Ploto

D7.1

The PLOTO Integrated System and Acceptance tests 1st version

Project name

Deployment and Assessment of Predictive modelling, environmentally sustainable and emerging digital technologies and tools for improving the resilience of IWW against Climate change and other extremes.

Horizon Innovation Actions | Project No. 101069941

Call HORIZON-CL5-2021-D6-01

Co-funded by the European Union

Dissemination level	Public (PU) - fully open
Type of deliverable	DEM - Demonstrator, pilot, prototype
Work package	WP7 – On site Integration, Demonstration and Validation Activities
Deliverable number	D7.1 The PLOTO Integrated System and Acceptance tests 1st version
Status - version, date	Final – V1.0, 30/04/2025
Deliverable leader	INTRA
Contractual date of delivery	30/04/2025
Keywords	Integration, functional and integration testing, CI/CD, hazards impact, mitigation solutions

Quality Control

	Name	Organisation	Date
Peer review 1	Christos Lachanas, Adam Malavazos-Siekierka	NTUA	25/04/2025
Peer review 2	Alexis Melitsiotis	EXUS	28/04/2025

Version History

Version	Date	Author	Summary of changes
0.1	18/02/2025	INTRA	Initial Version with ToC
0.2	26/02/2025	INTRA	Update initial version and provide instructions for contributions
0.3	10/03/2025	INTRA, RISA, STWS, AUTH, STWS, ULiege, Sorecc, UM, EXUS	Incorporate inputs for functional tests (sections 5) integration point and test specifications (section 6, section 9)
0.4	31/03/2025	UM, Sorecc, NTUA	Inputs for components description in section 5 and updates in sections 6 and 9
0.5	01/04/2025	UDG	Inputs in section 7
0.6	07/04/2025	RISA, INTRA, ULiege, NTUA, STWS, EXUS, AUTH	Inputs in sections 3 and 4, updates in section 5,6 and 9

0.7	11/04/2025	INTRA, NTUA	Inputs in sections 1 and 2, updates in section 7
0.8	14/04/2025	INTRA	Inputs in Executive summary and section 8
0.9	14/04/2025	INTRA	Preparation of version ready for internal review
0.10	29/04/2025	INTRA, UDG, NTUA	Addressing internal review comments and preparation of version for final QA.
1.0	30/04/2025	INTRA	Final submitted version

Legal Disclaimer

Table of contents

List of figures

List of tables

List of abbreviations and acronyms

Abbreviation	Meaning
AI	Artificial Intelligence
API	Application Programming Interface
BCM	Business Continuity Model
CA	Certificate Authority
CI/CD	Continuous Integration/Continuous Delivery and/or Continuous Deployment
CNN	Convolutional Neural Network
CVDD	Computer Vision for Damage Diagnosis
DSM	Design Structure Matrix
GCS	Ground Control Station
GeoXAI	Geo-Explainable Artificial Intelligence
Grad-CAM	Gradient-Weighted Class Activation Mapping
COP	Common Operational Picture
DT	Digital Twin
HTTPS	Hypertext Transfer Protocol Secure
HW	Hardware
IMS	Incident Management System
IWW	Inland WaterWays
IWAT	IWW Assessment Tool
MitM	man-in-the-middle
Middleware	MW
MHM	Multi-Hazard model
ML	Machine Learning
OMS	Operational Modelling System
OMSDA	Operational Modelling System Data Assimilation
Physical Impact	PI
RAE	Risk Assessment Engine

Abbreviation	Meaning
RBAC	Role-Based Access Control
SAR	Synthetic Aperture Radar
SCM	Source Code Management
SI	Socioeconomic Impact
SSH	Secure Shell
SSL	Secure Sockets Layer
SW	Software
TLS	Transport Layer Security
UAV	Unmanned Aerial Vehicle
UI	User Interface
VM	Virtual Machine
VPN	Virtual Private Network
YOLO	You Only Look Once

Executive Summary

This deliverable incorporates the outcomes of the integration activities that led to the first release of the PLOTO integrated system. In this context, it encapsulates the outputs of the development tasks in WP3 to WP6, focusing on the interoperability among the PLOTO components through the definition and specification of PLOTO integration points and relevant interfaces. This is accompanied by the documentation on the corresponding testing activities that have been put in place and refer to functional and bilateral integration tests.

Starting from the integration methodology, the reader is introduced to the main concepts of the framework that has been established for PLOTO. This refers to DevOps best practices that have been followed to streamline the development/testing and software release workflows through automation. These practices are based on the utilization of a Continuous Integration/ Continuous Delivery (CI/CD) system that has been implemented for PLOTO. Moreover, the integration methodology covers the different steps of the integration activities starting from the identification of the integration points to their implementation and validation through small and larger scale tests. Furthermore, the updated view of the PLOTO reference architecture (initially presented in [1]) is described, highlighting how it is instantiated as a set of bilateral integration points. The PLOTO integration environment which has been used for testing activities is detailed covering the different tools and services which are incorporated in the CI/CD system.

A summary of each component's functionality and relevance to the PLOTO integrated system is also given and followed by the corresponding functional testing activities. Then, the adopted integration plan is presented accompanied by the details of how the PLOTO components interoperate with each other. This starts from the identification of the integration points, and it is followed by their detailed specifications and listing of corresponding integration testing activities.

Finally, the outcomes of *T2.5 – "Customization and full characterization of the measures to be selected for simulation and implementation"* are described in detail, targeting the different PLOTO Use Cases A, B, and C. This covers, initially, the analytic description of the methodology used for customization and full characterization of these measures and then the specific impacts and mitigation solutions selected per use case.

1 Introduction

1.1 Project information

The project entitled **"Deployment and assessment of predictive modelling, environmentally sustainable and emerging digital technologies and tools for improving the resilience of IWW against climate change and other extremes (PLOTO)"** aims at increasing the resilience of the IWW and the connected hinterland infrastructures, especially under adverse conditions, such as extreme weather, accidents and other kinds of hazards. In doing this, downscaled climate change scenarios will be combined with simulation tools and actual data, to provide operators an integrated tool able to support more effective management of their infrastructures at strategic and operational levels.

PLOTO project consists in the deployment and assessment of predictive modelling, environmentally sustainable and emerging digital technologies and tools for improving the resilience of IWW against climate change and other extremes. An integrated tool is set up to allow relevant authorities to improve the efficiency of their infrastructures management. This tool is a combination of downscaled climate change scenarios with simulation tools and actual data. Six complementary avenues will be considered to achieve this integrated tool that will support relevant authorities and their operators for more effective management:

- Measure and use high-resolution modelling data for the determination and assessment of the climatic risk of the selected transport infrastructures and associated expected damages.
- Use existing data from various sources with new types of sensor-generated data (computer vision) to feed the used simulator.
- Utilise tailored weather forecasts (combining seamlessly all available data sources) for specific hot spots, providing real-time early warnings with corresponding impact assessment.
- Develop improved multi-temporal, multi-sensor UAV- and satellite-based observations with robust spectral analysis, computer vision and machine learning-based assessment for diverse transport infrastructures.
- Design and implement an integrated resilience assessment platform environment as an innovative planning tool that will permit a quantitative resilience assessment through an end-to-end simulation environment, running "what-if" impact/risk/resilience assessment scenarios. The effects of adaptation measures can be investigated by changing the hazard, exposure and vulnerability input parameters.
- Design and implement a Common Operational Picture (COP), including an enhanced visualisation interface and an Incident Management System (IMS).

The PLOTO integrated platform and its tools will be validated in three case studies in Belgium, Romania and Hungary.

1.2 Purpose of the deliverable

Deliverable 7.1 entitled *"The PLOTO Integrated System and Acceptance tests 1st version"* aims to describe in detail the integration aspects of PLOTO components that lead to the first release of the PLOTO unified platform. The integration framework is elaborated, covering the presentation of DevOps best practices that have been followed on top of a CI/CD platform which has been established to streamline the development, integration, and testing activities of PLOTO components. Moreover, the specification of all identified integration points, functional and integration tests that have been put in place is provided in detail.

The integration timeline covering the different steps of the integration procedures is highlighted. This plan has been fully achieved and marks the successful completion of the corresponding *Milestone 7 - "First Integrated version of the PLOTO system"*.

1.3 Intended audience

D7.1 dissemination level is public. It targets any audience interested in the specifics of the PLOTO integrated system, how the PLOTO components interact with each other, as well as the associated DevOps procedures and integration point objectives.

1.4 Structure of the deliverable and its relationship with other work packages/deliverables

The deliverable is structured as follows:

- **Section 1** presents the overall objectives of PLOTO, the purpose of this deliverable, as well as its intended audience and structure.
- **Section 2** introduces the reader to the main aspects of PLOTO integration methodology covering the concepts of Continuous Integration/ Continuous Delivery and Continuous Deployment (CI/CD), as well as the processes of integration points identification and testing definitions. Finally, it highlights the communication channels and technical management tools that have been used throughout the project.
- **Section 3** presents an updated view of the instantiated PLOTO architecture, which is based on the initial architecture provided from D2.2 and includes refinements following more recent project developments.
- **Section 4** presents in detail the integration environment that has been put in place to facilitate and optimize the PLOTO integration activities. It describes analytically the secured CI/CD infrastructure and services, as well as the associated software build, testing, and deployment automation procedures.
- **Section 5** provides an overview of all the software components integrated in the PLOTO platform, accompanied by the corresponding functional tests that have been carried out.
- **Section 6** delves into the integration specifics, covering the integration timeline which has been followed, the identification and specification of the bilateral integration points, as well as the associated integration tests. Finally, it provides an overview of the next integration steps.
- **Section 7** reports on the outcomes of *T2.5 – "Customization and full characterization of the measures to be selected for simulation and implementation"*.
- **Section 8** concludes the document and provides an overview of the next steps leading to the final release of the PLOTO integrated system.

This deliverable is related to the outputs of *T2.3 – Design of the overall system architecture* as detailed in [1], specifically regarding the instantiation of the PLOTO architecture through the unified PLOTO integrated platform. It also references the outcomes of WP3-WP6, which concern the development of individual PLOTO technical components, as documented in the corresponding deliverables. The PLOTO integration task T7.1 incorporates these outcomes as software artifacts that are continuously integrated into the platform according to the PLOTO integration methodology.

2 Integration methodology

The goal of the PLOTO integration framework is to integrate the PLOTO technical components and services from WP3-WP6 into a unified platform. This platform allows for flexible and secure instantiation of the PLOTO architecture following the varying requirements of different pilot executions. To streamline the testing and integration activities a CI/CD system has been set up, as a set of interconnected DevOps services that support the development activities. The CI/CD concept is described in Section 2.1 and the CI/CD services and relevant procedures are explained in detail in Section 4.

2.1 Continuous Integration/ Continuous Delivery

The CI/CD approach promotes seamless collaboration among development teams, enabling efficient code releases, updates, and the automation of testing and integration workflows.

Continuous Integration involves developers regularly pushing their code updates into a shared repository. Each code submission is automatically tested, facilitating the early detection of errors in the development process. Automated builds and tests (including unit, functional, and integration tests) validate each integration to ensure that the application remains stable as new changes are added to the main repository branch. This process includes packaging the software components into Docker container images and deploying them as containerized applications in the dedicated testing environment hosted by Docker-based servers. The primary objective of CI is to speed up the release cycle by identifying and addressing bugs early, thus improving the development workflow and reducing the need for extensive backtracking. This allows teams to focus more on development and integration efforts.

Continuous Delivery extends the principles of Continuous Integration and represents the next phase in the software release process. Its primary goal is to generate a deployable software artifact intended for end-user environments, such as pilot deployments. It's essential to keep this stage in a consistently "green" or healthy state, signifying that the artifact is always deployment ready. By ensuring that every code change is immediately releasable, Continuous Delivery helps minimize the risks associated with new software and feature releases. This methodology allows for smaller, more frequent updates. In a Continuous Delivery setup, automation is implemented up to the point where deployment-ready artifacts are produced, but the actual deployment to production is typically initiated manually.

