D3.1 ENHANCE Open One Health Core Platform

WP3: ENHANCE AI-enabled toolkit for coastal management

Authors: Iason Sotiropoulos (SLG), Stamatia Rizou (SLG), Eva Flo (CSIC), Meritxell Turo (CSIC), Jaume Piera (CSIC), Evmorfia Bataka (UTH)
Date: 30/12/2025

Title of Deliverable	D3.1 ENHANCE Open One Health Core Platform
Type of Deliverable	R, P
Dissemination Level	PU
Work Package	WP3 – ENHANCE Open One Health Core Platform
Lead Beneficiary	SingularLogic (SLG)
Author(s)	Dr. Stamata Rizou, Iason Sotiropoulos (SLG)
Contributors	Lien Bakelants, Eva Flo, Andreu Fornos, Meritxell Turó, Jaume Piera (CSIC), Evmorfia Bataka (UTH)
Primary Contact and Email	Dr. Stamatia Rizou srizou@singularlogic.eu
Data Due.	31/12/2025
Date Submitted	30/12/2025
Status (In progress / Draft / Completed)	Completed
Reviewed by (if applicable)	Completed
Suggested Citation	Iason Sotiropoulos, Stamatia Rizou (2025) ENHANCE Open One Health Core Platform. ENHANCE Deliverable 3.1, Horizon Europe grant no. 10113447

Date	Version	Contributors	Comments
27/11/2025	V1.0	Stamatia Rizou, Iason Sotiropoulos (SLG)	First draft version of the deliverable
19/12/2025	V.2.0	Lien Bakelants(NAZKA)	Contribution to Section 7 about the Toolkit's interface
19/12/2025	V2.0	Eva Flo, Andreu Fornos, Meritxell Turó, Jaume Piera (CSIC)	Filled in sections 5, regarding pressures, and 6, regarding impacts. Reviewed content.
23/12/2025	V2.1	Eva Flo (CSIC)	Internal Review
30/12/2025	V3.0	Antona Vronti, Iason Sotiropoulos (SLG)	Final Consolidation of the deliverable

Table of Contents

Table of Abbreviations

Abbreviations	Explanation
AI	Artificial Intelligence
a.k.a	Also Known As
API	Application Programming Interface
Chl-*a*	Chlorophyll-*a*
CMEMS	Copernicus Marine Environment Monitoring Service
CORINE	European Coordination of Information on the Environment
D	Deliverable
EGNSS	European Global Navigation Satellite Systems
EO	Earth Observation

ES	Spain
GIS	Geographic Information System
GR	Greece
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
I/O	Input/Output
IoT	Internet of Things
LECZ	Low Elevation Coastal Zone
OIDC	OpenID Connect
T	Task
SST	Sea Surface Temperature
ToC	Table of Contents
WP	Work Package
WFD	Water Framework Directive

Table of Figures

Table of Tables

Executive Summary

The ENHANCE Open One Health Core Platform is the foundational infrastructure developed under Work Package 3 of the ENHANCE project, designed to deliver actionable intelligence on environmental pressures, impacts, and risks in coastal environments.

The deliverable presents the first release of the core platform, which is built as a modular, cloud-native stack, the platform features secure identity and access management (Keycloak, OIDC), scalable object storage (MinIO), and a lightweight metadata registry (MongoDB), all orchestrated via Kubernetes for resilience and scalability. Its architecture is intentionally minimal for early integration with project services and toolkits, while remaining compatible with future data-sharing patterns and external inputs. In addition, the deliverable presents operational pipelines for environmental pressure and impact assessment, predictive modelling, and dynamic risk mapping, leveraging advanced AI and statistical models to process heterogeneous data and produce decision-ready insights.

The platform's user-centric design is reflected in the iterative development of interactive interfaces tailored to diverse stakeholders, including the public, researchers, and decision-makers. By establishing a secure, interoperable, and extensible foundation, the ENHANCE Core Platform is positioned to drive innovation and collaboration in coastal management and One Health initiatives, supporting both current pilot needs and future extensions.This deliverable reports the first release of the ENHANCE Open One Health Core Platform—the data and services backbone of Work Package (WP) 3 that enables the generation and delivery of decision-ready intelligence for Pressures, Impacts, and Risk in coastal environments. The Core Platform consolidates requirements emerging from WP2's co-creation cycle (personas, user stories, user requirements, technical use cases) and instantiates them as a modular, standards-aware, cloud-native stack that ingests, governs, registers, and exposes data and derived products to the ENHANCE Toolkit and external clients.

1. Introduction

1.1 Context and Purpose

ENHANCE brings a One Health approach to coastal management by combining Copernicus EO, in-situ, and citizen-science data into actionable insights for two pilots: Barcelona and Ebre coast (ES) and Pagasitikos Gulf (GR). WP2 distilled stakeholder needs into personas, user requirements, and technical use cases, and consolidated a Conceptual Architecture that frames the technical work of WP3. This deliverable (D3.1) documents how those inputs are translated into an operational Core Platform baseline that underpins the Pressures–Impacts–Risk services and the ENHANCE Toolkit.

1.2 Scope of Deliverable

D3.1 mainly focuses on the Core Platform (T3.1). It: (i) explains the process linking WP2 outputs to platform design; (ii) states requirements and design drivers (interoperability, sovereignty, scalability, provenance); (iii) details the Conceptual & High-Level Architecture and how it is instantiated in this first deployment; (iv) defines initial interfaces & conventions (metadata profiles, I/O contracts) for early integration and demonstration; and (v) outlines the near-term roadmap toward the M13 milestone. Finally, Pressures characterization and quantification description, corresponding to T3.2, and Impact assessment Tools description, corresponding to T3.3, will complete this deliverable.

2. ENHANCE High Level Architecture

ENHANCE aims to build an open, modular, interoperable platform that will enable the integration and processing of heterogeneous data, allowing the development of added-value services. Figure 1 shows the proposed platform's high-level architecture.

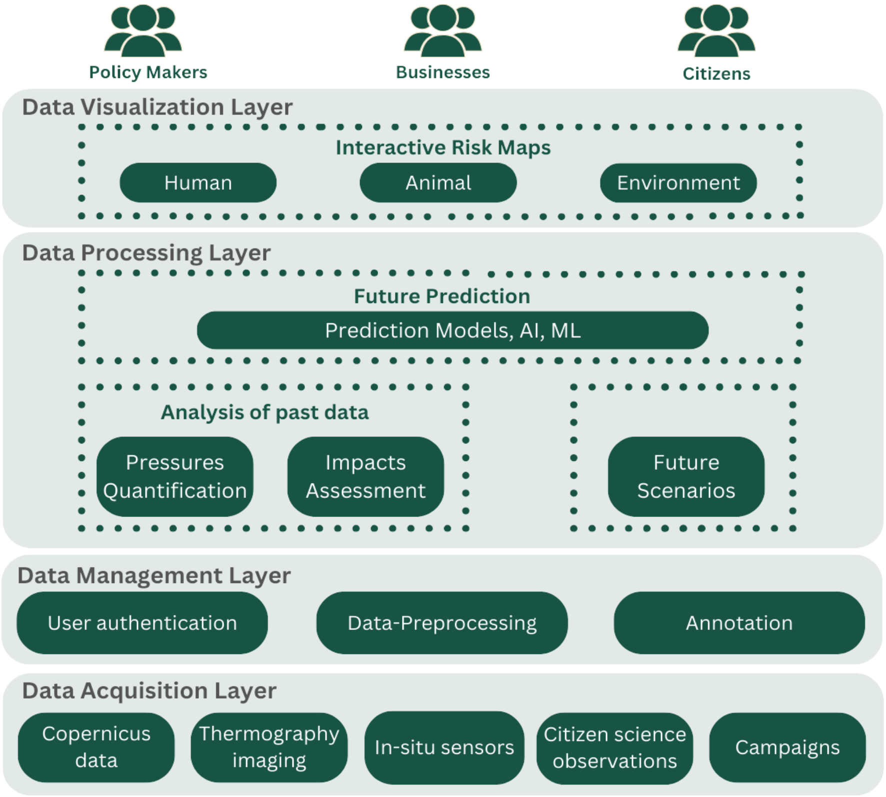


Figure 1: ENHANCE High-Level Architecture

Data Acquisition Layer: A baseline core platform will implement the data acquisition and the data management layers of the architecture as shown in Figure 1. The platform will implement interfaces to relevant Copernicus core services, citizen science observatories, and external data sources, such as external data repositories (e.g. open data repositories for demographic data) and Internet of Things (IoT) data (e.g. thermal images captured from drones).

Data Management: On top of the Data Acquisition Layer, ENHANCE will provide a Data Management Layer that is composed of the data integration and storage components of the platform. To address different user needs, the platform architecture will consider different databases, i.e., relational and/or non-relational. In addition, the platform will integrate in this layer a secondary deep storage component. The secondary storage component adds horizontal scalability and flexibility to the data management layer due to its high-performance, S3-compatible object storage.

Data Exchange: The platform will also support trustworthy data exchange by leveraging on standard-compliant connectors (in line with Data Space initiatives) to gather, standardize and make data available to application providers and external stakeholders. The platform will follow an open modular approach, enabling integration with external systems through appropriate RESTful APIs. In this line, ENHANCE will offer where applicable client libraries to facilitate the integration with the open APIs and allow external developers to leverage platforms' capabilities. Finally, through Data Discovery, the platform allows the querying of ENHANCE data catalogues through a common interface.

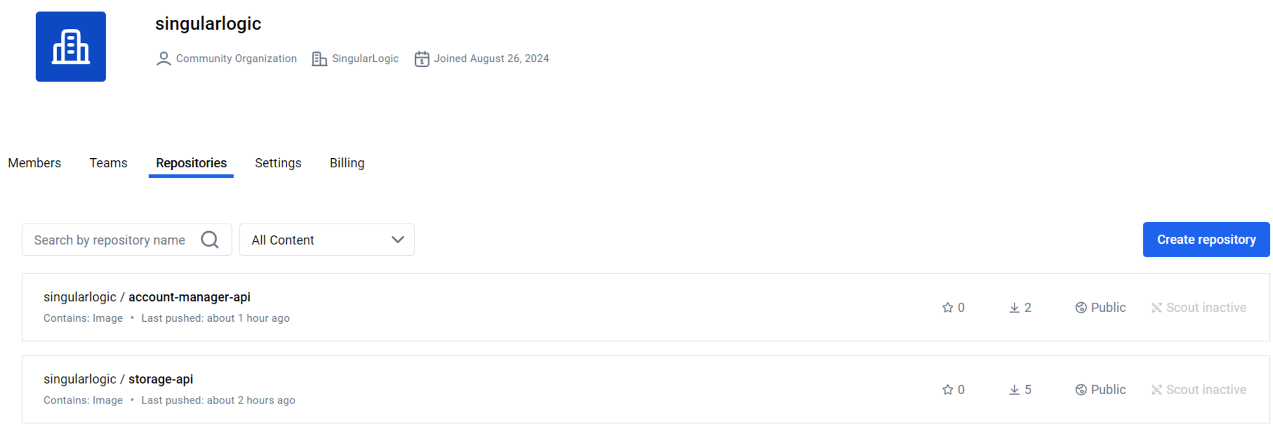
Identity Management: In addition, the ENHANCE platform will provide horizontal services related to User Management, Access Control and Data Discovery services. In this line, ENHANCE will provide the following mechanisms: 1) Identity management: component for user authentication that would enable users to use data and services based on predefined assigned roles; 2) Access Control: component responsible to check if the user has the rights to access/process/write specific data.

Data Discovery & AI: The platform will integrate components for the processing and fusion of data, along with appropriate AI-based prediction models.

Data Visualization & Services: At the application level, the platform will deliver services that will allow the visualization of data through dashboards and geo-spatial views. By relying on the underlying data and AI analytics, the platform will deliver decision support services, enabling simulation and analysis of future scenarios.

3. Runtime Environment

The ENHANCE Platform adopts Docker as the core containerization technology, which facilitates the creation of lightweight, self-contained software units encapsulating code and its dependencies. This approach ensures that microservices can run consistently across various computing environments, from development to production. Docker containers are highly portable, making it easy to deploy them on virtual machines, cloud platforms, and personal computers, ensuring maximum flexibility for stakeholders. Docker Desktop provides a user-friendly interface for managing containers, while the Docker Engine handles the execution of containers, supporting a wide range of operating systems including Windows, Linux, and macOS.



singularlogic
Community Organization · SingularLogic · Joined August 26, 2024

singularlogic / account-manager-api
Contains: Image • Last pushed: about 1 hour ago
0 · 2 · Public · Scout inactive

singularlogic / storage-api
Contains: Image • Last pushed: about 2 hours ago
0 · 5 · Public · Scout inactive

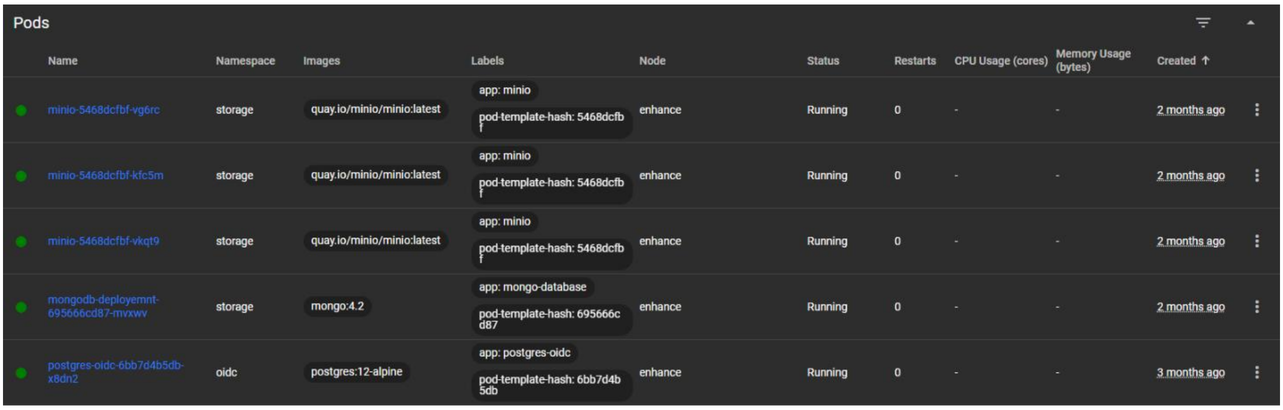
Figure 2: Docker repositories related to the ENHANCE Exchange Platform

3.1 Container orchestration through Kubernetes

To manage the complexity of deploying and scaling multiple containers, the platform uses Kubernetes for container orchestration. Kubernetes automates the deployment, scaling, and operation of containerized applications, ensuring high availability and efficient resource utilization. The choice of Kubernetes is driven by its cloud-agnostic nature, allowing the platform to be hosted on various cloud services without vendor lock-in. Kubernetes orchestrates the deployment of containers using YAML configuration files, defining the desired state of the system. This setup includes key components such as Deployments, Pods, Replica Sets, Services, and Ingresses, each playing a crucial role in maintaining the platform's operational integrity.

3.1.1 Platform Pods

In Kubernetes, a pod is the smallest deployable unit and can contain one or more containers that share storage and network resources. Pods are designed to support close collaboration between their containers, facilitating efficient communication and resource sharing. The current ENHANCE Kubernetes cluster comprises 19 pods (a screenshot presenting 5 of them is depicted in Figure 3), each encapsulating microservices critical to the platform's functionality. This modular approach ensures that the platform can scale horizontally, adding more pods to handle increased load without affecting performance.



