 | Environmental Pests | 1 |
| 7 | Improving Civil Protection Activities Using EO Acquired Datasets | 8 |

## 2.2 DoA Requirements

The process of developing requirements involves four (4) generic activities, namely: requirements elicitation, requirements analysis, requirements specification, and requirements verification and validation. In practice, this process is tailored based on the domain or organization where it is applied.

Classically, requirements elicitation involves identifying stakeholders, their needs, and constraints, and methodically extracting requirements. The outcome of elicitation is a set of raw requirements that are likely to be incomplete, ambiguous, inconsistent, and largely documented in natural language (or with graphical notations that may or may not be augmented with natural language). Subsequently, the raw requirements are analyzed to resolve the ambiguities, inconsistencies, and conflicts. Requirements analysis is considered a critical step to the success of a systems or software project. It should lead to requirements that are well documented, measurable, testable, and traceable to subsequent outputs (i.e., architecture design documents) of the project. Requirements should be defined to a level of detail sufficient for the system design that will follow.

Following the analysis phase, we generate a requirements specification document. This document encapsulates the analyzed requirements in various acceptable formats such as natural language (widely used in current practices), formal structures, or graphical representations. The requirements specification then undergoes validation, where it is confirmed that the specification is an accurate, correct, and complete statement of the stakeholders’ needs.

In fact, this progression of activities described above, abstractly represents the requirements development process, the outcome of which is a requirements specification that exhibits a set of desired quality properties e.g., those recommended by the various IEEE standards.

This progression of activities may be linearly ordered, in a waterfall-like process[1], as shown in Figure 1. Here, the activities are serially ordered such that requirements elicitation and analysis is followed by specification and preceded by a feasibility study. After specification, requirements validation occurs, and the outcome of the process is a requirements document containing a model of the system to be built, user and system requirements.

[1] I. Sommerville; Software Engineering, 7th Ed.; John Wiley and Sons, 2004.

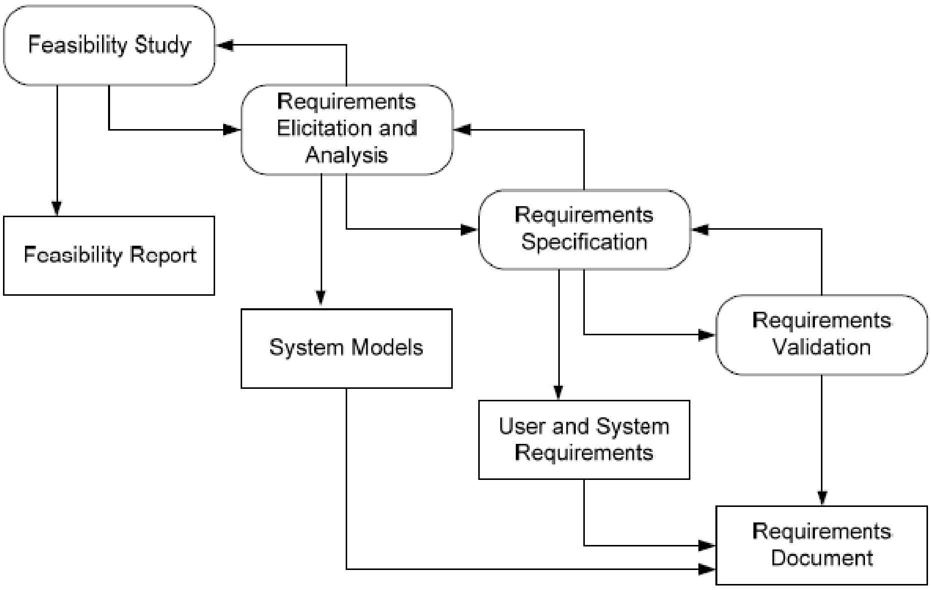


**Figure 1: Waterfall model of RE process.**

In practice the process may not follow such a strict linear progression; rather, elicitation is often observed to intertwine and iterate with analysis, specification, and validation. Besides these, some supporting activities exist for requirements development. Briefly, they include requirements discovery, classification, prioritization, and negotiation. Requirements discovery is performed, in part, during elicitation where interaction with stakeholders occurs to clarify user needs and discover requirements arising from the domain. During classification of requirements, coherency among requirements is achieved by organizing them according to the identified classification categories. Subsequently, prioritization and negotiation of requirements assists in identifying and resolving requirements conflicts.

In many cases, the creation of a complete requirement specification before moving to the next phases of system or software design is a pretty difficult task due to the inherent system complexity or the fact that certain constraints or issues will not become evident until development activities start or even a first version of the system is put in operation. In such situations acceptable best practices mandate the combination of linear and iterative systems development methodologies, with the primary objective of each being to reduce inherent project risk by breaking a project into smaller segments and providing more ease-of-change during the development process. Such processes are described as incremental or evolutionary.

The EO4EU project prescribes the use of two iterations during platform development, therefore it seems that an evolutionary development life cycle process is well suited. During each iteration, a requirement specification is provided that can be used for the delivery or evolution of certain operational capabilities of the final product which can be put to operation. Feedback will be provided that may lead to modifying or extending requirements in subsequent iterations. Each delivery in this model represents a full development cycle, including requirements analysis. The deliveries may overlap, as shown in Figure 2 or one delivery may be completed before the next is begun. The product of each requirements analysis phase is an addition or improvement to the product(s) of the requirements analysis phase of the previous delivery. Similarly, the implementation portions of each delivery may add to, or upgrade, products of earlier deliveries. With this understanding, each delivery may be looked at as a small example of a baseline management life cycle, with a development process and time span small enough to minimize the problems discussed above.

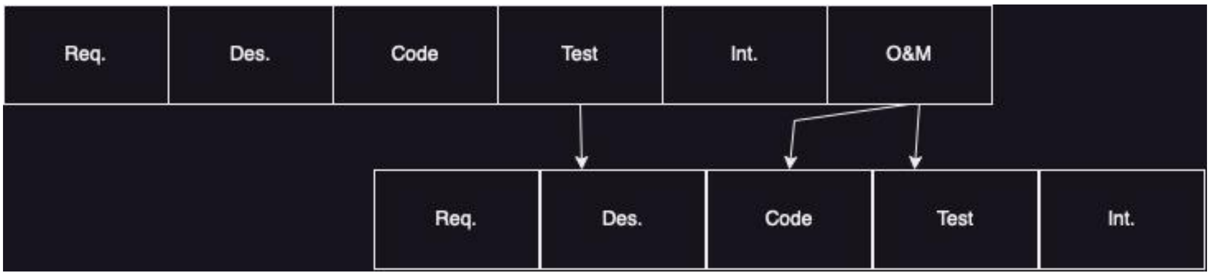
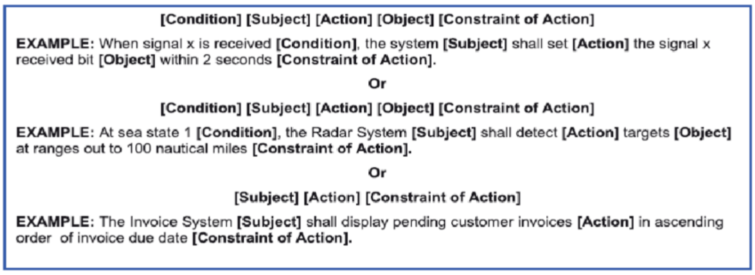


**Figure 2: Simplified schema of Evolutionary Development model.**

In modern requirement analysis, appropriate requirement management tools are often employed to facilitate:

- The definition of requirements using certain patterns or wordings (often based on predefined vocabulary or a domain specific ontology). This enables the creation of well-formed requirements (semi-formal or formal) which can be further supported by formal languages. The advantage is obvious. Requirements become clearer to the potential reviewer; tools can be applied for validating their correctness and consistency while identification of inconsistencies, ambiguities, missing requirements, and requirements conflicts can be facilitated. Figure 3 below presents an example style for writing requirements according to ISO WD 29148:2018 standard[2].



**[Condition] [Subject] [Action] [Object] [Constraint of Action]**

**EXAMPLE:** When signal x is received **[Condition]**, the system **[Subject]** shall set **[Action]** the signal x received bit **[Object]** within 2 seconds **[Constraint of Action]**.

**Or**

**[Condition] [Subject] [Action] [Object] [Constraint of Action]**

**EXAMPLE:** At sea state 1 **[Condition]**, the Radar System **[Subject]** shall detect **[Action]** targets **[Object]** at ranges out to 100 nautical miles **[Constraint of Action]**.

**Or**

**[Subject] [Action] [Constraint of Action]**

**EXAMPLE:** The Invoice System **[Subject]** shall display pending customer invoices **[Action]** in ascending order of invoice due date **[Constraint of Action]**.

**Figure 3: Examples of requirements statement construction styles recommended by ISO WD 29148:2011**

The standard recommends the usage of the words ‘shall’ and ‘should’ to be used to express whether a requirement is mandatory or not and exhorts the avoidance of vague or ambiguous terms and subjective language.

[2] International Organization for Standardization; ISO/IEC WD 29148.3; Software and Systems Engineering – Life Cycle Processes – Requirements Engineering, 2018.

- The linking and traceability of requirements to design and validation documents like architecture design, detailed design, test cases etc. This makes the detection of requirements that were overseen, not fulfilled, or not tested/verified much easier.

In EO4EU although we do not impose the use of a proper Requirement Management tool or strict formalization of requirements, we do propose to follow some general principles and adopt a common template for gathering and eliciting requirements from all the partners involved in this project. The VOLERE requirements specification template[3] has been chosen as a start point, adapting it to the particularities and needs of EO4EU.

Each of the partners involved in the project follows their own processes in the requirements definition phase of their activities. It was important to propose a common way to formalize the requirements that is easy to use and adapted to the needs of the project. In this context, VOLERE is a straightforward methodology that does not require a complex analysis to be applied. Furthermore, it guarantees the participation of all relevant actors, who are further involved in the design and development that must fulfill the requirements defined.

The adapted methodology used by EO4EU, will allow the identification and formalization of unambiguous requirements while it will ease the subsequent assessment and validation processes, during Architecture Definition, Development and Evaluation. It is important to make use of a common methodology to gather, classify and assess the requirements a priori. The management of the requirements depends on this common methodology, providing the means to trace the identification, definition, assessment, formalization and if necessary, improvement of the requirements gathered.

Moreover, requirements should be the key to evaluate the entire project at the end of the development phase. A set of well-defined and unambiguous requirements is needed, not only as input for any further specification and development, but also as part of the evaluation framework.

VOLERE defines the gathering process and the shell to register the requirements, classified in 27 categories in 5 main groups:

- **Project drivers**, the business-related forces. For example, the purpose of the project is a project driver, as are all the stakeholders - each for different reasons.
- **Project constraints**, restrictions on how the product must be designed. For example, it might have to be implemented in the hand-held device being given to major customers, or it might have to use certain existing servers and desktop computers, or any other hardware, software, or business practice.
- **Functional requirements**, the fundamental or essential subject matter of the product. They describe what the product must do or what processing actions it is to take.

[3] Volere Template Edition 20, 2023, https://www.volere.org/templates/volere-requirements-specification-template/

- **Non-functional requirements**, the properties that the functions must have, such as performance, security, and usability.
- **Project issues**, the conditions under which the project will be done. The reason for including them as part of the requirements is to present a coherent picture of all factors that contribute to the success or failure of the project and to illustrate how managers can use requirements as input when managing a project.

VOLERE methodology is a universal way to describe requirements. In the first version of the requirements deliverable, we expect to focus on user requirements more from user needs perspective rather than system requirements. Of course, it is not possible to present only user view on the EO4EU project because users are not aware of many details, constraints, and other assumptions. Therefore, for the first version of the Requirement Analysis document, we will deliver a list of user requirements prescribed by a set of potential scenarios and/or use cases as well as high level functional and non-functional requirements grouped in two major classes based on EO4EU envisaged architecture:

- Functional Requirements
- Non – Functional as Data Requirements

The first class includes requirements related to the sever side part of EO4EU that is middleware, front-end and cluster tier while the second class includes any kind of requirement that has to do with EO data needs from the users. For each of the above categories a further subgrouping may occur while requirement analysis and architecture definition proceed in detail and more components and/or modules are identified.

The VOLERE template prescribes description of a particular requirement with following characteristic:

- *Requirement Numbering*. Give each requirement a unique identifier to make it traceable throughout the development process. The numbering scheme suggested in the requirement shell is as follows:
    - Requirement number is the next unique requirement number.
    - Requirement Type is the section number from the template for this type of requirement. The inclusion of the section number is not necessary because we do have a unique requirement identifier. However, it serves as a reminder of what this requirement relates to and helps to remind us why the requirement is considered important. In addition, the ability to compare requirements of the same type makes it easier to identify contradictions and duplications.
- *Event/use case #* is the identifier of a business event or use case that contains this requirement. There might be several Event/use case #'s for one requirement because the same requirement might relate to several events. The terms event and use case are already widely used in the systems development world.
- *Customer Value* is a measure of how much your client cares about each requirement. Customer should grade each requirement for Customer Satisfaction or Customer Dissatisfaction. The point of having a satisfaction and a dissatisfaction rating is that it guides clients to think of the requirement from two different perspectives and helps

to uncover what they care about most deeply. Measurable criteria are expected to be developed and described in D5.7 “User acceptance impact assessment report”.

- *Dependencies* keep track of other requirements that have an impact on this requirement. If the dependency exists because requirements use the same information, then use of standard naming conventions and definitions will implement this dependency. Other dependencies exist because a solution to this requirement has a positive or negative effect on solutions to other requirements. Some requirements, especially project drivers and project constraints, have an impact on all the other requirements.
- *Conflicts* keep track of other requirements that disagree with this one.

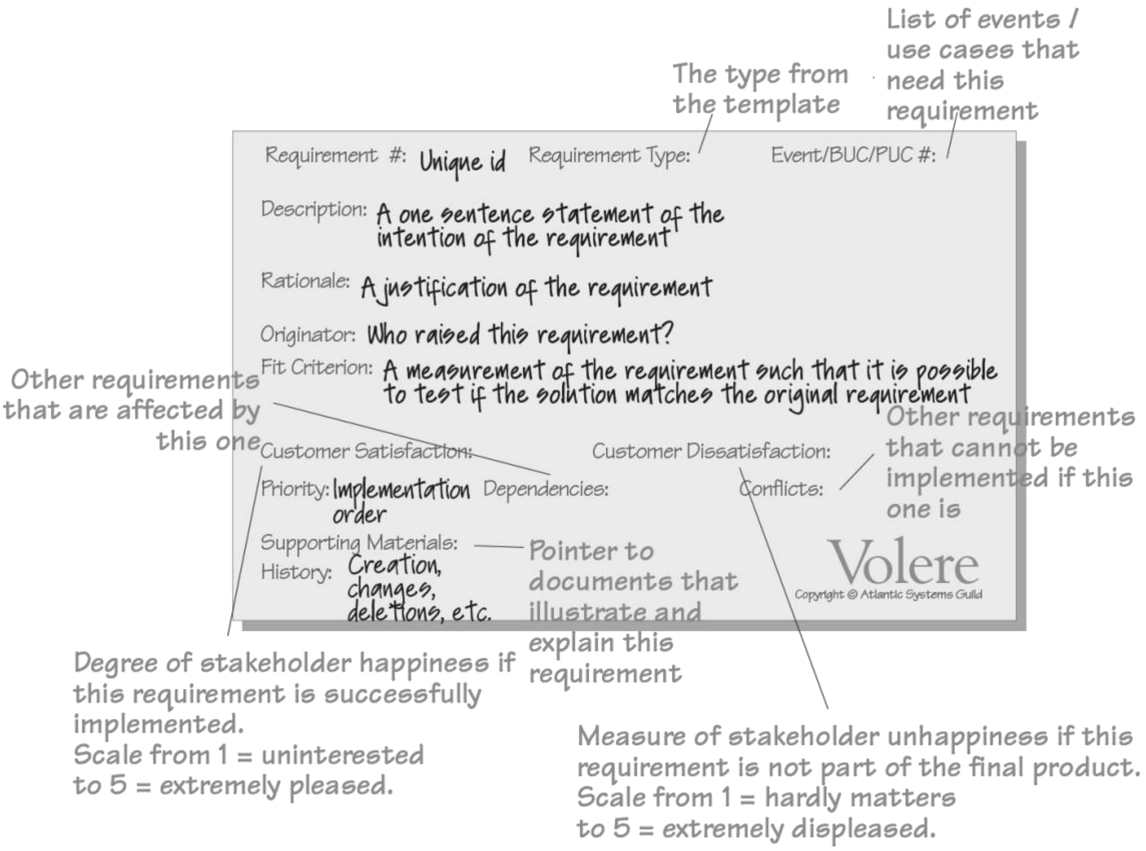

List of events / use cases that need this requirement
The type from the template
Requirement #: Unique id
Requirement Type:
Event/BUC/PUC #:
Description: A one sentence statement of the intention of the requirement
Rationale: A justification of the requirement
Originator: Who raised this requirement?
Fit Criterion: A measurement of the requirement such that it is possible to test if the solution matches the original requirement
Other requirements that are affected by this one
Customer Satisfaction:
Customer Dissatisfaction:
Other requirements that cannot be implemented if this one is
Priority: Implementation order
Dependencies:
Conflicts:
Supporting Materials:
History: Creation, changes, deletions, etc.
Pointer to documents that illustrate and explain this requirement
Volere
Copyright © Atlantic Systems Guild
Degree of stakeholder happiness if this requirement is successfully implemented. Scale from 1 = uninterested to 5 = extremely pleased.
Measure of stakeholder unhappiness if this requirement is not part of the final product. Scale from 1 = hardly matters to 5 = extremely displeased.

**Figure 4 Requirement card template based on Volere (extract from Volere Template Edition 20 —August 2023)**

In this project, a more simplified card is proposed compared to the one presented in Figure 4. This is presented in Table 2-Table 3 below.

**Table 2: Exemplary Requirement Card used in this Deliverable.**

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Id:** | {Class Id} -001 | **Type:** | follow categorization proposed in Volere template<br>FUNC | **Importance (priority):** | MEDIUM | **Source:** | Requirement origin e.g., Consortium Know-how, members, law regulation, standards etc.<br>DoA | **Ver:** | 1 |
| **Title:** | Requirement title/name (1 sentence) | | | | | | | | |
| **Description:** | More detailed description of particular requirement (textual form only).<br>If Requirement title is sufficient enough to understand the requirement, this filed can remain empty. | | | | | | | | |
| Additional Info (comments): | Any additional info to better clarify or illustrate concepts (pictures may be possible). | | | | | | | | |
| Related Scenario(s) | | | | | | | | | |
| Related Component | | | | | | | | | |

**Table 3: List of Requirements Types.**

| | | |
|---|---|---|
| **Functional** | Functional | FUNC |
| | Data | DATA |
| **Non-functional:** | Look and Feel Requirements | L&F |
| | Usability Requirements | USE |
| | Performance Requirements | PERF |
| | Operational - Environmental Requirements | ENV |
| | Maintainability and Support Requirements | SUP |
| | Security Requirements | SEC |
| | Other | OTH |

## 2.3 External User Requirements

The EO4EU platform will provide a single point of access to EO data not only for the use cases of the project but also for a wide range of users including researchers, engineers, students, and policymakers, among others. To account for the needs of these users, T2.2 has reached out to various user profiles using an online survey. This survey focuses on important aspects of the platform such as data accessibility, findability, computing requirements, platform access requirements, algorithms, etc. To increase the engagement of potential users, an early access to the platform was offered to those participating in the online survey. This early access will allow these users to test the platform using realistic scenarios and give their feedback of their experience on the platform, which will allow to refine the user requirements and produce the final version of user requirements and BPMN, which will be reported in D2.3.

Please note that the activities involved in the external user requirements collections are not isolated steps but part of an integral effort with WPs 3 and 4, aiming to create a platform that responds to the needs of a wide variety of users.

### *2.3.1 External user requirements collection and analysis sub-tasks*

The external user requirements task encompasses several sub-tasks, each contributing to the overall objective. Figure 5 depicts the logical flow of these sub-tasks. More details are provided in the following sections.