Continuous Deployment advances the release pipeline by fully automating the deployment of software to the production environment (in contrast to continuous delivery where, as explained above, the actual deployment to production is manual), provided that all automated tests have been successfully completed.

The PLOTO framework, is based on a **Continuous Integration/Continuous Delivery (CI/CD)** model to manage testing and deployments of containerized software components. The CI/CD environment has been established on Hetzner Cloud Linux Virtual Machines (VMs) (section 4.1). To support this setup, the specific hardware requirements of the various PLOTO software components were first assessed to ensure that the infrastructure could accommodate them within the CI/CD services and the PLOTO development/testing servers.

A technical workshop was conducted to familiarize the technical partners with the integration environment and framework developed by the INTRA team for the project. Comprehensive documentation and sample projects were provided to assist the technical partners in creating their own deployment and testing pipelines for their containerized solutions.

2.2 Integration points identification, specification, and testing

Identifying the integration points (Section 6.2) as a complete set of bilateral PLOTO components that interface with each other across the various PLOTO Use Cases was one of the initial steps of the integration activities. The aim of this step was to update the relevant initial inputs from the release of the PLOTO architecture [1], considering the refined technical data flows of the pilot scenarios.

After their identification, the integration points were further detailed including the selection of appropriate protocols/interface technologies to ensure interoperability and defining sequence diagrams to be implemented throughout the integration process (Section 9). Based on these specifications, exact testing plans have been executed at both the PLOTO component level (functional tests in Section 5) and integration point level (integration tests in Section 6.3). At the time of the first release of the PLOTO platform, the defined functional and bilateral integration tests have been largely implemented. Alongside the completion of these tests, the primary ongoing activity is the implementation of end-to-end tests based on the defined technical data flows of the operational use case scenarios, which will be reported in *D7.2: "The PLOTO Integrated System and Acceptance tests final version"*. The objective of these tests is to validate the PLOTO integrated platform at a system level.

2.3 Communication channels & management of integration activities

To support PLOTO platform's integration activities, various communication channels have been established. Regular bi-weekly T7.1 meetings are held to monitor the status of ongoing activities, along with ad-hoc meetings scheduled to address specific challenges, resolve technical issues, and more. Additionally, shared project planning and integration monitoring tools based on GitHub [2] have been implemented and are used extensively to provide a common view of the status, blocking issues, and progress of integration/testing activities among the technical partners. Furthermore, dedicated PLOTO communication channels on the Slack platform [3] have been set up, facilitating quick and effective exchanges for brainstorming and issue resolution among PLOTO component providers.

3 Instantiation of PLOTO reference architecture

As described in [1] (M10), the high-level architecture of the PLOTO platform aligns with the project's Work Packages (WP3, WP4, WP5, and WP6). Each WP contributes key models, data, and technological solutions that are seamlessly integrated and processed through the Middleware services, as further described in [4] (M20) and [5] (M28). This modular architecture ensures that all components interact smoothly, maximizing both the efficiency and scalability of the platform.

Since the initial definition of the PLOTO architecture in [1], submitted in M10 of the project, several updates have been implemented to better meet the evolving project needs and the specific requirements of the technical partners. Notably, some modules have been renamed for clarity and accuracy. For example, the module previously referred to as Validated Impact Indicators (T3.6) has been renamed to Operational Modelling System Data Assimilation (OMSDA, T3.6), reflecting its more focused functionality. Additionally, some modules have undergone integration of components and datasets, which were previously outlined separately in the earlier architecture. For instance, IWW Assessment Tool (IWAT, T6.1) and Standard Response Procedures (T6.3) are now integrated into the platform architecture, ensuring a more cohesive structure.

These updates are fully reflected in the revised architecture (as displayed in Figure 1), aligning with the latest project developments. The following sections of this document provide further technical details and descriptions on the modules, as well as on the integration points and their respective specifications, demonstrating how these changes contribute to the overall coherence and adaptability of the PLOTO platform.

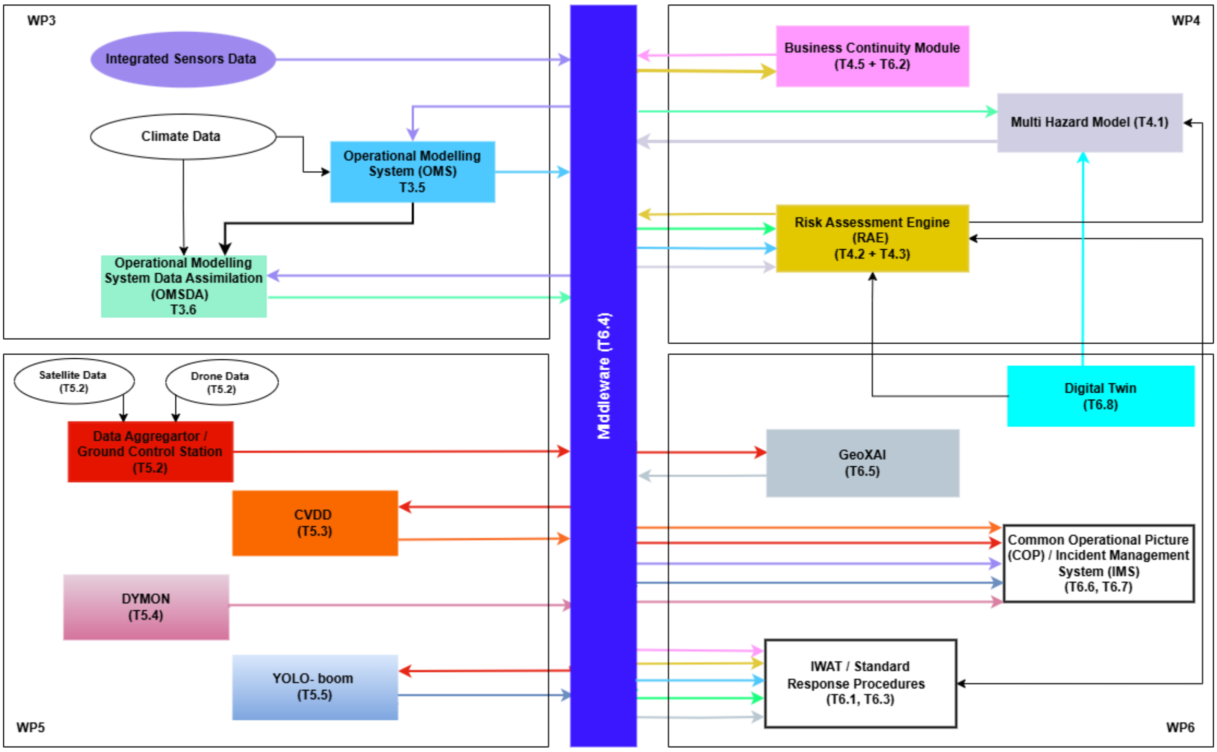


Figure 1 Updated PLOTO architecture

4 PLOTO Integration environment

Many software components which are integrated in the PLOTO platform use open-source DevOps technologies for build, testing, and deployment operations using automated workflows. These DevOps tools are integrated in a robust CI/CD system. This system brings together the various software components of the PLOTO platform and establishes the appropriate environments needed for the development and release processes.

This chapter describes the CI/CD system components in detail, covering both the installation and configuration processes as they were performed in the PLOTO project. Each CI/CD tool is discussed, highlighting its role and contribution to the overall project workflow.

The PLOTO CI/CD system has been implemented in a cloud-based environment, providing an automated build, testing, and software release system. This automation boosts the efficiency of the development process, allowing developers working with the CI/CD toolset to integrate and manage their services smoothly. This streamlined integration is essential for ensuring the consistency and reliability of the development lifecycle throughout the system.

4.1 Hetzner Cloud Infrastructure

The service approach known as "cloud hosting" depends on an external provider for compute and storage resources. Due to the flexible, pay-as-you-go nature of this provider's infrastructure, customers can increase resources horizontally as needed while maintaining high reliability. The software components participating in the CI/CD system of PLOTO project are hosted on virtual machines provided by Hetzner, a well-known public cloud provider [6].

The servers of Hetzner Cloud are hosted in German data centres. These servers are equipped with state-of-the-art hardware, such as Intel® Xeon® Gold and AMD EPYCTM 2nd Gen processors, as well as quick NVMe SSDs for peak performance. This provides the PLOTO platform with high-speed and reliable infrastructure.

Hetzner's cloud infrastructure provides a variety of beneficial features. These include snapshot capabilities, regular backups, and stringent data protection measures. These features are essential for ensuring the integrity and availability of the PLOTO software.

Figure 2 illustrates the project allocation of Hetzner Cloud's hosting services dedicated to the PLOTO platform. It highlights the virtual servers that are key components of the CI/CD system and the docker-based development/testing servers.

Rocky Linux, an open-source enterprise operating system, has been chosen as the operating system for these virtual machines. Rocky Linux is an essential option for business settings that require reliable and secure operating systems since it strives to offer full bug-for-bug compatibility with Red Hat Enterprise Linux. This choice demonstrates the project's dedication to preserving security and reliability in its cloud-hosting setup.

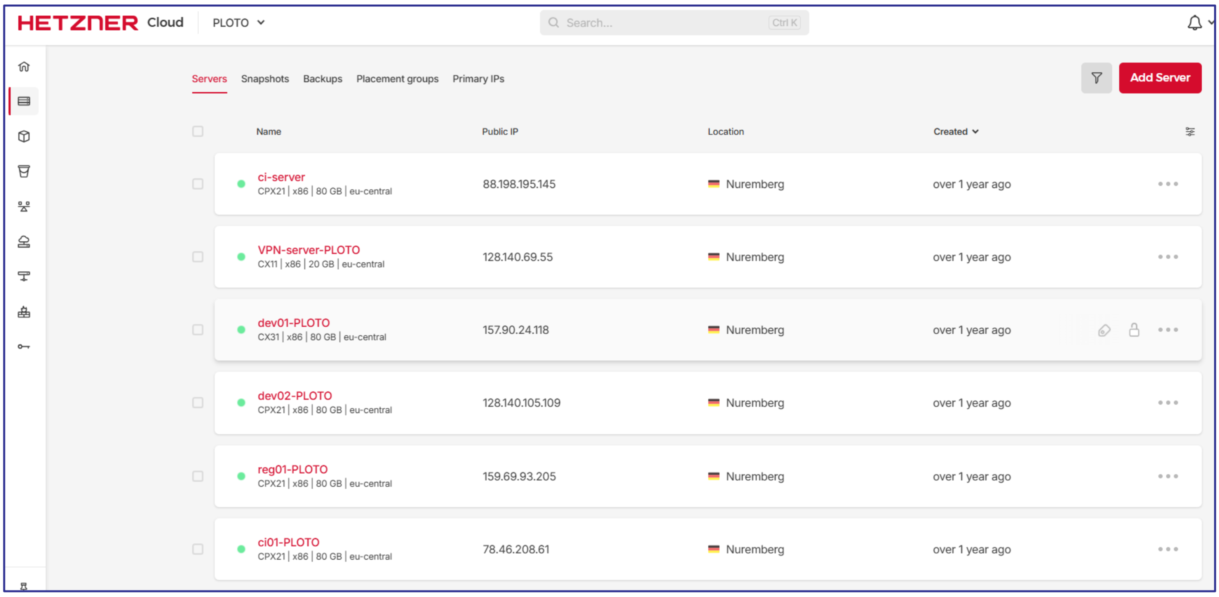


Figure 2: PLOTO project virtual servers' allocation in Hetzner cloud

4.2 CI/CD system

4.2.1. Overview of CI/CD services

As previously stated, many PLOTO software components have made use of the provided Continuous Integration & Continuous Delivery (CI/CD) system, which includes essential services like GitHub SCM (Source Code Management, Section 4.2.2), Jenkins continuous integration server (Section 4.2.3), Harbor private container registry (Section 4.2.4), and Portainer container management tool (Section 4.2.5). These integrated tools support the full software lifecycle from development to the release of extensively tested and fully functional software.

A training workshop has been held, and a *"CI/CD User Guide"* has been provided to the PLOTO technical partners to increase project participants' comprehension and use of the PLOTO CI/CD system. In order to obtain the best results, this guide provides thorough instructions on the general system architecture, the aforementioned CI/CD software tools used, and how to use them.