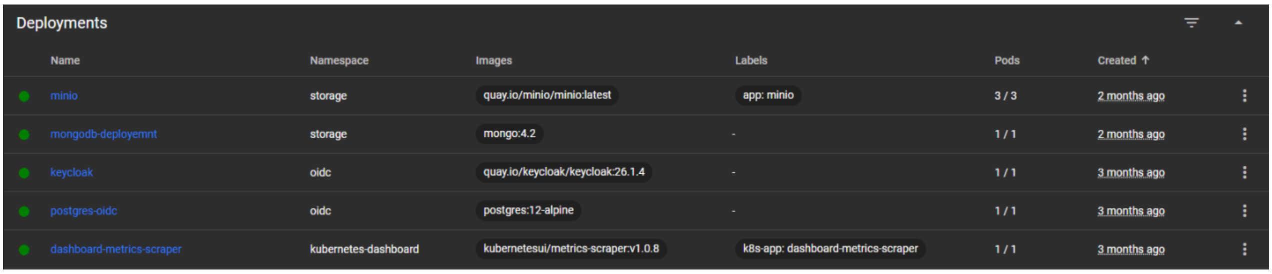
Pods

Name	Namespace	Images	Labels	Node	Status	Restarts	CPU Usage (cores)	Memory Usage (bytes)	Created ↑
minio-5468dcfbf-vg6rc	storage	quay.io/minio/minio:latest	app: minio pod-template-hash: 5468dcfbf	enhance	Running	0	-	-	2 months ago
minio-5468dcfbf-kfc5m	storage	quay.io/minio/minio:latest	app: minio pod-template-hash: 5468dcfbf	enhance	Running	0	-	-	2 months ago
minio-5468dcfbf-vkqt9	storage	quay.io/minio/minio:latest	app: minio pod-template-hash: 5468dcfbf	enhance	Running	0	-	-	2 months ago
mongodb-deployemnt-695666cd87-mvxwv	storage	mongo:4.2	app: mongo-database pod-template-hash: 695666cd87	enhance	Running	0	-	-	2 months ago
postgres-oidc-6bb7d4b5db-x8dn2	oidc	postgres:12-alpine	app: postgres-oidc pod-template-hash: 6bb7d4b5db	enhance	Running	0	-	-	3 months ago

Figure 3: Screenshot displaying 5 pods of the ENHANCE Exchange Platform

3.1.2 Platform Deployments

Deployments are higher-level abstractions in Kubernetes that manage the lifecycle of pods, including scaling, updating, and rolling back applications. A deployment specifies the desired state of the application, such as the number of pod replicas that should be running. Kubernetes ensures that the actual state matches the desired state, automatically replacing failed pods and updating configurations without downtime. The ENHANCE cluster includes 8 deployments (a screenshot presenting 5 of them is shown in Figure 4), each managing different microservices, ensuring seamless updates and high availability. Table A1, in Appendix, presents the YAML file declaring the exchange platform's deployment in the cluster.



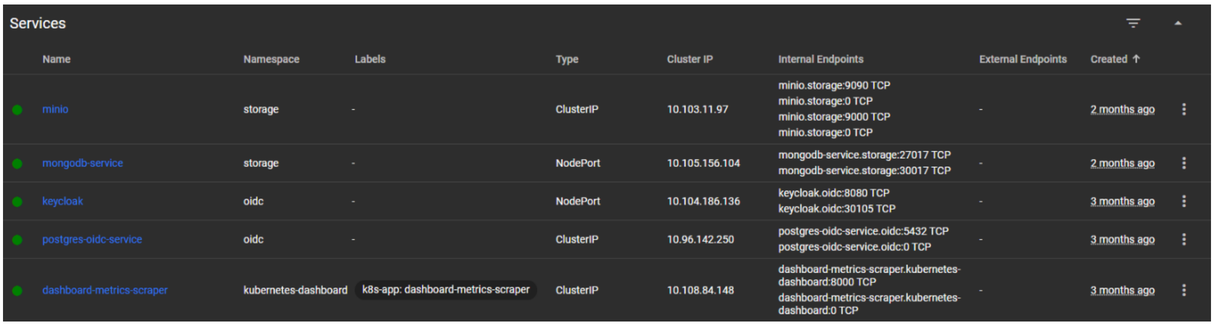
Deployments

Name	Namespace	Images	Labels	Pods	Created ↑
minio	storage	quay.io/minio/minio:latest	app: minio	3 / 3	2 months ago
mongodb-deployemnt	storage	mongo:4.2	-	1 / 1	2 months ago
keycloak	oidc	quay.io/keycloak/keycloak:26.1.4	-	1 / 1	3 months ago
postgres-oidc	oidc	postgres:12-alpine	-	1 / 1	3 months ago
dashboard-metrics-scraper	kubernetes-dashboard	kubernetesui/metrics-scraper:v1.0.8	k8s-app: dashboard-metrics-scraper	1 / 1	3 months ago

Figure 4: Screenshot displaying 5 deployments of the ENHANCE Exchange Platform

3.1.3 Exchange Platform Services

Kubernetes services provide stable endpoints to access pods, abstracting the underlying complexity of pod management. Services enable load balancing and facilitate communication between different parts of the application or with external clients. The platform currently operates 10 services (a screenshot presenting 5 of them is shown in Figure 5), each corresponding to a specific set of pods. These services ensure that applications remain accessible and performant, distributing traffic evenly and managing network routing efficiently. Table A2, in Appendix, presented the YAML file declaring the exchange platform's service in the cluster.



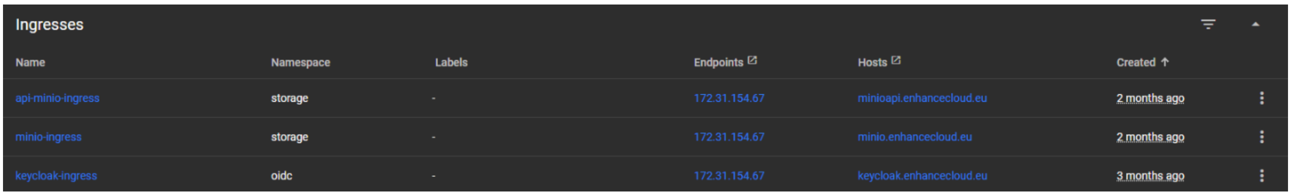
Services

	Name	Namespace	Labels	Type	Cluster IP	Internal Endpoints	External Endpoints	Created ↑
●	minio	storage	-	ClusterIP	10.103.11.97	minio.storage:9090 TCP minio.storage:0 TCP minio.storage:9000 TCP minio.storage:0 TCP	-	2 months ago
●	mongodb-service	storage	-	NodePort	10.105.156.104	mongodb-service.storage:27017 TCP mongodb-service.storage:30017 TCP	-	2 months ago
●	keycloak	oidc	-	NodePort	10.104.186.136	keycloak.oidc:8080 TCP keycloak.oidc:30105 TCP	-	3 months ago
●	postgres-oidc-service	oidc	-	ClusterIP	10.96.142.250	postgres-oidc-service.oidc:5432 TCP postgres-oidc-service.oidc:0 TCP	-	3 months ago
●	dashboard-metrics-scraper	kubernetes-dashboard	k8s-app: dashboard-metrics-scraper	ClusterIP	10.108.84.148	dashboard-metrics-scraper.kubernetes-dashboard:8000 TCP dashboard-metrics-scraper.kubernetes-dashboard:0 TCP	-	3 months ago

Figure 5: Screenshot displaying 5 services of the ENHANCE Exchange Platform

3.1.4 Exchange Platform Ingresses

Ingresses in Kubernetes manage external access to services within the cluster, defining rules for routing HTTP and HTTPS traffic to specific services based on domain names and paths. Ingress controllers provide advanced load balancing, SSL termination, and name-based virtual hosting. The ENHANCE platform employs 5 ingresses (a screenshot presenting 3 of them is shown in Figure 7), which handle incoming traffic, directing it to the appropriate services based on predefined rules. This setup enhances the platform's security and accessibility, providing a robust mechanism for managing external requests. Table A3, in Appendix, presents the YAML file declaring the core platform's ingress in the cluster.



Ingresses

Name	Namespace	Labels	Endpoints	Hosts	Created ↑
api-minio-ingress	storage	-	172.31.154.67	minioapi.enhancecloud.eu	2 months ago
minio-ingress	storage	-	172.31.154.67	minio.enhancecloud.eu	2 months ago
keycloak-ingress	oidc	-	172.31.154.67	keycloak.enhancecloud.eu	3 months ago

Figure 6: Screenshot displaying the ingresses of the ENHANCE Exchange Platform

4. ENHANCE Open One Health Core Platform

The ENHANCE data Exchange Platform is designed to enhance the management and exchange of data related to coastal areas. It employs a streamlined architecture that ensures robust security, efficient data management, and user-friendly interactions. The platform integrates advanced technologies such as Docker and Kubernetes for container orchestration and scalability.

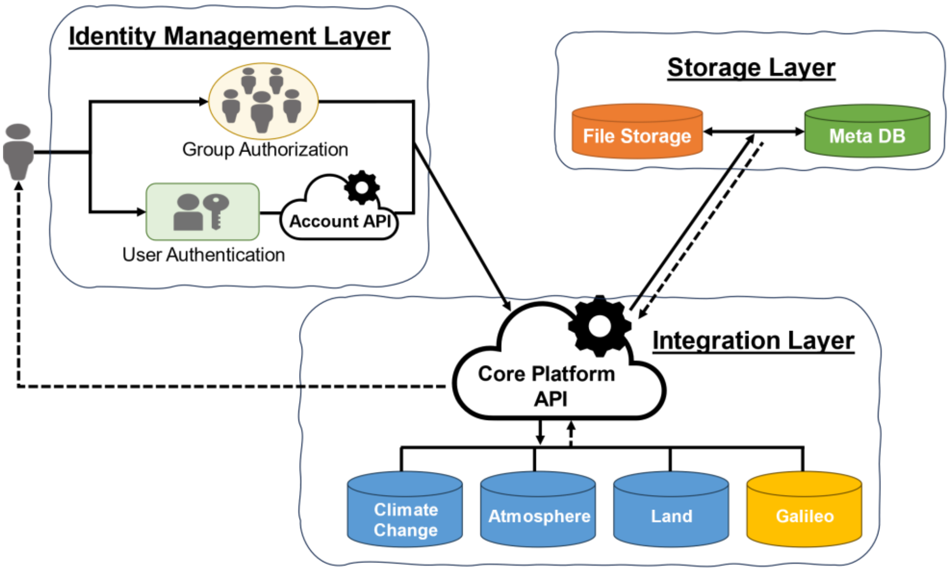


Figure 7: In-depth visualisation of the Exchange Platform's architecture

The primary objective is to provide a high-fidelity software solution that promotes trust in data storage, coupled with advanced access and sharing mechanisms. To achieve this, the Exchange Platform presents a REST API that serves as an integrator between services and data, implementing also logic for managing files, folders, and sharing mechanisms, ensuring a cohesive and sophisticated user experience. The Exchange Platform follows a three-tier architecture with the following layers: i) identity management (part of the Interconnection block 1); ii) storage; and iii) integration.

4.1 Identity Management Layer

The ENHANCE Core Platform adopts a modular and extensible identity management architecture based on OpenID Connect (OIDC) and Keycloak, designed to support both the ENHANCE ecosystem and its integration with external platforms such as MINKA. OIDC provides a secure, standardized authentication layer on top of OAuth 2.0, enabling users and services to authenticate and obtain short-lived access tokens used to protect all ENHANCE applications.

Within ENHANCE, Keycloak acts as the authoritative Identity Provider (IdP) for the Core Platform. It issues and validates OAuth2/OIDC tokens consumed by the different services and pipelines. Authorization is enforced through hierarchical group-based roles (admin, editor, viewer), allowing fine-grained access control aligned with organizational structures, pilots, and service scopes. This model ensures that users and services only access data and functionality explicitly permitted by their role and group membership, as well as supporting a GDPR-aligned, least-privilege access control approach.

To support programmatic identity operations and cross-platform integration, the platform introduces a dedicated Identity Management API, with a Swagger documentation avaliable at https://account.enhancecloud.eu/swagger/, implemented in Python as a controlled abstraction layer over the Keycloak Admin REST API. This Management API is responsible for non-interactive identity operations highly correlated with GDPR compliance such as:

- Token introspection/validation for protected backend operations.
- Organization and group lifecycle management.
- User and service account provisioning (creation/linking, status management).
- Role and group membership assignment and revocation.
- Client credential management for service-to-service access.
- Consent and policy state management when required by project governance.

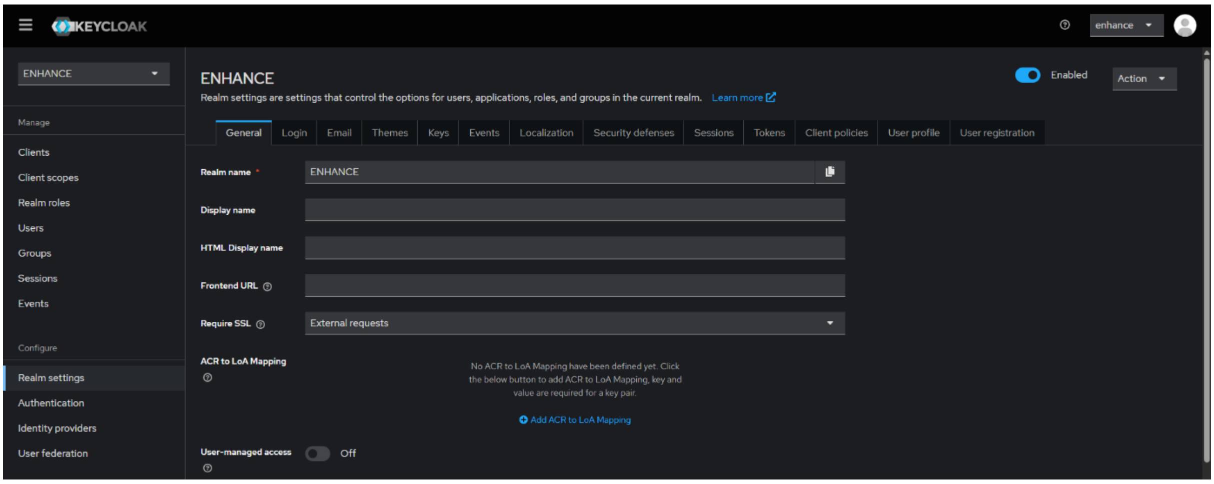
This Management API is designed to be reusable across deployments. In practice, both ENHANCE and MINKA will operate their own Keycloak IdP instances, each paired with an equivalent Management API layer, following the same architectural pattern. This ensures consistency in how identities, roles, and permissions are handled across platforms, while allowing each ecosystem to retain autonomy over its users and governance rules.

Integration between MINKA and ENHANCE is implemented through OIDC-based federation between the two Keycloak deployments where ENHANCE Keycloak is configured to delegate user authentication to MINKA Keycloak as an external Identity Provider, enabling Single Sign-On (SSO) without merging user directories. Users authenticate with their existing MINKA accounts, and after successful login ENHANCE issues access tokens that are used to authorize requests against ENHANCE services. Service-to-service interactions follow the same model using confidential clients and scoped roles, ensuring that all access to ENHANCE resources remains governed by ENHANCE-issued tokens and auditable authorization policies. The Management API layer plays a key role in mapping identities, synchronizing relevant group or role information where needed, and enabling controlled data and service exchange between the two systems.

This architecture directly supports several core use cases of the project:

- When citizen observations are submitted through MINKA and trigger AI-based processing (e.g. species identification), ENHANCE AI services authenticate using service credentials managed via the ENHANCE IdP and Management API.
- Access to observation data, execution of AI models, and publication of derived results are all protected through the same identity and authorization mechanisms.
- Data exchanged between MINKA and ENHANCE (validated observations, AI annotations, derived indicators, ...) governed by role-based access control and auditable identity actions.

Overall, the Identity Management Layer provides a secure, auditable, and interoperable foundation that enables ENHANCE to operate as a standalone platform while seamlessly integrating with MINKA and other external services. By combining standardized authentication (OIDC), centralized authorization (Keycloak), and a reusable Management API layer, the platform ensures consistent enforcement of access policies across users, services, and AI workflows, supporting both current pilot needs and future extensions.