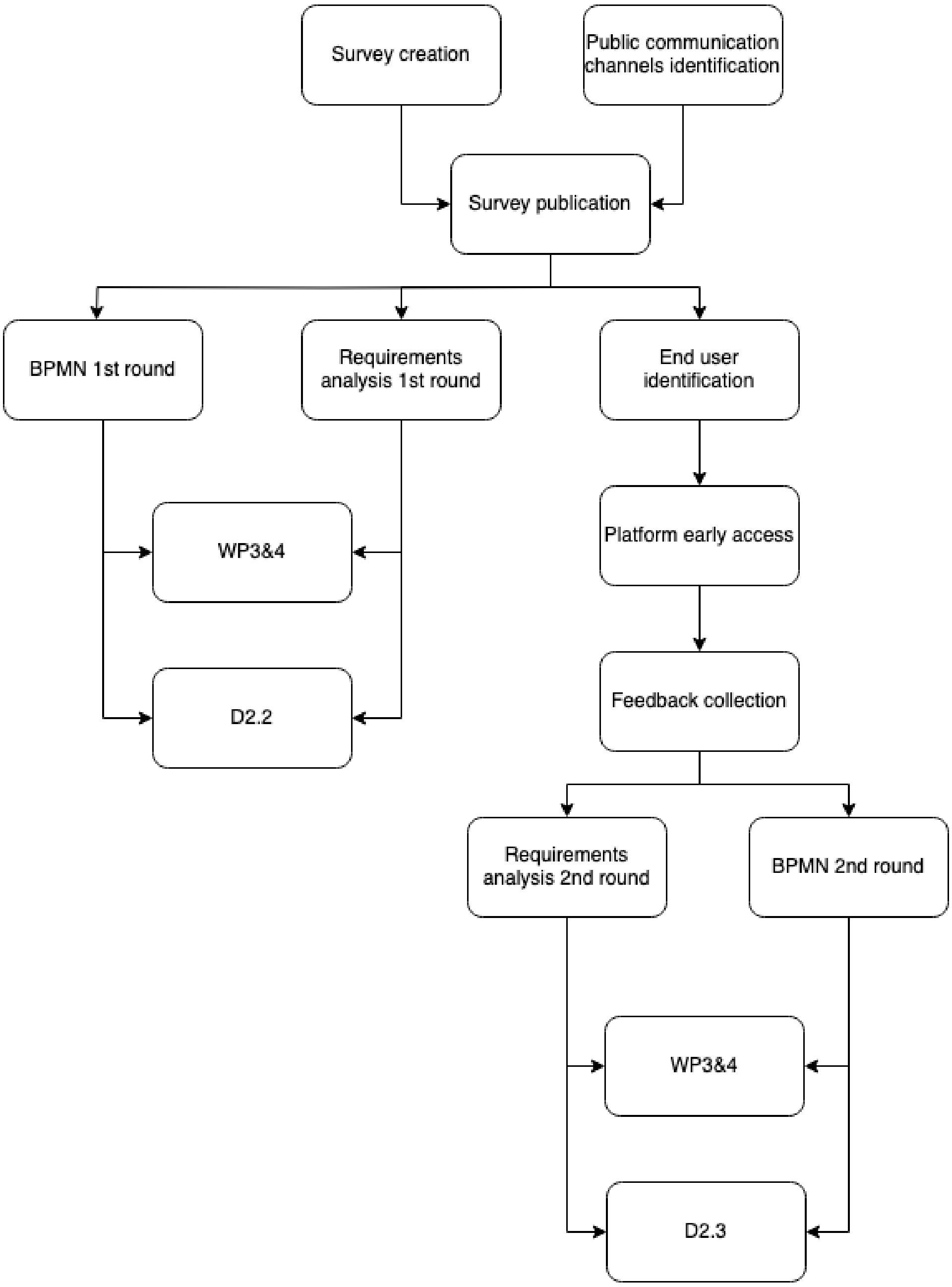


**Figure 5: External user requirements collection and analysis approach.**

### 2.3.1.1 End-user survey creation, review, and finalization

The first step involved was the creation of an end-user survey, designed to gather and assess external user needs. This survey underwent a rigorous review process with the other partners in Work Package (WP) 2 to ensure its effectiveness and was subsequently finalized. Google Forms tool was used to implement the survey, which provides as easy dissemination tool with a user-friendly interface and graphic analysis capabilities [4].

[4] https://forms.gle/hwCkSuPbJHyugRKH9

The survey focuses on multiple aspects that can be summarized as:

- Data requirements
- Storage and access
- Data search and indexing
- Hardware and other processing capabilities
- Algorithms
- Interfacing services
- Authentication and authorization

The questions in the following table help gather user requirements related to these aspects.

**Table 4: External user survey questions**

| Questions |
|---|
| What is your professional/academic position? |
| How familiar are you with Earth Observation data and analysis? |
| If you are familiar with EO data, which data sources have you already used? |
| If you are familiar with EO data analysis, do you use a specific platform to perform data search and processing? |
| What is the market/research segment your use case belongs to? |
| Which datasets do you think will be potentially useful for your use case? |
| How long is the time series data required for the use case? |
| What is the spatial extent of the analysis in your use case? |
| What is the expected size of your dataset? |
| What will you use the available storage associated with your account for? |
| Will the processing results be only accessible by you? or are they open access?<br>or would you like to share them with specific users? |
| How would you like to search for datasets? |
| How do you intend to search for data? |
| Which of these metadata items do you think will be useful when you search for data? (Is it possible to select multiple options?) |
| Do you think that having a two-level metadata system, Collection (dataset) and Granule (file-level metadata) will help improve the discoverability and accessibility of the available data? |

| |
|---|
| What would the use case require in terms of computing resources for data processing? |
| What would the use case require in terms of the available RAM resources? |
| Does the use case require data fusion from different sources? |
| Which are the data sources that you would like to use for the fusion? (e.g., Fusion of Sentinel 1 and Sentinel 2 data can be exploited for agriculture use cases) |
| Does the use case require the observation data (satellite measurements) to be downloaded? or meaningful representations of the data can be enough? For example, to assess plant vigor, instead of downloading all Sentinel-2 bands, the NDVI can be used. |
| Does the use case require real-time or near real-time analysis? |
| Will you use the platform to run the whole processing procedure? |
| Do you expect to access the intermediate processing results? |
| Do you prefer to manually select the data sources required for data fusion? |
| How would you like to interact with the different platform components?<br>(Is it possible to select more than one option?) |
| Do you need to build your own code and processing pipelines on the platform? |
| If your answer to the previous question is yes, what tools do you need to build your own code? |
| Following the two previous questions, do you need to create your code directly on the GUI or do you want to import external code or both (you can choose this by checking both answers)? |
| What is the best way of authenticating the Application Programming Interface (API) calls to EO4EU platform that best works with your institute's system requirements? |
| How would you like to visualize the results? |
| When would you like the analysis results to be delivered? |

#### 2.3.1.2 Public Communication Channels Identification

The identification of appropriate public communication channels was a crucial step to disseminate the survey effectively and reach a wide audience. Multiple channels were identified in collaboration with partners from WP5 and WP6. More precisely, the survey was published on the project's Linkedin page [5], official website [6], Twitter [7], and via email communication with potential users. Moreover, partners in WP5 and WP6 helped disseminate this survey in their networks. By involving diverse end-users, a comprehensive range of perspectives could be captured.

#### 2.3.1.3 Survey Publication

The survey was then published following the public communication channels identification to maximize its reach and encourage user participation. The deadline of submitting the responses was set to 14th April 2023.

#### 2.3.1.4 End-User Identification

After the deadline of response submission, potential end users of the platform were identified. These users could provide valuable insights through their participation in the survey and will be offered early access to the platform, which will allow collecting their feedback and refine user requirements.

#### 2.3.1.5 First Stage Requirements Collection and Analysis

Following the deadline of the survey, a thorough process of requirements collection and analysis took place. This initial round of data collection formed the foundation for subsequent analysis and refinement. Section3.3 provides more details on the analysis of the external user requirements.

#### 2.3.1.6 First Stage BPMN

Simultaneously with the requirements collection, a preliminary version of the BPMN flowcharts was developed. These flowcharts aimed to depict the interaction between end-users and the platform in various scenarios. As the project progressed, these flowcharts would continue to evolve and be updated accordingly.

#### 2.3.1.7 Facilitate End-User Access for Testing

As previously mentioned, early access to the platform will be offered to those participating in the external user requirements survey, to encourage active participation and obtain valuable feedback. Survey participants will be granted access to the EO4EU platform, allowing them to test its features and capabilities. The insights and feedback gathered through this early access will be instrumental in refining the final set of requirements.

#### 2.3.1.8 Second Stage End-User Feedback Collection, Requirements Refinement, and Analysis

Building upon the initial stage of external user requirements collection and analysis as well as the early access to the platform, the second round of feedback collection will be conducted. This stage will focus on further refining and analyzing the user requirements based on the feedback received.

[5] https://www.linkedin.com/posts/eo4eu_earth-activity-7052196240322625536-TZ0y

[6] https://eo4eu.eu/news/gain-early-access-eo4eu-platform

[7] https://twitter.com/EO4EU/status/1646430295837532162?s=20

### 2.3.1.9 Final Version of BPMN

The BPMN flowcharts' final version will be created in accordance with the clarified specifications. These flowcharts will offer a thorough and complete representation of the business process flows involving end users, taking into account their unique needs and requirements and describing the interaction between the end user and the platform.

## 2.4 Consolidation of User Requirements and Collaboration

Currently, ECMWF is collaborating with NKUA (T2.3), SISTEMA (T2.1), and CMCC (T5.1) to consolidate all user requirements, including internal requirements exhibited by the project's use cases, DoA requirements and external user requirements. This collaborative effort aims at providing a single reference for all the requirements, assigning responsibility to the relevant technical partners and following up with the satisfaction of these requirements. The consolidated table of all requirements is shown in D2.4.

# 3 Requirements Analysis

## 3.1 Use Case Requirements

The work performed during Task 2.1 and summarized in D2.1, allowed analyzing the technical needs of the use cases.

The 108 data requirements allowed breaking down the data needs as described in Figure 6

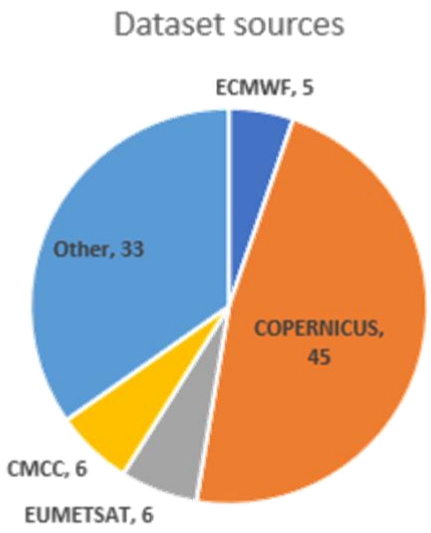


**Figure 6: Summary of the data source needs.**

For what regards data preparation and pre-processing (101 collected requirements), Table 5 shows the summary of the data accessibility and exploitability needs per use case.

**Table 5: Summary of data accessibility and exploitability requirements.**

| | UC1 | UC2 | UC3 | UC4 | UC5 | UC6 | UC7 | # UCs |
|---|---|---|---|---|---|---|---|---|
| Data download | X | X | X | X | X | X | X | **7** |
| remote data subsetting | | X | X | X | X | X | | **5** |
| Spatial subsetting / interpolation | | | X | X | X | X | X | **5** |
| temporal subsetting aggregation | | | | X | | X | X | **3** |
| processing close to data | D | | D | D | D | D | D | **0** |
| Synergistic exploitability | X | | X | X | X | X | X | **6** |

Hardware needs are summarized in Table 6 while algorithm capability needs are provided in Table 7.

**Table 6: Summary of computational needs per use case.**

| | UC1 | UC2 | UC3 | UC4 | UC5 | UC6 | UC7 | # UCs |
|---|---|---|---|---|---|---|---|---|
| CPU | X | X | X | X | X | X | X | **7** |
| GPU | | X | X | | | X | | **3** |
| Elastic processing | X | X | | | | X | | **3** |
| high computational resources | | | X | | X | | | **2** |
| Scale up possibility | | X | X | X | X | X | X | **6** |

**Table 7: Summary of algorithm capabilities need.**

| | UC1 | UC2 | UC3 | UC4 | UC5 | UC6 | UC7 | # UCs |
|---|---|---|---|---|---|---|---|---|
| waterfall | X | X | X | X | X | | X | **6** |
| AI processing | | D | D | | | **X** | | **3** |
| Automatic processing | X | X | | X | | X | X | **5** |
| Maturity level (TRL) | 7 | 9 | 4 | 7 | 4 | 4 | 9 | **6,3 (avg)** |
| 4. prototype | | | X | | X | X | | **3** |
| 7. pre-operational | X | | | X | | | | **2** |
| 9. Operational | | X | | | | | X | **2** |

Data presentation and provision needs (36 requirements) are summarized in Table 8.

**Table 8: Summary of data presentation need.**

| | UC1 | UC2 | UC3 | UC4 | UC5 | UC6 | UC7 | # UCs |
|---|---|---|---|---|---|---|---|---|
| Web app | X | X | X | X | X | X | | **6** |
| API | | X | | | | X | | **2** |
| Mobile APP | X | X | | | | | | **2** |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Data download | | | | X | X | X | | **3** |
| resuts push | | | | | | | X | **1** |

Figures from Figure 7 - Figure 17 visually show the results of the Ice-breaking questionnaire filled by the end-users of the Use Cases. Each figure reports the question and a summary of the results. The visualization is heterogeneous among the questions, which are quite different from each other and are often associated with open answers. The Ice-breaking questionnaire particularly aims to understand the general composition of the end-users and their ability/willingness to interact with the Project platform at different levels for their operations. For some questions the number of possible answers is so large that not all of them can be visualized: in such cases the most frequent answers are always highlighted.