The user guide also includes examples that demonstrate the build, deploy, and test procedures step-by-step. Additionally, it offers links to the resources made specifically for these purposes. Through adoption of these standards, the development teams working on the project have successfully integrated their software modules, tested them, and deployed them.

4.2.2. Version Control System

GitHub [7] acts as the version control system for PLOTO software. Through its powerful online platform, it supports software development and version control through Git. In this context it provides the basic distributed version control and SCM functionalities of Git but also enhances these capabilities with its own unique tools to optimize the development workflow. The platform's distributed version control system enables multiple developers to collaborate on the PLOTO software simultaneously, enabling them to clone the entire repository, make changes locally, and push their updated versions back to GitHub. This system ensures that there is always a complete version history and backup, helping to maintain the integrity and continuity of the software development process.

Moreover, GitHub offers advanced access control mechanisms, enabling PLOTO project administrators to precisely manage who can view or edit the project. This ensures that sensitive parts of the software remain accessible only to authorized individuals, which is crucial in collaborative environments with multiple partners or contributors. Additionally, GitHub enhances collaboration with a variety of features that support team interaction and project management. Bug tracking allows team members to report, monitor, and resolve issues efficiently, ensuring that bugs are addressed promptly.

The PLOTO Github organization (Figure 3) has been created and can be accessed at : https://github.com/PLOTO-PROJECT.

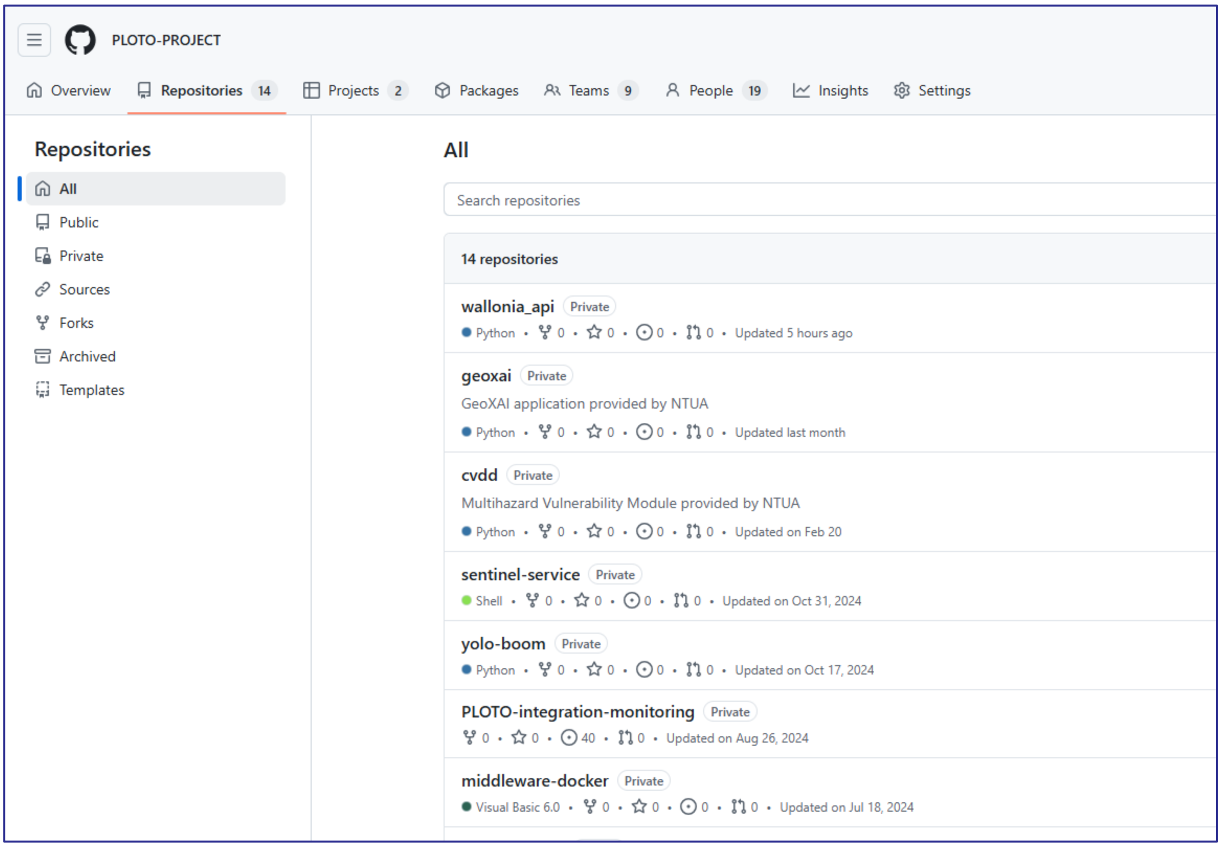


Figure 3: PLOTO Github organization

To contribute source code to the PLOTO repository, each technical partner must have a GitHub account and be a member of the PLOTO organization.

Additionally, distinct GitHub teams have been established for the technical partners to enhance collaboration and facilitate project management. This organizational strategy ensures that each partner has a dedicated space for their contributions, allowing for better management of permissions, more focused discussions, and tailored access to relevant repositories and workflows. By segmenting teams in this manner, each technical partner can operate within a controlled environment that is specifically aligned with their expertise and responsibilities, thereby optimizing the development process and facilitating more efficient and secure collaboration.

4.2.3. Continuous Integration server

Jenkins [8] has been selected as the CI server within the CI/CD framework for the PLOTO platform. The PLOTO Jenkins server is hosted as a Docker container on one of the virtual servers provided by Hetzner Cloud, and it is accessible via: https://jenkins.ploto.rid-intrasoft.eu/. To ensure secure connections, Let's Encrypt [9] has been chosen as the certificate authority for the server.

The Continuous Integration process on the platform is organized as follows (Figure 4):

1. **Local Development:** Developers make modifications to the source code in their local repositories.
2. **Commit Changes:** Once changes are made, they commit these updates to the shared repository.
3. **Trigger Build:** Upon committing, a notification is sent to the Jenkins CI server.
4. **Automated Build and Test:** The CI server retrieves the latest source code, builds the application, and runs both functional and integration tests.
5. **Artifact Generation:** If tests pass, the server creates deployable, testable artifacts.
6. **Tagging Builds:** The server labels the build with a tag corresponding to the version of the code it has just processed.
7. **Feedback and Reporting:** Jenkins provides the development team with detailed reports on successful builds and tests and alerts them if any build or test fails.
8. **Issue Resolution:** The development team quickly addresses any issues that arise.
9. **Ongoing Integration and Testing:** Throughout the project's lifecycle, the server continuously integrates new changes and runs tests to maintain stability and functionality.

This structured approach supported by Jenkins ensures a reliable, efficient, and continuous integration environment, greatly improving the development process for the PLOTO platform.

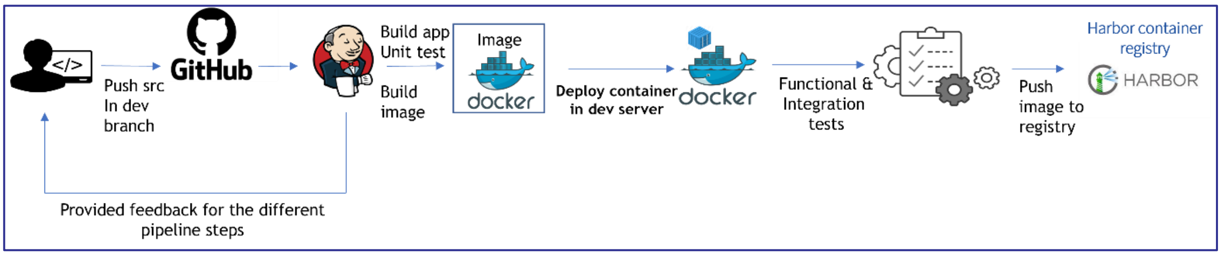


Figure 4: Continuous Integration workflow

Dedicated workspaces (folders) have been created that correspond to the different components of PLOTO as shown in Figure 5.

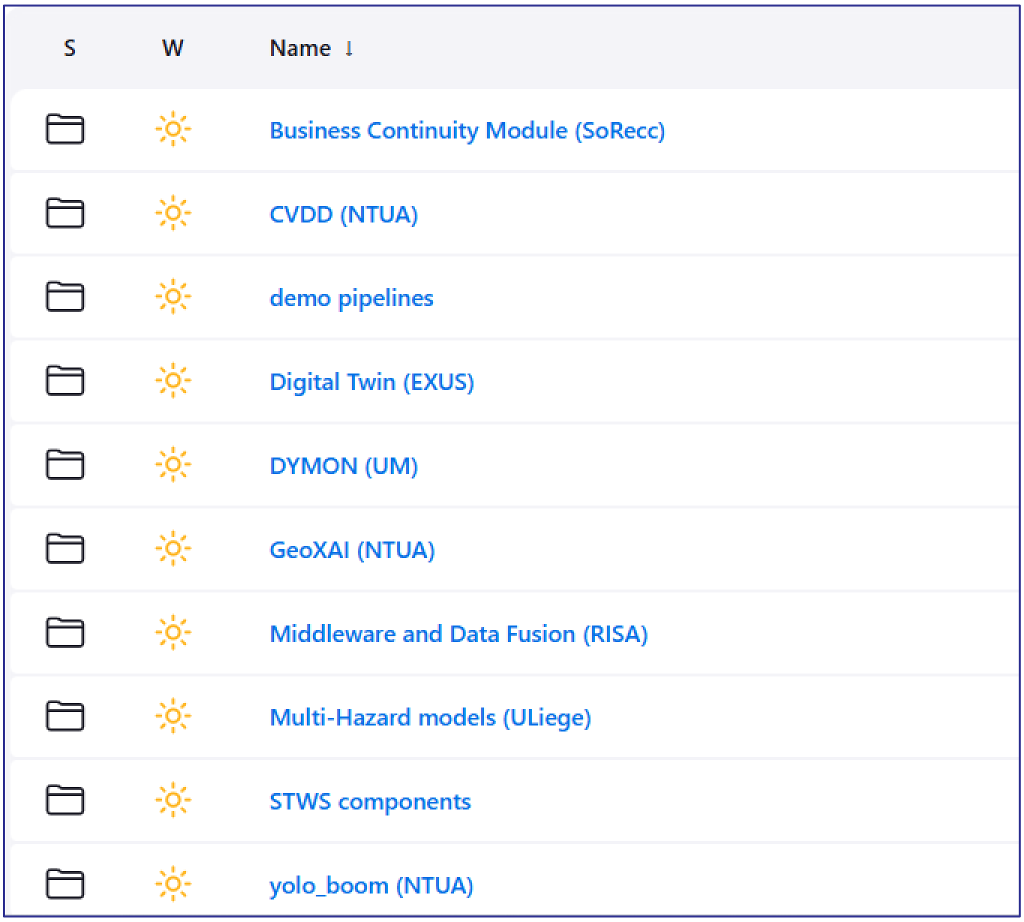
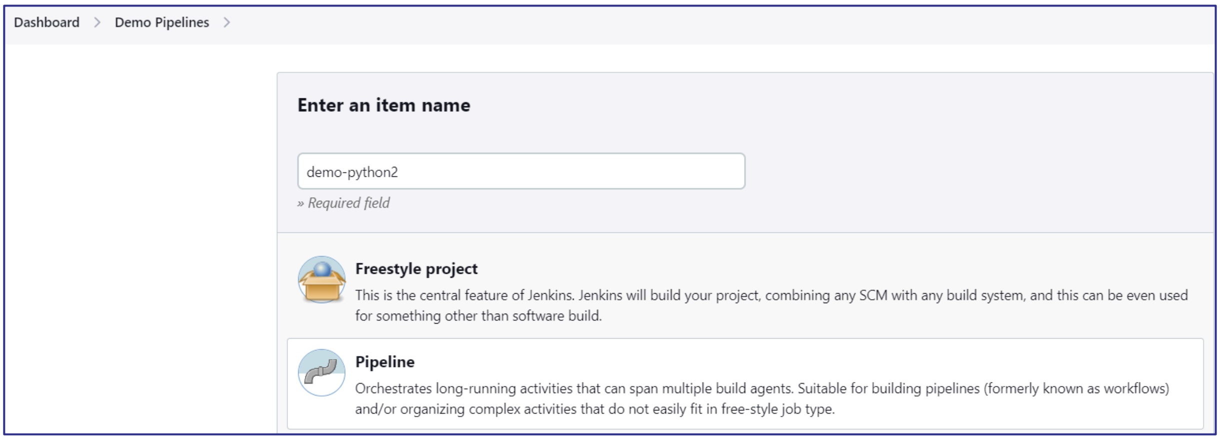