ENHANCE

Realm settings are settings that control the options for users, applications, roles, and groups in the current realm. Learn more

Enabled

Realm name * ENHANCE

Display name

HTML Display name

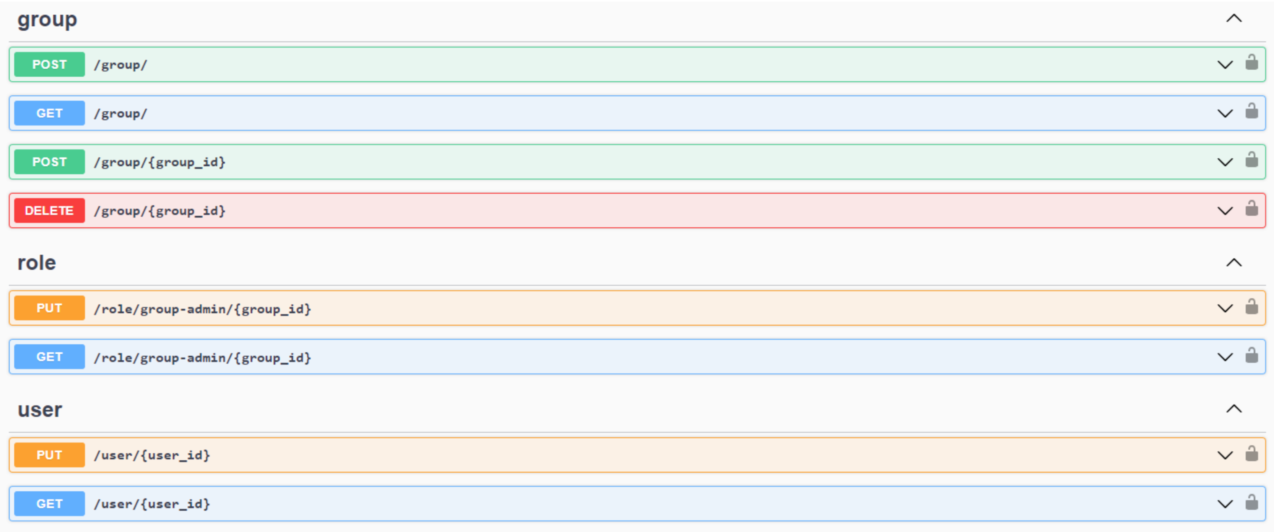
Frontend URL

Require SSL: External requests

ACR to LoA Mapping: No ACR to LoA Mapping have been defined yet. Click the below button to add ACR to LoA Mapping, key and value are required for a key pair.

User-managed access: Off

Figure 8: Implementation of Keycloak in the ENHANCE ecosystem



group

POST /group/

GET /group/

POST /group/{group_id}

DELETE /group/{group_id}

role

PUT /role/group-admin/{group_id}

GET /role/group-admin/{group_id}

user

PUT /user/{user_id}

GET /user/{user_id}

Figure 9: Swagger of the Accounts REST API

4.2 Storage Layer

The Storage Layer utilizes MinIO for object storage and MongoDB for metadata management (Figure 11). MinIO is an S3-compatible file system known for its high performance and scalability. It employs parallelization techniques and horizontal scalability by adding new nodes to the cluster. MinIO ensures data security through encryption in transit and at rest, supporting secure communication protocols. Files are stored in multiple parts to enhance confidentiality and security, aligning with the platform's commitment to data privacy. MongoDB, a NoSQL DB, stores metadata associated with these files. Its schema-less design allows for flexibility in handling diverse metadata structures, adapting to the platform's evolving needs. MongoDB's advanced query capabilities and organization-centric approach empower users to perform detailed searches based on metadata attributes. This integrated approach between MinIO and MongoDB ensures a synchronized workflow, enhancing data management efficiency and reliability.

4.3 Integration Layer

The Integration Layer is powered by a REST API developed in Golang (v1.20.3), which serves as the backbone of the ENHANCE Exchange Platform. This API manages core functionalities such as uploading, downloading, and deleting files, creating MinIO buckets, establishing folders for organized content, and executing operations like folder creation, deletion, and sharing. The API ensures seamless communication between the Identity Management and Storage layers, facilitating secure and efficient data handling. It also adheres to OpenAPI standards, providing a standardized interface for data retrieval and manipulation. This ensures that various services can efficiently consume and contribute to the shared data ecosystem. Strict data models, designed and implemented to enhance interoperability, establish a common language for data exchange, ensuring consistency and reducing ambiguity. Future iterations of the Core API will integrate additional services such as Copernicus APIs related to Land, Climate Change, and Atmosphere services, creating a unified and comprehensive data repository. The integration of the EGNSS Galileo service is also planned, enhancing the platform's geospatial and environmental data capabilities. The swagger documentation of the Core API is available at https://api.enhancecloud.eu/swagger and presented below.

Buckets

POST `/bucket` Create bucket.
DELETE `/bucket/{id}` Delete bucket with all contents.

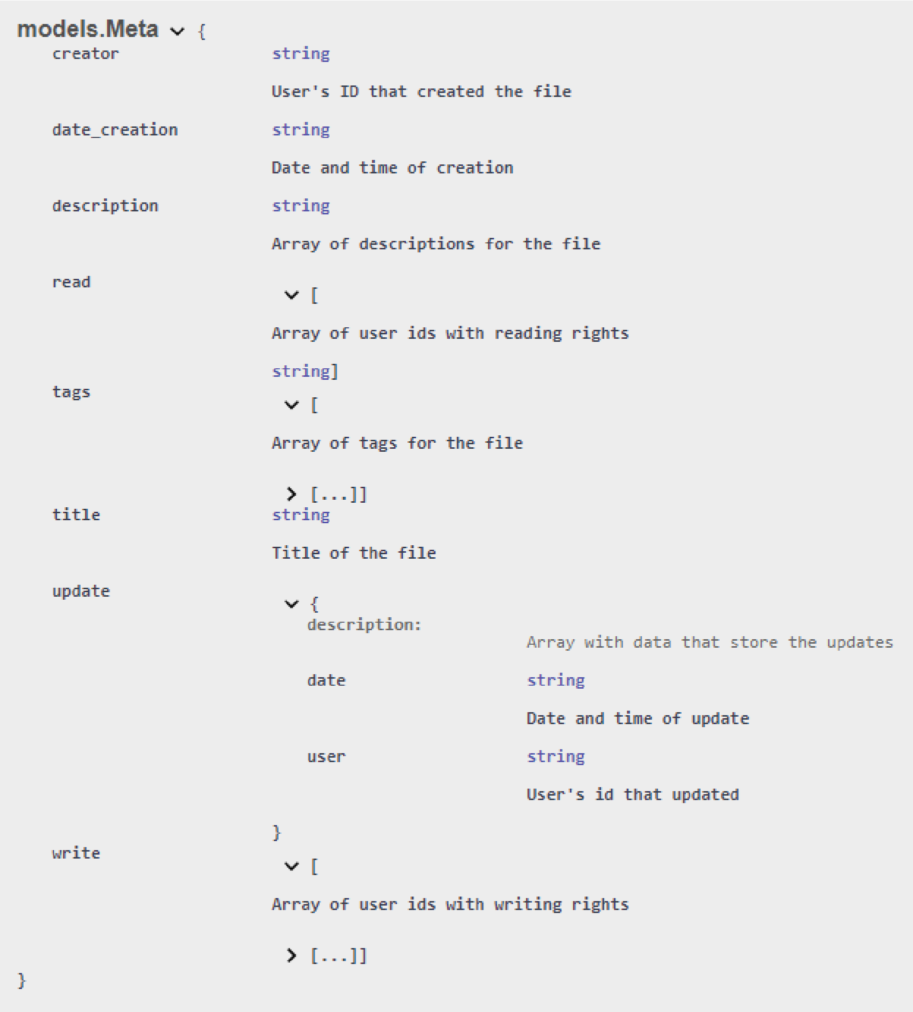
Files

PUT `/file` Update a file.
POST `/file` Upload a file.
GET `/file/{id}` Download a file.
DELETE `/file/{id}` Delete file by ID.
GET `/info/file` Get metadata of file.

Folders

GET `/folder` Get folder by id.
PUT `/folder` Update folder by ID.
POST `/folder` Create a new folder.
GET `/folder/list` List folder's items.
DELETE `/folder/{id}` Delete folder by id.

Figure 10: Swagger of the Exchange Platform's REST API



```
models.Meta ⌄ {
   creator                  string
                            User's ID that created the file

   date_creation            string
                            Date and time of creation

   description              string
                            Array of descriptions for the file

   read                      ⌄ [
                            Array of user ids with reading rights

                            string]
   tags                      ⌄ [
                            Array of tags for the file

                             > [...]]
   title                    string
                            Title of the file

   update                    ⌄ {
                               description:
                                                      Array with data that store the updates

                               date                   string
                                                      Date and time of update

                               user                   string
                                                      User's id that updated

                            }
   write                     ⌄ [
                            Array of user ids with writing rights

                             > [...]]
}
```

Figure 11: Example of the data model of the metadata stored in MongoDB

5. Pressures Characterization and Quantification

The One Health concept of the World Health Organization (WHO) addresses emerging social and environmental challenges through a holistic lens, integrating human, animal, and environmental health. When applied within the driver-pressure-state-impact-response (DPSIR) management framework it enables the identification of pressures and impacts and guides administrations and companies in decision-making to optimize the health of this triad.

On of the most threatening issues worldwide for this triad is eutrophication. According to Ferreira et al. (2011), eutrophication is “a process driven by the enrichment of water by nutrients, especially compounds of nitrogen and/or phosphorus, leading to increased growth, primary production and biomass of algae; changes in the balance of organisms; and water quality degradation. The consequences of eutrophication are undesirable if they appreciably degrade ecosystem health and/or the sustainable provision of goods and services”. Several policies have been enacted in Europe with the aim of restoring and protecting waters, such as the Water Framework Directive (WFD; 2000/60/EC) and the Marine Strategy Framework Directive (MSFD; 2000/60/EC). Both directives are based on the DPSIR framework, which allows for the identification of environmental problems and their causes. This link permits afterwards the proposal of management plans, which generally follow the Ecosystem Approach and aim for environmental sustainability, social well-being, and economic prosperity.

Eutrophication assessments rely on chlorophyll-a concentration (Chl-a) data from water bodies. Traditionally, these assessments are conducted using in-situ Chl-a measurements, typically expressed as mean or 90thpercentile values. However, ocean colour satellites, such as Sentinel-3 from the European Space Agency's (ESA) Copernicus Programme, have enhanced these assessments by providing Chl-a maps with higher spatial and temporal resolutions (300 m) compared to in-situ data. These maps have improved assessment capabilities, particularly for open marine waters. Coastal waters, however, present greater variability and require even higher spatial resolution. Mediterranean coastal waters, revealed not oligotrophic as Mediterranean open sea (Flo, 2017), are specially threatened by cultural (i.e., anthropogenic) eutrophication due to, among others, its intense human activities, high population densities, and low tidal range (20–40 cm), which limits the dilution of nutrient-rich continental inflows. Recent advancements, such as the Sentinel-2satellite equipped with the MultiSpectral Instrument (MSI) sensor, now enable ocean colour mapping with very high spatial resolution (10–60 m). This advancement allows for more precise eutrophication studies and assessments in microtidal coastal waters.

Within the ENHANCE project we aim to provide information on pressures (section 5) and impacts (section 6) regarding the eutrophication process based on, among others, remote sensing data from the Copernicus program of the ESA.

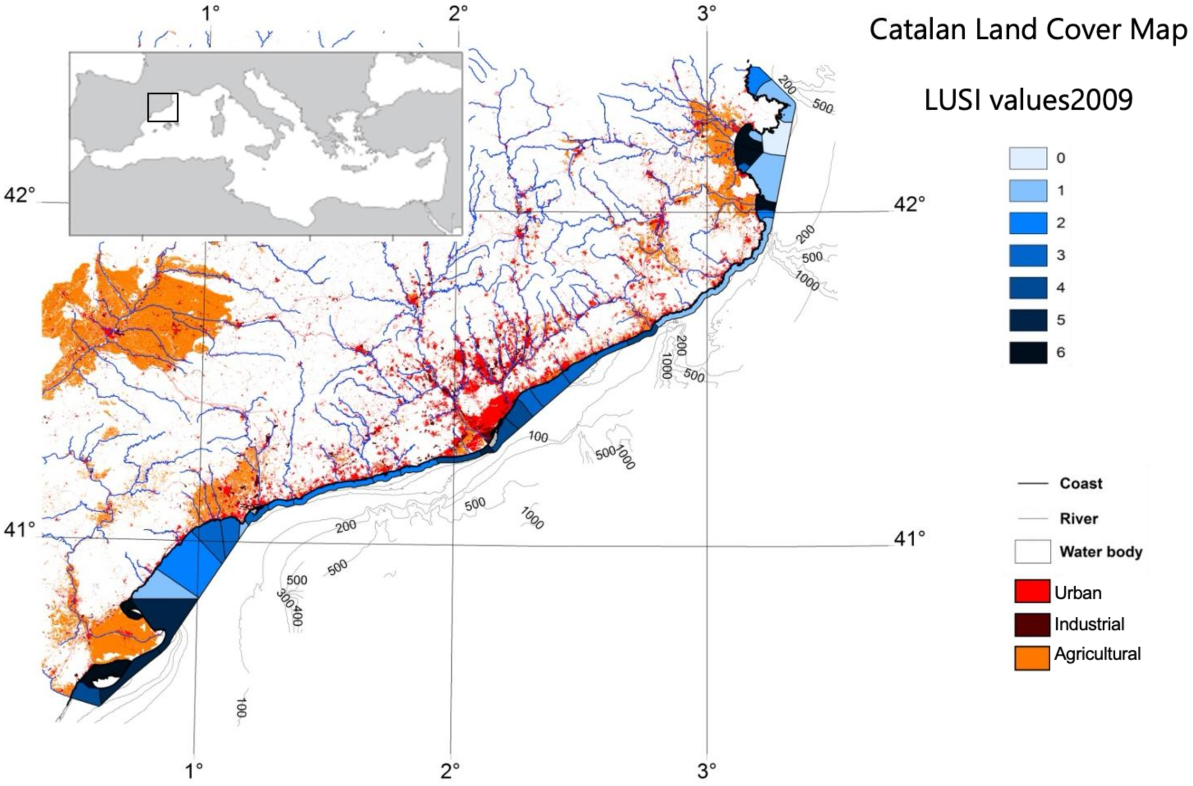
5.1. Products

Two products will be delivered in relation to pressures and in both cases related to the Environmental Health Indicator of the OH framework.

5.1.1. Risk of eutrophication of coastal waters measured in land

The first is a map of the risk of eutrophication of coastal waters measured in land. It will be based on the D-LUSI, which needs information of three kinds. First, information on Land Useand Land Cover (LULC), which will be obtained from the Copernicus Land Monitoring Service webpage (https://land.copernicus.eu/en/products/corine-land-cover). From there the European Coordination of Information on the Environment (CORINE) Land Cover will be downloaded. Second, this map will serve also as to gather information on the coastline morphology. Thirds, information on fluvial inflows, which will be obtained by local bodies. In relation to case study 1, for example, it will be obtained by the Catalan Institute of Statistics

(IDESCAT; www.idescat.cat). An example of the results of LUSI, which is the previous version of D-LUSI, is described in Figure 12. LUSI values provide a semi-quantitative assessment of continental pressures on the coastal waters of the studied area. They have no units and range from 0.75 to 8.75. A low LUSI value indicates that coastal waters are not or only slightly influenced by continental pressures and/or that these pressures are diluted. On the contrary, a high LUSI value indicates that coastal waters are strongly influenced by continental pressures and/or that these pressures are not diluted.



.

Figure 12: Example of results of LUSI for the Catalan coast in 2009 obtained by the Catalan Land cover Map. The location of the Catalan Coast in the Mediterranean Sea is indicated by a square in the top left map.

5.1.2. Risk of eutrophication on coastal waters

The second is a map of the risk of eutrophication on coastal waters assessed directly on coastal waters, thus, one step beyond D-LUSI maps. In this case the map will be a mosaic of several images in a specific time period with information of concentration of Chl-*a*. To be able to provide this product, first, ocean colour images of high spatial resolution obtained by Sentinel-2 will be downloaded from the Copernicus Data Space Ecosystem (https://dataspace.copernicus.eu/browser) of the ESA. Second, these images will be processed with the repository ACOLITE (https://github.com/acolite) developed by the Royal Belgian Institute of Natural Sciences (https://www.naturalsciences.be). Third, the single images will be combined in a mosaic. The result of a test performed in the Catalan Cost and similar to what the product will be is depicted in Figure 13.