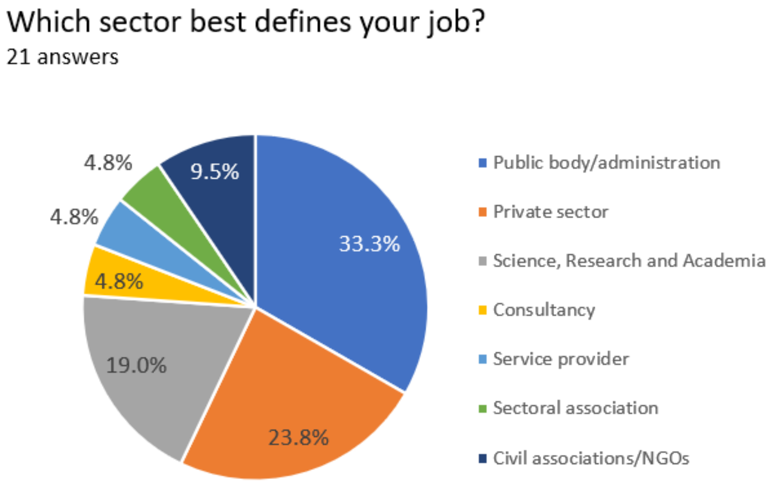


**Figure 7: Question 1 of the ice-breaking questionnaire.**

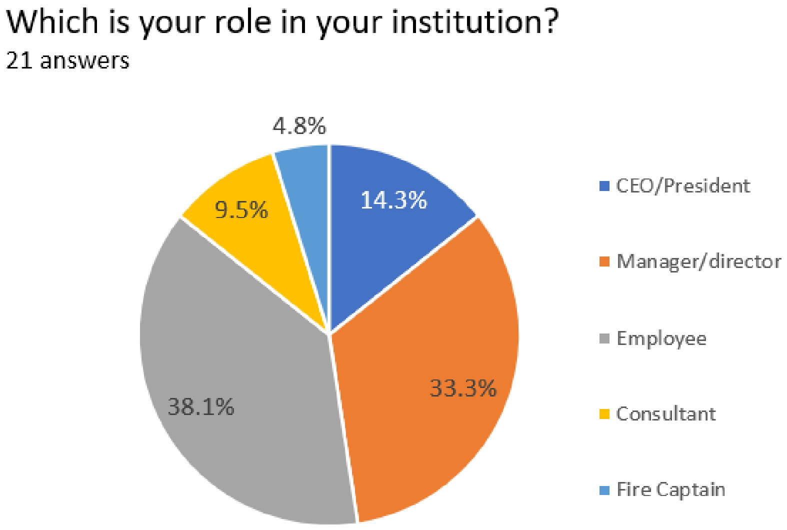


**Figure 8: Question 2 of the ice-breaking questionnaire.**

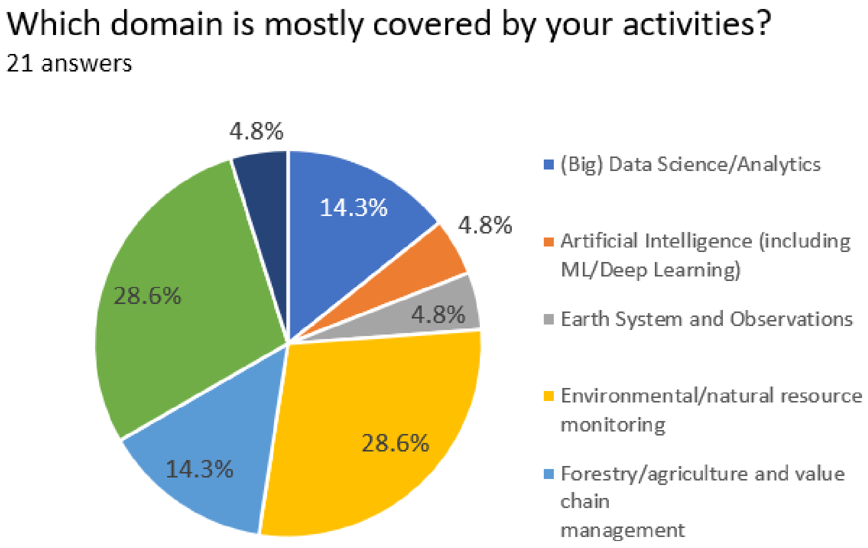


**Figure 9: Question 3 of the ice-breaking questionnaire.**

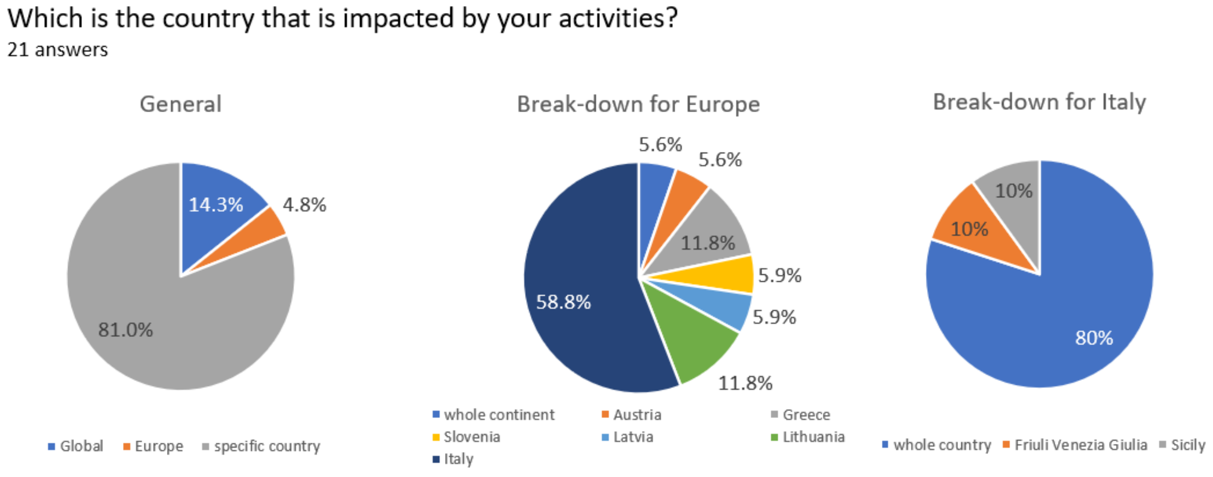


**Figure 10: Question 4 of the ice-breaking questionnaire.**

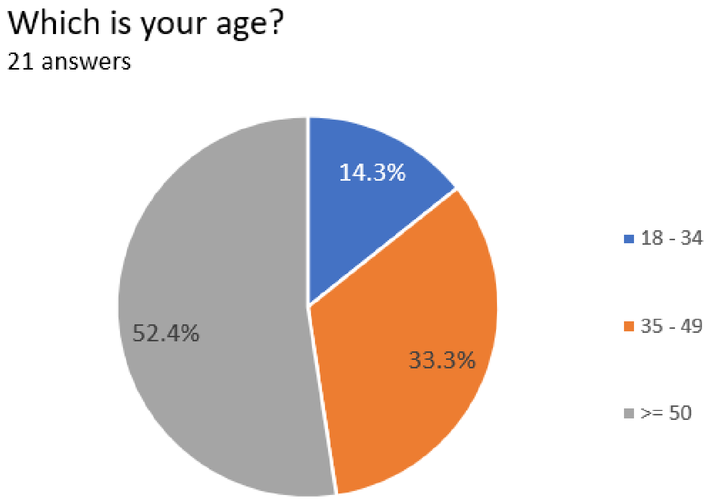


**Figure 11: Question 5 of the ice-breaking questionnaire.**

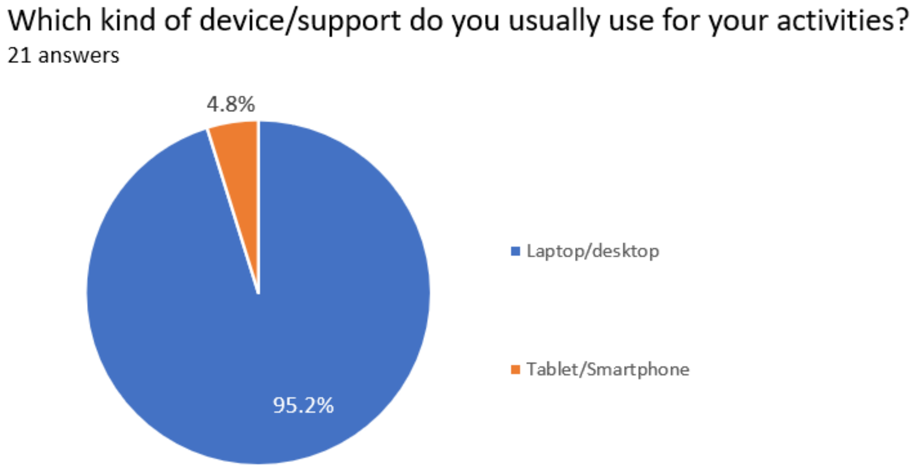


**Figure 12: Question 6 of the ice-breaking questionnaire.**

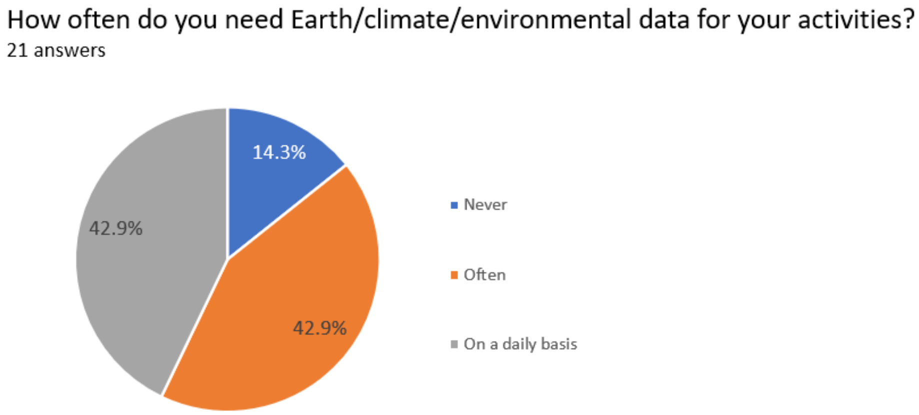


**Figure 13: Question 7 of the ice-breaking questionnaire.**

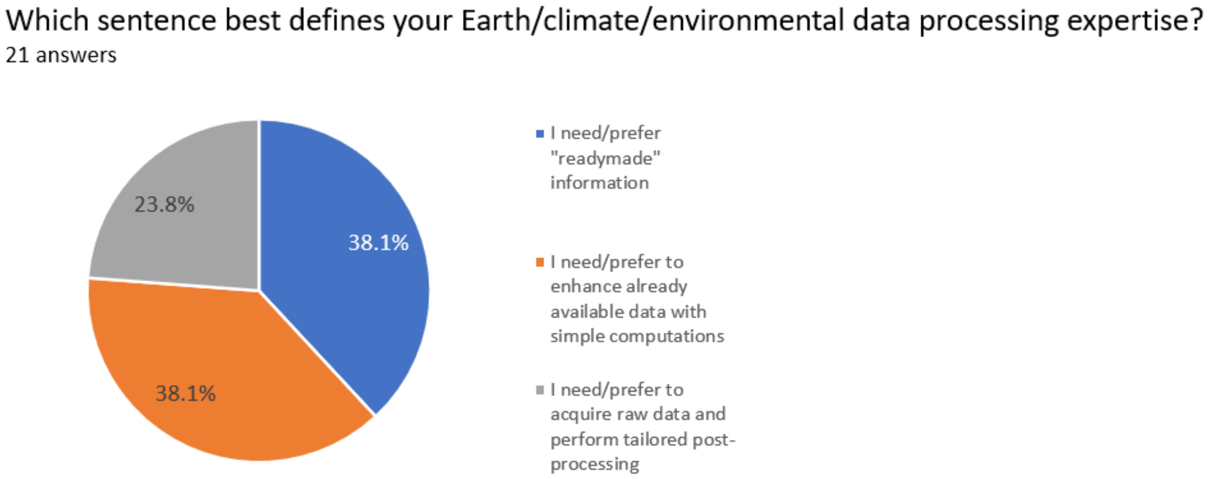


**Figure 14: Question 8 of the ice-breaking questionnaire.**

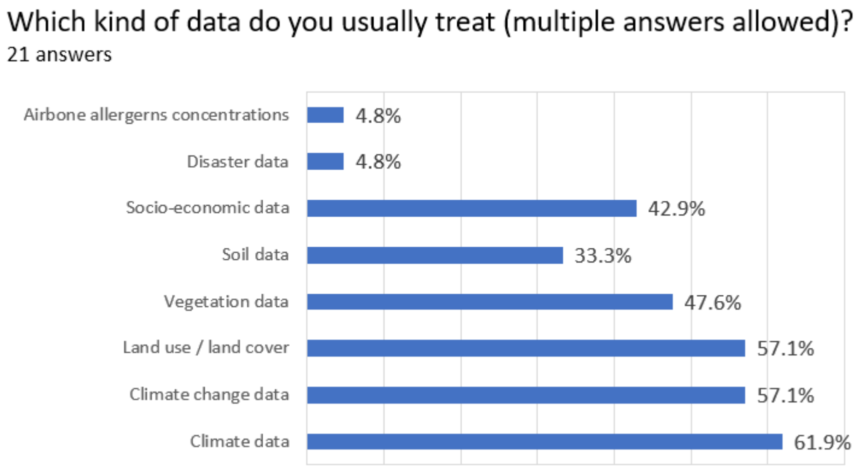


**Figure 15: Question 9 of the ice-breaking questionnaire.**

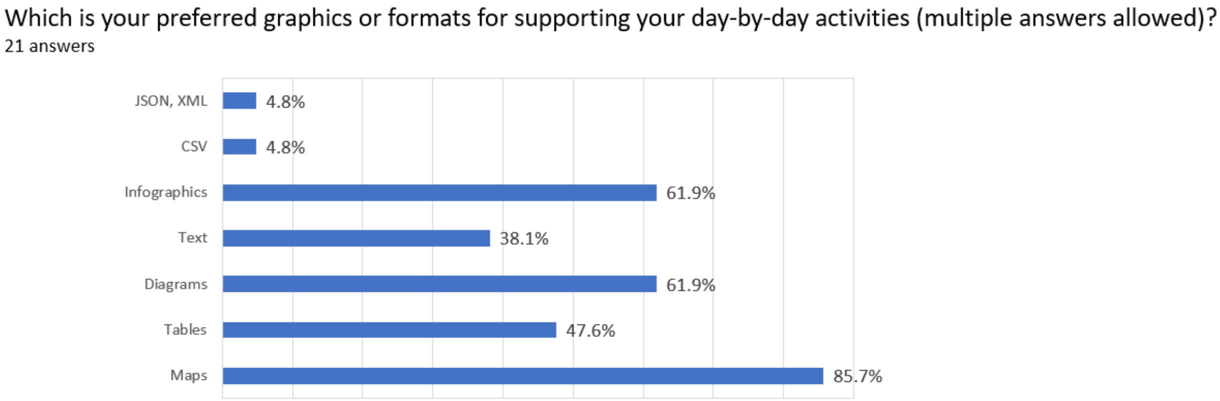


**Figure 16: Question 10 of the ice-breaking questionnaire.**

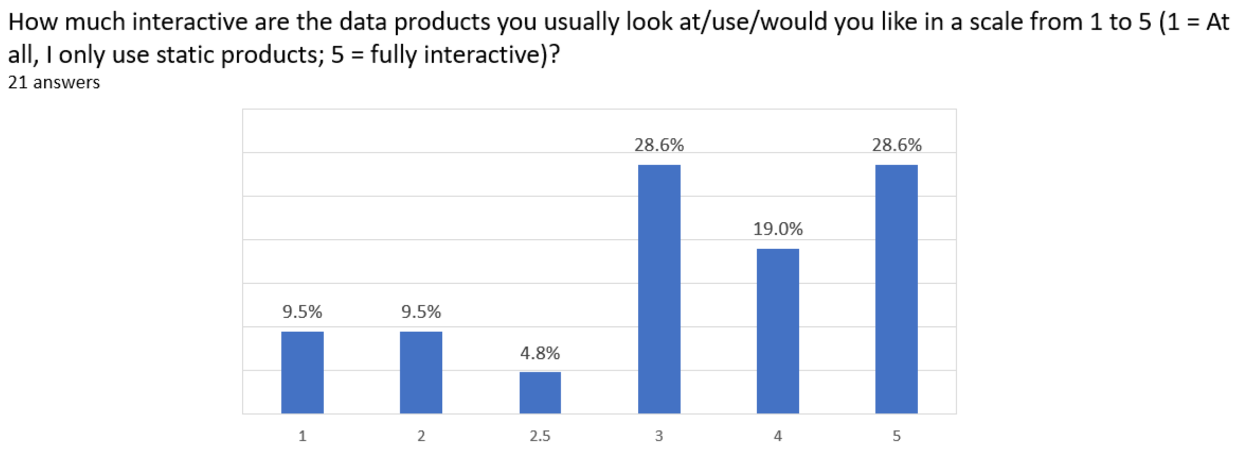


**Figure 17: Question 11 of the ice-breaking questionnaire.**

## 3.2 DoA Requirements

This section includes all the requirements coming from the description of DoA following the VOLERE template as described and analyzed in Section 2.2.

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Id:** | SS-KG-01/ NVCR | **Type:** | FUNC | **Importance (priority):** | HIGH | **Source:** | DoA | **Ver:** | 1 |
| **Title:** | | To augment the FAIRness of EO data knowledge | | | | | | | |
| **Description:** | | The integration of existing and newly created earth observation related datasets with the EO4EU Knowledge Graph will further enhance the FAIRification process of EO data at a EU level. The knowledge graph provides a structured and interconnected framework to organize, link, and semantically enrich the data that bridges the gap between the technical aspects of data and the needs of the end-users. | | | | | | | |
| Additional Info (comments): | | N/A | | | | | | | |
| Related Scenario(s) | | Applicable horizontally to all use cases | | | | | | | |
| Related Component | | Knowledge Graph | | | | | | | |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Id:** | SS-KG-02/ NVCR | **Type:** | FUNC | **Importance (priority):** | MEDIUM | **Source:** | DoA | **Ver:** | 1 |
| **Title:** | To support a sophisticated representation of EO data entities and their dynamics | | | | | | | | |
| **Description:** | We introduce a Graph-Based Text Representation in which each document is represented as a single graph. Extracted features that directly or indirectly are associated with a dataset include structural or textual information. By representing data as nodes and relationships as edges, knowledge graphs provide a flexible and scalable structure that can accommodate diverse types of data, including satellite imagery, sensor measurements, environmental variables, and geospatial information. | | | | | | | | |
| Additional Info (comments): | N/A | | | | | | | | |
| Related Scenario(s) | Applicable horizontally to all use cases | | | | | | | | |
| Related Component | Knowledge Graph | | | | | | | | |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Id:** | SS-KG-03 /NVCR | **Type:** | FUNC | **Importance (priority):** | MEDIUM | **Source:** | DoA | **Ver:** | 1 |
| **Title:** | To semantically relate different resources and descriptions | | | | | | | | |
| **Description:** | The main advantage of knowledge graphs is their ability to integrate and harmonize disparate data sources. Earth observation data comes from a wide range of providers, each with its own formats, schemas, and standards. Storing this data in a knowledge graph allows for seamless integration, ensuring consistency and interoperability across different | | | | | | | | |

| | |
|---|---|
| | datasets. By mapping data to a common schema and establishing relationships, the knowledge graph enables researchers, policymakers, and stakeholders to access and analyze integrated data from multiple sources in a unified and coherent manner. |
| Additional Info (comments): | N/A |
| Related Scenario(s) | Applicable horizontally to all use cases |
| Related Component | Knowledge Graph |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Id:** | SS-KG-04/NVCR | **Type:** | FUNC | **Importance (priority):** | MEDIUM | **Source:** | DoA | **Ver:** | 1 |
| **Title:** | To investigate and map data and services transparently and efficiently | | | | | | | | |
| **Description:** | EO4EU's facilitates the accessibility of EO datasets through semantically enhanced access points, enabling users to search, identify, browse, test and exploit EO data and environmental related knowledge. | | | | | | | | |
| Additional Info (comments): | N/A | | | | | | | | |
| Related Scenario(s) | Applicable horizontally to all use cases | | | | | | | | |
| Related Component | Knowledge Graph | | | | | | | | |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Id:** | SS-KG-05/ NVCR | **Type:** | FUNC | **Importance (priority):** | MEDIUM | **Source:** | DoA | **Ver:** | 1 |

| | |
|---|---|
| **Title:** | To merge descriptions/ontologies by expressing semantic information |
| **Description:** | Graph data modelling and schema design for Earth observations involves representing Earth observation data, entities, attributes, and relationships in the form of a graph structure. It involves analyzing the different data sources, entities, and relationships that are relevant for the domain and the goal of the graph. The design of the models may involve the reuse or engineer ontologies, which are formal specifications of the concepts and their properties and relations in a domain, as well as the rules and constraints that govern them.<br>Mainly it involves the use of standards like RDF Schema and OWL to formalize the data model using a graph-structured data model or topology, where nodes are entities and edges are relationships between them. It can be beneficial to use semantic web technologies, such as URIs, vocabularies, and linked data principles, to enable interoperability, reusability, and integration of the data across different sources and formats. |
| Additional Info (comments): | n/a |
| Related Scenario(s) | Applicable horizontally to all use cases |
| Related Component | Knowledge Graph |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Id:** | SS-KG-06/ NVCR | **Type:** | FUNC | **Importance (priority):** | HIGH | **Source:** | DoA | **Ver:** | 1 |
| **Title:** | To conduct queries to discover EO resources that match users' requirements | | | | | | | | |
| **Description:** | Users will have the capacity to query the Knowledge graph in a three-step approach:<br>Step 1: The user makes a search about a query (i.e., soil erosion; a query, "soil erosion"). The KG will result the most similar datasets in terms of the description of the dataset. | | | | | | | | |

| | |
|---|---|
| | Step 2: The user chooses one of them, which he considers more useful, and sends a second request to the KG. As a response, the prioritization of the particular dataset into products, features, options is returned.<br>Step 3: The user selects the items he is interested in and sends a third and final request to the KG. As a response, the code corresponding to step 2 is returned with the appropriate API call. |
| Additional Info (comments): | N/A |
| Related Scenario(s) | Applicable horizontally to all use cases |
| Related Component | Knowledge Graph |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Id:** | SS-DDDAV-01/EBOS | **Type:** | FUNC | **Importance (priority):** | HIGH | **Source:** | DoA | **Ver:** | 1 |
| **Title:** | Usability and operability: The EO4EU platform shall provide a user-friendly interface which can be operated in an easy and intuitive way. | | | | | | | | |
| **Description:** | Usability and operability are critical aspects of the proposed EO4EU platform system that provides and exposes the system to be dynamic in terms of data access and processing, but also based on the openEO API in terms of the various types of end users, such as the internal end-users, external end-users, administrator users, and use case end-users. They ensure that the platform is user-friendly, easy to operate, and provides an intuitive interface for users to interact with. The following are the defined requirements, technologies, and features that contribute to achieve high usability and operability in the EO4EU platform system:<br>Requirements for Usability and Operability:<br>• Intuitive Interface: The EO4EU platform is based on a clear and intuitive user interface that is easy to navigate and understand, even for end-users with minimal technical knowledge.<br>• Responsive Design: The platform is being designed to work seamlessly on various devices and screen sizes, providing a | | | | | | | | |

consistent user experience across desktops, laptops, tablets, and mobile phones.

- Accessibility: The EO4EU platform is accessible to any end-users including external users and use case users, always following the standards of the user’s registration, and ensuring the security and authentication mechanism that is support by the UMM component.
- Efficiency: The EO4EU platform enables users to accomplish tasks-functionalities-operations efficiently, minimizing the number of steps required to complete common operations.
- Error Handling: The platform provides clear and informative error messages to users when issues arise, (such as in the case of EO Sources availability, or process of executing functionalities in a pipeline) helping them troubleshoot and resolve problems effectively.
- Consistency: The EO4EU platform's design, terminology, and interactions are based on a consistent throughout, promoting a cohesive user experience.
- Flexibility: The EO4EU platform provides customizable settings and options to accommodate individual user preferences and workflows.

EU Features-Technologies for Usability and Operability:

- Drag-and-Drop Functionality: Users can interact with the platform by dragging and dropping elements, widgets, making it easier to customize dashboards, workflows, and visualizations.
- Interactive Tutorials and Help Guides: The platform can include interactive tutorials and context-sensitive help guides to assist users in learning and using the system.
- Smart Search and Filters: Implementing a powerful search function and advanced filters enables users to find data, models, and resources quickly.
- Personalization: Allowing users to personalize their dashboard layouts, preferences, and settings ensures a more user-centric experience.
- Undo/Redo and Revision History: Providing an undo/redo feature and revision history allows users to revert changes and track their actions.
- Context Menus: Context menus offer quick access to relevant actions and options based on the user's current context within the platform.
- Collaboration and Sharing: Enabling collaboration features like real-time editing, sharing dashboards, and collaborative annotations fosters teamwork and knowledge sharing.

| | |
|---|---|
| | The above EO4EU requirements, technologies, and features, define and describe the proposed EO4EU software dynamic platform system, that will enable a user-friendly and highly operable interface that empowers users to interact with the platform seamlessly and efficiently. |
| Additional Info (comments): | N/A |
| Related Scenario(s) | |
| Related Component | Graphical User Interface |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Id:** | SS- DDAV - 02/ EBOS | **Type:** | FUNC | **Importance (priority):** | MEDIUM | **Source:** | DoA | **Ver:** | 1 |

| | |
|---|---|
| **Title:** | Adaptability: EO4EU shall provide a fully configurable dashboard that will be able to personalize the viewable components depending on the user's needs. |
| **Description:** | The EO4EU platform software system is based on an adaptive, interactive, learning, and communication system that offers a fully configurable dashboard with data visualization graphs-charts. This system has the capability to personalize the viewable components based on the specific needs of each end-user. The key features and functionalities of this adaptive platform as the following:<br>• Adaptive System: An adaptive system is designed to adjust and tailor its behavior based on user interactions, preferences, and feedback. In this context, the platform can adapt its interface, features, and data visualization based on the individual requirements of each end user and based on the WF models' selections. This adaptability ensures a more personalized and user-centric experience.<br>• Fully Configurable Dashboard: The platform provides a dashboard that can be customized and configured according to the user's preferences. Users can choose which data elements, widgets, and components they want to see on their dashboard. They can rearrange, add, or remove components to suit their specific needs and priorities. When the end-user will choose the |

| | |
|---|---|
| | EO Sources and the AI/ML models to be deployed for the analysis and interpretation of the outcome results, the WFE will dynamically generate the data type configuration files, that will automatically generate the Data Visualization Charts. However, if the end-user would like to select different data visualization charts, he/she will have this option without executing the initial WFE again.<br>• Data Visualization: The platform includes data visualization capabilities, allowing users to represent data in graphical or chart formats such as a pi-plot, xy-poly-line graphs, bars, and other data chart styles. In addition, an image chart will be deployed to visualize Data Fusion images or Grip Images processed by the WFE. This enables users to gain insights from the data more easily, understand patterns, and make data-driven decisions.<br>• Personalization: Each user can personalize their dashboard and data visualizations to align with their unique requirements and preferences. Personalization options may include selecting data sources, specifying key performance indicators (KPIs), choosing the types of graphs or charts, setting filters, and adjusting time periods.<br>• Viewable Components: Viewable components refer to the data visualization elements, widgets, or modules that can be added or removed from the dashboard. Depending on the user's role, responsibilities, and interests, they can choose to display specific components relevant to their tasks. For example, an end-user might prefer to see some EO data charts based on specific time duration, while a different end-user may desire to observe statistics several years ago.<br>• User Profiles and Permissions: To support personalization and configuration, the platform likely incorporates user profiles and permissions. Different users have different roles and access levels, as this will be defined based on the user’s rights, which determine what access level will they be granted. However, in general, any EO data and dashboard components will be available to be seen and analyzed by any end-user.<br>• Ease of Interaction: The system ensures that users can interact with the dashboard and data visualization tools with ease. It may include intuitive drag-and-drop functionality for arranging components, filters to refine data views, and tooltips to provide additional information on data points.<br>Real-time Updates: The platform can provide real-time or near-real-time updates for the data and visualizations, allowing users to stay informed about the latest developments and make timely decisions.<br>Overall, this adaptive EO4EU platform system will offer a flexible and user-friendly environment for users to access relevant data and insights, |

| | |
|---|---|
| | fostering a more efficient and effective workflow tailored to individual needs and preferences. |
| Additional Info (comments): | N/A |
| Related Scenario(s) | |
| Related Component | Graphical User Interface |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Id:** | SS-DDAV-03/EBOS | **Type:** | FUNC | **Importance (priority):** | MEDIUM | **Source:** | DoA | **Ver:** | 1 |
| **Title:** | The EO4EU platform's dashboard shall be developed to offer robustness, high performance, cross-platform support, improved quality and easy maintenance. | | | | | | | | |
| **Description:** | The EO4EU platform software system's dashboard has ambitious goals in terms of its features and capabilities. There are many key features and technologies that will contribute to achieve the desired characteristics:<br>• Robustness and High Performance: The dashboard should be built with a focus on robustness and high performance. This entails using efficient coding practices, optimizing algorithms, and leveraging technologies that can handle large datasets and complex computations without sacrificing responsiveness. Furthermore, it will provide the essential charts and graphics for interacting with the EO processed data as described in the project.<br>• Cross-platform Support: To ensure broad accessibility, the dashboard should be designed to work seamlessly on various platforms and devices, including desktops, laptops, tablets, and mobile phones. This may involve using responsive design techniques and adhering to web standards.<br>• Improved Quality and Easy Maintenance: To achieve improved quality, the development process should include rigorous testing and quality assurance measures. Code reviews, automated testing, and continuous integration can contribute to maintaining | | | | | | | | |