Figure 5: Component-specific allocation of workspaces in PLOTO CI server

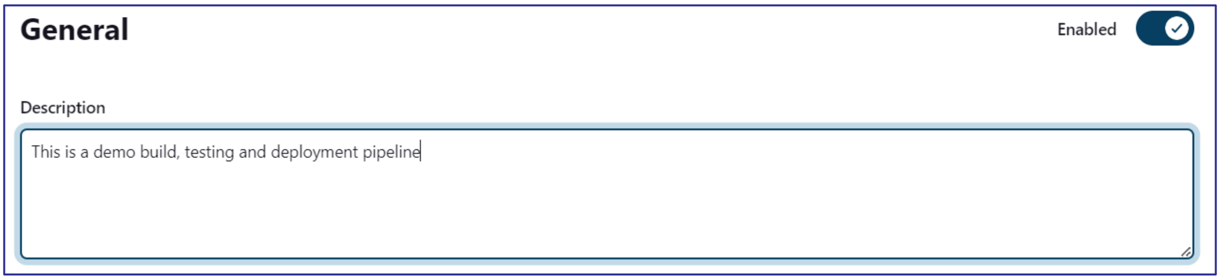
A Jenkins account has been provided to each partner with write permissions on specific workspaces.

In the following, the steps to create a Jenkins pipeline are described in detail.

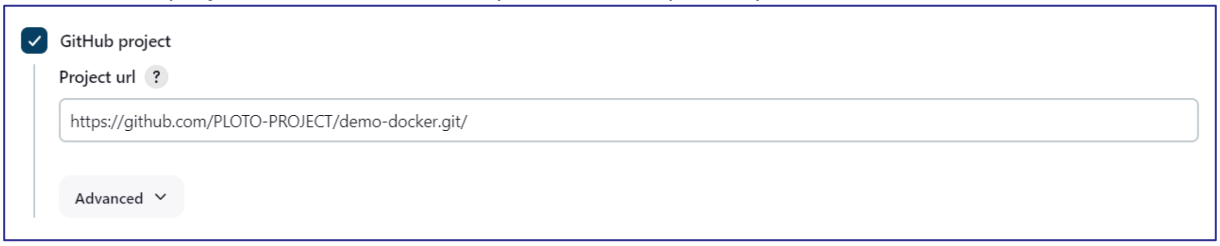
1. Create a **New Item** in Jenkins under the folder that corresponds to your component.
    - Enter an item name. It is recommended to give a name that matches the Github project and the associated branch (e.g., /master or /development)
    - Select Pipeline and click OK



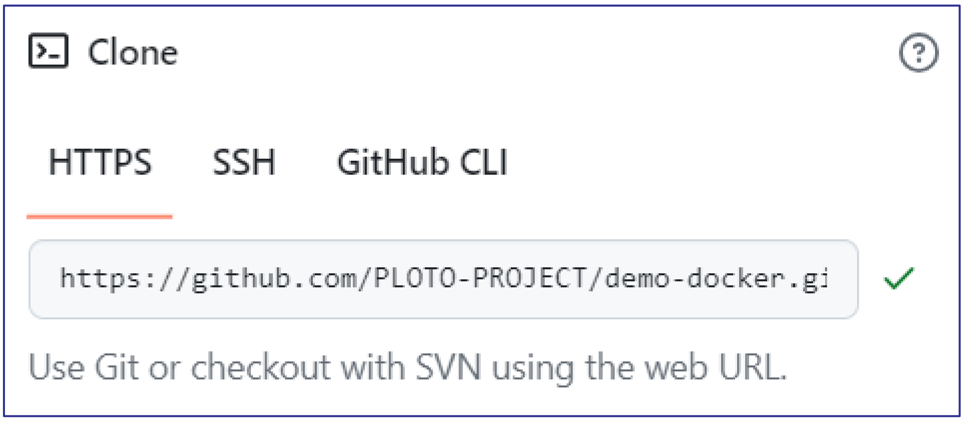
2. In the General tab add a description for your pipeline.



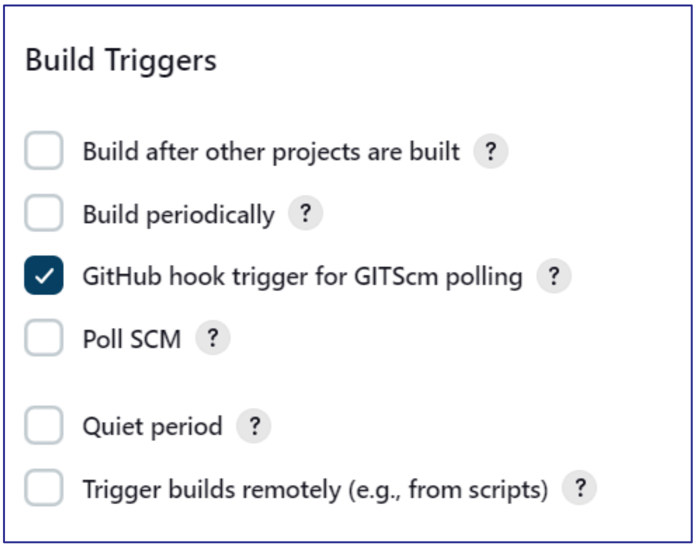
3. Select Github project and add the URL of your Github repository.



You can easily retrieve your repository's URL from PLOTO Github as follows:



4. Select the Github hook trigger option from the list of Build triggers

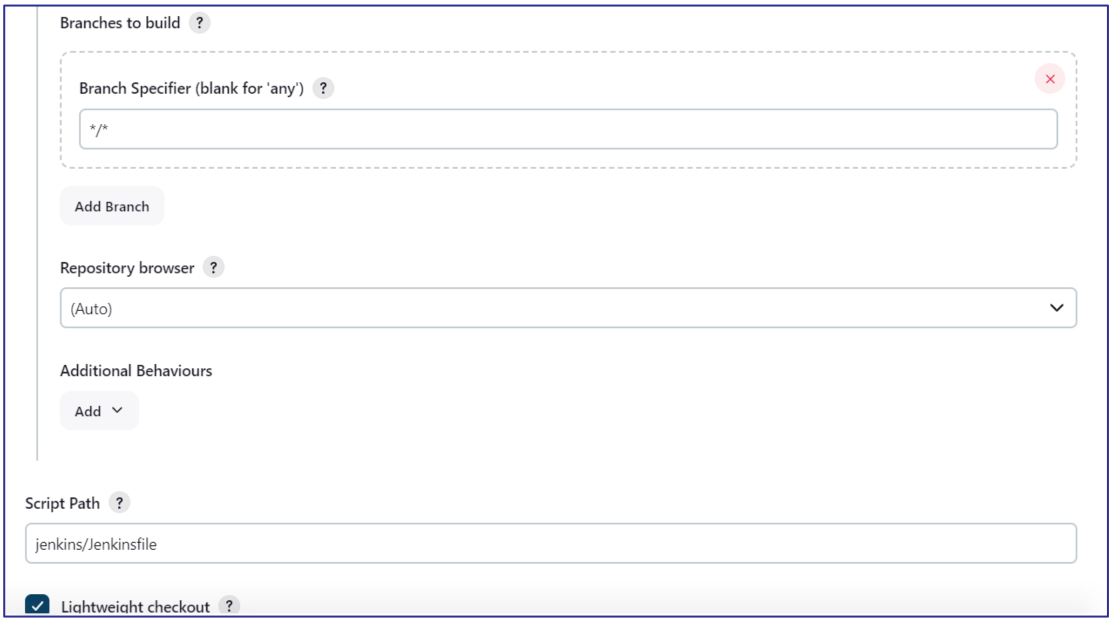


5. There is also the option to launch a job after another project is finished. In such a scenario, where the job needs to be launched after another job is finished (e.g., functional/integration test-jobs,

acceptance test jobs, etc.), the following option needs to be checked, referencing the job that precedes:

6. In the pipeline section select:
    a. Definition: Pipeline script from SCM
    b. SCM->Git
    c. Insert the Github repository URL as in step 3.
    d. Select “Github credentials for user rid-devops-admin”

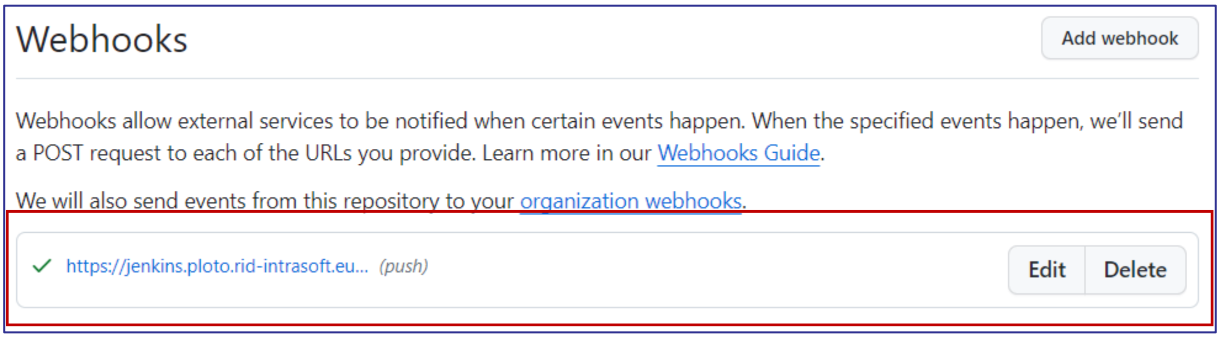
7. In "Branches to build", add the target branch(es) (e.g., */master, */development, or */* for any branch).
8. In the script path, add the path of your jenkinsfile relative to the repository’s parent directory (e.g., jenkins/Jenkinsfile).



9. Save the configuration.

Check if the webhook is registered in Github to automatically trigger the Jenkins pipeline after github push events:

10. On your Github repository go to Settings -> Webhooks and verify that the Webhook is registered with your pipeline as shown below:



4.2.4. Private container registry

Harbor [10] is an open-source project designed to provide a comprehensive container image registry with advanced management and security features for Docker images. It goes beyond the fundamental functionalities of a container registry by incorporating Role-Based Access Control (RBAC), vulnerability scanning, and image signing and verification. Harbor enables the replication of container images across multiple registry instances, ensuring consistent image management across various environments or geographic locations. It is built for high availability and can be deployed in a multi-cluster setup, offering resilience and scalability suitable for enterprise-level applications. Additionally, Harbor's architecture is highly extensible and includes a user-friendly web interface, making container image management and tracking straightforward and efficient.

A private Docker Registry has been established, allowing users to push and pull Docker images. This registry uses Secure Sockets Layer (SSL) encryption and user authentication. The URL of the Docker Registry is: https://harbor.ploto.rid-intrasoft.eu.

Different projects have been created for each technical component (Figure 6), similarly to the organization described for Jenkins in section 4.2.3.