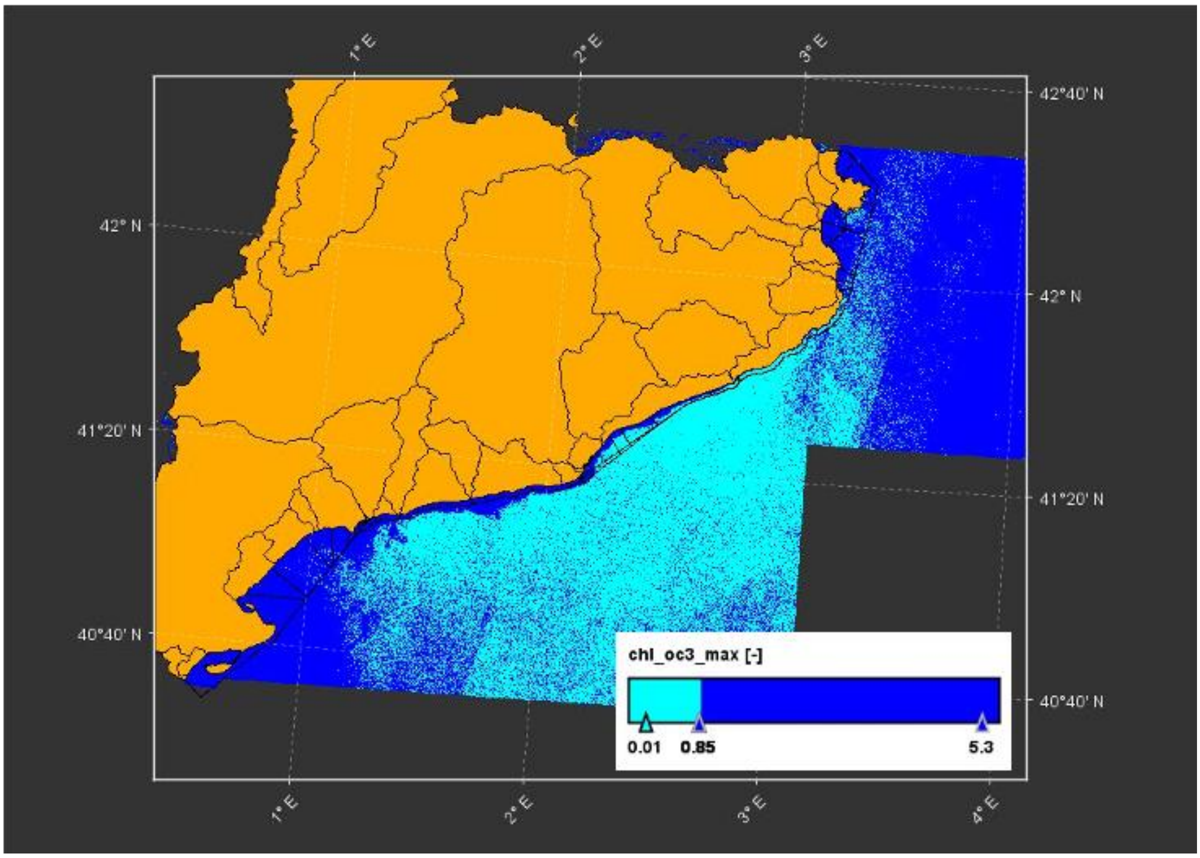


Figure 13: Mosaic of several Images of Sentinel-2 processed and where the concentration of Chl-a had been retrieved with the OC3 algorithm. A threshold of 0.85 was used to define the areas with high and low Chl-a concentration.

Both maps will be categorised, based on specific thresholds, to make the information more comprehensive to the stakeholders.

5.2. Pipelines

5.2.1. Pipeline for the assessment of the risk of eutrophication of coastal waters measured in land

The pipeline to obtain the maps of the risk of eutrophication on coastal waters from land, will be as follows:

1. **DOWNLOAD AND STORAGE PHASE**
    - Define the area of the study case
    - Define the time period to assess
    - Define where LULC maps will be stored
    - Download available Corine Land Cover Map from the Copernicus Land Monitoring Service webpage (https://land.copernicus.eu/en/products/corine-land-cover)

2. **PRESSURES CHARACTERISATION PHASE**
    - Calculate the area corresponding to urban pressures and assign a score according to Table 1.
    - Calculate the area corresponding to agricultural pressures and assign a score according to Table 1.
    - Calculate the area corresponding to industrial pressures and assign a score according to Table 1.
    - Calculate the riverine pressure from fluvial inflows and assign a score according to Table 1.

Urban	Agricultural (irrigated)	Industrial	Riverine	Score
	<10%	<10%	Low	0
<33%	40%	>10%	Medium	1
66 %	>40%		High	2
>66%				3

Table 1: Scores and categories of land pressures to be considered in D-LUSI.

An example of a map where only the land pressures considered to calculate D-LUSI are shown in Figure 14.

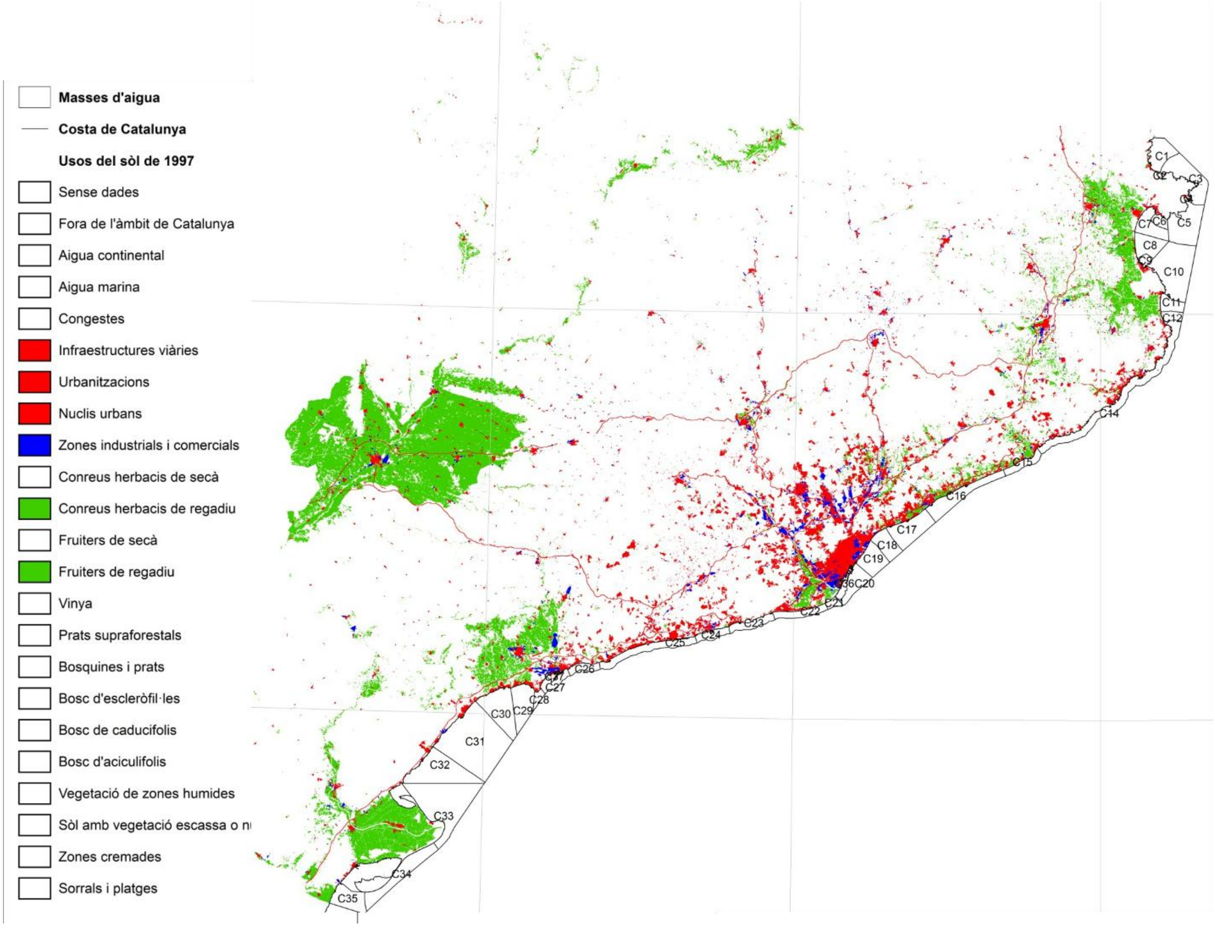


Figure 14: Catalonia map where only land pressures considered to calculate the D-LUSi are shown. Catalan Coastal Water Bodies of the Water Framework Directive (WFD) are indicated.

3. COASTLINE MOROHOLOGY CHARACTERISATION PHASE

- Define the coastline morphology and assign a correction number according to Table 2

Confinement	Correction number
Concave	1.25
Convex	0.75
Straight	1.00

Table 2: Correction numbers according to coastline morphology

4. CALCULATE D-LUSI VALUE

- Summing scores
- Dividing by the correction number

- Maps creation

5.2.2. Pipeline for the assessment of the risk of eutrophication on coastal waters

This product is based on the **Mission Sentinel-2** information from the Copernicus Program of the ESA.

Figure 15: Illustrative image of the Sentienel-2 satellites.

This mission is based on a constellation of three identical satellites (Figure 15) in the same orbit (180º), for optimal coverage and data delivery. Each satellite carries an innovative wide swath high-resolution MSI(Figure 16 with 13 spectral bands (Figure 17). The combination of high resolution, novel spectral capabilities, a swath width of 290 km and frequent revisit times provides unprecedented views of Earth. Sentinel-2A was launched on 23 June 2015, followed by Sentinel-2B on 7 March 2017. On 5 September 2024, Sentinel-2C was launched into orbit to join its siblings and ensure the continuous provision of high-resolution data from the mission.

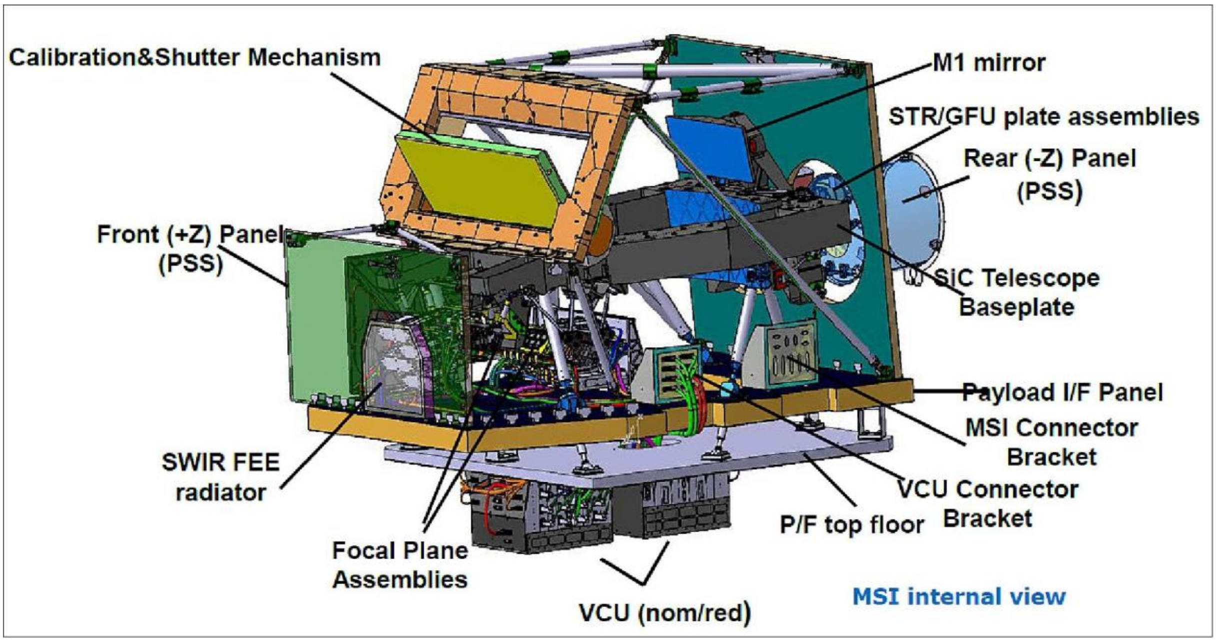


Figure 16: MSI architecture (credit: ESA)

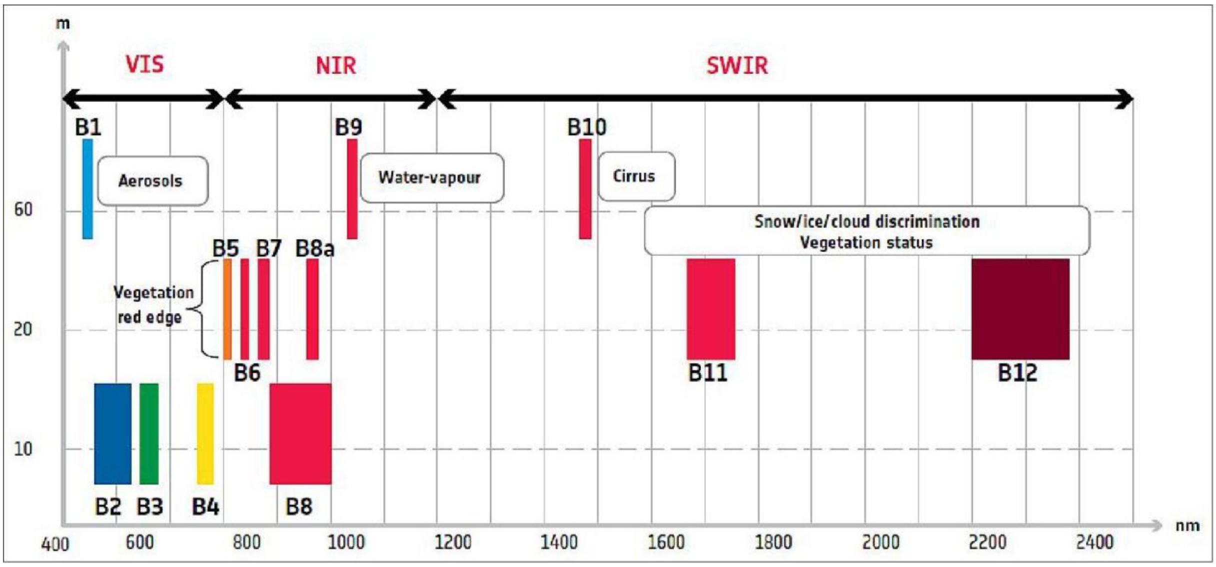


Figure 17: Sentinel-2's 13 spectral bands, from visible to shortwave infrared, offer spatial resolutions of 10–60 m, enabling high-precision land monitoring (credit: ESA).