| | |
|---|---|
| | high code standards. Additionally, modular, and well-documented code can make maintenance tasks more straightforward.<br>• AI/ML Marketplace (Library of ML Models-Algorithms): The integration of an AI/ML marketplace allows users to access and utilize the project’s new developed and designed machine learning models and algorithms. The dashboard should provide a seamless interface for users to browse, select, and integrate these models into their workflows.<br>• Earth Observation Data (Images and Data Values): The platform should support the ingestion, storage, and processing of Earth Observation data, which may include satellite images, geospatial data, and associated metadata. Efficient data management and integration with relevant tools are essential for handling this specialized data type.<br>• Knowledge Graph Tool: The knowledge graph tool allows users to create, manage, and visualize complex relationships between different entities and concepts. Integrating this tool into the dashboard enables users to explore connections and derive insights from the data, as the outcome of the KG generates the necessary description files and documents.<br>• WorkFlow Editor: The workflow editor empowers users to create, customize, and automate workflows using a graphical interface. This feature enables non-technical users to design and execute complex processes efficiently.<br>• Web XR-VR Application: The integration of Web XR-VR capabilities enables users to experience and interact with immersive 3D environments directly within the dashboard. This feature can be beneficial for visualizing spatial data or creating engaging data representations.<br>• S3 Bucket Storage: is utilized as a software instance, and it is integrated with S3 bucket storage from cloud providers like Amazon Web Services (AWS) allows efficient storage and retrieval of large datasets and media assets. This ensures scalability and reliability in handling diverse data types.<br>• Elastic Search: Elasticsearch, as a powerful search and analytics engine, can enhance the dashboard's search capabilities and facilitate efficient querying and filtering of large datasets.<br>• Data Visualization for Plotting Data on Various Charts: The dashboard should offer a variety of data visualization options, including line charts, bar charts, pie charts, scatter plots, heatmaps, and more. Users should be able to select and customize visualizations to represent their data effectively. |

| | |
|---|---|
| | By incorporating these features and technologies, the EO4EU platform software system can create a very powerful and user-friendly dashboard that empowers users to access, analyze, and leverage diverse data sources and AI/ML models for decision-making and problem-solving. |
| Additional Info (comments): | N/A |
| Related Scenario(s) | |
| Related Component | Graphical User Interface |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Id:** | SS-DDAV-04/EBOS | **Type :** | FUNC | **Importance (priority):** | HIGH | **Source:** | DoA | **Ver:** | 1 |

| | |
|---|---|
| **Title:** | The EO4EU platform's dashboard shall rely on open standards and shall enable easy and secure communication and data transfer between the interfacing and the backend system. |
| **Description:** | It is always strongly recommended to use open standards and technical requirements that are commonly accepted for creating a dynamic EO4EU software platform system with dashboard-data visualization, user interfaces, and secure communication between frontend and backend systems. When designing and developing a dynamic AI/Machine Learning platform system, such as the EO4EU platform, with dashboard-data visualization and a user data interface, it involves several components and considerations. To ensure easy and secure communication and data transfer between the frontend and backend systems, we can follow the following open standards, technical requirements, and specifications:<br>Open Standards for Communication and Data Transfer:<br>HTTP/HTTPS: Use the HTTP protocol for communication between the frontend and backend. HTTPS should be enforced to ensure secure data transmission.<br>WebSocket: If real-time interaction is required, WebSocket can be used to establish a persistent connection between the frontend and backend for instant updates. |

| | |
|---|---|
| | JSON: Utilize JSON as the data interchange format for transmitting structured data between the systems.<br>OAuth 2.0: Implement OAuth 2.0 for user authentication and authorization. It provides a secure and standardized way to allow third-party applications limited access to a user's resources.<br>JWT (JSON Web Tokens): Use JWT for token-based authentication and secure information exchange. It allows the transmitting of claims between two parties in a compact and secure manner.<br>CORS (Cross-Origin Resource Sharing): Configure CORS to specify which origins are allowed to access your backend resources. This prevents unauthorized access.<br>Technical Requirements and Specifications:<br>API Design: Define clear and well-documented APIs that outline how the frontend and backend systems will communicate. This includes specifying endpoints, request/response formats, and authentication mechanisms.<br>Authentication and Authorization: Implement strong user authentication and authorization mechanisms to ensure that only authorized users can access and manipulate data. Common methods include OAuth, JWT, and API keys.<br>Data Encryption: Implement data encryption using technologies like Transport Layer Security (TLS) for secure communication between frontend and backend systems.<br>Data Validation and Sanitization: Validate and sanitize user inputs to prevent security vulnerabilities such as SQL injection and cross-site scripting (XSS).<br>Error Handling: Design robust error handling mechanisms to provide meaningful error messages to users and prevent exposing sensitive system information.<br>CORS: If the frontend and backend are hosted on different domains, configure CORS to control which origins are allowed to access resources.<br>Rate Limiting: Implement rate limiting to prevent abuse or overload of the backend system by a single user or client.<br>Data Serialization: Ensure that data sent between frontend and backend is properly serialized and deserialized. Common formats include JSON. |

| | |
|---|---|
| | Logging and Monitoring: Set up logging and monitoring tools to track system behavior, detect anomalies, and identify potential security breaches.<br>Secure Storage: Safely store sensitive information such as API keys, passwords, and tokens. Use secure storage mechanisms and avoid hardcoding these values in source code.<br>Data Privacy and Compliance: Ensure compliance with relevant data protection regulations, such as GDPR, depending on the nature of the data being handled.<br>Performance Optimization: Optimize data transfer and communication for performance, considering factors like data compression and efficient query design.<br>It's important to note that choices of technologies can vary depending on the specific requirements and constraints of the EO4EU platform. These open standards and requirements may need to be adjusted based on the unique context and use case of our AI/Machine Learning EO4EU platform and based on the level of security standards to be defined for the end-users' access and data processing.  A good and safe practice is to always stay up to date with the latest security best practices and standards in the field of software development to ensure the security and reliability of our system.<br>The EO4EU platform shall provide a multi-layered graphical user interface (GUI) for visual analytics. |
| Additional Info (comments): | N/A |
| Related Scenario(s) | |
| Related Component | Graphical User Interface |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Id:** | SS-DDAV-05/EBOS | **Type:** | FUNC | **Importance (priority):** | HIGH | **Source:** | DoA | **Ver:** | 1 |

| | |
|---|---|
| **Title:** | The EO4EU platform shall provide a multi-layered user interface (GUI) for visual analytics. |
| **Description:** | Creating a comprehensive multi-layered GUI such as the Dashboard-Data Visualization for the EO4EU Observation dynamic software platform, involves integrating a variety of complicating functionalities-operations, such as Data Fusion, Function as a Service (FaaS), Machine Learning models/algorithms, S3 Bucket access, and Elastic Search access. Below are the technical requirements and specifications for each of these components within the GUI-Dashboard-Data Visualization.<br>Data Fusion, Function as a Service, and Machine Learning models/Algorithms:<br>Workflow Editor Integration: Integrate a visual workflow editor where users can drag and drop components representing Data Fusion, FaaS, and ML algorithms.<br>Component Library: Provide a library of pre-built components representing various fusion techniques, functions, and ML models.<br>Parameterization: Allow users to customize the parameters for each component within the workflow editor.<br>Real-time Feedback: Provide real-time feedback on data flow, component connections, and possible errors during workflow creation.<br>Connection Flexibility: Support the connection of components in a flexible manner to create complex processing pipelines.<br>Visual Representation: Visualize the workflow as a block diagram, making it easy for users to understand the data flow.<br>Dashboard-Data Visualizations:<br>Interactive Dashboard: Design an interactive dashboard where users can arrange, customize, and resize widgets.<br>Widget Types: Offer a range of widget types, including charts (line, scatter, bar), maps, histograms, xy-line plots, and image viewers.<br>Data Binding: Allow users to bind widgets to specific data sources or output from workflow components.<br>Data Filtering: Enable users to filter data within widgets based on attributes or time periods. |

| | |
|---|---|
| | Data Overlay: Support overlaying multiple datasets and visualizing them together.<br>Export and Sharing: Allow users to export dashboards as images, PDFs, or shareable links.<br>S3 Bucket Access and Reading Data:<br>Credentials Management: Provide a secure way for users to input their S3 bucket credentials or utilize Identity Access Management (IAM) roles.<br>File Browser: Include a file browser to navigate S3 buckets and select files for processing.<br>Data Import: Allow users to select files and import them into the workflow editor or data visualization widgets.<br>Progress Indicators: Display progress bars or indicators during data upload and processing.<br>Elastic Search Access and Reading Data:<br>Connection Setup: Allow users to configure the connection to their Elastic Search cluster by providing necessary endpoints and credentials.<br>Index Selection: Enable users to select specific indices or data sources from the Elastic Search cluster.<br>Query Builder: Provide a query builder interface to help users construct complex queries easily.<br>Data Retrieval: Fetch and display data from Elastic Search in various visualization widgets.<br>Search Filters: Allow users to filter data based on specific criteria before visualization.<br>Prioritizing user-friendly interfaces, responsiveness, and seamless integration between different software components withing the Dashboard as software application services is a great advantage but also very useful for the end-users when interacting with data files and data visualizations. Regular testing, user feedback, and iterative design improvements will be crucial to ensure the GUI-Dashboard-Data Visualization meets users' needs effectively and efficiently. |

<table>
<tr><td colspan="2">Additional Info (comments):</td><td colspan="8">N/A</td></tr>
<tr><td colspan="2">Related Scenario(s)</td><td colspan="8"></td></tr>
<tr><td colspan="2">Related Component</td><td colspan="8">Graphical User Interface</td></tr>
<tr><td><b>Id:</b></td><td>SS-DDAV-06/EBOS</td><td><b>Type:</b></td><td>FUNC</td><td><b>Importance (priority):</b></td><td>HIGH</td><td><b>Source:</b></td><td>DoA</td><td><b>Ver:</b></td><td>1</td></tr>
<tr><td colspan="2"><b>Title:</b></td><td colspan="8">The EO4EU platform shall provide a Command Line Interface (CLI).</td></tr>
<tr><td colspan="2"><b>Description:</b></td><td colspan="8">The proposed CLI is a text-based interface that allows users to interact with the EO4EU platform system by entering commands or instructions as lines of text. The CLI provides a way for users to execute specific tasks, functionalities, and operations by typing commands, which are then interpreted and executed by the underlying software components. The actual aim here is to provide the internal end-user, use case user, external user, and administrator user the ability to interact and to execute EU4EU operations from a line interface through specific commands.<br><br>The main purpose of the proposed CLI is to provide a streamlined and efficient way as an assistive tool for users to interact with the EO4EU platform software application by communicating with both the front-end and back-end. The introduced CLI allows users to perform various tasks without the need for a GUI, and is often preferred and used mainly by developers, system administrators, and power users who prefer keyboard-based interactions and automation.<br><br>The CLI, is capable to communicate with various software components that are being developed in the EO4EU platform system, such as:<br><br>Web API: A Web API is a set of rules and protocols that allows different software applications to communicate with each other over the internet. To interact with a Web API through the CLI, the user can use a tool like cURL or HTTPie to make</td></tr>
</table>

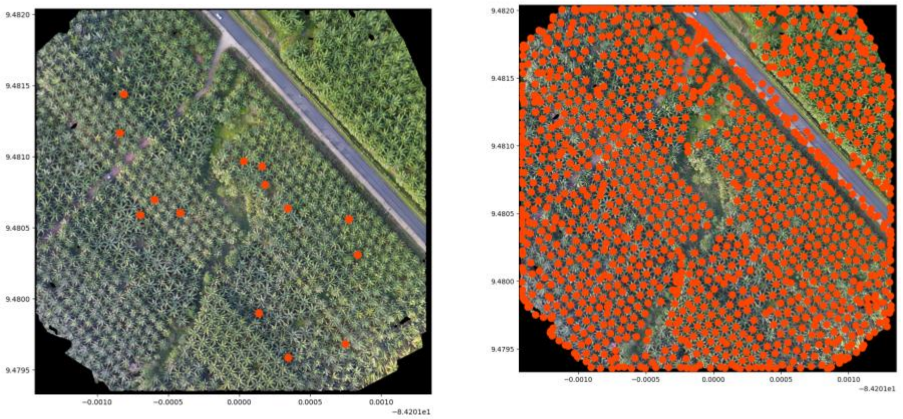
| | |
|---|---|
| | HTTP requests to specific API endpoints.<br>• Dashboard: Interacting with a dashboard through the CLI may involve using specific command-line tools provided by the dashboard software. These tools might allow users to fetch data, configure settings, or trigger specific actions related to the dashboard. The exact commands will depend on the dashboard software and its CLI capabilities.<br>• Database (S3 Bucket and Elastic Search): To communicate with a database through the CLI, users can use Structured Query Language (SQL) commands. For instance, if the database is running PostgreSQL, the user can use the psql command-line tool to execute SQL queries:<br>• Data Visualization: Data visualization libraries usually come with APIs that can be accessed through programming languages like Python or JavaScript. To interact with these libraries through the CLI, users can create scripts or use CLI tools that encapsulate the visualization functionalities and provide appropriate command-line parameters to customize the visual output.<br>• AI/Machine Learning Library: Interacting with AI/Machine Learning libraries through the CLI typically involves running pre-trained models, training models on custom data, or performing various machine learning tasks. Command-line tools or Python scripts can be used to invoke the library's functions and pass the required data and parameters.<br>• User Management Model: A user management model, responsible for handling user authentication and authorization, might have CLI tools to manage user accounts and permissions. These tools would allow administrators to create, modify, or delete user accounts and set their access rights using command-line commands.<br>• End-User: The end-users can interact with the application or system via the CLI by executing specific commands related to their needs or tasks. These commands should be intuitive and well-documented to ensure ease of use for the end-users.<br>It's essential to provide clear documentation and help messages within the CLI to guide users on the available commands and their usage. Additionally, incorporating tab-completion and interactive prompts can enhance the user experience while interacting with the CLI. |
| Additional Info (comments): | N/A |

| | |
|---|---|
| Related Scenario(s) | |
| Related Component | Graphical User Interface |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Id:** | SS-DDAV-07/ CINECA | **Type:** | FUNC | **Importance (priority):** | HIGH | **Source:** | DoA | **Ver:** | 1 |
| **Title:** | The EO4EU platform shall provide an extended reality (XR) web-based application interface. | | | | | | | | |
| **Description:** | The EO4EU Graphical User Interface will include, among the different visualization tools, an extended reality (XR) web-based application interface that will enable the possibility to visualize and explore the EO data in a more immersive way. The XR interface will include a Virtual Reality (VR) web application to explore the data in an immersive environment and an Augmented reality (AR) web application to visualize and analyze the EO data in the physical world, making advantage of the geolocation system of the device where the user will run the application to blend the virtual scientific data representation with the actual user's ambient environment. Both the applications will be accessible by web and should be compatible with the main web browsers. Additional requirement for the XR interface will be:<br>• Cross platform compatibility: the AR and VR applications should be designed to work seamlessly on different devices. Their GUI should be implemented using a responsive design. Moreover, the VR application should be compatible with different hardware designed for supporting VR applications, ranging from cardboard to VR head-mounted devices.<br>• Optimized performance and scalability: the architecture of the XR interface should be designed to assure optimized performance in order to ensure a high-quality experience of user's navigation on different devices. Furthermore, the architecture should be scalable and capable of handling large amounts of data or vast areas of interest.<br>• Navigation and data interaction: the XR interface should allow the user to navigate the virtual environment and to interact with the EO data in order to explore and to have a better | | | | | | | | |

|  |  |
|---|---|
|  | comprehension of the data themselves. For instance, the user might be able to move inside the virtual world, to query the data related to a specific point of interest or to customize the way the data is represented or visualized. The navigation system and the different possibilities of user interactions should be implemented following all the best practices and design principles to give the user a high-quality experience and to avoid any kind of the so-called motion-sickness in the scope of the VR application. The kind of visualization and interactions should be designed to meet the specific demands and needs of an XR experience.<br>• Ease of interaction: the XR interface should have a minimal GUI that should be clear and easy to understand for the user. The way the user will navigate the application should be intuitive as well as the way he/she can interact with the EO data and the virtual world. User's actions should be supported by visual feedback capable of guiding and helping the user during his/her interactions within the application. |
| Additional Info (comments): | N/A |
| Related Scenario(s) |  |
| Related Component | Graphical User Interface |

|  |  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|
| **Id:** | SS-FE-001/NKUA | **Type :** | FUNC | **Importance (priority):** | HIGH | **Source:** | DoA | **Ver:** | 1 |
| **Title:** | The fusion engine shall integrate heterogeneous data sources. |  |  |  |  |  |  |  |  |
| **Description:** | Fusion Engine supports multiple data sources needed to support geospatial interpolation functionalities. The possible data sources are NetCDF, XML, CSV, ESRI Shapefiles, GeoTIFFs and TXT. |  |  |  |  |  |  |  |  |
| Additional Info (comments): | In the example below the correlated information from GeoTIFFs and shapefiles is shown and then a cluster analysis is presented. |  |  |  |  |  |  |  |  |

| | |
|---|---|
| |  |
| Related Scenario(s) | All use cases |
| Related Component | Fusion Engine |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Id:** | SS-FE-002/NKUA | **Type:** | FUNC | **Importance (priority):** | HIGH | **Source:** | DoA | **Ver:** | 1 |

| | |
|---|---|
| **Title:** | Fusion engine shall provide preprocessing assessment and refinement of data. Preprocessing of data includes spatiotemporal interpolation. |
| **Description:** | The functionalities supported by FE are considered in the preprocessing and the refinement step of the workflow. A user selects its data, selects from a toolset of FE possible refinement and preprocessing tasks described in D5.1 and then he/she can select the ML model that needs to run. If FE can do all the cleaning and reshaping of data, the new dataset is stored in order for the rest components to have direct access. |
| Additional Info (comments): | N/A |
| Related Scenario(s) | All use cases |
| Related Component | Fusion Engine |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Id:** | SS-FE-003/ NKUA | **Type:** | FUNC | **Importance (priority):** | HIGH | **Source:** | DoA | **Ver:** | 1 |
| **Title:** | Fusion engine comprises the guidelines from Domain Specific Language (DSL). | | | | | | | | |
| **Description:** | FE receives the workflow of user described in DSL and then this workflow is “translated” to FE workflow in order to execute the correct steps of the user chain. | | | | | | | | |
| Additional Info (comments): | N/A | | | | | | | | |
| Related Scenario(s) | All use cases | | | | | | | | |
| Related Component | Fusion Engine | | | | | | | | |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Id:** | SS-FSO-001/NKUA | **Type:** | FUNC | **Importance (priority):** | HIGH | **Source:** | DoA | **Ver:** | 1 |
| **Title:** | FaaS shall provide computational resources and services as a per service instance of a containerized environment. | | | | | | | | |
| **Description:** | Based on its distributed character, the present function will give end-user the ability to dynamically create a block of code that can be executed in parallel and asynchronously, speeding up the possible computational demands of the user's algorithm. The user will have the ability through the graphical use interface to write an algorithm (e.g., in python) which will then be built and deployed for the specific workflow including the necessary components to become available for processing. | | | | | | | | |
| Additional Info (comments): | N/A | | | | | | | | |

| | |
|---|---|
| Related Scenario(s) | All use cases |
| Related Component | Function as a Service Orchestrator |

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Id: | **SS-FSO-002/ENG** | Type: | **FUNC** | Importance (priority): | **MEDIUM** | Source: | **DoA** | Ver: | **1** | | |
| **Title:** | | Provide GPU capabilities supporting accelerated AI and High-Performance Computing (HPC) | | | | | | | | | |
| **Description:** | | GPUs will be used for the training of the ML models on HPC resources provided by CINECA's infrastructure. The ML model training is not completed yet and the progress of its implementation will be reported on the next version of the deliverable. | | | | | | | | | |
| Additional Info (comments): | | N/A | | | | | | | | | |
| Related Scenario(s) | | Training of ML models | | | | | | | | | |
| Related Component | | ML model training | | | | | | | | | |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Id:** | SS-FSO-003/ENG | **Type:** | FUNC | **Importance (priority):** | HIGH | **Source:** | DoA | Ver: | 1 |
| **Title:** | Virtualization layer based on Linux/KVM | | | | | | | | |
| **Description:** | All the eo4eu resources are deployed on OpenStack from two clouds infrastructures, WEkEO DIAS and CINECA's cloud (ada cloud). OpenStack virtualization based on KVM  and is the default setting for | | | | | | | | |

| | |
|---|---|
| | the hypervisors comprising an OpenStack cloud. The KVM hypervisor supports the following virtual machine image formats: Raw<br>QEMU Copy-on-write (qcow2), QED Qemu Enhanced Disk and VMware virtual machine disk format (vmdk). |
| Additional Info (comments): | OpenStack components and their usage are analysed in various deliverables |
| Related Scenario(s) | N/A |
| Related Component | The entire eo4eu platform is based on OpenStack. |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Id: | SS-FSO-004/ENG | **Type:** | FUNC | **Importance (priority):** | HIGH | **Source**: | DoA | **Ver:** | 1 |

| | |
|---|---|
| Title: | Infrastructure as a Service (IaaS) layer using Openstack |
| Description: | IaaS is based on OpenStack provided from two different infrastructures ( WEkEO DIAS and CINECA’s cloud). |
| Additional Info (comments): | N/A |
| Related Scenario(s) | - |
| Related Component | Eo4eu platform is based on OpenStack. |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Id:** | SS-FSO-005/ENG | **Type:** | FUNC | **Importance (priority):** | HIGH | **Source:** | DoA | **Ver**: | 1 |

| | |
|---|---|
| Title: | Kubernetes for running the operational workloads |
| Description: | All EO4EU services and components are deployed on Kubernetes clusters which ensure the resilience and scalability of the various service s |
| Additional Info (comments): | - |
| Related Scenario(s) | - |
| Related Component | |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Id:** | SS-FSO-006/ENG | **Type:** | FUNC | **Importance (priority):** | MEDIUM | **Source:** | DoA | **Ver:** | 1 |
| **Title:** | Deployment isolation for every service/workflow with minimum footprint to the system. | | | | | | | | |
| **Description:** | Each service within the EO4EU platform is deployed in its dedicated namespace within the Kubernetes cluster. This practice aligns with the microservices architecture employed in the design of EO4EU, where the application is crafted as a collection of services. Such an architectural approach offers a robust framework for the development, deployment, and maintenance of microservices independently. It ensures isolation and controlled access to and from services through granulated Role-Based Access Control (RBAC) provided by the Kubernetes cluster. | | | | | | | | |
| Additional Info (comments): | N/A | | | | | | | | |
| Related Scenario(s) | - | | | | | | | | |
| Related Component | Eo4eu platform | | | | | | | | |