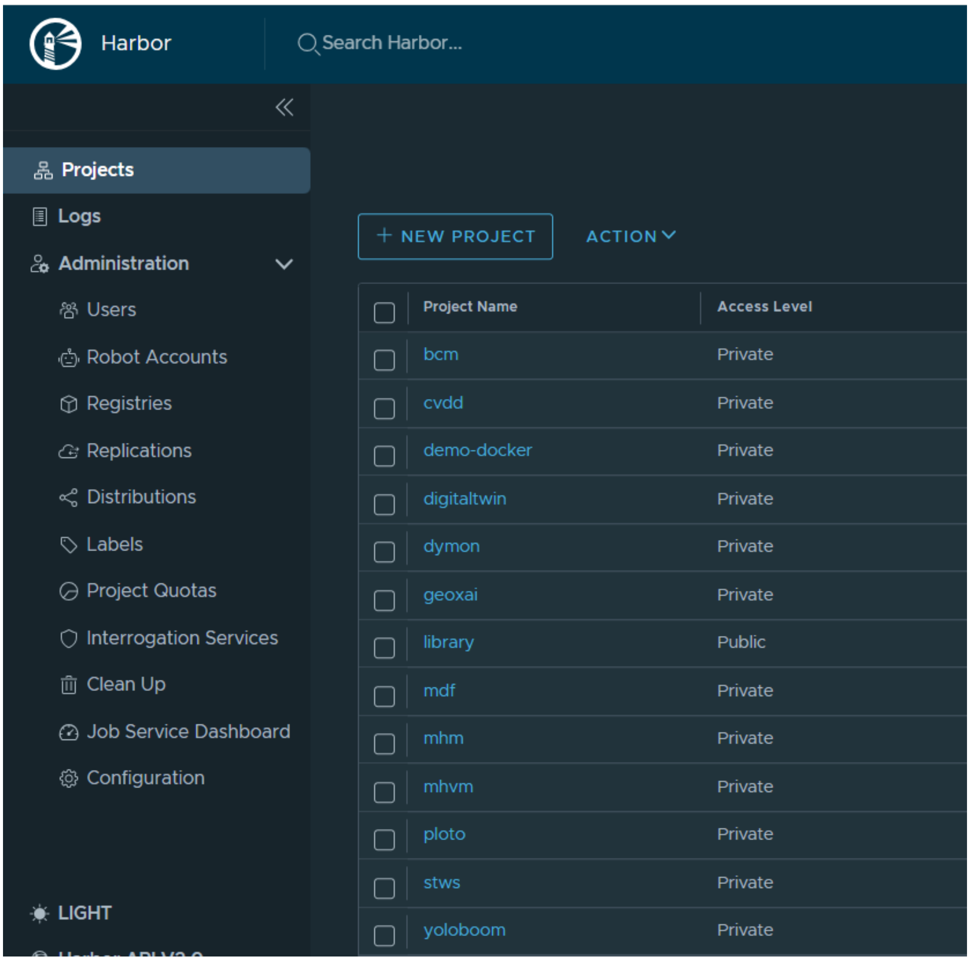


Figure 6: PLOTO component-specific projects in Harbor container image registry

Each partner has been assigned an account with access to the GUI and permissions to push and pull images within specific projects related to their developments. Within each project, partners can create various registry repositories to host their Docker images.

The Harbor GUI provides capabilities to:

- View the history of each image.
- View different versions of Docker images securely.
- Delete tags that are no longer needed.

A retention policy has been set to keep the latest 5 Docker image tags per project and per repository, and this policy is automatically executed every hour. Users can push and pull images either through Jenkins (using an automated pipeline) or from their own remote hosts.

4.2.5. Container Management tool

Portainer [11] is a lightweight container management User Interface (UI) that simplifies working with Docker. It provides an intuitive web-based interface that helps users manage containers, images, networks, volumes, and more without needing to use command-line tools. It incorporates RBAC features allowing teams to collaborate securely. In this context, fine-grained permissions can be defined based on users, teams, and environments. Portainer also provides real-time container logs and stats (CPU, memory, network), while also providing a built-in web terminal for container console access. Finally, it provides service management capabilities, allowing to start, stop, remove, or inspect containers.

A Portainer instance has been set up on top of PLOTO development servers (Figure 7) and is accessible through: https://portainer.ploto.rid-intrasoft.eu/

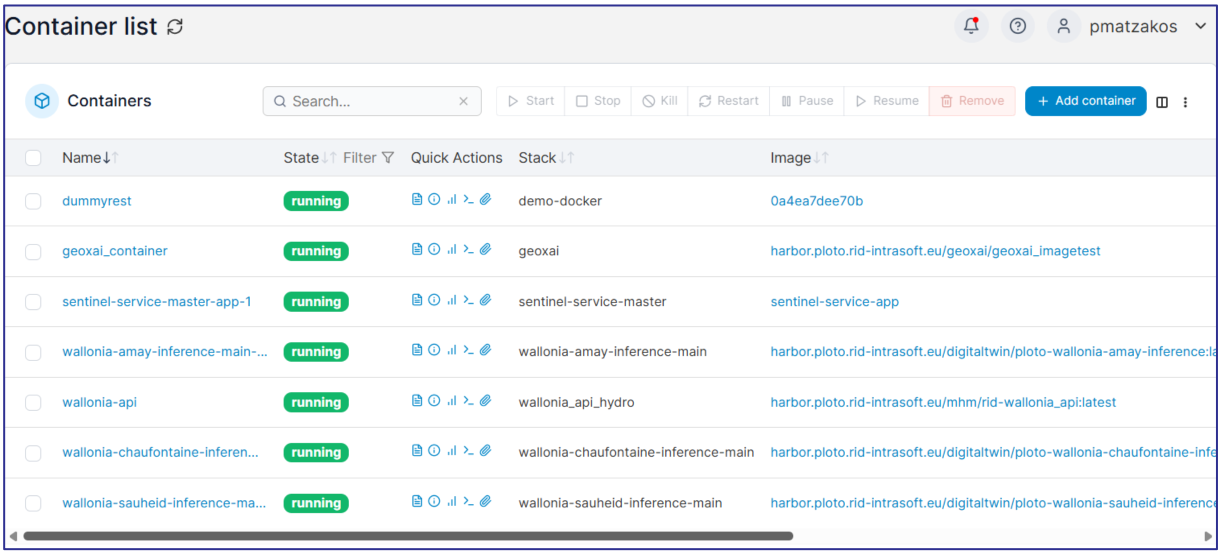


Figure 7: Portainer environment for managing PLOTO components' containers

4.3 Security features of PLOTO integration environment

The PLOTO CI/CD system incorporates a set of security features to safeguard the CI/CD infrastructure and services, as well as the deployed project's artifacts. In the following subsections these security features are described in detail.

4.3.1. Encryption & Secure communication over HTTPS

Access to the CI/CD services is secured with Hypertext Transfer Protocol Secure (HTTPS). Specifically, the continuous integration server (Jenkins), the artifact repository (Harbor Container Registry), as well as the container management tool (Portainer) are all secured with HTTPS. Data transmitted via HTTPS is protected by the Transport Layer Security (TLS) protocol, which offers three key layers of protection:

- **Encryption**: Protects data transmitted between parties by encrypting it, ensuring that communications are secure from eavesdropping. With HTTPS-secured services, users' interactions are safeguarded against interception, tracking, and data theft.

- **Data Integrity**: Guarantees that data remains unaltered and uncorrupted during transfer, whether through accidental means or malicious actions, allowing any tampering to be detected.
- **Authentication**: Verifies that users are connecting to the intended service, preventing man-in-the-middle (MitM) attacks and building user trust in the security of their communications.

Let's Encrypt [9] has been used as the Certificate Authority (CA) for the issuance of X.509 certificates offering TLS encryption to secure the CI/CD services. Moreover, the CI/CD services' certificates are configured to be automatically renewed through a cron job.

4.3.2. User authentication and Role-based Access Control

The services provided in the PLOTO CI/CD environment, including the PLOTO GitHub organization, the continuous integration server (Jenkins), and the container image registry (Harbor), are secured through user authentication and RBAC. RBAC ensures that all partners can access only their own and any agreed-upon collaborative development and testing projects/repositories.

4.3.3. VPN configuration

An OpenVPN [12] server has been configured via the PFSense software firewall [13] to enable PLOTO partners to access the deployed services and applications hosted on the virtual servers in Hetzner public cloud. By utilizing the OpenVPN server, external partners can connect to their deployed services through a secure and encrypted Virtual Private Network (VPN) tunnel.

For every user of the CI/CD environment, a unique SSL/TLS certificate has been generated and distributed. With this certificate and their login credentials, users can authenticate to the PLOTO VPN using the OpenVPN client application.

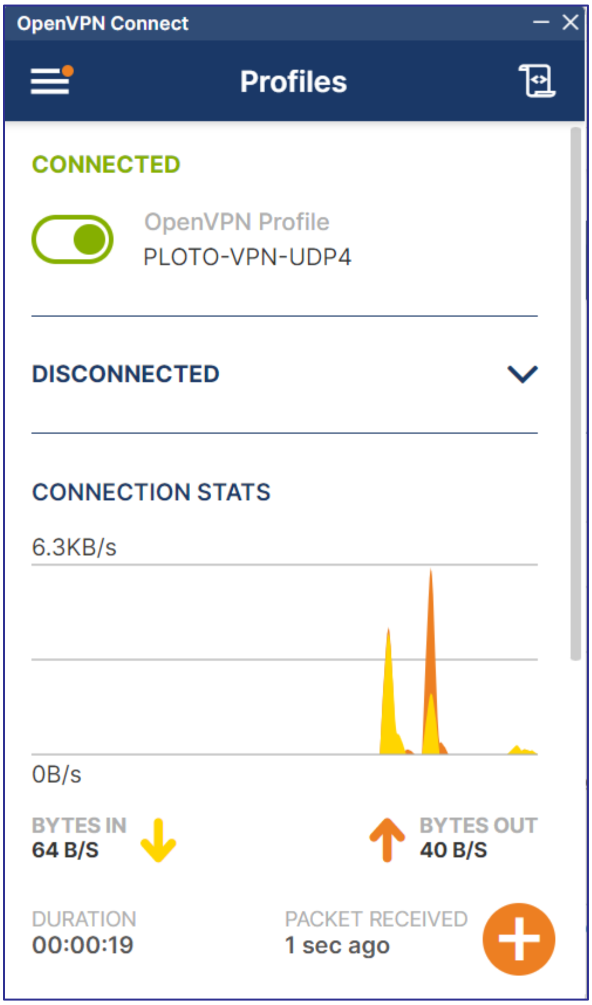


Figure 8: Access to PLOTO VPN

4.3.4. Firewall protection of CI/CD environment

A series of firewalls within the Hetzner cloud has been positioned in front of the PLOTO VM infrastructure to limit access to authorized external IPs/subnets and ports exclusively (Figure 9), as well as via the PLOTO VPN, while permitting internal server communication among the PLOTO servers.

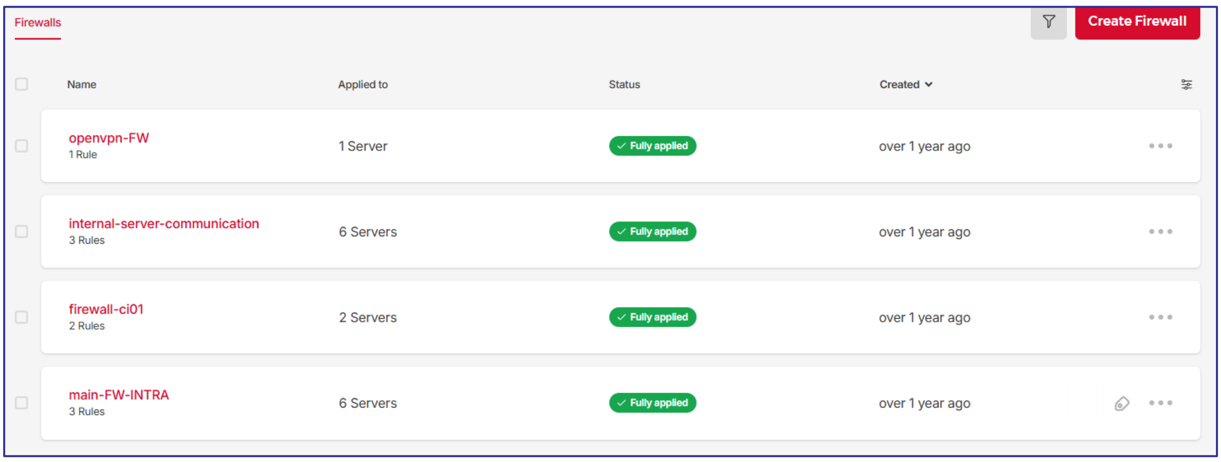


Figure 9: Hetzner firewalls to restrict access to PLOTO servers

4.3.5. SSH key-based authentication

Secure Shell (SSH) access for administering and managing the PLOTO VM infrastructure servers has been set up to use key-based authentication exclusively. This approach provides enhanced security compared to traditional password authentication methods. Although passwords are transmitted securely to the server, they can be susceptible to attacks if they are not sufficiently complex or lengthy. On the other hand, SSH public key authentication, which involves creating and storing a pair of cryptographic keys and configuring the servers to recognize and accept these keys, offers a more secure and reliable method.