The pipeline to obtain the **maps of the risk of eutrophication** on coastal waters assessed directly on coastal waters will be as follows:

1. **DOWNLOAD AND STORAGE PHASE**
    a. Define the area of the study case
    b. Define the time period to assess
    c. Define where imagery will be stored
    d. Download available images of Sentinel-2 from the Copernicus Data Space Ecosystem (https://dataspace.copernicus.eu/browser) of the ESA with the following characteristics (Figure 18)
        - 100% Cloud coverage
        - L1C
        - All platforms
        - Immediate availability

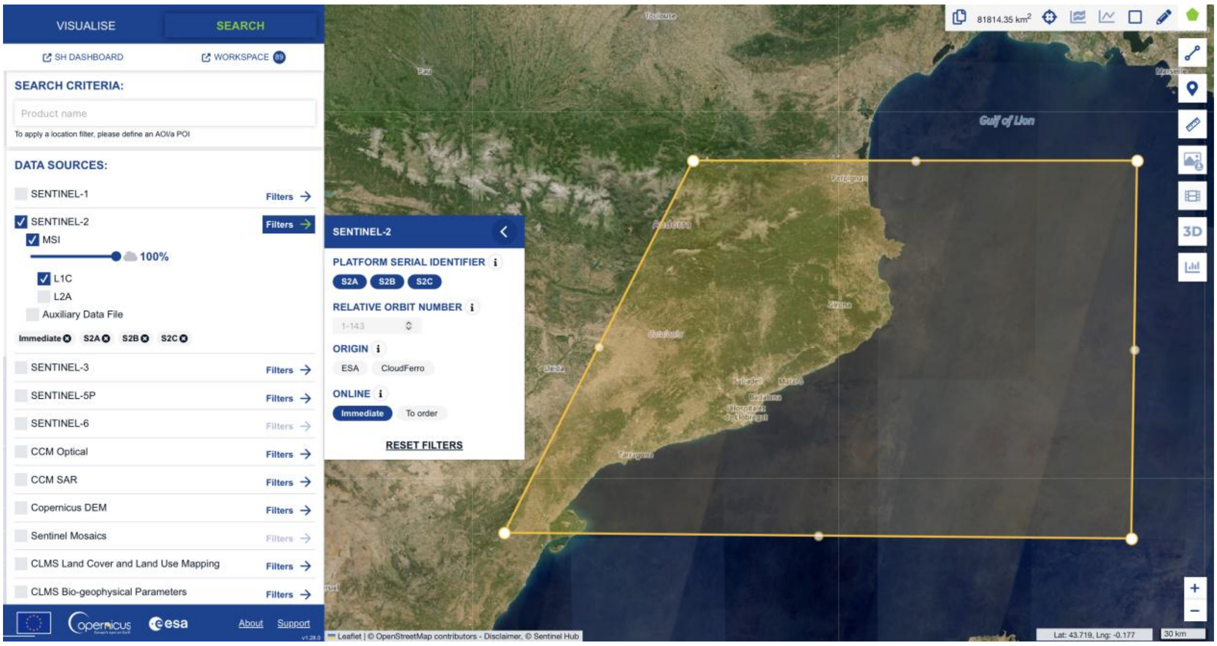


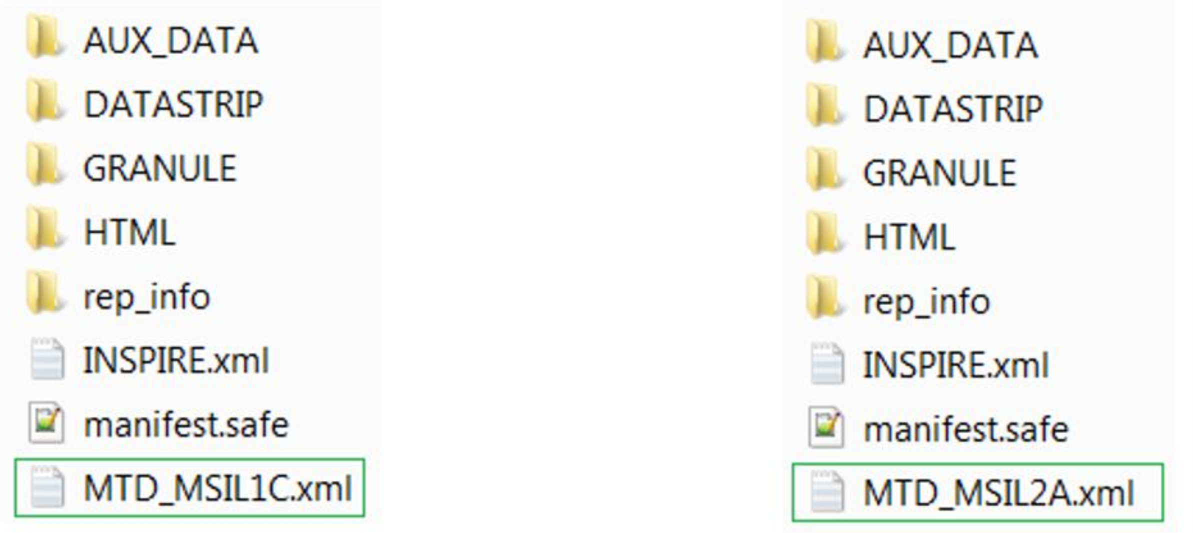
Figure 18: Capture of the Copernicus Data Space Browser

2. **PROCESSING PHASE**
    a. Define where the images will be processed
    b. Set up the workstation or supercomputer
        i. Download and Instal CONDA from www.anaconda.com
        ii. Download and install ACOLITE repository from GitHub (https://github.com/acolite)
        iii. Create an environment for ACOLITE
        iv. Activate the ACOLITE environment

    v. Install required packages
    vi. Create the necessary folders (usually in and out)

c. Set up imagery

    i. Select images to be processed
    ii. Unzip them to *.SAFE and check the format. Should be as depicted in Figure 19. These files could be checked by using specialized software as the Sentinel Application Platform (SNAP) from the ESA (https://step.esa.int/main/toolboxes/snap/) or directly in python.



AUX_DATA
DATASTRIP
GRANULE
HTML
rep_info
INSPIRE.xml
manifest.safe
MTD_MSIL1C.xml

AUX_DATA
DATASTRIP
GRANULE
HTML
rep_info
INSPIRE.xml
manifest.safe
MTD_MSIL2A.xml

*Figure 19: Capture of the files inside the *.SAFE file of images of Sentinel-2. In green, files that can be opened in specialized software as SNAP from ESA.*

d. Create a settings_file.txt with a text editor using a UTF-8 encoding. Several parameters could be specified. However, three are mandatory:

    i. Input file
    ii. Output file
    iii. Output resolution (for Sentinel-2 it could be 60, 20 and 10m)

If nothing else is included, two results are obtained:

- rhot_*: ρt, the top-of-atmosphere reflectance as derived from the original input file (L1R)
- rhos_*: ρs, the surface reflectance (L2R)

Whoever to process Sentinel-2 to achieve our goals we need to specify in adition the next parameters:

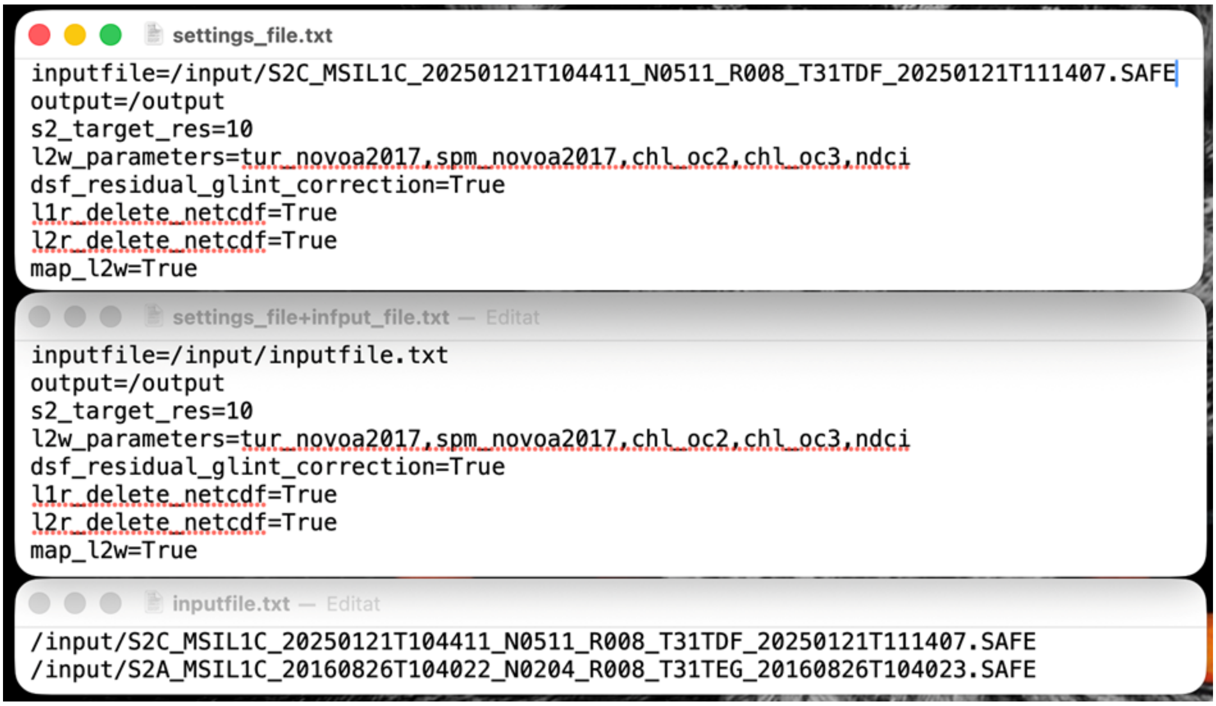
    iv. Output parameters

For Case Study 1 the next retrieval algorithms will be used:

- tur_novoa2017, for Turbidity in FNU
- spm_novoa2017, for Suspended Matter concentration in gm−3
- chl_oc2, for Chl-a concentration in µg/l using 2 bands
- chl_oc3, for Chl-a concentration in µg/l using 3 bands
- ndci, for The Normalised Difference Chlorophyll Index algorithm by Mishra and Mishra (2012)

    v. Correction of sun glint
    vi. Deletion of intermediate netcdf files (to decrease storage)
    vii. Output maps (Useful at the testing phase of the processing)

An option to optimize the processment of the images it to create an **inputfile.txt** with the paths and images to be processed within the settings file instead of a single image. In the Figure 20 there is an example of a settings file ready to process a single image with all the parameters indicated following the notations of the ACOLITE manual. Besides, there are examples of a settings file with an input file included to process two images, and of an input file with the paths of the images.



```
inputfile=/input/S2C_MSIL1C_20250121T104411_N0511_R008_T31TDF_20250121T111407.SAFE
output=/output
s2_target_res=10
l2w_parameters=tur_novoa2017,spm_novoa2017,chl_oc2,chl_oc3,ndci
dsf_residual_glint_correction=True
l1r_delete_netcdf=True
l2r_delete_netcdf=True
map_l2w=True
```

```
inputfile=/input/inputfile.txt
output=/output
s2_target_res=10
l2w_parameters=tur_novoa2017,spm_novoa2017,chl_oc2,chl_oc3,ndci
dsf_residual_glint_correction=True
l1r_delete_netcdf=True
l2r_delete_netcdf=True
map_l2w=True
```

```
/input/S2C_MSIL1C_20250121T104411_N0511_R008_T31TDF_20250121T111407.SAFE
/input/S2A_MSIL1C_20160826T104022_N0204_R008_T31TEG_20160826T104023.SAFE
```

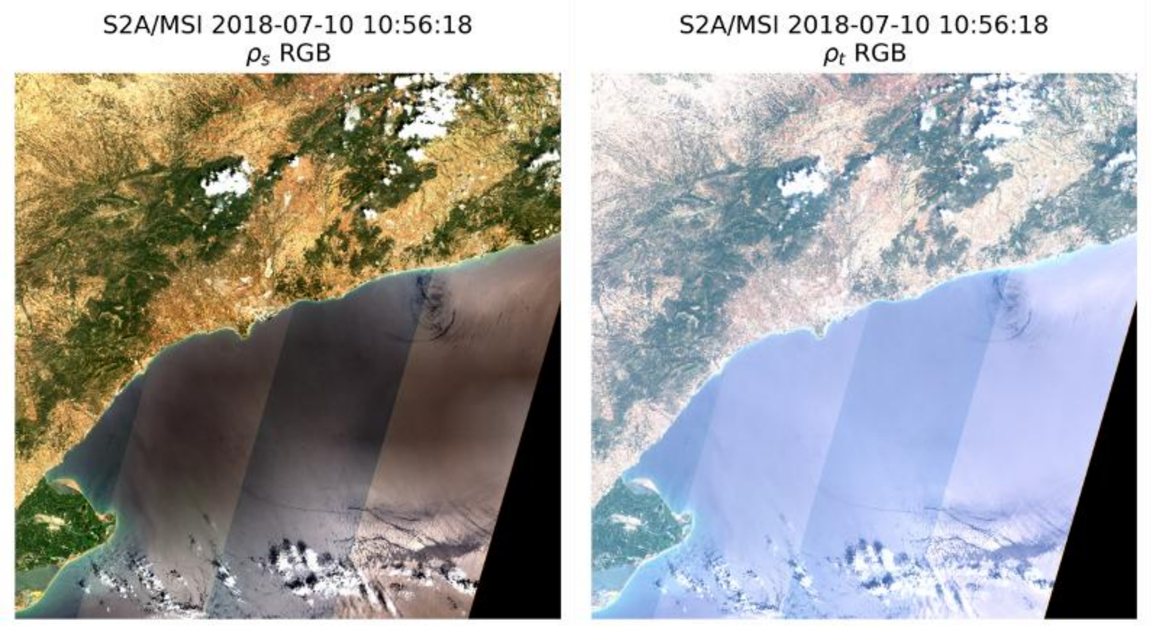
Figure 20: Example of a settings file ready to process a single image (top), a settings file with an input file included to process two images (middle), and an input file with the paths of the images (bottom).

e. Launch ACOLITE with the previous settings specified in the settings_file.txt. The process will apply the atmospheric and sun glint corrections. In addition, we will use the selected algorithms to retrieve the desired variables. The output of the process will be several files. This is described in Figure 21.

```
acolite_run_20211129_202755_l1r_settings.txt
acolite_run_20211129_202755_l2r_settings.txt
acolite_run_20211129_202755_log_file.txt
S2B_MSI_2021_11_26_10_59_36_T31TCF_L1R.nc
S2B_MSI_2021_11_26_10_59_36_T31TCF_L2R_rgb_rhos.png
S2B_MSI_2021_11_26_10_59_36_T31TCF_L2R_rgb_rhot.png
S2B_MSI_2021_11_26_10_59_36_T31TCF_L2R.nc
S2B_MSI_2021_11_26_10_59_36_T31TCF_L2W_ndci.png
S2B_MSI_2021_11_26_10_59_36_T31TCF_L2W.nc
```

Figure 21: Capture of the files resulting of processing images of Sentine-2 with ACOLITE.

Note that the L1R, which is the top of the atmosphere (TOA) image, is used as an input to process L2R, which is the bottom of the atmosphere (BOA) image; and that this last file is used as an input to process L2W, which are the images of the selected output parameters. In Figure 22 a comparison between TOA and BOA images are shown.



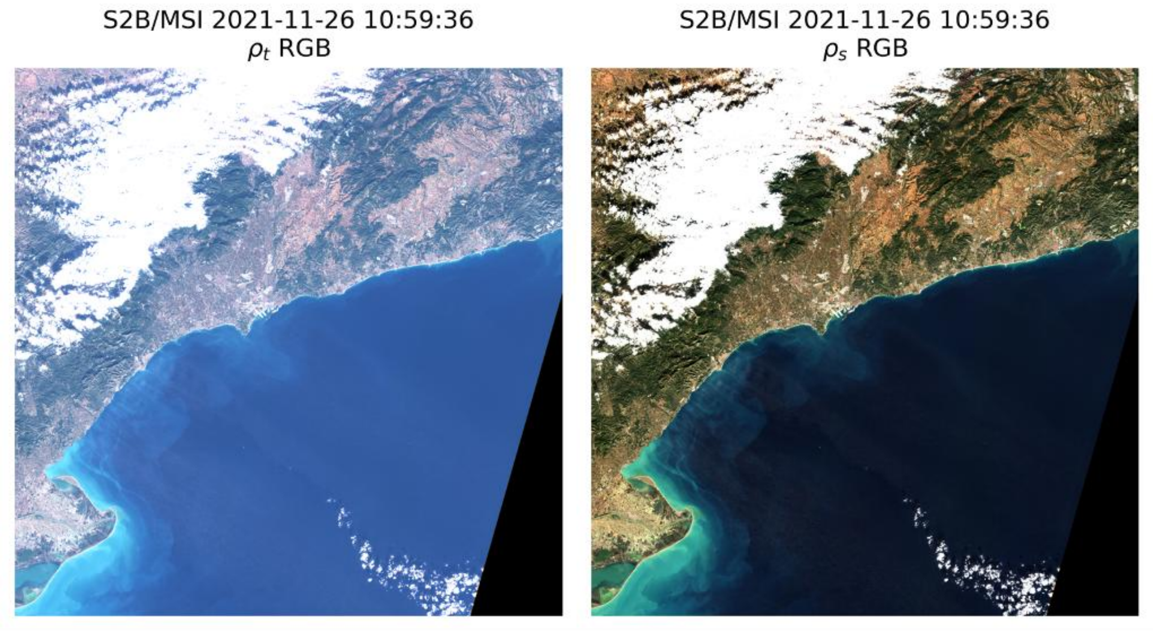


Figure 22: Comparison of four images: two from the Central (top) and SW (bottom) Catalan coast, showing Barcelona and the Ebro Delta, and two from Sentinel-2 before (TOA, left) and after (BOA, right) atmospheric and sun glint correction using the ACOLITE repository

An example of retrieved parameters maps (Chl-a and SPM) is shown in Figure 23.