Funded by the European Union

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Id:** | SS-FSO-007/ENG | **Type:** | FUNC | **Importance (priority):** | MEDIUM | **Source:** | DoA | **Ver:** | 1 |
| **Title:** | Function as a Service (FaaS) orchestrator with resource reservation capability | | | | | | | | |
| **Description:** | FaaS orchestrator will facilitate the execution of various functions needed and triggered by the submitted user workloads. Eo4eu platform will investigate further the functionality/capability of the resource reservation and it progress will be reported on the next version of the deliverable. | | | | | | | | |
| Additional Info (comments): | N/A | | | | | | | | |
| Related Scenario(s) | Eo4eu workload | | | | | | | | |
| Related Component | Function as a Service Orchestrator | | | | | | | | |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Id:** | SS-ML-001/ HES-SO | **Type :** | FUNC | **Importance (priority):** | HIGH | **Source:** | DoA | **Ver:** | 1 |
| **Title:** | A self-supervised learning model based on contrastive learning, utilizing the vast volume of unlabeled data, by learning robust representation that will facilitate downstream tasks minimizing their labelled data budget requirement will be developed | | | | | | | | |
| **Description:** | In Deliverable 3.4, a detailed evaluation and documentation of the self-supervised learning model will be provided. Based on existing literature, it is anticipated that this approach will enhance the performance of | | | | | | | | |

| | |
|---|---|
| | downstream tasks compared to training solely on the data specific to those tasks. Nevertheless, it is crucial to validate this claim specifically on EO data. |
| Additional Info (comments): | N/A |
| Related Scenario(s) | |
| Related Component | Machine Learning |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Id:** | SS-ML-002/ HES-SO | **Type :** | FUNC | **Importance (priority):** | HIGH | **Source:** | DoA | **Ver:** | 1 |
| **Title:** | | A generic ML pipeline will be designed and developed. The Generic ML pipeline of EO4EU will enable the learning of a robust and transferable representation of the input data in a latent space, in an unsupervised way. | | | | | | | |
| **Description:** | | Utilizing the trained self-supervised learning (SSL) model as specified in SS-ML-001/HES-SO, the aim is to showcase the quality and effectiveness of our representation across various downstream tasks. The pipeline spans from receiving the data through the platform, performing inference using the SSL model, subsequently feeding the SSL output to the supervised model of a downstream task, ending up with a prediction of the latter, which assigns semantic content to the data (for instance a classifier assigning classes of a data instance). | | | | | | | |
| Additional Info (comments): | | N/A | | | | | | | |
| Related Scenario(s) | | | | | | | | | |
| Related Component | | Machine Learning | | | | | | | |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Id:** | SS-ML-003/ HES-SO | **Type:** | FUNC | **Importance (priority):** | HIGH | **Source:** | DoA | **Ver:** | 1 |
| **Title:** | Machine learning models will facilitate the volume reductions of the data that will be transferred. | | | | | | | | |
| **Description:** | The models and tools to carry out the compression and decompression will be delivered. | | | | | | | | |
| Additional Info (comments): | N/A | | | | | | | | |
| Related Scenario(s) | | | | | | | | | |
| Related Component | Machine Learning | | | | | | | | |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Id:** | SS-ML-004/ HES-SO | **Type:** | FUNC | **Importance (priority):** | HIGH | **Source:** | DoA | **Ver:** | 1 |
| **Title:** | An encoder-decoder pair will serve the compression process in our platform. | | | | | | | | |
| **Description:** | A compression-decompression model will be trained as an encode-decoder pair building on a vector-quantized variational auto-encoder (VQ-VAE) pattern. It will be documented in detail in deliverable 3.4. | | | | | | | | |
| Additional Info (comments): | N/A | | | | | | | | |
| Related Scenario(s) | | | | | | | | | |

| | |
|---|---|
| Related Component | Machine Learning |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Id:** | SS-ML-005/ HES-SO | **Type:** | FUNC | **Importance (priority):** | HIGH | **Source:** | DoA | **Ver:** | 1 |
| **Title:** | The final version of all developed models will be shaped in the form that will allow their publication in the EOSC | | | | | | | | |
| **Description:** | All models will be meticulously documented, ensuring that both the code and the resulting trained models are provided. Everything will be made accessible in a universally recognized standard format, facilitating reusability. | | | | | | | | |
| Additional Info (comments): | N/A | | | | | | | | |
| Related Scenario(s) | | | | | | | | | |
| Related Component | Machine Learning | | | | | | | | |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Id:** | SS-ML-006/ HES-SO | **Type:** | FUNC | **Importance (priority):** | HIGH | **Source:** | DoA | **Ver:** | 1 |
| **Title:** | Deep learning models following a self-supervised learning approach, and in particular contrastive learning will be developed, tuned, evaluated, and optimized. | | | | | | | | |
| **Description:** | In Deliverable 3.4, a detailed evaluation and documentation of the self-supervised learning model using a modern and very deep architecture will be provided. Based on existing literature, it is anticipated that this approach will enhance the performance of downstream tasks compared | | | | | | | | |

| | |
|---|---|
| | to training solely on the data specific to those tasks. Nevertheless, it is crucial to validate this claim specifically on EO data. |
| Additional Info (comments): | N/A |
| Related Scenario(s) | |
| Related Component | Machine Learning |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Id:** | SS-ML-007/ HES-SO | **Type :** | FUNC | **Importance (priority):** | MEDIUM | **Source:** | DoA | **Ver:** | 1 |
| **Title:** | Deep learning models will exploit HPC infrastructure for the computational aspects | | | | | | | | |
| **Description:** | We will utilize GPU resources within a High-Performance Computing (HPC) cluster to train various machine learning models. These models require significant resources and time for training, with GPUs and TPUs being the most efficient hardware options. The training process will take place on the HPC cluster, after which the trained models will be deployed to our platform for user access. It's important to note that model training is an offline process, separate from the platform operations, but the outcomes of this training — the models themselves — will be integrated and utilized within the platform. | | | | | | | | |
| Additional Info (comments): | N/A | | | | | | | | |
| Related Scenario(s) | | | | | | | | | |
| Related Component | Machine Learning | | | | | | | | |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Id:** | SS-MP-01/EBOS | **Type:** | FUNC | **Importance (priority):** | HIGH | **Source:** | DoA | **Ver:** | 1 |

| | |
|---|---|
| **Title:** | Developed, tested, and evaluated models will be publicized in a public repository. |
| **Description:** | To develop, test, and evaluate AI/ML models for publicizing in a repository like an AI/ML Marketplace (library) and integrating them into a dashboard, as a software instance indicating a list of the AI/Machine learning models: 1. Self-Supervised Learning Algorithms, 2. Features to Classification ML Algorithm, 3. Compression ML Algorithm, 4. Decompression ML Algorithm, we need to consider the following technical requirements, software architecture’s requirements and specifications. An important consideration we need to take into account is that these AI/ML algorithms can be locally stored inside the S3 Bucket on Cloud or Local Archive server on the Cluster or can be defined as function calls through wrapper codes (python script) communicating with the inference server through Kubernetes or KAFKA streaming platform.<br>Technical Requirements:<br>Algorithm Development: Choose appropriate programming languages (e.g., Python) and frameworks (e.g., TensorFlow, PyTorch) for implementing the machine learning algorithms.<br>Implement each algorithm according to its specific requirements, whether it's SSL, classification, compression, or decompression.<br>• Model Evaluation: Define evaluation metrics suitable for each algorithm's task (e.g., accuracy, F1-score, compression ratio).<br>• Use appropriate datasets for training, validation, and testing.<br>• Implement thorough testing and cross-validation to ensure model performance and generalization.<br>Documentation:<br>• Document each algorithm's purpose, implementation details, parameters, and usage instructions. |

| | |
|---|---|
| | • Include examples of how to use the models and interpret their results.<br>• Code Quality and Maintainability: Follow coding best practices and maintain a clean, well-structured codebase.<br>• Include comments, docstrings, and version control (e.g., Git) to facilitate collaboration and future updates.<br>Software Architecture Requirements:<br>Modularity and Reusability:<br>• Design algorithms as modular components to allow easy integration and reuse.<br>• Consider creating a common interface for algorithms to ensure consistency and ease of use.<br>API/Wrapper Design:<br>• If using wrapper scripts, design APIs that encapsulate the algorithm's functionality and handle data input/output.<br>• Ensure wrappers are compatible with the dashboard's expected data formats.<br>Version Control:<br>• Implement versioning for models to track changes and updates.<br>• Ensure backward compatibility with older versions if possible.<br>Dependency Management:<br>• Clearly define dependencies (libraries, frameworks) required for each algorithm.<br>• Utilize virtual environments (e.g., Python's VirtualEnv) to manage dependencies.<br>Deployment Options:<br>S3 Bucket or Local Archive Server:<br>• Store model files in your S3 bucket or local archive server.<br>• Provide download links or API endpoints for users to access and integrate models.<br>Wrapper Scripts and Inference Servers:<br>• Create wrapper scripts (Python scripts) that encapsulate the models' functionality. |

| | |
|---|---|
| | • Deploy wrapper scripts on an inference server, such as Kubernetes or Kafka, to provide a scalable and efficient inference solution.<br>Security and Privacy Considerations:<br>Access Control:<br>• Implement appropriate access controls to protect the models' intellectual property and data.<br>• Utilize authentication mechanisms to ensure only authorized users can access the models.<br>Data Privacy:<br>• Ensure that sensitive data is properly anonymized or encrypted, especially if the models involve personal or sensitive information.<br>Code Review and Vulnerability Scanning:<br>• Conduct code reviews to identify security vulnerabilities or potential risks.<br>• Perform regular vulnerability scans to ensure the models and deployment platforms are secure.<br>By addressing these technical requirements, software architecture considerations, and deployment options, we can develop AI/Machine Learning models that are well-documented, accessible, and ready for integration into a public repository and dashboard environment, such as the AI/ML Marketplace. |
| Additional Info (comments): | N/A |
| Related Scenario(s) | |
| Related Component | AI/ML Marketplace |

<table>
<tr><td><b>Id:</b></td><td>SS-MP-02/EBOS</td><td><b>Type :</b></td><td>FUNC</td><td><b>Importance (priority):</b></td><td>MEDIUM</td><td><b>Source:</b></td><td>DoA</td><td><b>Ver:</b></td><td>1</td></tr>
<tr><td colspan="2"><b>Title:</b></td><td colspan="8">Other FAIR, open repositories will also be potential places of publicizing.</td></tr>
<tr><td colspan="2"><b>Description:</b></td><td colspan="8">Creating a FAIR (Findable, Accessible, Interoperable, Reusable) and open repository for scientific publications and associated materials involves several technical requirements, software architecture considerations, and specifications. Furthermore, to enhance the full interoperability and reusability of our computational AL/ML models, we will utilize relevant platforms such as Acumos and  OpenML, which facilitate the sharing of AI models, simplifying our EO4EU cross-software platform’s reusability.<br>Technical Requirements:<br>• Storage Infrastructure: Ensure you have sufficient storage space to accommodate various data types, including research papers, datasets, models, code, and other supplementary materials.<br>• Data Formats: Support a wide range of data formats to cater to diverse research materials, such as PDFs for papers, CSV/JSON for datasets, various programming languages for code, and standardized formats for machine learning models.<br>• Metadata Standardization: Implement standardized metadata schemas (e.g., Dublin Core, DataCite) to make data and materials more discoverable.<br>• Versioning: Enable version control for datasets, code, and models to track changes over time and facilitate reproducibility.<br>• Access Control: Implement access controls to ensure materials are appropriately shared according to licensing terms and data sharing policies.<br>• Authentication and Authorization: Secure user authentication and authorization mechanisms to control who can upload, access, and modify data.<br>Software Architecture Requirements:<br>• Modularity: Design the architecture with modularity in mind, allowing you to add new features and functionalities without disrupting the entire system.<br>• Scalability: Create a scalable architecture that can handle increasing volumes of data and users over time.</td></tr>
</table>

| | |
|---|---|
| | • APIs: Develop well-documented APIs that allow integration with other platforms and tools, such as GitLab, Acumos, and OpenML.<br>• Search and Discovery: Implement a robust search and discovery system, enabling users to find relevant research materials quickly.<br>• Interoperability: Ensure compatibility with existing standards (e.g., RESTful APIs) to facilitate integration with other platforms.<br><br>Specifications:<br><br>• Metadata Standards: Define metadata standards for different types of content (papers, datasets, models, code) to ensure consistency and ease of discovery.<br>• Licensing: Specify the licensing terms under which data and materials are shared. Consider using open licenses (e.g., Creative Commons) to promote reuse.<br>• Data Access and Usage Policies: Clearly outline data access policies, usage guidelines, and citation requirements to maintain ethical and legal use of the materials.<br>• API Specifications: Define the APIs for interacting with the repository, including uploading, retrieving, and searching for content.<br>• Integration Guidelines: Provide guidelines for integrating with external platforms like GitLab, Acumos, and OpenML, ensuring seamless interoperability.<br>• User Interfaces: Detail the user interfaces for uploading, browsing, and accessing different types of materials.<br>• Version Control: Specify how version control will be implemented, including how changes to materials will be tracked and displayed.<br>By adhering to these technical requirements, software architecture considerations, and specifications, we can create a robust and interoperable repository that promotes the FAIR principles and facilitates the sharing and reuse of scientific materials in the field of AI/ML and Earth observation (EO4EU). |
| Additional Info (comments): | N/A |
| Related Scenario(s) | |
| Related Component | AI/ML Marketplace |

<table>
<tr><td><b>Id:</b></td><td>SS-MP-03/EBOS</td><td><b>Type:</b></td><td>FUNC</td><td><b>Importance (priority):</b></td><td>MEDIUM</td><td><b>Source:</b></td><td>DoA</td><td><b>Ver:</b></td><td>1</td></tr>
<tr><td colspan="2"><b>Title:</b></td><td colspan="8">Openly sharing the Models and relevant material with the European Open Science Cloud (EOSC) Marketplace.</td></tr>
<tr><td colspan="2"><b>Description:</b></td><td colspan="8">There are several specific requirements that need to be considered for openly sharing the AI/ML Models, and relevant material with the EOSC Marketplace. All AI/ML models including the Data Fusion, FaaS, pipeline processing and data compression, that will be developed, tested, and evaluated during the project will be publicized, accompanied with all relevant assisting material to support their reusability (metadata, configuration files, documentation, etc.) in a public repository. The Marketplace of the EOSC portal, will be our primary portal for openly sharing the Models and relevant material, but we will also share these AI/ML Models-Algorithms on a local storage called the AI/ML Marketplace for internal and external use, depending on the access rights of the registered end-users.<br>Sharing AI/ML models and relevant materials with the EOSC Marketplace requires careful consideration of technical requirements, software architecture, and specifications.<br>Technical Requirements:<br>1 Data Storage and Management: Set up a robust storage infrastructure to house AI/ML models, datasets, documentation, and related materials. This should include sufficient capacity for both current and future storage needs.<br>2 Data Formats: Support standard data formats for models (e.g., ONNX, TensorFlow, PyTorch), datasets (e.g., CSV, JSON), and documentation (e.g., PDF, Markdown).<br>3 Metadata Standards: Implement standardized metadata schemas for models and related materials to enhance discoverability and reusability.<br>4 Version Control: Integrate versioning mechanisms for models and related files, enabling users to track changes and access previous versions.</td></tr>
</table>

| | |
|---|---|
| | 5 Authentication and Authorization: Implement secure user authentication and authorization to control access to different materials based on user roles and permissions.<br>6 Data Access Policies: Define access policies and permissions for both the public EOSC Marketplace and the internal AI/ML Marketplace, considering varying levels of access rights.<br>Software Architecture Requirements:<br>• Modular Architecture: Design a modular architecture that allows for easy integration with the EOSC Marketplace and the local AI/ML Marketplace.<br>• API Integration: Develop well-documented APIs that facilitate communication between your platform and the EOSC Marketplace, enabling seamless data exchange.<br>• Scalability: Create a scalable architecture to handle growing volumes of models and data as the project progresses.<br>• Interoperability: Ensure compatibility with EOSC's existing infrastructure and technologies for smooth integration.<br>• User Interfaces: Design user-friendly interfaces for uploading, browsing, and accessing AI/ML models, datasets, and documentation.<br>Specifications:<br>• Metadata Specifications: Define specific metadata standards for AI/ML models, datasets, and documentation. This includes fields such as title, description, authors, licenses, keywords, and more.<br>• API Specifications: Detail the APIs required for interacting with the EOSC Marketplace, including endpoints for uploading, updating, and retrieving models and data.<br>• Access Control Specifications: Specify how access control and permission management will be implemented for both the public EOSC Marketplace and the internal AI/ML Marketplace.<br>• Version Control Specifications: Define how versioning will be handled, including how users can access different versions of models and associated materials.<br>• User Management Specifications: Detail how user accounts will be managed, including registration, roles, and permissions.<br>• Integration Guidelines: Provide step-by-step guidelines for researchers and users on how to integrate your models and data from the EOSC Marketplace into their workflows.<br><br>Considering the above technical requirements, software architecture considerations, and specifications, we can ensure the successful integration of your AI/ML models and materials into the European Open |

|  |  |
|---|---|
|  | Science Cloud (EOSC) Marketplace while also catering to our internal AI/ML Marketplace needs. |
| Additional Info (comments): | N/A |
| Related Scenario(s) |  |
| Related Component | AI/ML Marketplace |

## 3.3 External User Requirements

This section summarizes the external user requirements for a viable EO platform, collected from 30 participants. These requirements are organized into 8 categories: data requirements, storage and access, data search, indexing and catalogs, hardware processing capabilities, other processing capabilities, algorithms, interfacing services, authentication, and authorization.

A general analysis was conducted on the collected external user requirements using a word cloud. Figure 18 depicts the word cloud of these requirements, where some important insights can be observed. For instance, one can notice the importance of data availability for those users, including data coming from the Copernicus missions (Sentinel-1, Sentinel-2, Sentinel-3), ERA-5, MODIS, among others. Additionally, the users expressed the importance of the metadata associated with the different datasets, which will increase their discoverability and accessibility. Other highlights include the vital role of the interfacing services of the platform, e.g., an API, processing algorithms and computing resources.

More details on these requirements are provided in the following sections.

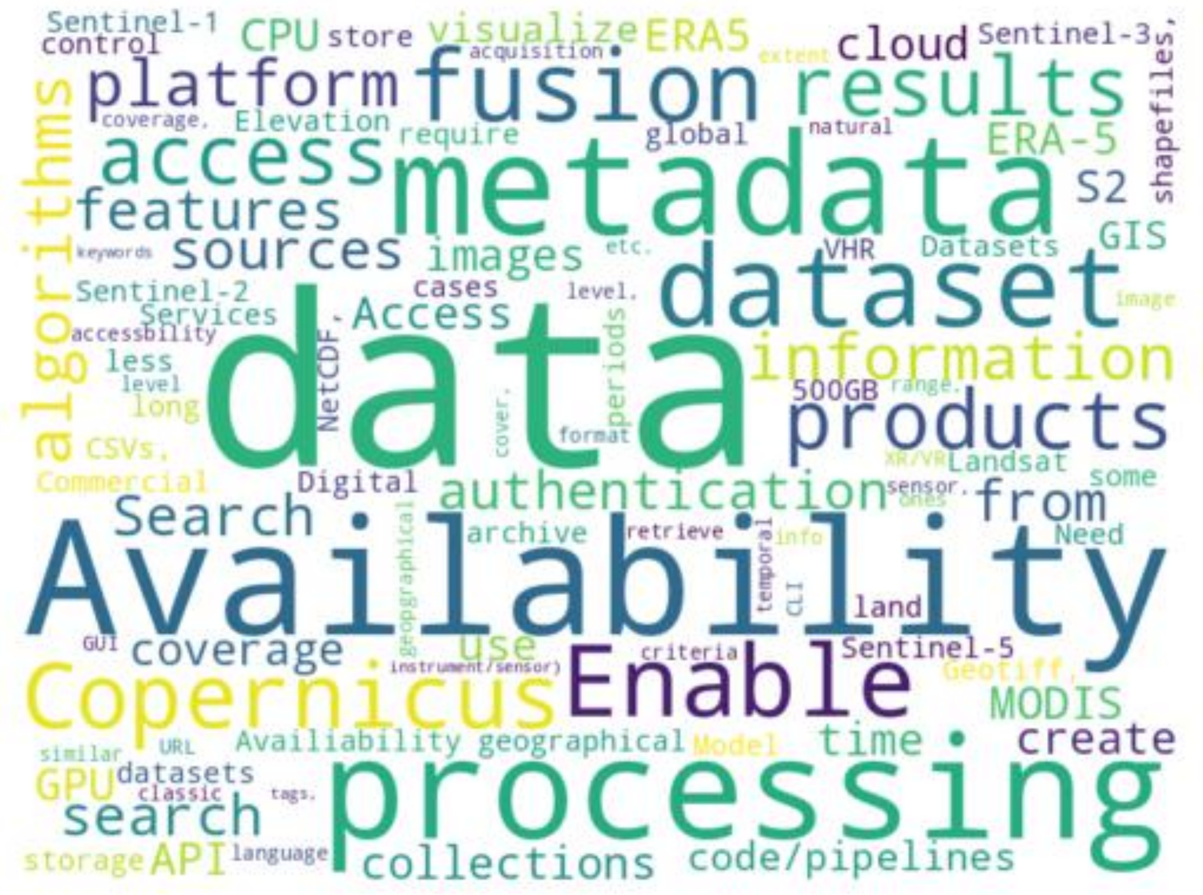


Figure 18: Word cloud on the requirements collected from external users.