5 PLOTO components and functional testing

In this section, a summary of each PLOTO component involved in the integrated platform is provided in subsections 5.1-5.14, highlighting its role in the overall system. This is accompanied by the description of each component's functional tests.

5.1 Middleware

The Middleware (MW) serves as a core component within the PLOTO platform, ensuring the seamless acquisition, storage, and management of data from various sources, including information systems, satellites, drones, and weather stations. Acting as a broker, it facilitates interoperability between diverse network interfaces, guaranteeing that all services and modules exchange data in a standardized format. By leveraging the FIWARE platform, the Middleware integrates time-series databases, enabling efficient storage, retrieval, and processing of raw and aggregated data. Through its modular and scalable architecture, it supports real-time data streaming, spatial data management, and complex event processing, enhancing decision-making across the platform. Its layered structure ensures high availability, adaptability, and seamless integration with external systems, making it a robust solution for managing heterogeneous data sources and enabling smart data fusion.

The architecture of the Middleware, fully detailed in [5], submitted at M28 (as well as in the first version of this deliverable, namely [4], submitted at M20), comprises **multiple layers** designed to ensure interoperability, security, and advanced analytics. The ***Abstraction Layer*** handles data ingestion and retrieval, incorporating tools such as NiFi and Kafka for efficient data streaming and transformation. The ***Data Storage Layer*** utilizes PostgreSQL with TimescaleDB and PostGIS to manage extensive time-series and spatial datasets, while the ***Interoperability Layer*** provides APIs that enable seamless integration between system components. Additionally, the ***Processing and Analytics Layer*** supports real-time data filtering, aggregation, and complex event processing, ensuring timely insights for informed decision-making. Security is maintained through the ***Security Layer***, leveraging FIWARE's Keyrock for access control and authentication.

In Table 1, a summary of the functional tests performed to ensure the expected functionality of the Middleware is provided.

Table 1: Middleware functional tests

Component's acronym	Test identifier	Objective	Execution Method (CI/CD pipeline, manual)
MW	MW_F001	Middleware stores data from sensors.	CI/CD pipeline
	MW_F002	Middleware sends data from sensors (upon request and upon receipt).	CI/CD pipeline
	MW_F003	Middleware retrieves historical data and accumulated values.	CI/CD pipeline
	MW_F004	Middleware sends notifications upon the receipt of new data.	CI/CD pipeline
	MW_F005	Middleware sends alerts after the fusion of data from multiple sources.	Manual
	MW_F006	Middleware stores geospatial data into the database.	CI/CD pipeline

Component's acronym	Test identifier	Objective	Execution Method (CI/CD pipeline, manual)
	MW_F007	Middleware sends upon request geospatial data.	CI/CD pipeline
	MW_F008	Middleware stores other data formats (e.g., static, HDF5).	Manual
	MW_F009	Middleware sends upon request other data formats (e.g., static, HDF5).	Manual
	MW_F010	Middleware receives notifications through the Kafka.	CI/CD pipeline

5.2 COP/IMS

ENGAGE IMS/COP offers a wide range of capabilities, including real-time situational awareness, multi-source data integration, incident tracking, and coordinated response planning. In the context of PLOTO project, ENGAGE IMS/COP has been adapted to support the monitoring and management of critical infrastructures alongIW (IWW). It delivers a Common Operational Picture tailored to PLOTO's specific requirements, enabling stakeholders to detect, assess, and respond to incidents impacting transportation, logistics, and safety across river corridors—facilitating timely and informed decision-making. Data from sensors and other available sources (e.g., software services) are integrated into the platform and presented to CI operators through a modern, intuitive Graphical User Interface. Further details on the design and supported functionalities of the COP/IMS component are to be provided in Deliverable *D6.6: Customized COP, IMS, and IWAT for IWW Operators*.

A summary of the functional tests performed for the COP/IMS component is provided in Table 2.

Table 2: COP/IMS functional tests

Component's acronym	Test identifier	Objective	Execution Method (CI/CD pipeline, manual)
COP/IMS	COP/IMS_F001	Incident creation on UI.	Manual
	COP/IMS_F002	The form prompts the user to fill in all mandatory fields.	Manual
	COP/IMS F003	Real-time data updates	Manual
	COP/IMS F004	Verify that users can filter and sort incidents based on criteria.	Manual
	COP/IMS F005	Incidents are accurately represented on a map with correct geolocation.	Manual
	COP/IMS F006	Sensor data is correctly displayed in the relevant sections.	Manual

5.3 OMS

The Operational Modelling System (OMS) is an advanced forecasting module developed by AUTH to predict extreme climate indicators and atmospheric stressors linked to specific hazards. Operating in nowcasting and forecasting modes, OMS employs a scheduler-driven infrastructure to manage real-time meteorological simulations. At its core, OMS integrates the mesoscale meteorological model MEMO, producing high-resolution hourly datasets of key atmospheric parameters, such as wind speed, temperature, cloud cover, and turbulence characteristics. The system is designed to simulate local meteorological phenomena, including mountain-valley winds, sea/lake breezes, and urban heat islands, enhancing its applicability in regional-scale environmental studies. To ensure reliable and timely predictions, OMS retrieves meteorological forecasts from the ICON-EU numerical weather prediction model, incorporating updates every three hours. An automated data pipeline facilitates the seamless processing and integration of these forecasts, optimizing the model's performance. By providing continuous, high-resolution meteorological insights, OMS supports risk assessment, disaster preparedness, and climate-sensitive decision-making.

A summary of the functional tests performed for the OMS component is provided in Table 3.

Table 3: OMS functional tests

Component's acronym	Test identifier	Objective	Execution Method (CI/CD pipeline, manual)
OMS	OMS_F001	OMS receives real-time data from the Middleware	Manual
	OMS_F002	OMS receives delayed data from the Middleware	Manual
	OMS_F003	OMS performs scheduled 3-hourly model run	Manual
	OMS_F004	OMS uploads 3-hourly snapshot to the Middleware	Manual
	OMS_F005	OMS uploads delayed 3-hourly snapshot to the Middleware	Manual

5.4 OMSDA

The Operational Modelling System Data Assimilation (OMSDA) module enhances the predictive capabilities of OMS by integrating real-time measurements with archived meteorological forecasts. By combining data fusion (assimilation) and computational reanalysis techniques, OMSDA refines meteorological fields across targeted areas, improving the accuracy of climate and hazard assessments. A key feature of OMSDA is its ability to continuously reconstruct meteorological conditions using sensor data and MEMO model outputs, ensuring high-resolution, dynamically updated atmospheric fields. To improve reliability, the system accounts for uncertainties in both measurements and model predictions, quantifying potential errors in the derived meteorological fields. The assimilation process is fully automated and integrated into the OMS operational workflow. A scheduler-driven mechanism ensures real-time corrections during nowcasts and forecasts, while a dedicated downloader module continuously retrieves observational data from weather stations across the pilot areas. By refining numerical outputs with high-quality measurements, OMSDA enhances environmental monitoring, risk assessment, and decision-making in climate-sensitive applications.

A summary of the functional tests performed for the OMSDA component is provided in Table 4.

Table 4: OMSDA functional tests

Component's acronym	Test identifier	Objective	Execution Method (CI/CD pipeline, manual)
OMSDA	OMSDA_F001	OMSDA receives real-time data from the Middleware	Manual
	OMSDA_F002	OMSDA receives delayed data from the Middleware	Manual
	OMSDA_F003	OMSDA performs range and sanity checks to retrieved sensor data	Manual
	OMSDA_F004	OMSDA performs scheduled 3-hourly model run	Manual
	OMSDA_F005	OMSDA uploads hourly snapshot to the Middleware	Manual

5.5 IWAT

IWAT is a key component designed to support the evaluation of the resilience of IWW. It acts as the central orchestrator in the context of the PLOTO risk assessment process, ensuring seamless connectivity between real-time data and analytical services. IWAT monitors the Kafka message broker for newly available data and, upon detection, automatically triggers predefined workflows that invoke the appropriate components responsible for identifying vulnerable infrastructure, planning cost-effective adaptation measures, and integrating data from various sensors and sources. Further to automated workflows, IWAT additionally supports manual execution of selected workflows, enabling the simulation of "what-if?" scenarios to assess potential risks and adaptation strategies. Through this functionality, IWAT enables comprehensive, data-driven decision-making. More detailed information regarding the IWAT component is available in [14] and [15].

A summary of the functional tests performed for the IWAT component is provided in Table 5.

Table 5: IWAT functional tests

Component's acronym	Test identifier	Objective	Execution method (CI/CD pipeline, manual)
IWAT	IWAT_F001	Show workflows on UI	Manual
	IWAT_F002	Execute workflow	Manual
	IWAT_F003	Show workflow execution results on UI	Manual

5.6 Multi-Hazard model

The Multi-Hazard model module focuses on flood hazard modelling in Use Case C. It includes automated precipitation data retrieval, multiple requests to the Digital Twin (DT) API to predict discharge time series at Amay, Sauheid, and Chaudfontaine, and an operational 1D hydraulic model. This model estimates water levels in the Meuse River and Albert Canal, considers potential breach discharges, assesses dike overtopping probability assumptions, and provides 2D flood propagation results. Further details are provided in [16] and [17].

A summary of the functional tests performed for the Multi-Hazard model is provided in Table 6.

Table 6: Multi-Hazard model functional tests

Component's acronym	Test identifier	Objective	Execution Method (CI/CD pipeline, manual)
Multi-hazard model	MHM_F001	Execute the downloading of precipitation and discharge sensors from Wallonia stations on a short period	Manual
	MHM_F002	Load data from a specific climatic model and a specific scenario at a specific hydrometry station	Manual
	MHM_F003	Make several requests to the EXUS API to produce a prediction timeseries of discharge at a specific hotspot with controlled inputs	Manual
	MHM_F004	Run a 1D hydrodynamic model	Manual
	MHM_F005	Interpolate flood maps to compute flood propagation time series maps	Manual

5.7 RAE

Task *T4.3 "Holistic Risk Assessment Propagation model"* aims at developing a comprehensive risk assessment framework by reviewing and refining existing methodologies to align precisely with the PLOTO project's needs. Leveraging this enhanced framework, the outcome of T4.3 is a software denoted as Risk Assessment Engine (RAE) that integrates and convolves: a) weather data (outcome of WP3), b) weather, hydrological, and seismic hazards (outcome of T4.1), c) exposure models (outcome of T4.2), and d) vulnerability data for assets and infrastructure at risk for both seismic and weather hazards (outcome of T4.2). The risk assessment propagation model will allow the computation of risk of individual assets subjected to multiple hazards, the propagation of risk at the system level, with the latter being formulated by individual assets as an integrated system (e.g., inland port), and finally the propagation of risk to the connected hinterland infrastructure (e.g., surrounding town or city). The propagation of risk from the asset-level to the system-level will offer key data among others to evaluate the operational status of the system (i.e., business operation continuity in T6.2), as well as inside and outside interactions when subjected to natural hazards (T4.1) and extreme weather events due to climate change (WP3). These enable the software to accurately model and quantify risk propagation across interconnected IWW assets and their hinterland. The engine supports the evaluation of both pre-event and trans-post event impacts.