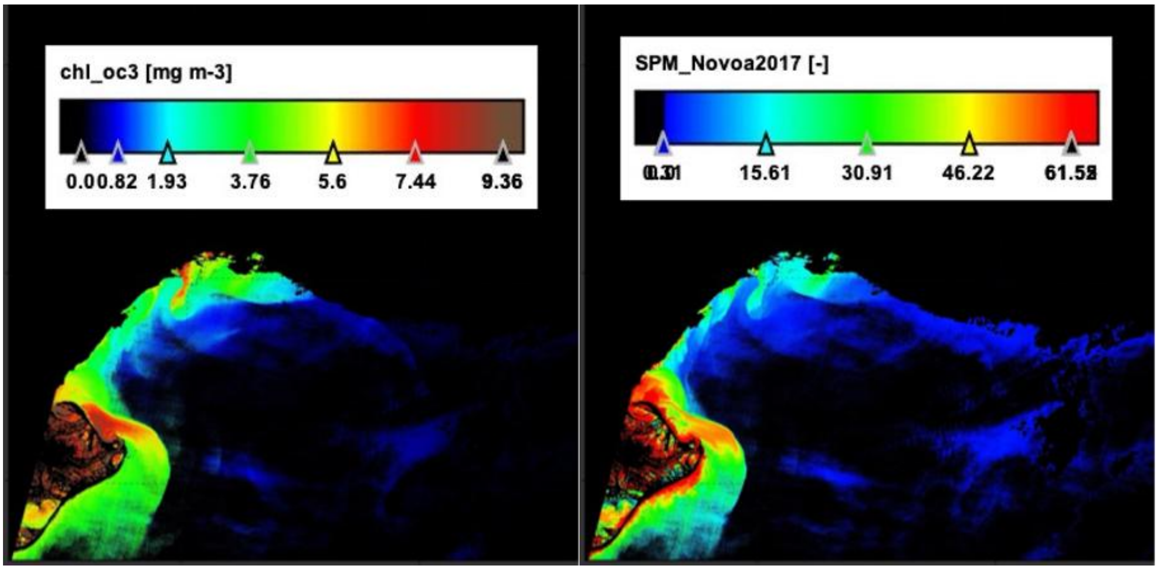


*Figure 23: Example of maps of Chl-a concentration (left) and total suspended matter (SPM; right) retrieved from the BOA image of Sentinel-2 from the SW Catalan coast, shown in Figure 22. Used retrieval algorithms were OC3 for Chl-a concentration and Novoa2017 for SPM. Units for Chl-a concentration are mg*m3.*

Mosaicking

i. Select images to be mosaicked
ii. Perform the mosaic. This could be done by using specialized software, as SNAP, or directly with a python script.

An example of a mosaic for Chl-*a* concentration is shown in Figure 24.

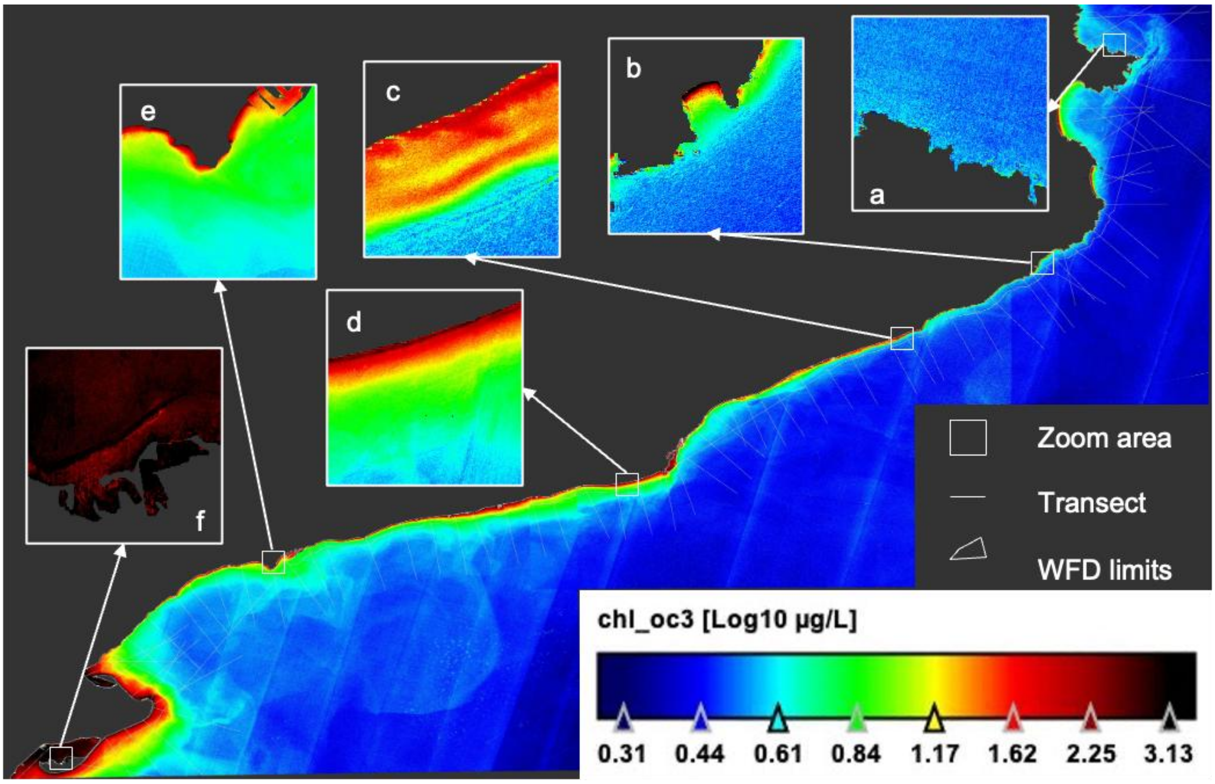


Figure 24: Example of a mosaic map obtained with several single images of Chl-a concentration from the Catlan Coast. Some areas are zoomed in to show how remote sensing images can reveal spatial information in detail.

f. Applying thresholds. This final step results in the ENHANCE product.

6. Impacts Assessment Tools

Under the implementation of the One Heath concept within the DPSIR framework, the EHNACE project not only aims at providing information related to the eutrophication process (see section 5) it also aims at evaluating the impacts of other threatening issues for the human, animal and environmental health. Specifically, biodiversity loss and the possibility of bacterial contamination or algal blooms presence.

6.1. Products

Four products will be delivered in relation to impacts, three related to the Animal Health Indicator and oneto the Human Health Indicator for the OH framework.

Two products will inform about **biodiversity loss** and are related to the Animal Health Indicator. These will be assessed by comparing **fish** occurrences in Urban Beaches (Case Study 1-A) and **plant** occurrences in Ebre coastal areas (Case Study 1-B) reported in the MINKA Citizen Observatory (minka-sdg.org).

Biodiversity loss is assessed by examining changes in species richness and the presence of invasive species detected through MINKA observations, complemented by AI systems that support species identification. For fish in Barcelona, the observed richness is compared with the expected richness using saturation curves to determine whether there is a relative decrease in the number of species detected. For coastal plants in the Ebre Delta, the analysis focuses on expanding and monitoring the species inventory, as the baseline is still being consolidated. In both cases, the emergence or increase of invasive species is also considered a potential indicator of biodiversity alteration.

The third product will inform about the **possibility of bacterial contamination** of coastal waters and is related to the Human Health Indicator. This product will be in the form of maps of turbidity and its categorization with a threshold will serve as an early warning.

The fourth product will inform about the **possibility of the presence of algal blooms** on coastal waters and is related to the Animal Health Indicator. This product will be in the form of maps of phytoplanktonic biomass (a.k.a. concentration of Chl-*a*) and its categorization with a threshold will also serve as an early warning.

6.2. Pipelines

6.2.1. Pipeline for the AI-based identification of animal and plant species from citizen observations

This pipeline supports the Animal Health Indicator by processing citizen-submitted photos through a multi-model AI system. The objective is to extract structured species identification data from images collected via the MINKA Citizen Observatory, enabling scalable biodiversity tracking, including the monitoring of invasive species.

6.2.1.1 Overview

The ENHANCE platform uses two complementary AI pipelines:

- A modular, multi-stage AI flow for fish classification, built on the AMOVALIH (Advanced Marine Observations Validation-Identification based on Hybrid intelligence) architecture.
- A direct API-based integration with Pl@ntNet for plant identification.

An AI Agent that can manage these models will be triggered automatically when users upload new observations through MINKA. Their output is integrated into the ENHANCE registry and later consumed by downstream biodiversity analysis tools (Section 7).

6.2.1.2 AI Pipeline for Fish Classification

The animal pipeline is designed to classify fish in images taken in the coastal environments of Case Study 1 (Barcelona). It relies on three sequential models:

1. Detection: A bounding-box detector identifies the location of each fish in the image. It supports multi-fish scenes by returning coordinates for each instance.
2. Segmentation: Each detected region is passed to a segmentation model that extracts a binary mask of the fish. This step removes background noise and isolates only the relevant pixels (e.g. tail fin, body shape), significantly improving classification accuracy.
3. Species classifier: The segmented fish mask is then classified using a trained fish species model.

Optional human validation (by the community or experts) can be triggered if confidence is below a set threshold or somebody doesn’t trust the classification. All outputs (bounding box, mask, species name, confidence, validation status) are stored in the metadata registry and linked to the original observation.

6.2.1.3 API Pipeline for Plant Classificatin Using Pl@ntNet

Plant identification relies on the external Pl@ntNet service. The pipeline is simpler but highly effective :

1. Forwarding: A plant photo from MINKA is sent to an API of AI models that integrates Pl@ntNet's using a secure channel.
2. Candidate extraction: Pl@ntNet returns a list of likely species with confidence scores.
3. Selection: The platform selects the top species match, provided its confidence exceeds a quality threshold. If uncertain, optional user or expert review is triggered.

As with fish, the identified species, confidence, and validation status are stored with the observation and used to update plant biodiversity metrics.

6.2.1.4 Output and Integration

Both classification outputs are automatically linked to their source observations and published into the registry via the ENHANCE service APIs. These structured species records are used in:

- Richness and presence/absence metrics (Section 7.1)
- Detection of new or invasive species
- Baseline inventory building in the Ebre delta (plants)

All predictions include metadata (e.g. model used, timestamp, processing status) to support traceability, reproducibility, and human verification.

6.2.2. Pipeline for detection of bacterial contamination and algal blooms

Turbidity and phytoplanktonic biomass maps will be retrieved from single (meaning from 1 specific day) ocean colour images obtained by Sentinel-2. First, ocean colour images of high spatial resolution obtained by Sentinel-2 will be downloaded from the Copernicus Data Space Ecosystem (https://dataspace.copernicus.eu/browser) of the ESA . Second, these images will be processed with the repository ACOLITE (https://github.com/acolite) developed by the Royal Belgian Institute of Natural Sciences (https://www.naturalsciences.be). The pipeline is the same as with pressures measured in coastal waters, however in this case the mosaicking phase is skipped (detailed pipeline is described in Section 5.2.2.). Results will be similar as those shown in Figure 23.

7. Prediction models and Risk Maps Production

7.1 Potential Prediction Models

To support operational monitoring of Chl-*a* dynamics, we will explore the development of a set of predictive/statistical models that estimate Chl-*a* concentrations using a combination of environmental and oceanographic predictors. These predictors, such as sea surface temperature (SST), salinity, turbidity and meteorological variables, will be selected based on their demonstrated relevance in Chl-*a* variability. The models may include statistical, machine-learning, or hybrid approaches (for example ANN, SVM and others), depending on performance and interpretability requirements. Each model will be trained and validated using historical datasets (for example the DDBB of the Catalan water agency and the Hellenic Centre for Marine Research and Marine Copernicus services), *in-situ* samples, lab data and satellite data, ensuring robustness across different spatial and temporal conditions. This will be achieved with in-situ and lab data primarily used for local calibration and independent validation

These models will be fully integrated into the project platform, enabling automated updates. All relevant variables will be retrieved directly from the platform’s data ingestion pipeline, which sources harmonized and pre-processed inputs from Copernicus Marine and relevant Earth Observation services. This integration will ensure consistency, traceability, and scalability, while enabling seamless deployment of model outputs.

For model development, python and R will be used. Possible Python libraries that will be used are scikit-learn, geopandas, scipy, statsmodels, XGBoost, pytorch, matplotlib, FastAPI, Docker (for platform deployment) and others. For R, libraries like sf, terra, raster, tidyverse, lubridate, imputeTS, caret, xgboost, keras, tensorflow and others will be used.

The steps that we plan to follow are divided into two sections.

First, the offline training pipeline. Second, operational/real-time pipeline (platform integration).

For the offline training pileline, Figure 25 describes the possible steps.

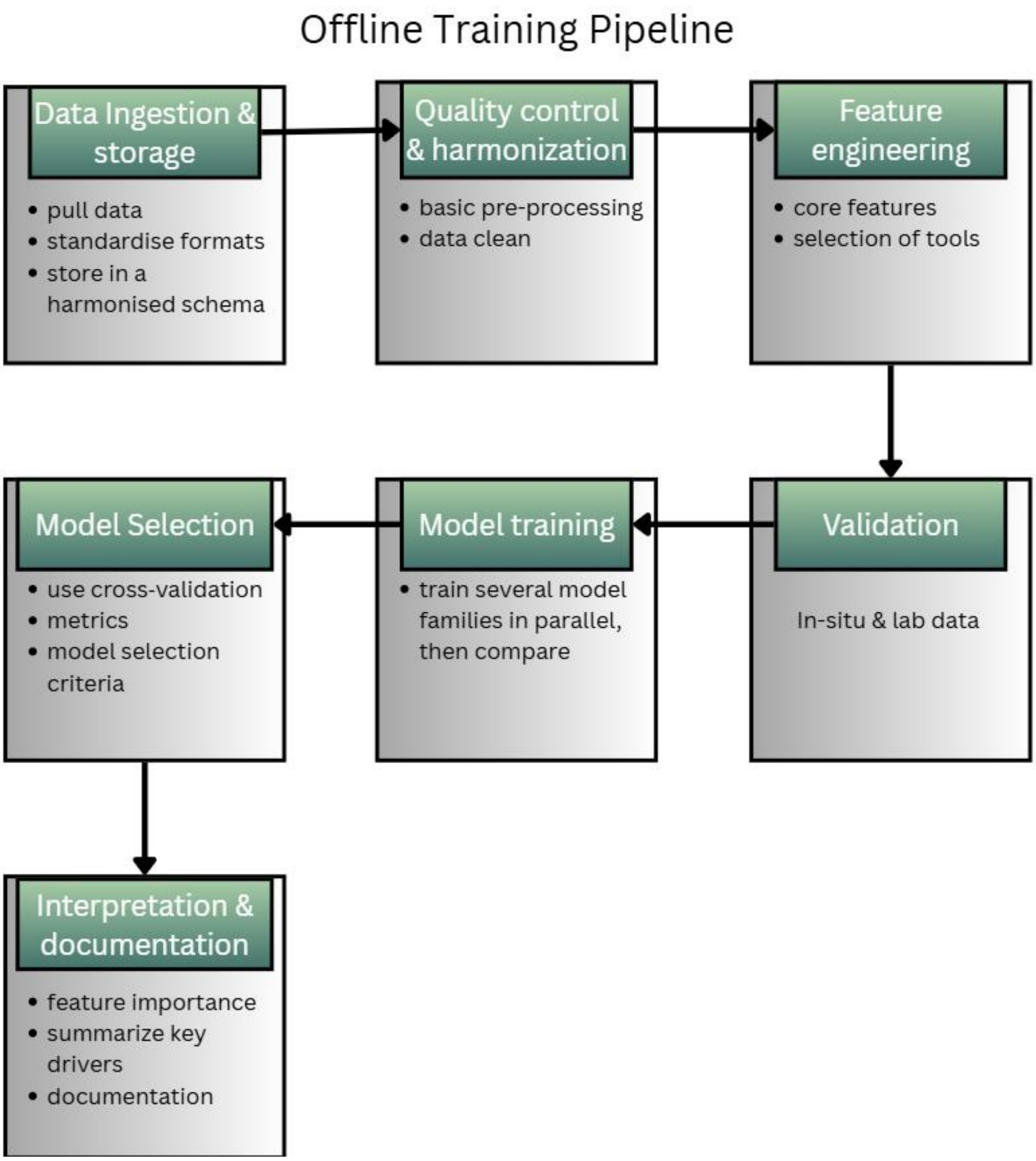


Figure 25: Offline Pipeline

1.Data ingestion and storage:

Bring all relevant data (e.g. CMEMS, EO, in-situ, lab) into a consistent structure that the models can use. Possible data sources include:

  - Copernicus Marine (CMEMS): Chl-a, SST, salinity, turbidity/TSM, currents, mixed-layer depth and meteorological variables.
  - Earth Observation surface reflectances and automatic Chl-a products.
  - Lab and in-situ data: surface temperature, turbidity, salinity, pH, Chl-a.