### 3.3.1 Data requirements

Table 9 summarizes the data requirements collected from the external user survey along with the received votes. The data requirements category includes requirements for the types of EO datasets that the platform should provide access to, as well as the temporal and spatial coverage of the data. The most important requirements in this category are:

- **Availability of a wide range of EO datasets:** The platform should provide access to a wide range of EO datasets, including those from Sentinel-2, Sentinel-1, Digital Elevation Model (DEM), Landsat collection, ERA-5, Copernicus Services, MODIS, and commercial Very High Resolution (VHR) images.
- **Availability of archive data for long time periods:** The platform should provide access to archive data for long time periods, such as 1-5 years. This will allow users to study long-term changes in the Earth's surface.
- **Availability of data for various geographical coverage:** The platform should provide data for various geographical coverage, up to global for some use cases. This will allow users to study a wide range of phenomena, from local to global scales.

**Table 9: External user data requirements.**

| Category | Subcategory | Requirement | Votes |
|---|---|---|---|
| Data requirements | Dataset type | Availability of Copernicus Sentinel-2 all products | 70 % |
| Data requirements | Dataset type | Availability of Copernicus Sentinel-1 all products | 63,30 % |
| Data requirements | Dataset type | Availability of Digital Elevation Model | 60 % |
| Data requirements | Dataset type | Availability of Landsat collections | 56,70 % |
| Data requirements | Dataset type | Availability of Copernicus Sentinel-3 all products | 46,70 % |
| Data requirements | Dataset type | Availability of ERA-5 land data | 46,70 % |
| Data requirements | Dataset type | Availability of ERA-5 data | 43,30 % |
| Data requirements | Dataset type | Availability of Copernicus Services datasets | 43,30 % |
| Data requirements | Dataset type | Availability of MODIS collections | 43,30 % |
| Data requirements | Dataset type | Availability of Copernicus Sentinel-5 all products | 36,70 % |
| Data requirements | Dataset type | Availability of Commercial VHR images | 33,30 % |
| Data requirements | Dataset coverage | Availability of archive data for long time periods | 100 % |
| Data requirements | Dataset coverage | Availability of data for various geographical coverage up to global for some use cases | 100 % |

### 3.3.2 *Storage and access*

The storage and access category includes requirements for the storage capacity of the platform, the file formats that the platform should support, and the ways in which users can access the data. The most important requirements in this category are:

- **Storage capacity of at least 500GB:** The platform should have a storage capacity of at least 500GB to store the EO datasets for the user use cases.
- **Support for GeoTIFF, NetCDF, Shapefiles, CSVs, etc:** The platform should support the most popular file formats of geospatial datasets, such as GeoTIFF, NetCDF, shapefiles, CSVs, GeoJSON, etc.
- **Support for multiple ways of accessing data:** The platform should support multiple ways of accessing data, such as API, GUI and CLI. This will allow users to access the

data in the way that best suits their needs. Additionally, the platform should provide a permission control mechanism on the datasets produced by the users to allow sharing them with others.

Table 10 describes the storage and access requirements expressed by the external users along with their associated votes.

**Table 10: External user storage and access requirements.**

| Category | Subcategory | Requirement | Votes |
|---|---|---|---|
| Storage and access | Dataset storage | Datasets require less than 500GB of storage | 60 % |
| Storage and access | File format | Need to store Geotiff, NetCDF, shapefiles, CSVs, etc | 100 % |
| Storage and access | Results accessibility | Allow the user to control the accessibility and sharing of his/her processing results | 100 % |

### *3.3.3 Data search, indexing and catalogs*

The data search, indexing and catalogs category includes requirements for the way in which users can search for and browse the data on the platform. Moreover, it includes the users' point of view on the most important metadata that will help increase the discoverability and accessibility of EO datasets. The most important requirements in this category are:

- **Ability to search for data using classic criteria:** The platform should allow users to search for data using classic criteria, such as time range, sensor, processing level, coverage, and cloud cover.
- **Ability to search for data using natural language:** The platform should allow users to search for data using natural language. This will make it easier for users to find the data they are looking for using natural expressions.
- **Ability to search for data using images:** The platform should allow users to search for data using images. This will allow users to find similar images to the ones they are looking for.
- **Use of a two-level metadata system:** The platform should use a two-level metadata system, a collection (dataset) and granule (file) levels, to improve discoverability and accessibility of the data. This will allow users to find the data they are looking for more easily.
- **Useful metadata that can help increase the discoverability and accessibility of EO datasets:** such as the geographical and temporal extents, processing level, keywords, etc.

Table 11 provides more details on the requirements associated with the data search, indexing and catalogs category.

**Table 11: External user data search, indexing and catalogs requirements.**

| Category | Subcategory | Requirement | Votes |
|---|---|---|---|
| Data search, indexing and catalogs | Data search tools | API access to search for available data | 73,30 % |
| Data search, indexing and catalogs | Data search tools | GUI to search for available data | 70 % |
| Data search, indexing and catalogs | Data search tools | CLI and XR/VR to search for available data | 20 % |
| Data search, indexing and catalogs | Data search options | Search available data using classic criteria such as time range, sensor, processing level, coverage, cloud cover, etc. | 93,30 % |
| Data search, indexing and catalogs | Data search options | Search available data using natural language | 36,70 % |
| Data search, indexing and catalogs | Data search options | Search available data using an image to retrieve similar ones | 30 % |
| Data search, indexing and catalogs | Data search metadata | Add geographical extent in the dataset metadata | 86,70 % |
| Data search, indexing and catalogs | Data search metadata | Add temporal coverage in the dataset metadata | 80,00 % |
| Data search, indexing and catalogs | Data search metadata | Add the dataset URL to the dataset metadata | 63,30 % |
| Data search, indexing and catalogs | Data search metadata | Add data format the dataset metadata | 60,00 % |
| Data search, indexing and catalogs | Data search metadata | Add processing level to the dataset metadata | 60,00 % |
| Data search, indexing and catalogs | Data search metadata | Add the acquisition info to the metadata (e.g., instrument/sensor) | 56,70 % |
| Data search, indexing and catalogs | Data search metadata | Add keywords to the metadata (e.g., tags, application domain, etc) | 53,30 % |
| Data search, indexing and catalogs | Data search metadata | Add information about the content granularity to the metadata | 46,70 % |
| Data search, indexing and catalogs | Data search metadata | Add DOI to the dataset metadata | 33,30 % |
| Data search, indexing and catalogs | Data search metadata | Add dataset version to the metadata | 16,70 % |
| Data search, indexing and catalogs | Data search metadata | Add information about FAIR compliant, license type to the metadata | 3,30 % |

| | | | |
|---|---|---|---|
| Data search, indexing and catalogs | Data search metadata | Add information on data quality (e.g., cloud cover) to the metadata | 3,30 % |
| Data search, indexing and catalogs | Data search metadata | Use a two-level metadata system, a collection (dataset) and granule (file) levels, to improve discoverability and accessibility of the data | 96,70 % |

### 3.3.4 Hardware processing capabilities

The hardware processing capabilities category includes requirements for the hardware that the platform should use to process the EO data. The most important requirements in this category are:

- **Support for CPU and GPU processing:** The platform should support both CPU and GPU processing. This will allow the platform to process large datasets more efficiently.
- **Availability of at least 30-50GB of memory:** The platform should have at least 30-50GB of memory to process large datasets.

The detailed requirements as collected from the external user survey are provided in Table 12.

**Table 12: External user computing resources requirements.**

| Category | Subcategory | Requirement | Votes |
|---|---|---|---|
| Hardware processing capabilities | Processor requirements | Availability of CPU supported processing | 36,70 % |
| Hardware processing capabilities | Processor requirements | Availability of CPU supported multiprocessing | 30,00 % |
| Hardware processing capabilities | Processor requirements | Availability of GPU supported processing | 20,00 % |
| Hardware processing capabilities | Processor requirements | Availability of GPU supported multiprocessing | 13,30 % |
| Hardware processing capabilities | Memory requirements | Availability of memory of ~ 30-50GB for processing | 100 % |

### 3.3.5 Other processing capabilities

Table 13 shows the collected requirements in the other processing capabilities category which includes requirements for other processing capabilities that the platform should provide, such as the ability to perform near-real-time processing of data, the ability to run part/all of a processing pipeline on the platform, and the ability to access intermediate results of the processing. The most important requirements in this category are:

- **Ability to perform near-real-time processing of data:** The platform should be able to perform near-real-time processing of data. This will allow users to get results quickly.
- **Ability to run part/all of a processing pipeline on the platform:** The platform should allow users to run part/all of a processing pipeline on the platform. This will allow users to customize the processing to their specific needs. Additionally, it will allow users to combine their local processing chains with processing chains running on the platform.
- **Ability to access data using API/CLI:** The platform should allow users to access data using different methods such as API, CLI, etc.
- **Ability to access intermediate results of the processing:** The platform should allow users to access intermediate results of the processing. This will allow users to verify the validity of their results and debug any possible problem.

**Table 13: Other processing requirements derived from the external user survey.**

| Category | Subcategory | Requirement | Votes |
|---|---|---|---|
| Other processing capabilities | Generic | Provide processing capabilities that allow users to perform NRT processing of data | 63,30 % |
| Other processing capabilities | Generic | Allow the users to run part/all of their processing pipeline on the platform with easy integration to their existing setups via APIs/ CLIs | 70,00 % |
| Other processing capabilities | Generic | Allow users to use the platform for data access using APIs/CLIs or any other method | 30,00 % |
| Other processing capabilities | Generic | The ability to access intermediate results of the processing to verify the processing steps | 70,00 % |
| Other processing capabilities | Generic | Allow the user to build own code/pipeline on the platform | 56,70 % |
| Other processing capabilities | Generic | Allow the user to export local code to be run and executed on the platform | 82,60 % |

### 3.3.6 Algorithms

The algorithms category includes requirements for processing algorithms and capabilities that users insisted to be useful on a platform for EO data access and processing. Indeed, users expressed that the platform should provide algorithms for data fusion and feature extraction that will allow to extract added value from EO datasets. The most important requirements in this category are:

- **Data fusion algorithms:** The platform should provide data fusion algorithms that allow aggregating data coming from multiple sources such as Sentinel-1 and Sentinel-2, Sentinel-2 and ERA-5 and MODIS and ERA-5, among others.
- **Feature extraction algorithms:** The platform should provide algorithms for thematic feature extraction such as vegetation indices (e.g., the Normalized Difference Vegetation Index (NDVI), the Normalized Difference Index (NDI)) and Machine Learning (ML) extracted features.

Table 14 provides more details on the requirements associated with the algorithms category.

**Table 14: External user algorithms requirements.**

| Category | Subcategory | Requirement | Votes |
|---|---|---|---|
| Algorithms | Data fusion | Provide data fusion algorithms to aggregate information come from different sources | 70,00 % |
| Algorithms | Data fusion | Provide S1 and S2 data fusion algorithms | 100 % |
| Algorithms | Data fusion | Provide MODIS and ERA5 data fusion algorithms | 100 % |
| Algorithms | Data fusion | Provide S2 and ERA5 fusion algorithms | 100 % |
| Algorithms | Data fusion | Provide a mechanism for automatic selection of data sources for fusion and allow the user to manually select the sources for data fusion | 100,00 % |
| Algorithms | Feature extraction | Allow the users to download features extracted from the datasets. This can be classical RS features such as NDVI from optical images or more sophisticated ML-learned features | 53,30 % |

### 3.3.7 Interfacing services

Table 15 provides details on the requirements associated with the interfacing services category, which includes requirements for interfacing services of the platform that would facilitate the platform access and use. Users expressed that the platform should provide different interfaces that will allow users to implement their code and visualize processing results and other datasets. The most important requirements in this category are:

- **Code editor:** The platform should provide interfaces to allow users to implement their code, such as JupyterLab and RStudio.
- **Visualization:** The platform should provide a variety of visualization interfaces such as WebGIS and XR/VR.

**Table 15: External user interfacing services requirements.**

| Category | Subcategory | Requirement | Votes |
|---|---|---|---|
| Interfacing services | Code editor | Access to JupyterLab to allow the user to create own code/pipelines | 66,70 % |
| Interfacing services | Code editor | Access to RStudio to allow the user to create own code/pipelines | 6,70 % |
| Interfacing services | Visualization | Enable web GIS to visualize the processing results via a webapp | 56,70 % |
| Interfacing services | Visualization | Allow downloading the processing results where the user can choose a tool of his/her choice to visualize them | 40,00 % |
| Interfacing services | Visualization | Enable GIS visualization via VR/XR | 3,30 % |

## 3.3.8 Authentication and authorization

Table 16 provides details on the requirements associated with the authentication and authorization category. Users insisted on the importance of implementing different authentication methods that are compatible with their existing platforms and environments.

**Table 16: External user authentication and authorization requirements.**

| Category | Subcategory | Requirement | Votes |
|---|---|---|---|
| Authentication and authorization | Authentication method | Enable authentication via API access token | 46,70 % |
| Authentication and authorization | Authentication method | Enable authentication via HTTPs basic auth | 26,70 % |
| Authentication and authorization | Authentication method | Enable authentication based on institutional credentials | 13,30 % |
| Authentication and authorization | Authentication method | Enable OAuth with OpenID | 6,70 % |

Funded by the European Union

# 4 Business Process Modeling and Notation

BPMN provides a modeling language that can be easily shared and understood among various stakeholders involved in business processes including technical developers responsible for implementation, business analyst who shape and refine processes and business users overseeing and managing operations. To be more specific, BPMNs enable organizations to:

- Achieve quicker consensus by employing clear and unambiguous models.
- Engage stakeholders using visually expressive notations.
- Simplify the process of analyzing and improving the operational workflows.
- Provide training materials for new employees, users, and other stakeholders by establishing a repository of process flows, case definitions and business rules.

The following sections provide BPMN workflows for different user journeys on the EO4EU platform. These workflows depict the interaction between the platform user and the underlying processes executed on the platform side. Figure 19 explains the set of notions used in these workflows.

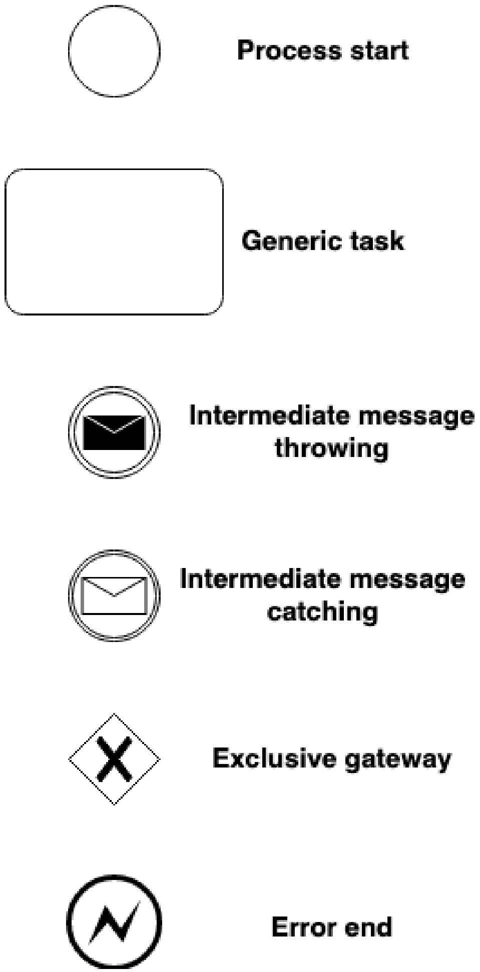


Figure 19: The set of notions used to create the BPMN workflows.

## 4.1 User Signup

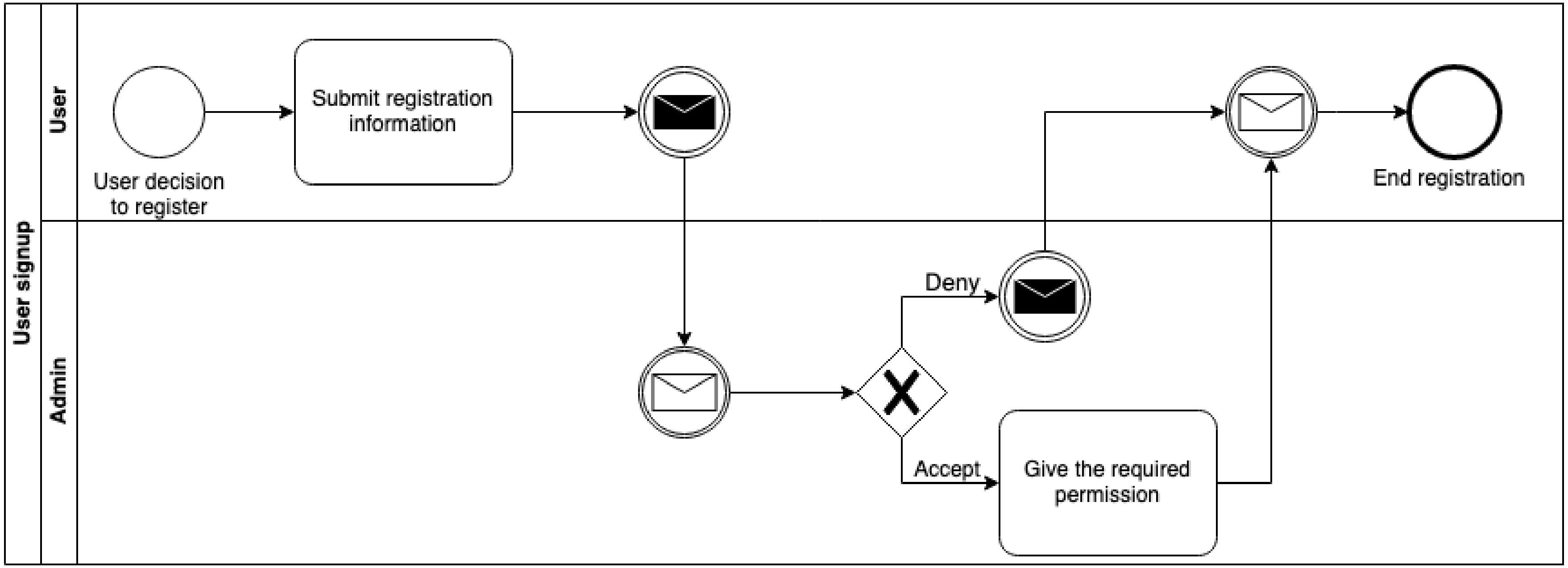

User signup
User
Admin
User decision to register
Submit registration information
End registration
Deny
Accept
Give the required permission

**Figure 20: BPMN of the user signup process.**

Figure 20 depicts the BPMN of the user signup process. A detailed description of this figure is provided in what follows:

**Start Event**

The diagram starts with a **start event**, which indicates the beginning of the process. The process starts when a user decides to create a new account in the platform, where the start event is labeled “user decision to register”.

**Task: “Submit Registration Information”**

Once the user decides to register, information about the account to be created must be filled and submitted in this task, which may include the user’s first and last names, email address, etc.

**Intermediate Message Throwing: “Sending Registration Information”**
The filled registration information in the registration form is sent to the platform to be further validated.

**Intermediate Message Catching: “Receiving Registration Information”**
The platform receives the registration information from the user.

**Exclusive Gateway: “Validating the Registration”**
The next element in the diagram is an exclusive gateway, which splits the process into two paths depending on the validity of the information sent by the user. If the information is valid the process continues by creating a user account and sending the registration confirmation to the user. On the other hand, if the information is not valid, the user is also informed that the information is not correct by sending an email.

**Task: “Assign the Required Permissions”**
Depending on the user profile, different permissions might be needed. In this task, the diagram proceeds with assigning these permissions on the right path of the exclusive gateway, which means that the registration information is valid. This task also includes the sending of the account creation confirmation to the user.

**Intermediate Message Throwing: “Rejection of Registration”**
If there is a problem with the user’s registration information, the exclusive gateway’s left path will lead to the sending of an email to the user stating the registration failure.

**Intermediate Message Catching: “Receiving Registration Status”**
At this stage, the user will receive the status of the registration (pass or fail) from the platform. This step leads to the end of the registration event.

**End Event:**
The process is terminated when the user receives information regarding the status of the registration.

Please note that the above diagram depicts two pools and two lanes, which represent the user and the platform (represented by its administrator), hence providing more information on the high-level interaction between them but not diving into the details of the communication and interaction between the platform components, which is described in D2.4.