Regarding WP4 high-level architecture, RAE relates to key WP4 modules, such as Business Continuity Model (BCM) [T4.4, T4.5], the Physical Impact (PI) [T4.3], and the Socioeconomic Impact (SI) [T4.4, T4.5]. Specifically, the system initiates with the Multi-Hazard Model [WP3 and T4.1], which generates quantified data on seismic, weather, and hydro hazards. This data is fed into the Physical Vulnerability Assessment [T4.2]. The resulting vulnerability data, combined with the BCM [T4.5], is then processed by the RAE [T4.3]. This engine synthesizes hazard and vulnerability data to quantify the PI [T4.3] and subsequently, through the BCM, determines the SI [T4.5]. RAE interacts with IWAT platform [T6.1] to display the results, which receives data for on-demand analysis for "what-if" scenarios. The Middleware [T6.4] facilitates the data exchange between these modules, ensuring seamless information flow.

The methodology and the insight regarding RAE are the content of Deliverable *D4.5: "Impact Assessment Model and Overall Organisational Resilience".*

A summary of the functional tests performed for the RAE component is provided in Table 7.

Table 7: RAE functional tests

Component's acronym	Test identifier	Objective	Execution Method (CI/CD pipeline, manual)
RAE	RAE_F001	Execute Impact assessment what-if seismic scenario	Manual
	RAE_F002	Execute Impact assessment what-if flood scenario	Manual
	RAE_F003	Execute Impact assessment what-is seismic scenario	Manual
	RAE_F004	Execute Impact assessment what-is flood scenario	Manual
	RAE_F005	Produce impact assessment results	Manual

5.8 GeoXAI

The GeoXAI module is an explainable, fully convolutional neural network designed for segmenting Sentinel-1 SAR images into flooded and non-flooded areas. It employs a compact encoder-decoder CNN architecture to achieve precise flood segmentation while maintaining computational efficiency. To enhance interpretability and build trust among stakeholders, the module integrates a Grad-CAM explainer, which provides visual insights into the model's decision-making process. This approach helps mitigate reliance on naive thresholding selection during post-processing, ensuring more transparent and reliable flood mapping. By bridging the gap between complex deep learning models and user interpretability, GeoXAI supports informed decision-making in disaster response and flood risk management. Additionally, its explainability features enable domain experts to validate model predictions, fostering confidence in real-world applications.

The GeoXAI, CVDD (Section 5.10) and YOLO-Boom (Section 5.12) tools have been packaged into Docker containers in a unified manner. The standard approach relies on the principle that each application receives one or more files as input and produces one or more files as output. The Middleware is responsible for retrieving all input files and expects the corresponding output files to be returned. The primary distinctions between these applications stem from the varying types of inputs and outputs, as well as the specific processing tasks performed by each tool.

The Docker structure for GeoXAI, CVDD, and YOLO-Boom modules, implemented by NTUA, is originally described in [18].

As depicted in Figure 10 the Docker container architecture ensures that the application has direct access only to the input/output folders, with no other form of external communication. Even the connection from the 'Organizer' to the application is strictly one-directional, solely responsible for initiating the triggering actions described below. The 'Organizer' serves as the core component of the API section, orchestrating all necessary operations to ensure the seamless functionality of the container.

The 'Kafka Topic Subscriber' module facilitates subscription to a predefined Kafka topic, continuously polling it for incoming messages. Upon receiving a message, this module extracts its contents and forwards them to the 'Organizer' for further processing. Additionally, the 'GeoServer/FTP Server Connector' module enables connections to GeoServer and an FTP server, allowing file exchanges when the corresponding paths are specified.

Finally, the API endpoints are categorised into three main groups—Functional Tests, Integration Tests, and Executional Endpoints—ensuring a structured and efficient routing of requests.

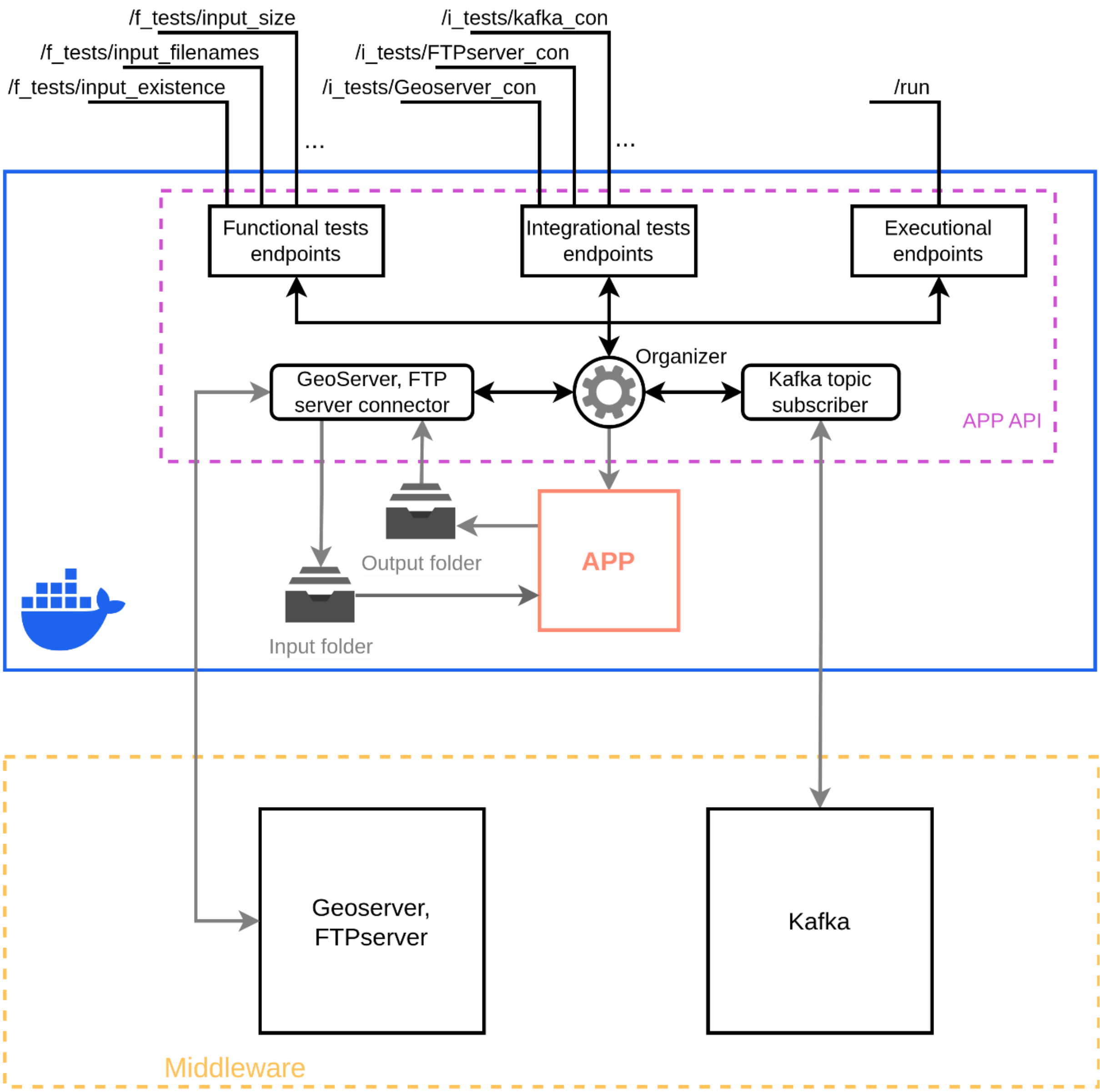


Figure 10: Docker container architecture for GeoXAI, CVDD, and YOLO-Boom components

The sequence diagram in Figure 11 provides a detailed representation of the processes occurring both inside and outside the Docker container when the 'run' command is executed. It demonstrates how the submodules interact under the coordination of the 'Organizer' to maintain a logical sequence of events, ensure proper task execution, and uphold workflow integrity, ultimately enabling the robust operation of the entire module. Additionally, the diagram outlines key communication flows between components, illustrating how data is exchanged and processed at various stages. It also highlights potential failure-handling mechanisms, enhancing the system's resilience and reliability during execution.

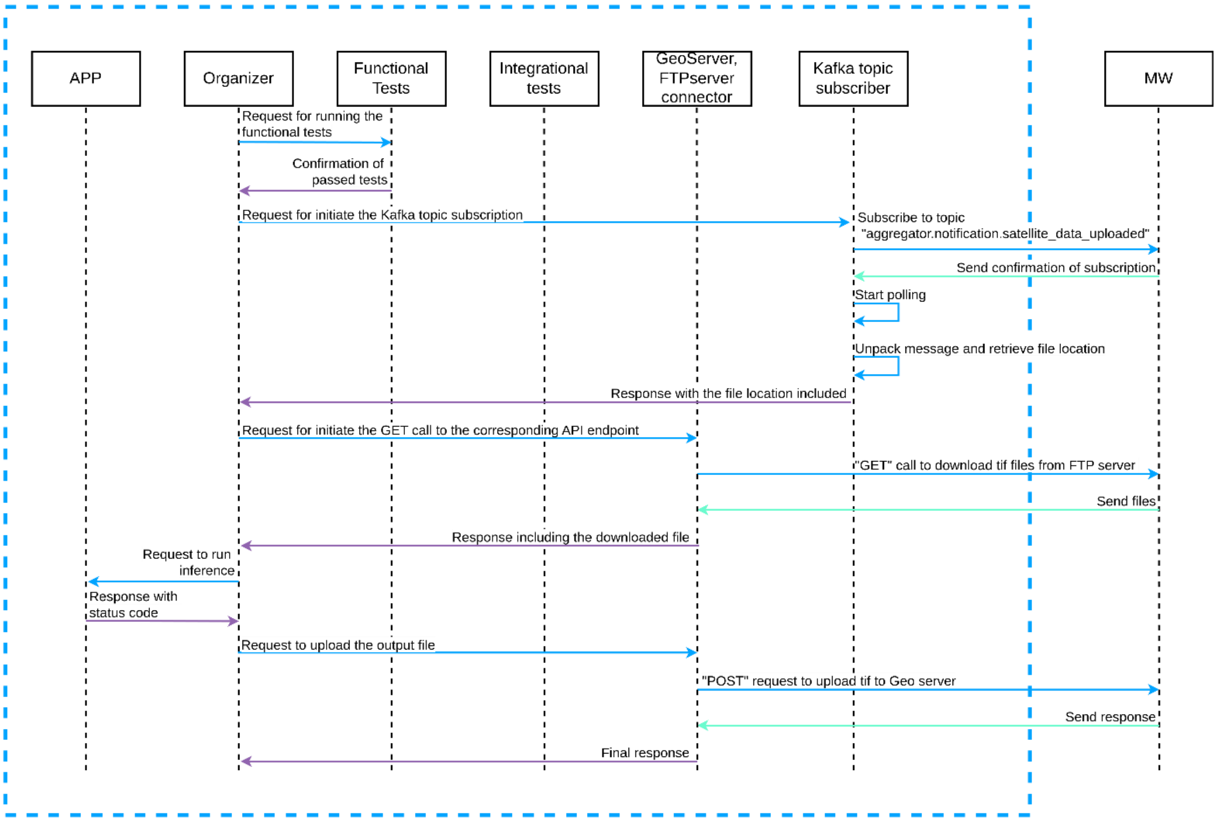


Figure 11: GeoXAI, CVDD, Yolo-Boom: Detailed representation of the processes relevant to the Docker container after running the command for execution.

A summary of the functional tests performed for the GeoXAI component is provided in Table 8.