All data are stored in a common format.

2. Quality Control and Harmonisation:

checking and cleaning your data to make sure it’s realistic, consistent, and usable before modelling. For example, range checks, use provider quality flags, outlier detection, missing data handling, consistency checks

3. Feature Engineering:

turn raw predictors into informative features. Identify core features and choose appropriate tools.

4. Model Validation

Ensure that independent in-situ/lab data are reserved for final validation of both the satellite-based products and the possible prediction models.

5. Model Training

Train several model families in parallel, then compare:

- Statistical baselines: for example, linear regression, GLM, etc.
- Machine-learning models: for example, Random Forest, XGBoost, SVM, ANN/MLP
- Hybrid models

6. Model Selection

- Use cross-validation
- Metrics: for example RMSE, MAE, bias, R-square etc
- Accuracy
- Stability
- Runtime
- Interpretability

7. Interpretation & Documentation

- Compute feature importance
- Summarise key drivers
- Documentation: for example input variables, units, preprocessing steps etc.

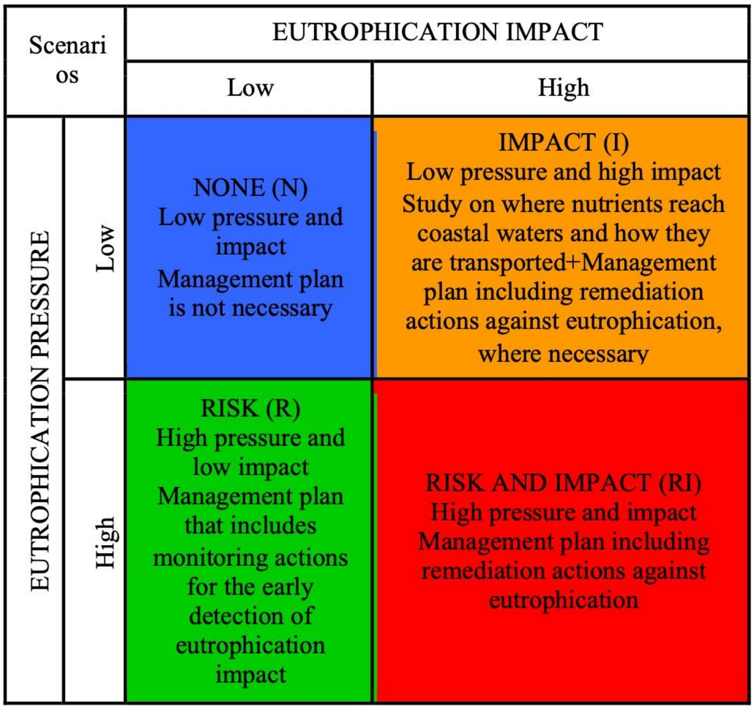
7.2 Dynamic risk maps

ENHANCE project will provide two kinds of Dynamic Risk Maps. One is based on a simple methodology, which crosses information about pressures and impacts, and the other is much more complex but provides more information to stakeholders on pressures and impacts. The first methodology will be tested in both case studies and the second only in case study 2.

7.2.1 Simple Dynamic Risk Maps: the PIM table

Eutrophication is a major threat to coastal waters, which has led to the world-wide adoption of policies aimed at protecting and restoring them. Many of these laws are based on the driver-pressure-state-impact-response (DPSIR) framework and follow the Ecosystem Approach. To apply them, assess eutrophication pressure and impact and provide useful information for managers are necessary. On one hand, eutrophication pressure on land is based on land uses, land cover, and fluvial influences (which can be estimated based on coastal water salinity). On the other hand, eutrophication impact is based on Chl-a concentration. This information allows to establish the pressure-impact relationship and fill in the Pressure-Impact-Management (PIM) table (Flo, E., et al., 2024). The last, guides the design of management plans for coastal zones by identifying where monitoring and remediation actions should be implemented.

The methodology to implement the PIM table and obtain a simple dynamic risk map is very straightforward. First, values of pressure and impact should be categorized in two: high or low. Second, this information should be crossed within the PIM table (Figure 26) to establish which management actions should be implemented. Third, a map should be drawn with the results.



Scenarios		EUTROPHICATION IMPACT	
		Low	High
EUTROPHICATION PRESSURE	Low	NONE (N) Low pressure and impact Management plan is not necessary	IMPACT (I) Low pressure and high impact Study on where nutrients reach coastal waters and how they are transported+Management plan including remediation actions against eutrophication, where necessary
	High	RISK (R) High pressure and low impact Management plan that includes monitoring actions for the early detection of eutrophication impact	RISK AND IMPACT (RI) High pressure and impact Management plan including remediation actions against eutrophication

Figure 26: The Pressure-Impact-Management (PIM) table

An example of the application of the PIM table in the Catalan Coast (Case Study 1) following the requirements of the Water Framework Directive is depicted in Figure 27. The assessment involves all the coastal and transitional water bodies in a 6-year period (2011-2016) and it is based on in-situ data.

Within the ENHANCE project, we aim to apply this methodology with remote sensing data. The results will support the WFD's Biological Quality Element for Phytoplankton and the MSFD's Descriptor 5 on Eutrophication, in the next River Basin Management Plans (4th cycle; 2028–2033) and the next Marine Strategy for the Levantine-Balearic Marine Demarcation (3rd cycle; 2024–2030) respectively.

This information will not only be key for admistrations related to coastal management but also to other stakeholders. These include the Catalan Water Agency (ACA), the Spanish Ministry for the Ecological Transition and the Demographic Challenge (MITERD), the European Commission (EC), the scientific community, and enterprises and organizations related to the Mediterranean coastal zone.

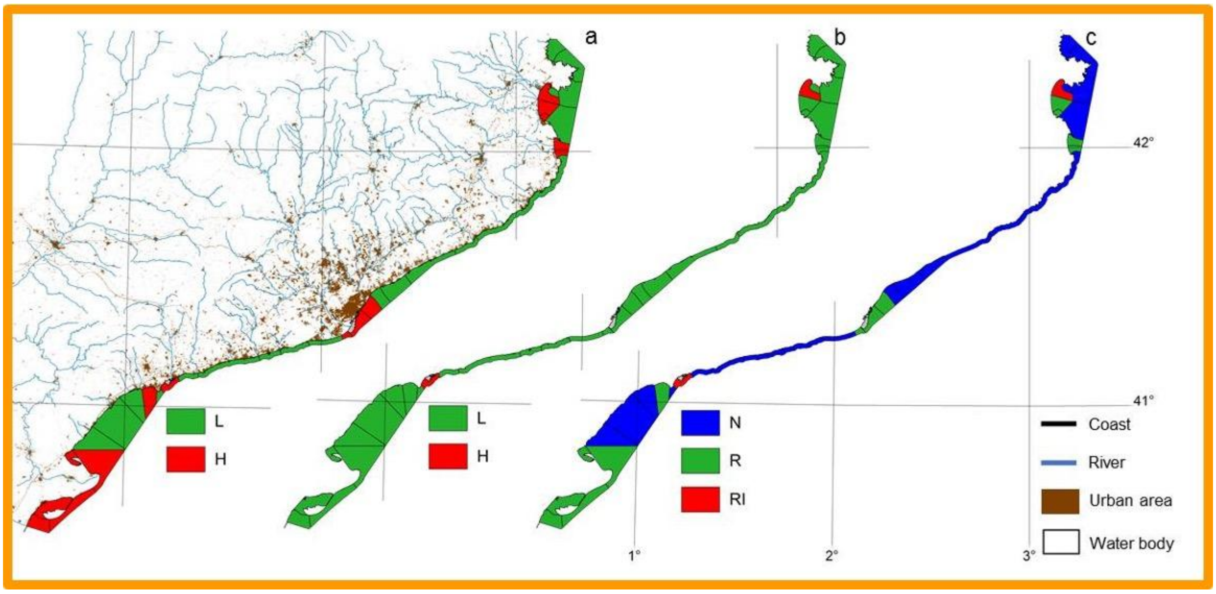


Figure 27: Example of the results obtained with the application of the PIM table. a) map of the measured pressures categorized in low (L) and high (H); b) map of the measured impacts categorized in low (L) and high (H); c) map of the categories of management actions that should be implemented in every water body: none (N), risk (R), and risk and impact (RI). In this example, none of the assessed water bodies were classified as presenting only impact (I), the fourth category of the PIM table. In the map of the pressures rivers and urban land use areas are indicated.

7.2.2 Complex Dynamic Risk Maps

To support the prioritisation of coastal locations for adaptation, we will develop an operational multi-hazard coastal risk index that compares municipalities and coastal administrative units against a set of peer areas. The assessment will be implemented at the level of municipal or coastal administrative polygons, with optional 1–2 km coastline segmentation to capture along-shore hotspots. Risk will be decomposed into three interacting pillars: Hazard (H), capturing the intensity and frequency of coastal, hydro-climatic and geohazards; Exposure (E), representing people, assets and land uses located in harm's way, both across the municipality and within the ≤10 m Low Elevation Coastal Zone (LECZ); and Vulnerability (V), reflecting both Sensitivity and Adaptive Capacity, i.e. how susceptible populations and systems are to damage and how capable they are of preparing for, absorbing and recovering from impacts.

Indicators for each pillar will be compiled from authoritative coastal, hydro-climatic, land-use and socio-economic datasets and harmonised to municipal boundaries and the ≤10 m LECZ. All indicators will be normalised to the [0,1] range using min–max scaling against a benchmark set of coastal peers, with adaptive-capacity indicators inverted after scaling to ensure consistent directionality. Pillar scores will be calculated as equal-weighted means, and overall risk will be derived as a multiplicative composite (EMI_Risk = AC_H × AC_E × AC_V).

Robustness will be assessed through sensitivity analyses (e.g. ±10% weight perturbations and leave-one-out indicator tests) and validation against local event records.

Data processing will start with the delineation of the study boundary. Municipal polygons and coastlines will be acquired and, where relevant, a 0–10 km inland coastal belt will be defined to focus the analysis. Using a high-quality digital elevation model (EU-DEM), the Low Elevation Coastal Zone (LECZ) will be derived by masking elevations ≤10 m and retaining only cells that are hydraulically connected to the sea. Hazard inputs will be compiled from reanalysis products and European hazard layers clipped to the study area, and transformed into event-based extents (e.g., RP50 footprints).

Land-use information from Urban Atlas or CORINE will be reclassified into residential, transport, critical infrastructure, agriculture, wetlands, natural/forest and open areas, and corresponding shares will be calculated both at municipal scale and within the LECZ. Finally, socio-economic and infrastructure datasets (e.g., population, labour, education, GDP from Eurostat and national statistics) will be assembled, and locations of ports, airports, power plants and rail stations will be extracted from official registries and OpenStreetMap, completing the exposure and vulnerability layers for the composite risk assessment.

To bridge the methodological narrative with implementation, the assessment is organised as a coherent, end-to-end pipeline that transforms heterogeneous spatial and socio-economic inputs into operational risk scores and dynamic maps. The pipeline begins by defining the spatial analysis units (municipal/coastal administrative polygons, with optional 1–2 km coastline segmentation to capture along-shore hotspots) and delineating the ≤10 m Low Elevation Coastal Zone (LECZ) that is hydraulically connected to the sea. It then compiles and harmonises indicators across the three interacting risk pillars—Hazard (H), Exposure (E), and Vulnerability (V, combining Sensitivity and Adaptive Capacity)—so they are comparable across a benchmark set of coastal peers through min–max normalisation to the [0,1] range (with adaptive-capacity indicators inverted after scaling to ensure consistent directionality). Pillar scores are computed as equal-weighted means and integrated into a multiplicative overall index (EMI_Risk = AC_H × AC_E × AC_V), which is subsequently stress-tested through robustness checks (e.g., ±10% weight perturbations and leave-one-out indicator tests) and basic validation against local event records. In this way, the pipeline ensures transparency, reproducibility, and decision-readiness, supporting the prioritisation of coastal locations for adaptation and the identification of risk hotspots.

1) Data acquisition

Core layers

- Administrative boundaries (municipal polygons)

- Coastline (polyline)
- DEM (EU-DEM or equivalent)
- Hazard layers (reanalysis + EU hazard layers; flood/surge/waves/erosion/heat/heavy rain/landslides etc.)
- Land use/land cover (Urban Atlas or CORINE)
- Socio-economic & infrastructure (population, GDP, education, labour, etc.)
- Critical infrastructure points (ports/airports/power plants/rail) + OSM extras

Output: data_catalog.csv (source, date, resolution, license, geometry type)

2) Spatial preprocessing (standardise)

1. **Reproject** all layers to a single CRS.
2. **Clip** to study boundary (municipalities + optional coastal belt).
3. **Topology/QC:** fix invalid geometries, remove slivers, harmonise coastlines.

Output: clean geodatabase: 01_clean_inputs.gpkg

3) Build the LECZ (≤10 m, sea-connected)

1. From DEM: mask **elevation ≤10 m**
2. Keep only cells **hydraulically connected to the sea** (sea-connected component filtering)
3. Convert to polygon and intersect with municipalities → **LECZ-by-municipality**

Output: LECZ_10m_connected layer + muni_LECZ (polygon)

4) Optional coastline segmentation (hotspot resolution)

- Generate 1–2 km **coastline segments**
- Create segment "influence zone" (e.g., buffer inland or intersect with coastal belt/LECZ)
- Maintain a crosswalk: segment_id ↔ municipality_id

Output: coast_segments + segment_muni_xwalk

5) Indicator engineering (H, E, V) at two spatial supports

Compute each indicator **(a) at municipal scale** and **(b) within LECZ** (and optionally per segment).

5A) Hazard (H) indicators

- Clip hazard rasters/footprints to municipality and LECZ
- Convert hazards to **event-based extents** (e.g., RP50 footprints)
- Summarise to unit: e.g., mean/max intensity, % area affected, frequency proxies

Output: H_table (rows=units, cols=hazard indicators)

5B) Exposure (E) indicators

- Population in municipality + population in LECZ
- Land-use shares (residential/transport/critical infra/agriculture/wetlands/natural/open) in municipality and LECZ
- Counts/density of critical infrastructure (ports/airports/power plants/rail stations), especially inside LECZ

Output: E_table

5C) Vulnerability (V) indicators (Sensitivity + Adaptive Capacity)

- Sensitivity: age structure, unemployment, deprivation, health proxies, etc.
- Adaptive capacity: income/GDP, education, services access, preparedness proxies (these will be **inverted after scaling**)

Output: V_table

6) Harmonise into one analysis matrix

- Join H/E/V tables into a single "wide" table: unit_id, geometry_id, indicators…
- Handle missingness (document rules: impute / drop / keep with flags)
- Check distributions and outliers (winsorise if you choose—document it)

Output: 02_indicator_matrix.parquet/csv + indicator_dictionary.xlsx

7) Normalisation against peer coastal areas (min–max to [0,1])

For each indicator *j*:

- Compute min/max **across the peer set**

- For **adaptive capacity** indicators (where "higher is better"), invert after scaling:

Output: 03_scaled_indicators

8) Pillar scores + overall risk (EMI_Risk)

Equal-weighted pillar scores (mean of scaled indicators):

Multiplicative overall risk:

(Optional) rescale EMI_Risk to [0,1] for visual comparability.

Output: 04_scores (AC_H, AC_E, AC_V, EMI_Risk, ranks)

9) Mapping & visual products (dynamic risk maps)

- Join 04_scores back to polygons and/or coastline segments
- Produce:
    - **Pillar maps** (H/E/V) + **overall risk**
    - **LECZ-only risk** (separate layer or side-by-side)
    - Hotspot maps alongshore (if segments are enabled)
- Classify (quantiles / natural breaks) and build an interactive web map/dashboard

Output: risk_maps.gpkg + web tiles / interactive map package

10) Robustness, sensitivity, validation

Sensitivity

- ±10% weight perturbations (even if baseline is equal weights)
- Leave-one-out indicators (drop one indicator at a time; recompute ranks)
- Rank stability metrics (Spearman correlation, top-k overlap)

Validation

- Compare high-risk units with **local event records** (impacts, claims, flooded areas)
- Simple diagnostics: hit rate for impacted municipalities, correlation with loss proxies

Output: 05_sensitivity_report + 06_validation_report

11) Operationalisation (repeatable updates)

- Package scripts (Python/R + QGIS/ArcGIS model builder)
- Version datasets + configs
- Run pipeline for baseline + follow-up periods → “dynamic” time slices (2010–2023, or your chosen windows)

Output: reproducible run logs + dated releases (/outputs/2020_baseline/, /outputs/2023_followup/)

The results of the Complex Dynamic Risk Map will be used by the Municipality of Volos and the Region of Thessaly to support evidence-based decision-making, enabling them to prioritise coastal communities facing increased risk and to target prevention, preparedness, and investment measures where they are most needed. In parallel, the outputs will be valuable for the academic community as a robust research resource, supporting spatial and temporal analyses of risk dynamics, the evaluation of scenarios and policy options, and the development of transferable methodological insights for coastal risk assessment and resilience planning.