## 4.2 Access token management

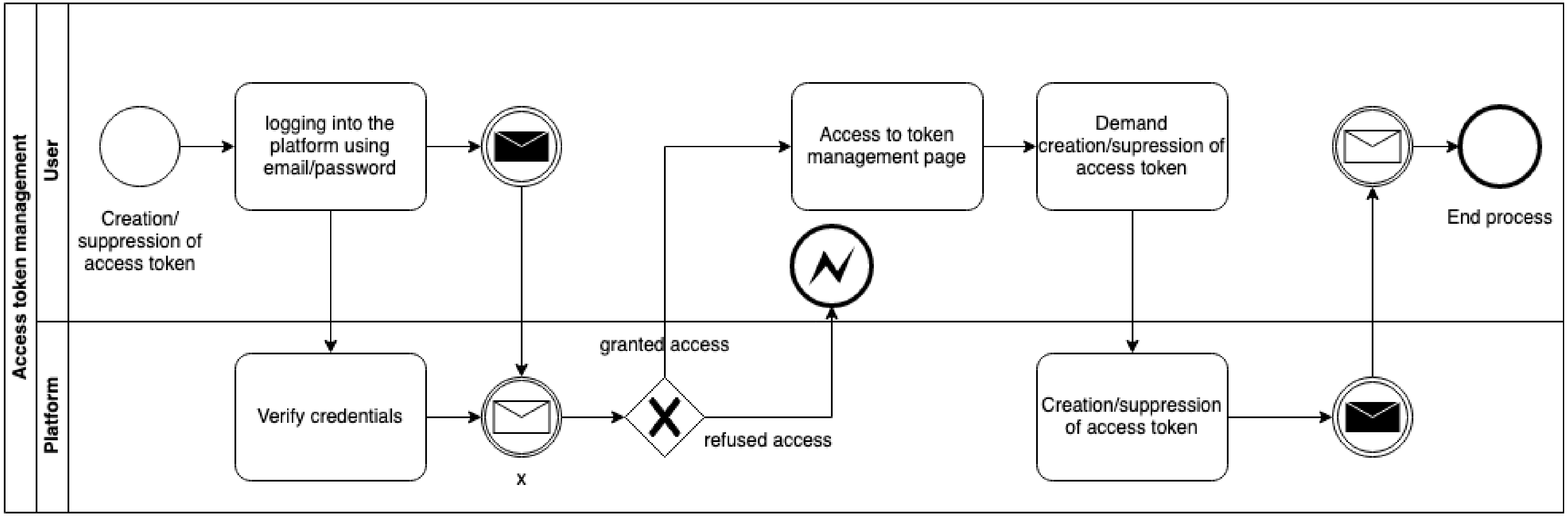

Access token management
User
Platform
Creation/ suppression of access token
logging into the platform using email/password
Access to token management page
Demand creation/supression of access token
End process
granted access
Verify credentials
x
refused access
Creation/suppression of access token

**Figure 21: BPMN of the access token management process.**

Figure 21 depicts the BPMN of access token management, which can be described as follows:

**Start Event**

The diagram starts with the user decision to create or delete an access token, which marks the starting event of the BPMN. This event is labeled as “creation/suppression of access token”, which can be found in the **user lane**.

**Task: “Logging into the Platform”**
In this task, the user uses email/password login to access the platform. The credentials are then sent to the platform to verify the identity and permissions of the user. This task is depicted in the **user lane**.

**Intermediate Message Throwing: “Sending logging information”**
The credentials are sent to the platform to be verified and to get the user’s permissions. This task is depicted in the **user lane**.

**Task: “Credentials Verification”**
The platform proceeds with the verification of the credentials to allow the user to login and access the access token management page.

**Exclusive Gateway: “Verifying User’s Permission to Access Token Management”**
An **exclusive gateway** describes the process in the platform (**platform lane**) to give the user the access to the token access management or not. This gateway leads to two paths. The right path, which proceeds with the user accessing the access token management page and the left one, which represents the user who does not have access to create/delete access tokens.

**Error End**
If the user does not have the permissions to access management, the exclusive gateway directs the process to an **Error End Event**, which marks the end of this process.

**Task: “Navigate to Access Token Management Page”**
The right path of the **exclusive gateway**, which represents the users who have permission to access token management, leads to a **task** where the user navigates to the access token management page.

**Task: “Create/Delete Access Tokens”**

Once the user is landed on the correct page, creation/suppression of access tokens is possible, where the user can create access tokens to integrate his/her apps with the platform processing capabilities. In this task, the user sends a request to create/delete an access token to the platform. This task can be found on the user lane of Figure 21.

**Task: "Creation/Suppression of Access Tokens"**
In this task (which can be found in the platform lane), the platform receives the request from the user to create/delete access token and proceeds with the requested task.

**Intermediate Message Throwing: "Sending Information on the Creation/Suppression of Access Token"**
The platform sends the response to the request of the user through this task, which is depicted on the platform lane of Figure 21.

**Intermediate Message Catching: "Receiving Information on the Creation/Suppression of Access Token"**
On the user lane, the response of the platform is received, which contains the access token if the request is successful or the information on why the request has not gone through if the request fails.

**End Event**
The process ends after receiving the response of the platform on the creation/suppression of access token. This event can be found in the user lane of Figure 21.

## 4.3 EO data search via the API

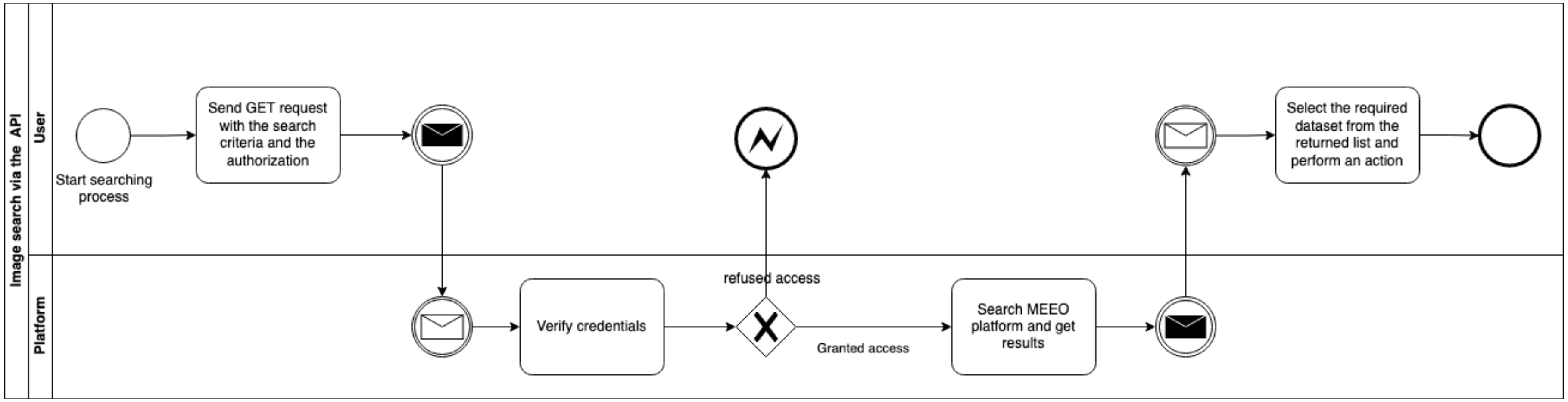


Image search via the API
User
Platform
Start searching process
Send GET request with the search criteria and the authorization
refused access
Select the required dataset from the returned list and perform an action
Verify credentials
Granted access
Search MEEO platform and get results

**Figure 22: BPMN of image search process via the API**.

Figure 22 represents the BPMN of using the API to search EO data. The figure represents two lanes, one associated with the user and the other one associated with the platform. The process and events in these two lanes are described in what follows.

### Start Event

The process starts when a user represents the willing to search for images/datasets via the platform.

### Task: "Send GET Request with the Search Criteria and Authorization"

Using a programming language of the user's choice, the user sends a GET request to the platform API that includes the search parameters, such as the dates, cloud cover, tile ID, if searching for Sentinel-2 images, as well as a proper authorization mechanism. This will only allow users with the permission to search the platform for images and datasets.

### Intermediate Message Throwing: "Send the GET Request from Client to Server"

This step represents the GET request sending from the client (user code) to the server.

**Intermediate Message Receiving: “Receive the GET Request on the Server Side”**
The step represents the reception of the GET request on the platform side.

**Task: “Verify Credentials”**
In this task, the platform verifies the credentials in the GET request to ensure that the user can use the platform and this specific service. This task can be found on the platform lane.

**Exclusive Gateway: “Allow/Deny User Access”**
The exclusive gateway represents a right path, where the user is granted access to the platform and this specific resource and proceeds with the process, as well as a left path, where the user does not have the permissions. This gateway can be found on the platform lane.

**Error End**
This error event follows the left path in the exclusive gateway for users without sufficient permissions. The process then ends at this point on the user lane of the diagram.

**Task: “Search MEEO Platform for Datasets”**
Upon the success of the authentication and authorization of the user, the platform internally communicates with MEEO platform with the search criteria to get the datasets. This task can be found on the platform lane on the diagram.

**Intermediate Message Throwing: “Send Search Results to the User”**
The platform then sends the search results obtained from MEEO platform to the user using an intermediate message throwing step. This step can be found on the platform lane of the diagram.

**Intermediate Message Receiving: “Receive Search Results from the Platform”**
The user receives the response of the search request from the platform. This step can be found on the user lane of the diagram.

**Task: “Select the Required Dataset from the Returned List and Perform Action”**

The user then selects the datasets/images from the list returned by the search process to further process/download it. This task is located on the user lane of the diagram.

**End Event**
Following the user task of process/download, the process ends with and end event.

## 4.4 EO data search via the GUI

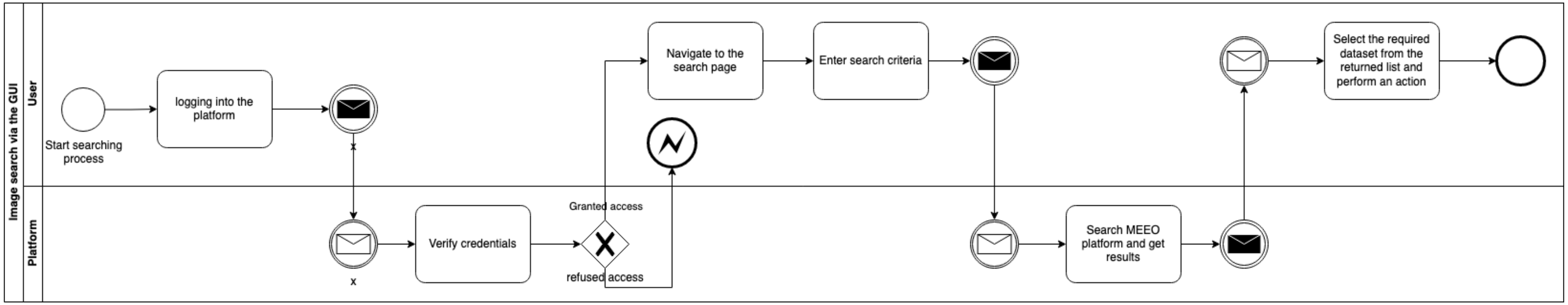


Image search via the GUI
User
Platform
Start searching process
logging into the platform
x
x
Verify credentials
Granted access
refused access
Navigate to the search page
Enter search criteria
Search MEEO platform and get results
Select the required dataset from the returned list and perform an action

**Figure 23: BPMN of image search process via the GUI.**

Figure 23 depicts the BPMN of image search via the GUI. More details on this process can be found in what follows.

**Start Event**
The start event marks the beginning of this process, which represents the user decision to start search for datasets/images using the platform GUI. The start event is located on the user lane of the diagram.

**Task: "Logging into the Platform"**
On the user lane, the user logs into the platform using their credentials and the GUI.

**Intermediate Message Throwing: “Sending User Credentials from the Client Side to the Server Side”**
Then, an intermediate message throwing step is responsible for sending the user credentials to the platform in order to be verified.

**Intermediate Message Receiving: “Receiving User Credentials from the Client Side”**
The platform receives the credentials from the client side, which will be verified in the next step.

**Task: “Verify Credentials”**
In this task, the platform verifies the credentials in the GET request to make sure that the user is allowed to use the platform and can use this specific service. This task can be found on the platform lane.

**Exclusive Gateway: “Allow/Deny User Access”**
The exclusive gateway represents a right path, where the user is granted access to the platform and this specific resource and proceeds with the process, as well as a left path, where the user does not have the permissions. This gateway can be found on the platform lane.

**Error End**
This error event follows the left path in the exclusive gateway for users without sufficient permissions. The process then ends at this point on the user lane of the diagram.

**Task: “Navigate to Image/Dataset Search Page on the Platform”**
If the login is successful, the user navigates to the image/dataset search page on the platform, where the user will use the GUI to search for the required datasets.

**Task: “Enter Search Criteria”**
On the GUI the user enters search criteria to be sent to the platform such as the dataset name, dates, temporal and spatial coverage.

**Intermediate Message Throwing: “Sending Search Criteria to the Platform”**
When the search criteria entered by the user are ready, the user click on a button “Search/Submit” to send the request to the platform.

**Intermediate Message Receiving: “Receiving Search Criteria from the User”**
The platform receives the request from the GUI, which includes information about the required dataset.

**Task: “Search MEEO Platform for Datasets”**
Upon the success of the authentication and authorization of the user, the platform internally communicates with MEEO platform with the search criteria to get the datasets. This task can be found on the platform lane on the diagram.

**Intermediate Message Throwing: “Send Search Results to the User”**
The platform then sends the search results obtained from MEEO platform to the user using an intermediate message throwing step. This step can be found on the platform lane of the diagram.

**Intermediate Message Receiving: “Receive Search Results from the Platform”**
The user receives the response of the search request from the platform. This step can be found on the user lane of the diagram.

**Task: “Select the Required Dataset from the Returned List and Perform Action”**
The user then selects the datasets/images from the list returned by the search process to further process/download it. This task is located on the user lane of the diagram.

**End Event**
Following the user task of process/download, the process ends with and end event.

## 4.5 Data fusion via the API

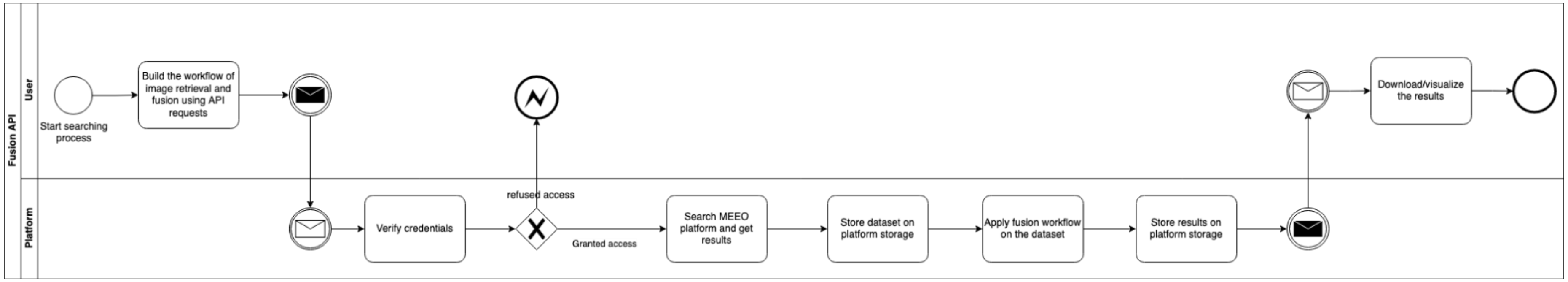

Fusion API
User
Platform
Start searching process
Build the workflow of image retrieval and fusion using API requests
refused access
Granted access
Verify credentials
Search MEEO platform and get results
Store dataset on platform storage
Apply fusion workflow on the dataset
Store results on platform storage
Download/visualize the results

**Figure 24: BPMN of the data fusion process via the API.**

Figure 24 describes the data fusion BPMN via the API. More details about this process are provided in what follows.

**Start Event**
The process starts when the user decides to run a data fusion process on the platform. This is highlighted using a start event on the user lane of the previous diagram.

**Task: “Build the Workflow of Image Retrieval and Data Fusion using the API”**
The user then proceeds with building the workflow using the platform API and a programming language of their choice. The request also includes the authentication/authorization mechanism required to access the platform capabilities. Note that this task is on the user lane of the previous diagram.

**Intermediate Message Throwing: “Send Workflow and Authentication/Authorization Information”**
On the user lane, a POST request is sent to the server side (platform lane), which contains the information required to build the workflow as well as the authentication/authorization information.

**Intermediate Message Receiving: “Receive Workflow and Authentication/Authorization Information”**
On the platform lane, the server side receives the POST request from the client side.

**Task: “Verify Credentials”**

In this task, the platform verifies the credentials in the GET request to make sure that the user is allowed to use the platform and can use this specific service. This task can be found on the platform lane. This step also includes the validation of the request sent by the user.

**Exclusive Gateway: “Allow/Deny User Access”**

The exclusive gateway represents a right path, where the user is granted access to the platform and this specific resource and proceeds with the process, as well as a left path, where the user does not have the permissions. This gateway can be found on the platform lane.

**Task: “Search MEEO Platform for Datasets”**

Upon the success of the authentication and authorization of the user, the platform internally communicates with MEEO platform with the search criteria to get the datasets. This task can be found on the platform lane on the diagram.

**Task: “Store the Dataset Temporarily on the Platform Storage”**

The required dataset is retrieved and stored on the platform temporary storage in order to be further processed by the fusion component. This task can be found on the platform lane of the previous diagram.

**Task: “Apply Data Fusion on the Retrieved Dataset”**

The data fusion component is then applied on the dataset stored on the temporary storage. This task is located on the platform lane of the previous diagram.

**Task: “Store the Results on the Platform Storage”**

The results of the data fusion process are stored on the platform storage to prepare the results delivery to the user. This task can be found on the platform lane of the previous diagram.

**Intermediate Message Throwing: “Sending the Data Fusion Workflow Results and Status”**

On the platform lane, a message is prepared to be delivered to the user. This message contains the processing final status, data fusion results and any additional information that can be useful. If the whole process is synchronous, this message will be delivered in the POST response. Otherwise, the user will be informed by other means.

**Intermediate Message Receiving: “Receiving the Data Fusion Workflow Results and Status”**
Depending on how the previous step is executed, the receiving of the message can be by different means. For synchronous request, this will be received via the response of the POST request performed by the user. Otherwise, other means will be used to receive the message.

**Task: “Download/Visualize the Results”**
The user then proceeds by either downloading the results directly or visualizing the results of the platform and downloading them from the GUI.

**End Event**
This event marks the end of the image retrieval and data fusion workflow.

## 4.6 Data fusion via the GUI

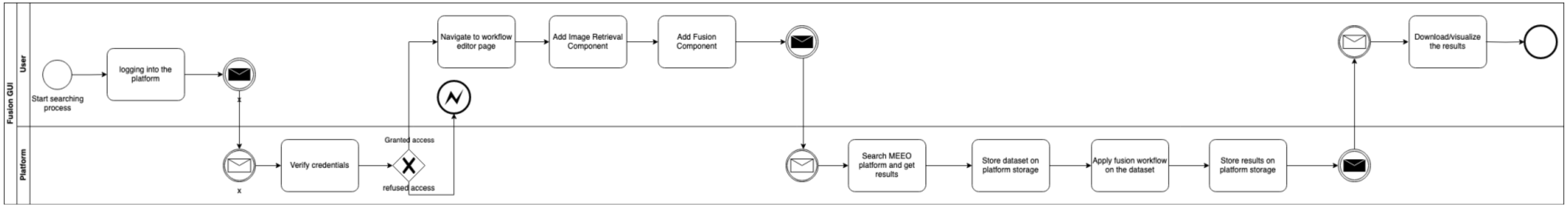

Fusion GUI
User
Platform
Start searching process
logging into the platform
x
Verify credentials
Granted access
refused access
Navigate to workflow editor page
Add Image Retrieval Component
Add Fusion Component
Search MEEO platform and get results
Store dataset on platform storage
Apply fusion workflow on the dataset
Store results on platform storage
Download/visualize the results

**Figure 25: BPMN of the data fusion process via the GUI.**

The BPMN of the data fusion process via the GUI is depicted in Figure 25. More details on this figure are provided in what follow.

**Start Event**
The process starts when the user decides to run a data fusion process on the platform. This is highlighted using a start event on the user lane of the previous diagram.

**Task: “Logging into the Platform”**
On the user lane, the user logs into the platform using their credentials and the GUI.

**Intermediate Message Throwing: “Sending User Credentials from the Client Side to the Server Side”**
Then, an intermediate message throwing step is responsible for sending the user credentials to the platform in order to be verified.

**Intermediate Message Receiving: “Receiving User Credentials from the Client Side”**
The platform receives the credentials from the client side, which will be verified in the next step.

**Task: “Verify Credentials”**
In this task, the platform verifies the credentials in the GET request to make sure that the user is allowed to use the platform and can use this specific service. This task can be found on the platform lane.

**Exclusive Gateway: “Allow/Deny User Access”**
The exclusive gateway represents a right path, where the user is granted access to the platform and this specific resource and proceeds with the process, as well as a left path, where the user does not have the permissions. This gateway can be found on the platform lane.

**Error End**
This error event follows the left path in the exclusive gateway for users without sufficient permissions. The process then ends at this point on the user lane of the diagram.

**Task: “Navigate to Workflow Editor Page on the Platform”**
If the login is successful, the user navigates to the workflow editor page on the platform, where the user will use the GUI to create the image retrieval and data fusion workflow.

**Task: “Workflow Creation”**
On the user lane, the workflow creation task comprises two subtasks: adding the image retrieval component and adding the data fusion component to the workflow using the workflow editor.

**Intermediate Message Throwing: “Send Workflow Information”**

On the user lane, the user clicks on a “Submit/Send” button to send the workflow request to the server side (platform lane), which contains the information required to build the workflow.

**Intermediate Message Receiving: “Receive Workflow Information”**

On the platform lane, the server side receives the POST request from the client side (GUI) to create the workflow.

**Task: “Search MEEO Platform for Datasets”**

Upon the success of the authentication and authorization of the user, the platform internally communicates with MEEO platform with the search criteria to get the datasets. This task can be found on the platform lane on the diagram.

**Task: “Store the Dataset Temporarily on the Platform Storage”**

The required dataset is retrieved and stored on the platform temporary storage in order to be further processed by the fusion component. This task can be found on the platform lane of the previous diagram.

**Task: “Apply Data Fusion on the Retrieved Dataset”**

The data fusion component is then applied on the dataset stored on the temporary storage. This task is located on the platform lane of the previous diagram.

**Task: “Store the Results on the Platform Storage”**

The results of the data fusion process are stored on the platform storage to prepare the results delivery to the user. This task can be found on the platform lane of the previous diagram.