8. ENHANCE toolkit and interactive user interfaces

8.1. Design of the interactive user interface

The process of creating an interactive user interface is divided into two distinct phases : the analysis or design phase and the development phase. In the design phase, a user-centred approach is employed to make sure the needs of the final end users are taken into account. The work on getting a better overview of the end users with persona's, user stories etc., is described in more detail in deliverable D2.2. For the interface, a first user flow was developed to get an overview of the steps a user can take in the application to reach their goals. In deliverable D2.2 this user flow was described in detail. Since this deliverable Nazka Mapps organised 3 co-creation activities (see Table 3) with the consortium partners to move forward with the design of the tool.

Date	Workshop
September 17 2025	Target audience workshop
October 14 2025	User flow refinement workshop
November 4 2025	Prototype feedback workshop

Table 3: Co-creating activities for the user interface

The first two activities were important to refine this first user flow to fit it more to the needs of the different user groups. In the first co-creation activity, an exercise was completed to share inspirational applications in the field of One Health (like for instance MINKA and MyOcean Pro). In a second step the consortium partners identified which stakeholders are the target audience of the ENHANCE user interface, in other words, which stakeholders will use the application directly? With this information, it was possible to create specific user flows for each of these target groups and discuss them in the second workshop. These separate user flows were combined in one overall user flow (see Figure 28) that serves as an input for the next step: creating the first prototype of the application.

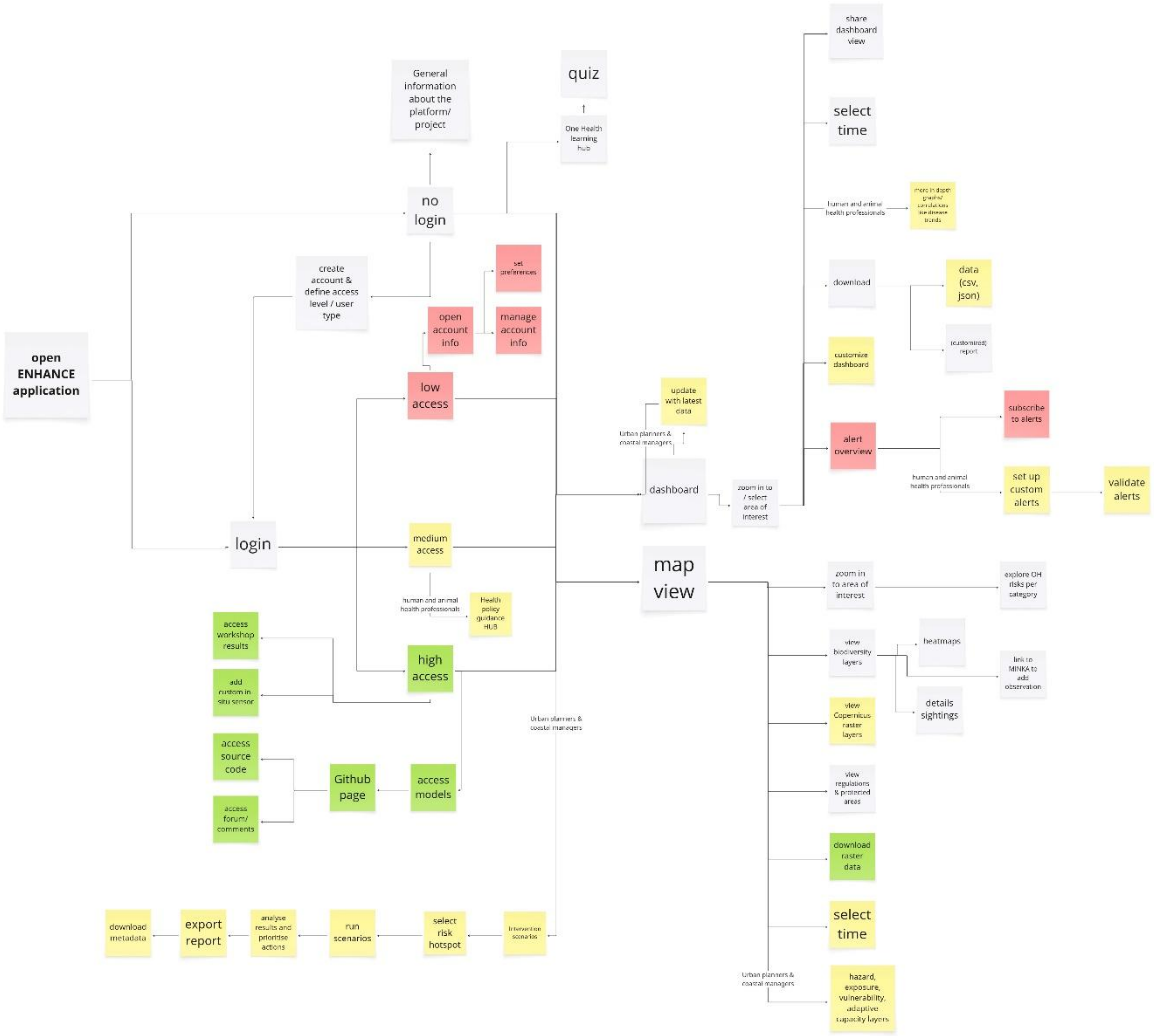


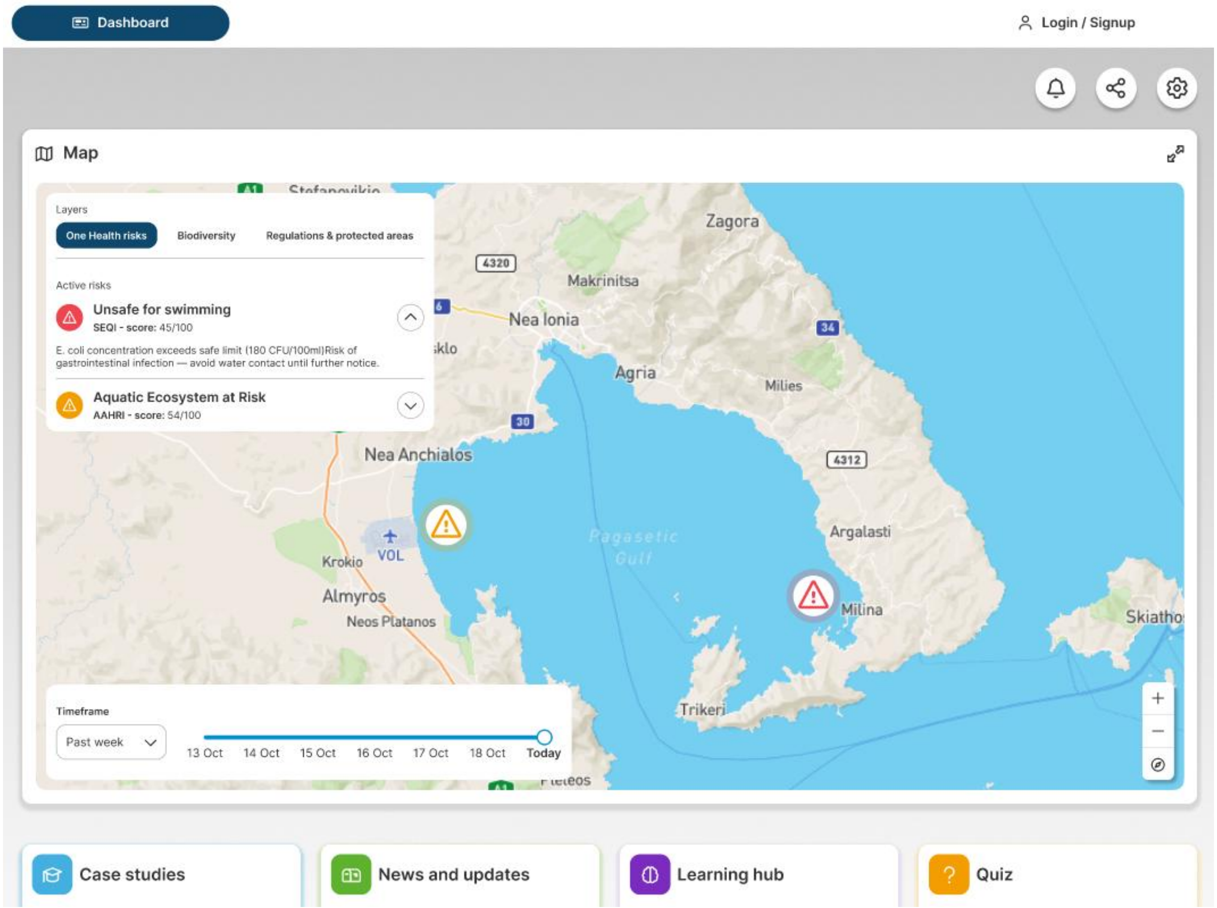
Figure 28: ENHANCE toolkit user-flow diagram

A medium fidelity prototype was developed to gather feedback from consortium partners during the plenary meeting in Barcelona. All the work from deliverable D2.2 as well as the extra co-creation activities (see Table 3) served as a backbone for this first visualisation of the tool. This prototype simulates what the application could look like e.g. the navigation, the look and feel, etc. Nazka Mapps created a first structure for the tool with a dashboard style that integrates a map, summary numbers and alerts into a user-friendly overview for the end users (see example in Figure 29). Different prototypes were made for 4 different target groups: general public and divers (no login users), urban planners & coastal managers, researchers & scientists and animal and human health professionals. During the workshop all consortium partners could share the following feedback on these prototypes:

- Likes
- Improvements

- Ideas
- Questions

This feedback will be explained in more detail in a next deliverable. With this helpful feedback the prototype can be improved further and will be ready in a next step for gathering feedback from end users of the different user groups.



Unsafe for swimming
SEQI - score: 45/100
E. coli concentration exceeds safe limit (180 CFU/100ml)Risk of gastrointestinal infection — avoid water contact until further notice.

Aquatic Ecosystem at Risk
AAHRI - score: 54/100

Figure 29: ENHANCE toolkit mockup

8.2 Development approach

In the development phase, an iterative approach will be used based on scrum methodology. Thanks to the Nazka Mapframe we can make rapid progress in setting up the application: the back-end is based on modern web frameworks and will be connected to the ENHANCE data platform, the front-end will be developed based on the final prototype. This framework is developed in-house by Nazka Mapps and is compatible with the best-known mapping libraries. It avoids wasting time on developing basic components. The Mapframe is an API-centric JavaScript architecture and is 100% based on web technology (and compatible with modern browsers). It is built with reliable open-source technologies such as Node.js, React and Nginx, etc.

Nazka's IT infrastructure is organized in the cloud and makes use of Amazon Web Services with environments that operate in the EU (region 'EU-West', Ireland). We work with different environments in the development process:

- Development (internal testing environment)
- Staging (external testing environment)
- Production (public environment after launch)

We work in sprints of 2 weeks towards a testable version. Each sprint starts with a sprint planning to prioritize the work of the next sprint. We plan to develop multiple versions of the interface, starting with a Minimum Viable Product (MVP). The scrum methodology makes it possible to be very flexible and integrate feedback if necessary. We plan to test the different versions of the applications with users and also incorporate that feedback in each new version. The status of the development will be reported in the next deliverable.

9. Conclusion

This deliverable documents the successful development and initial deployment of the ENHANCE Open One Health Core Platform, which forms the backbone for data integration, analysis, and decision support in coastal management. The platform's modular, cloud-native architecture—built on Docker and Kubernetes—enables secure, scalable, and interoperable management of diverse environmental datasets.

Key components such as robust identity management (via Keycloak and OIDC), high-performance storage (MinIO and MongoDB), and a standards-based REST API ensure that the platform can support a wide range of user needs, from secure data access to seamless integration with external systems. The deliverable also details operational pipelines for environmental pressure and impact assessment, predictive modeling, and dynamic risk mapping, all of which are grounded in stakeholder requirements and real-world use cases.

The ENHANCE platform's user-centric approach—reflected in the design and iterative development of its toolkit and interfaces—ensures that it is accessible and valuable to a broad spectrum of stakeholders, including the public, researchers, and decision-makers. By establishing a secure, interoperable, and extensible foundation, the platform is positioned to drive innovation and collaboration in coastal management and One Health initiatives.

In the next phase of the project, this initial prototype will be extended to implement actual integration points among the ENHANCE components that will be validated in the two case studies. The data pipelines along with the delivered ENHANCE services and relevant backend/frontend integrations will be reported in the next deliverable D3.2 and will provide the baseline for the testing of the ENHANCE solution.

10. Appendix

Tables with examples of the YAML files

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: storage-api-app
  namespace: storage
  labels:
    app: storage-api-app
spec:
  replicas: 3
  template:
    metadata:
      name: storage-api-app
      labels:
        app: storage-api-app
    spec:
      containers:
        - name: storage-api
          image:  singularlogic/storage-
api
          ports:
            - containerPort: 8006
          env:
          - name: ACCESS_KEY
            valueFrom:
              secretKeyRef:
                name: minio-keys
                 key: accessKey
          -      name:
                 SECRET_ACCESS_KEY
            valueFrom:
              secretKeyRef:
                name: minio-keys
                key: secretAccessKey
          - name: DATABASE
            value: minio
          - name: MONGO_URL
          - name: MINIO_URL
            value: minio.storage:9000
          - name: OIDC_PROVIDER
            value: https://keycloak-
ENHANCE.euinno.eu/realms/ENHANCE
          - name: COP_BUCKET_ID
            value: b3d2feda-5337-490c-b2dd-bbcbbb8f7f0d
          - name: CLIENT_ID
            valueFrom:
              secretKeyRef:
                name: kc-minio-client
                key: id
          - name: CLIENT_SECRET
            valueFrom:
              secretKeyRef:
                name: kc-minio-client
                key: password
          - name: ADS_UID
            valueFrom:
              secretKeyRef:
                name: secret-ads
                key: UUID
          - name: ADS_KEY
            valueFrom:
              secretKeyRef:
                name: secret-ads
                key: key
          - name: CDS_UID
            valueFrom:
              secretKeyRef:
                name: secret-cds
                key: UUID
          - name: CDS_KEY
            valueFrom:
              secretKeyRef:
```

```
        value:
mongodb://mongodb-
service.storage:27017
```

```
              name: secret-cds
              key: key
    restartPolicy: Always
selector:
  matchLabels:
    app: storage-api-app
```

Table A1: YAML file declaring the ENHANCE Exchange Platform API deployment

```
kind: Service
apiVersion: v1
metadata:
  name: storage-api
  namespace: storage
spec:
  selector:
    app: storage-api-app
  ports:
  - protocol: TCP
    port: 80
    targetPort: 30000
```

Table A2: YAML file declaring the ENHANCE Exchange Platform API service

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: api-core-ingress
  namespace: storage
  annotations:
    kubernetes.io/ingress.class: nginx
    nginx.org/client-max-body-size: "10m"
    nginx.ingress.kubernetes.io/proxy-body-size: "10m"
spec:
  rules:
  - host: api.enhancecloud.eu
```

```
http:
  paths:
  - path: /
    pathType: Prefix
    backend:
      service:
        name: storage-api
        port:
          number: 80
```

Table A3: YAML file declaring the ENHANCE Exchange Platform API ingress

ENHANCE
One Planet, One Health for Coastal Resilience

enhance-onehealth.eu
#ENHANCE_eu

@enhanceeu

@enhance_eu

Funded by
the European Union

EUSPA
European Union Agency for the Space Programme

"This project has received funding from the European Union's Horizon Europe under grant agreement No 101180146 —ENHANCE. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the granting authority. Neither the European Union nor the granting authority can be held responsible for them."