**Intermediate Message Throwing: “Sending the Data Fusion Workflow Results and Status”**

On the platform lane, a message is prepared to be delivered to the user. This message contains the processing final status, data fusion results and any additional information that can be useful. If the whole process is synchronous, this message will be delivered in the POST response. Otherwise, the user will be informed by other means.

**Intermediate Message Receiving: “Receiving the Data Fusion Workflow Results and Status”**
Depending on how the previous step is executed, the receiving of the message can be by different means. For synchronous request, this will be received via the response of the POST request performed by the user. Otherwise, other means will be used to receive the message.

**Task: “Download/Visualize the Results”**
The user then proceeds by either downloading the results directly or visualizing the results of the platform and downloading them from the GUI.

**End Event**
This event marks the end of the image retrieval and data fusion workflow.

## 4.7 ML via the API

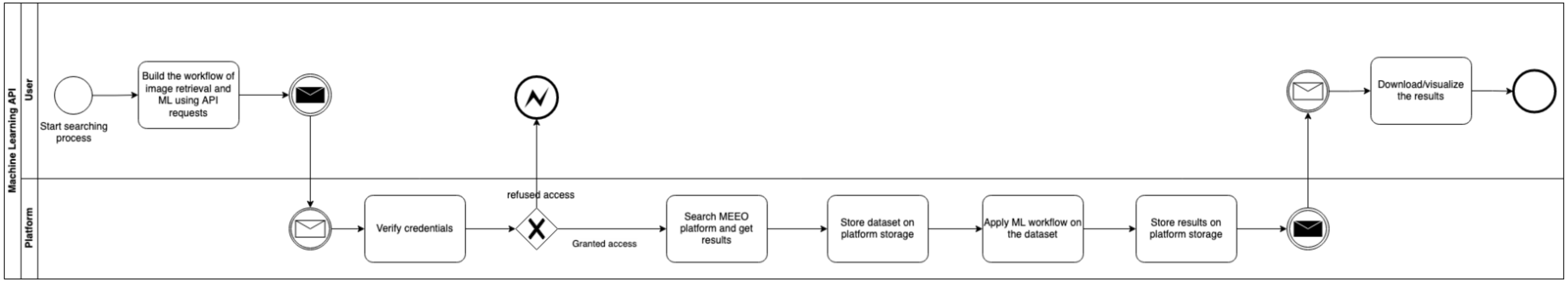

Machine Learning API
User
Platform
Start searching process
Build the workflow of image retrieval and ML using API requests
refused access
Granted access
Verify credentials
Search MEEO platform and get results
Store dataset on platform storage
Apply ML workflow on the dataset
Store results on platform storage
Download/visualize the results

**Figure 26: BPMN of the ML process via the API.**

The BPMN of the ML process via the API is depicted in Figure 26 and described in what follows.

**Start Event**
The process starts when the user decides to run an ML algorithm process on the platform. This is highlighted using a start event on the user lane of the previous diagram.

**Task: “Build the Workflow of Image Retrieval and ML using the API”**
The user then proceeds with building the workflow using the platform API and a programming language of their choice. The request also includes the authentication/authorization mechanism required to access the platform capabilities. Note that this task is on the user lane of the previous diagram.

**Intermediate Message Throwing: “Send Workflow and Authentication/Authorization Information”**
On the user lane, a POST request is sent to the server side (platform lane), which contains the information required to build the workflow as well as the authentication/authorization information.

**Intermediate Message Receiving: “Receive Workflow and Authentication/Authorization Information”**
On the platform lane, the server side receives the POST request from the client side.

**Task: “Verify Credentials”**
In this task, the platform verifies the credentials in the GET request to make sure that the user is allowed to use the platform and can use this specific service. This task can be found on the platform lane. This step also includes the validation of the request sent by the user.

**Exclusive Gateway: “Allow/Deny User Access”**
The exclusive gateway represents a right path, where the user is granted access to the platform and this specific resource and proceeds with the process, as well as a left path, where the user does not have the permissions. This gateway can be found on the platform lane.

**Task: “Search MEEO Platform for Datasets”**
Upon the success of the authentication and authorization of the user, the platform internally communicates with MEEO platform with the search criteria to get the datasets. This task can be found on the platform lane on the diagram.

**Task: “Store the Dataset Temporarily on the Platform Storage”**
The required dataset is retrieved and stored on the platform temporary storage in order to be further processed by the ML component. This task can be found on the platform lane of the previous diagram.

**Task: “Apply ML on the Retrieved Dataset”**
The ML component is then applied on the dataset stored on the temporary storage. This task is located on the platform lane of the previous diagram.

**Task: “Store the Results on the Platform Storage”**
The results of the ML process are stored on the platform storage to prepare the results delivery to the user. This task can be found on the platform lane of the previous diagram.

**Intermediate Message Throwing: “Sending the ML Workflow Results and Status”**

On the platform lane, a message is prepared to be delivered to the user. This message contains the processing final status, ML results and any additional information that can be useful. If the whole process is synchronous, this message will be delivered in the POST response. Otherwise, the user will be informed by other means.

**Intermediate Message Receiving: “Receiving the ML Workflow Results and Status”**
Depending on how the previous step is executed, the receiving of the message can be by different means. For synchronous request, this will be received via the response of the POST request performed by the user. Otherwise, other means will be used to receive the message.

**Task: “Download/Visualize the Results”**
The user then proceeds by either downloading the results directly or visualizing the results of the platform and downloading them from the GUI.

**End Event**
This event marks the end of the image retrieval and ML workflow.

## 4.8 ML via the GUI

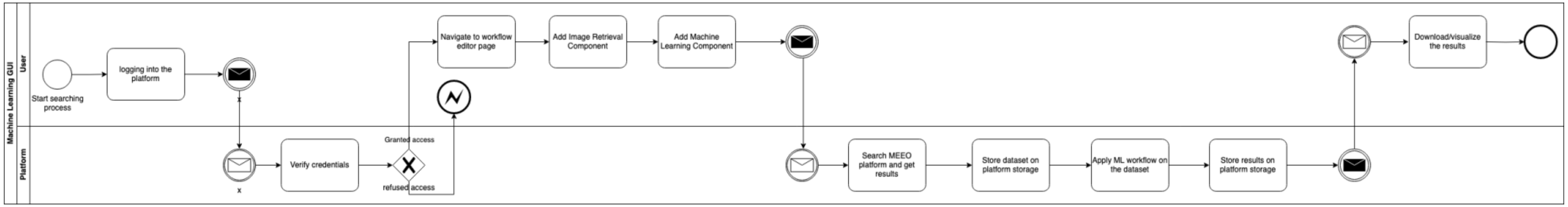

Machine Learning GUI
User
Platform
Start searching process
logging into the platform
x
x
Verify credentials
Granted access
refused access
Navigate to workflow editor page
Add Image Retrieval Component
Add Machine Learning Component
Search MEEO platform and get results
Store dataset on platform storage
Apply ML workflow on the dataset
Store results on platform storage
Download/visualize the results

**Figure 27: BPMN of the ML process via the GUI.**

The BPMN of the ML process via the GUI is depicted in Figure 27. More details on this figure are provided in what follow.

**Start Event**
The process starts when the user decides to run an ML process on the platform. This is highlighted using a start event on the user lane of the previous diagram.

**Task: "Logging into the Platform"**
On the user lane, the user logs into the platform using their credentials and the GUI.

**Intermediate Message Throwing: "Sending User Credentials from the Client Side to the Server Side"**
Then, an intermediate message throwing step is responsible for sending the user credentials to the platform in order to be verified.

**Intermediate Message Receiving: "Receiving User Credentials from the Client Side"**
The platform receives the credentials from the client side, which will be verified in the next step.

**Task: "Verify Credentials"**
In this task, the platform verifies the credentials in the GET request to make sure that the user is allowed to use the platform and can use this specific service. This task can be found on the platform lane.

**Exclusive Gateway: "Allow/Deny User Access"**
The exclusive gateway represents a right path, where the user is granted access to the platform and this specific resource and proceeds with the process, as well as a left path, where the user does not have the permissions. This gateway can be found on the platform lane.

**Error End**
This error event follows the left path in the exclusive gateway for users without sufficient permissions. The process then ends at this point on the user lane of the diagram.

**Task: "Navigate to Workflow Editor Page on the Platform"**
If the login is successful, the user navigates to the workflow editor page on the platform, where the user will use the GUI to create the image retrieval and data fusion workflow.

**Task: "Workflow Creation"**

On the user lane, the workflow creation task comprises two subtasks: adding the image retrieval component and adding the ML component to the workflow using the workflow editor.

**Intermediate Message Throwing: "Send Workflow Information"**

On the user lane, the user clicks on a "Submit/Send" button to send the workflow request to the server side (platform lane), which contains the information required to build the workflow.

**Intermediate Message Receiving: "Receive Workflow Information"**

On the platform lane, the server side receives the POST request from the client side (GUI) to create the workflow.

**Task: "Search MEEO Platform for Datasets"**

Upon the success of the authentication and authorization of the user, the platform internally communicates with MEEO platform with the search criteria to get the datasets. This task can be found on the platform lane on the diagram.

**Task: "Store the Dataset Temporarily on the Platform Storage"**

The required dataset is retrieved and stored on the platform temporary storage in order to be further processed by the ML component. This task can be found on the platform lane of the previous diagram.

**Task: "Apply ML on the Retrieved Dataset"**

The ML component is then applied on the dataset stored on the temporary storage. This task is located on the platform lane of the previous diagram.

**Task: "Store the Results on the Platform Storage"**

The results of the ML process are stored on the platform storage to prepare the results delivery to the user. This task can be found on the platform lane of the previous diagram.

**Intermediate Message Throwing: "Sending the ML Workflow Results and Status"**

On the platform lane, a message is prepared to be delivered to the user. This message contains the processing final status, ML results and any additional information that can be useful. If the whole process is synchronous, this message will be delivered in the POST response. Otherwise, the user will be informed by other means.

**Intermediate Message Receiving: “Receiving the ML Workflow Results and Status”**
Depending on how the previous step is executed, the receiving of the message can be by different means. For synchronous request, this will be received via the response of the POST request performed by the user. Otherwise, other means will be used to receive the message.

**Task: “Download/Visualize the Results”**
The user then proceeds by either downloading the results directly or visualizing the results of the platform and downloading them from the GUI.

**End Event**
This event marks the end of the image retrieval and data fusion workflow.

## 4.9 Data fusion and ML via the API

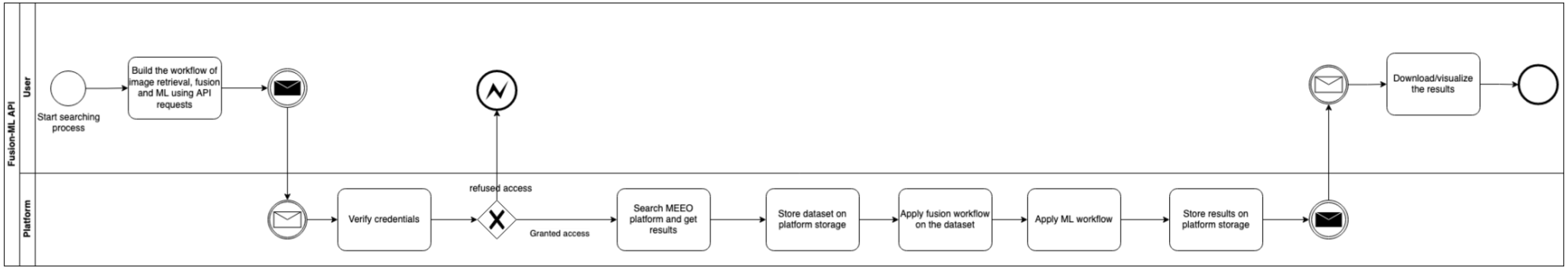

Fusion-ML API
User
Platform
Start searching process
Build the workflow of image retrieval, fusion and ML using API requests
refused access
Verify credentials
Granted access
Search MEEO platform and get results
Store dataset on platform storage
Apply fusion workflow on the dataset
Apply ML workflow
Store results on platform storage
Download/visualize the results

**Figure 28: BPMN of the data fusion and ML workflow via the API.**

Figure 28 describes the data fusion and ML workflow BPMN via the API. More details about this process are provided in what follows.

**Start Event**

The process starts when the user decides to run a data fusion process on the platform. This is highlighted using a start event on the user lane of the previous diagram.

**Task: "Build the Workflow of Image Retrieval, Data Fusion and ML using the API"**

The user then proceeds with building the workflow using the platform API and a programming language of their choice. The request also includes the authentication/authorization mechanism required to access the platform capabilities. Note that this task is on the user lane of the previous diagram.

**Intermediate Message Throwing: "Send Workflow and Authentication/Authorization Information"**

On the user lane, a POST request is sent to the server side (platform lane), which contains the information required to build the workflow as well as the authentication/authorization information.

**Intermediate Message Receiving: "Receive Workflow and Authentication/Authorization Information"**

On the platform lane, the server side receives the POST request from the client side.

**Task: "Verify Credentials"**

In this task, the platform verifies the credentials in the GET request to make sure that the user is allowed to use the platform and can use this specific service. This task can be found on the platform lane. This step also includes the validation of the request sent by the user.

**Exclusive Gateway: "Allow/Deny User Access"**

The exclusive gateway represents a right path, where the user is granted access to the platform and this specific resource and proceeds with the process, as well as a left path, where the user does not have the permissions. This gateway can be found on the platform lane.

**Task: "Search MEEO Platform for Datasets"**

Upon the success of the authentication and authorization of the user, the platform internally communicates with MEEO platform with the search criteria to get the datasets. This task can be found on the platform lane on the diagram.

**Task: "Store the Dataset Temporarily on the Platform Storage"**
The required dataset is retrieved and stored on the platform temporary storage in order to be further processed by the fusion component. This task can be found on the platform lane of the previous diagram.

**Task: "Apply Data Fusion on the Retrieved Dataset"**
The data fusion component is then applied on the dataset stored on the temporary storage. This task is located on the platform lane of the previous diagram.

**Task: "Store the Results on the Platform Storage"**
The results of the data fusion process are stored on the platform storage to prepare the results for the ML workflow. This task can be found on the platform lane of the previous diagram.

**Task: "Apply ML on the Data Fusion Results"**
The ML component is then applied on the dataset stored on the temporary storage, which is produced by the data fusion component. This task is located on the platform lane of the previous diagram.

**Task: "Store the Results on the Platform Storage"**
The results of the ML process are stored on the platform storage to prepare the results delivery to the user. This task can be found on the platform lane of the previous diagram.

**Intermediate Message Throwing: "Sending the Data Fusion and ML Workflow Results and Status"**
On the platform lane, a message is prepared to be delivered to the user. This message contains the processing final status, data fusion and ML results and any additional information that can be useful. If the whole process is synchronous, this message will be delivered in the POST response. Otherwise, the user will be informed by other means.

**Intermediate Message Receiving: “Receiving the Data Fusion and ML Workflow Results and Status”**

Depending on how the previous step is executed, the receiving of the message can be by different means. For synchronous request, this will be received via the response of the POST request performed by the user. Otherwise, other means will be used to receive the message.

**Task: “Download/Visualize the Results”**

The user then proceeds by either downloading the results directly or visualizing the results of the platform and downloading them from the GUI.

**End Event**

This event marks the end of the image retrieval and data fusion workflow.

## 4.10 Data fusion and ML via the GUI

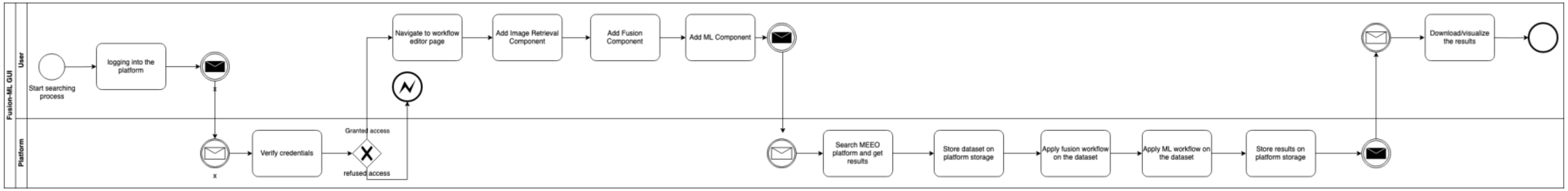

Fusion-ML GUI
User
Platform
Start searching process
logging into the platform
Navigate to workflow editor page
Add Image Retrieval Component
Add Fusion Component
Add ML Component
Download/visualize the results
x
Verify credentials
Granted access
refused access
Search MEEO platform and get results
Store dataset on platform storage
Apply fusion workflow on the dataset
Apply ML workflow on the dataset
Store results on platform storage

**Figure 29: BPMN of data fusion and ML workflow via the GUI.**

The BPMN of the data fusion and ML workflow via the GUI is depicted in Figure 29. More details on this figure are provided in what follow.

**Start Event**

The process starts when the user decides to run a data fusion and ML workflow on the platform. This is highlighted using a start event on the user lane of the previous diagram.

**Task: “Logging into the Platform”**

On the user lane, the user logs into the platform using their credentials and the GUI.

**Intermediate Message Throwing: “Sending User Credentials from the Client Side to the Server Side”**

Then, an intermediate message throwing step is responsible for sending the user credentials to the platform in order to be verified.

**Intermediate Message Receiving: “Receiving User Credentials from the Client Side”**

The platform receives the credentials from the client side, which will be verified in the next step.

**Task: “Verify Credentials”**

In this task, the platform verifies the credentials in the GET request to make sure that the user is allowed to use the platform and can use this specific service. This task can be found on the platform lane.

**Exclusive Gateway: “Allow/Deny User Access”**

The exclusive gateway represents a right path, where the user is granted access to the platform and this specific resource and proceeds with the process, as well as a left path, where the user does not have the permissions. This gateway can be found on the platform lane.

**Error End**

This error event follows the left path in the exclusive gateway for users without sufficient permissions. The process then ends at this point on the user lane of the diagram.

**Task: “Navigate to Workflow Editor Page on the Platform”**

If the login is successful, the user navigates to the workflow editor page on the platform, where the user will use the GUI to create the image retrieval, data fusion and ML workflow.

**Task: “Workflow Creation”**

On the user lane, the workflow creation task comprises three subtasks: adding the image retrieval component, adding the data fusion component and adding the ML component to the workflow using the workflow editor.

**Intermediate Message Throwing: “Send Workflow Information”**

On the user lane, the user clicks on a “Submit/Send” button to send the workflow request to the server side (platform lane), which contains the information required to build the workflow.

**Intermediate Message Receiving: “Receive Workflow Information”**

On the platform lane, the server side receives the POST request from the client side (GUI) to create the workflow.

**Task: “Search MEEO Platform for Datasets”**

Upon the success of the authentication and authorization of the user, the platform internally communicates with MEEO platform with the search criteria to get the datasets. This task can be found on the platform lane on the diagram.

**Task: “Store the Dataset Temporarily on the Platform Storage”**

The required dataset is retrieved and stored on the platform temporary storage in order to be further processed by the fusion component. This task can be found on the platform lane of the previous diagram.

**Task: “Apply Data Fusion on the Retrieved Dataset”**

The data fusion component is then applied on the dataset stored on the temporary storage. This task is located on the platform lane of the previous diagram.

**Task: “Store the Results on the Platform Storage”**

The results of the data fusion process are stored on the platform storage to prepare the results for the ML component. This task can be found on the platform lane of the previous diagram.

**Task: “Apply ML on the Data Fusion Results”**

The ML component is then applied on the dataset stored on the temporary storage, which is produced by the data fusion component. This task is located on the platform lane of the previous diagram.

**Task: “Store the Results on the Platform Storage”**

Funded by
the European Union

The results of the ML n process are stored on the platform storage to prepare the results delivery to the user. This task can be found on the platform lane of the previous diagram.

**Intermediate Message Throwing: “Sending the Data Fusion and ML Workflow Results and Status”**
On the platform lane, a message is prepared to be delivered to the user. This message contains the processing final status, data fusion and ML results and any additional information that can be useful. If the whole process is synchronous, this message will be delivered in the POST response. Otherwise, the user will be informed by other means.

**Intermediate Message Receiving: “Receiving the Data Fusion and ML Workflow Results and Status”**
Depending on how the previous step is executed, the receiving of the message can be by different means. For synchronous request, this will be received via the response of the POST request performed by the user. Otherwise, other means will be used to receive the message.

**Task: “Download/Visualize the Results”**
The user then proceeds by either downloading the results directly or visualizing the results of the platform and downloading them from the GUI.

**End Event**
This event marks the end of the image retrieval and data fusion workflow.

# 5 Conclusion

This document has presented a comprehensive overview of the requirements collection undertaken in the EO4EU project with the overarching goal of making EO data more accessible, usable, and valuable to a diverse range of users.

Requirements were extracted and synthesized from various sources, including the DoA, the use cases proposed in the project, and external user inputs. These requirements have served as the foundation upon which the EO4EU platform is being designed in such a way to be a single point of accessing and processing EO data using tools tailored to meet the unique needs of its user base.

The collaborative efforts of partners involved in Tasks 2.1 ,2.2, 2.3 and 5.1 have allowed gaining a comprehensive understanding of both external and internal user requirements. This synergy has been instrumental in the effective design and implementation of the EO4EU, bridging the gap between user expectations and technological capabilities.

BPMN has also been used to provide a holistic view of the user experience by mapping out user journeys across multiple scenarios. These BPMNs offer high-level insights into the interactions between users and the platform, ensuring that efficiency, effectiveness, and scalability are at the forefront of the EO4EU platform design considerations.