Table 8: GeoXAI functional tests

Component's acronym	Test identifier	Objective	Execution Method (CI/CD pipeline, manual)
GeoXAI	GeoXAI _F001	The input image exists	CI/CD pipeline
	GeoXAI _F002	The input image includes geo-reference information	CI/CD pipeline
	GeoXAI _F003	The input image is at least of size 2500 x 2500	CI/CD pipeline
	GeoXAI _F004	The size of the output image is non-zero	CI/CD
	GeoXAI _F005	The .h5 file exists	CI/CD pipeline
	GeoXAI _F006	The API call to retrieve the input image is working	CI/CD pipeline

Component's acronym	Test identifier	Objective	Execution Method (CI/CD pipeline, manual)
	GeoXAI _F007	The API call to post the output image to the Middleware is working	CI/CD pipeline

5.9 Data Aggregator

Data Aggregator is a core service within the PLOTO platform, responsible for managing data originating from both satellite image repositories and the Ground Control Station (GCS), as well as overseeing UAV mission operations. For satellite data, the service periodically retrieves imagery from external repositories and uploads it to the PLOTO Middleware, ensuring that up-to-date remote sensing data is available to the platform. In terms of mission management, the Data Aggregator serves as the core service of GCS. It handles the creation of UAV missions and manages the data generated during UAV flights, ensuring it is uploaded to the PLOTO Middleware and made available to all PLOTO components. Additionally, the service is responsible for disseminating notification messages to other PLOTO components that need to consume the data as soon as it becomes available. The full capabilities of the Data Aggregator are detailed in [19].

A summary of the functional tests performed for the Data Aggregator component is provided in Table 9.

Table 9: Data Aggregator functional tests

Component's acronym	Test identifier	Objective	Execution Method (CI/CD pipeline, manual)
Data Aggregator	DA_F001	Periodic satellite data acquisition	Manual
	DA_F002	Satellite data storage on Middleware	Manual
	DA_F003	UAV mission creation	Manual
	DA_F004	UAV mission kml export (to be uploaded on the UAV)	Manual
	DA_F005	Upload UAV mission data on Middleware	Manual

5.10 CVDD

The Computer Vision for Damage Diagnosis (CVDD) module leverages state-of-the-art computer vision techniques to monitor inland water areas and their surroundings with high precision. It first employs the WatNet and DeepWaterMap algorithms [20] to accurately map inland water bodies. Once the water areas are identified, the CVDD module extracts shorelines in both raster and vector formats, ensuring detailed spatial representation. Finally, it detects and classifies river islets, generating comprehensive vector data for further analysis. By automating these processes, CVDD enhances environmental monitoring, supports hydrological studies, and aids in assessing changes in water bodies over time, contributing to more effective water resource management and disaster response planning.

A summary of the functional tests performed for the CVDD component is provided in Table 10.

Table 10: CVDD functional tests

Component's acronym	Test identifier	Objective	Execution Method (CI/CD pipeline, manual)
Computer Vision for Damage Diganosis	CVDD_F001	The 6-bands images exist as B02, B03, B04, B08, B11 and B12	CI/CD pipeline
	CVDD _F002	The 6-bands images include georeference information	CI/CD pipeline
	CVDD _F003	The size of the output riverside extraction images is non-zero	CI/CD pipeline
	CVDD _F004	The size of the output water polygons is not zero	CI/CD pipeline
	CVDD _F005	The size of the output water lines is not zero	CI/CD pipeline
	CVDD _F006	The size of the output islets polygons is not zero	CI/CD pipeline
	CVDD _F007	The size of the output islets lines is not zero	CI/CD pipeline
	CVDD _F008	The size of the output islets points is not zero	CI/CD pipeline
	CVDD _F009	The river gpks exists	CI/CD pipeline
	CVDD _F010	The SCL image exists	CI/CD pipeline

5.11 DYMON

Dynamic (Periodic) IWW monitoring (or DYMON) utilizes spatial, satellite and UAV data to analyse disaster impacts, such as abnormal water coverage and infrastructure damage. By integrating Sentinel-2 imagery (indices like NDWI, NDVI, BSI) and LiDAR data, the system detects flood-affected areas and assesses damage to buildings, roads, and railways through spatial intersection techniques. Geometry extraction methods, including raster vectorization and marching squares, refine flood boundaries for precise analysis. The tool also detects obstructions on infrastructure, such as fallen trees, using LiDAR-based surface error analysis and geometry buffering. While effective in pre-disaster assessment, the lack of post-hazard LiDAR data limits direct observation of damage, highlighting the need for comparative pre- and post-event datasets to improve accuracy in disaster monitoring. Methodology and findings of the approach are detailed in [19].

A summary of the functional tests performed for the DYMON component is provided in Table 11.

Table 11: DYMON functional tests

Component's acronym	Test identifier	Objective	Execution Method (CI/CD pipeline, manual)
DYMON	DYMON_F001	Pilot site agnostic test of the damaged infrastructure assessment tool performed on the sample data deployed within	CI/CD pipeline

Component's acronym	Test identifier	Objective	Execution Method (CI/CD pipeline, manual)
		CI/CD. If executable runs successfully without the errors, then assessment works. (T5.4)	
	DYMON_F002	Pilot site agnostic test of the deployed damaged infrastructure assessment tool performed on the sample data on UM platform over HTTPS. If RPC service with provided sample data executes and returns 200 OK with the JSON response then service works. (T5.4)	Manual
	DYMON_F003	Monitoring output endpoints should be accessible for all three pilot sites. (T5.4)	Manual

5.12 YOLO-Boom

The YOLO-Boom is an AI-powered system that utilizes the YOLO (You Only Look Once) object detection framework to identify and highlight structural defects in industrial buildings. By analysing input images, the module efficiently detects regions of interest associated with damages such as discontinuities (cracks), corrosion, and other potential structural issues. Engineered for both speed and accuracy, YOLO-Boom facilitates rapid inspection and assessment, making it a valuable tool for predictive maintenance and infrastructure monitoring. By automating defect detection, it enhances safety, reduces manual inspection efforts, and supports proactive decision-making in industrial asset management. Additionally, its scalability allows it to be deployed across various industrial settings, ensuring reliable and consistent monitoring over time.

A summary of the functional tests performed for the YOLO-Boom component is provided in Table 12.

Table 12: YOLO-Boom functional tests

Component's acronym	Test identifier	Objective	Execution Method (CI/CD pipeline, manual)
YOLO-boom (T5.5, NTUA)	YOLO-boom_F001	Verify that the model weights .pt file exist	CI/CD pipeline
	YOLO-boom _F002	Verify that the input image exists	CI/CD pipeline
	YOLO-boom _F003	Verify that the model weights correspond to the existing model architecture	CI/CD pipeline

5.13 Digital Twin

The deployment phase integrates the Machine Learning (ML) model's real-time discharge predictions as upstream boundary conditions into a 1D hydrodynamic model, which computes water levels and flow dynamics along the river channel. These 1D outputs are then spatially interpolated using a 2D floodplain model to generate maps that estimate both flood extent and propagation rates. The pipeline effectively bridges ML predictions with physics-based hydraulic modelling, creating a closed-loop system where discharge estimates directly inform flood impact projections for emergency management. The RAE ingests the ML model's hydrological predictions as a key input for its multi-hazard risk calculations. The ML-generated discharge forecasts provide the hydrological hazard component that RAE utilizes within its integrated risk assessment framework, which addresses a range of different hazard sources including seismic and meteorological events.

As flood events evolve, the RAE continuously processes the ML model's updated discharge predictions to dynamically reassess risk levels. RAE generates time-dependent risk projections that reflect the latest hydrological conditions predicted by the ML system.

A summary of the functional tests performed for the Digital Twin component is provided in Table 13.

Table 13: Digital Twin functional tests

Component's acronym	Test identifier	Objective	Execution Method (CI/CD pipeline, manual)
DT (T6.8, EXUS)	DT_F001	DT for Sauheid (Belgium use case) endpoint is live and the model predict the expected water discharge	CI/CD pipeline
	DT_F002	DT for Amay (Belgium use case) endpoint is live and the model predict the expected water discharge	CI/CD pipeline
	DT_F003	DT for Chaudfontaine (Belgium use case) endpoint is live and the model predict the expected water discharge	CI/CD pipeline
By the time that D7.1 is submitted it is expected that the DT for the Hungarian and the Romanian Use Case will be manual.			

5.14 BCA

The Business Continuity Assessment (BCA) component evaluates the functionality of IWW ports under disruption scenarios caused by natural hazards and climate change. It operates within the Business Continuity Module, which uses critical facility downtime and reduced functionality as input to estimate product delivery delays. The BCA integrates data from the MW [T6.4] about the potential damage of assets due to natural hazard and climate change and evaluates the business continuity of IWW ports. BCA interacts with the IWAT platform [T6.1] to visualize results and facilitate on-demand "what-if" analyses.

This component plays a crucial role in PLOTO Demonstration Use Case A: Romania, where Business Operation Models for five ports have been developed and analysed. The methodology and findings of the BCA are the content of Deliverable [21].

A summary of the functional tests performed for the BCA component is provided in Table 14.

Table 14: BCA functional tests

Component's acronym	Test identifier	Objective	Execution Method (CI/CD pipeline, manual)
BCA	BCA_F001	Scenario-based business operation [T6.2] impact assessment	Manual

6 Integration of PLOTO components

6.1 Integration timeline

The implementation of the PLOTO integrated platform follows an iterative approach, leading to two major releases as specified in the integration timeline shown in Table 15. As detailed in Section 2.1, the integration methodology of the PLOTO platform complies with CI/CD best practices to minimize errors during the integration and deployment phases of software components and to expedite software release.

Table 15: PLOTO integration timeline

Iteration	Integration activities	Pertners	Date
Init	Collection of component-specific SW/HW requirements to be accommodated from the integration framework	WP3-WP6 component providers	M13-M14
	Implementation of CI/CD platform and provision of technical workshop and associated tutorial material	INTRA	M15-M17
	Initial identification of bilateral integration points and end-to-end workflows	WP3-WP6 component providers, INTRA	M14-M17
	Main specification phase of bilateral integration points, functional and integration tests	All WP3-WP6 component providers, INTRA	M18-M22
	Main phase of PLOTO components functional and bilateral integration testing using automated CI/CD workflows or manual testing	**All WP3-WP6 component providers, INTRA**	**M18-M31**
First release of PLOTO integrated system	First release of PLOTO integrated platform including components' docker images in the private PLOTO registry and implemented testing and deployment workflows on CI/CD system	All WP3-WP6 component providers, INTRA	M32
	Updated specifications and execution of end-to-end (system-level) tests	All WP3-WP6 component providers, INTRA	M30-M33
	Development and testing updates based on feedback received from pilot executions (T7.2)	All WP3-WP6 component providers, INTRA	M34-M37
	Collection of testing execution results	All WP3-WP6 component providers, INTRA	M37
Final release of PLOTO			M38

Iteration	Integration activities	Pertners	Date
integrated system			

6.2 Integration Points identification

Integration points consist of pairs of PLOTO components that communicate with each other through clearly established interfaces and exchange data or information following specific sequence diagrams.

To document and identify the integration points for the PLOTO platform, an integration matrix inspired by the Design Structure Matrix (DSM) methodology was employed. DSM is a popular technique in systems integration used to model complex system architectures and their interactions. This matrix visually illustrates the connections between system elements, typically arranged with rows on the left and columns above. These elements often represent the individual components of a system. Similarly to an adjacency matrix in graph theory, DSM is utilized in systems engineering and project management for analysing complex systems, conducting system studies, and organizing project plans.

The integration matrix indicates the bilateral communication among pairs of components. Each software component within the PLOTO platform is represented by both a row and a column. Table 16 showcases the integration matrix. This matrix is formatted as upper triangular square matrix, with each entry denoting an integration point. The naming convention for each point follows the format X.Y, where X and Y refer to the respective row and column of the involved software components.

Table 16: PLOTO Integration matrix

Integration Points	**1. Middleware**	**2. Sensors**	**3. COP/IMS**	**4. OMS**	**5. OMSDA**	**6. IWAT**	**7. Multi-hazard model**	**8. Risk Assessment Engine**	**9.Geo XAI**	**10. Data Aggregator**	**11. CVDD**	**12. YOLO-Boom**	**13. Digital Twin**	**14. BCA**	**15. DYMON**
1. Middleware		1,2	1,3	1,4	1,5	1,6	1,7	1,8	1,9	1,10	1,11	1,12		1,14	1,15
2. Sensors															
3. COP/IMS															
4. OMS															
5. OMSDA															
6. IWAT								6,8							
7. Multi-Hazard model													7,13		
8. Risk Assessment Engine															
9. GeoXAI															
10. Data Aggregator															
11. CVDD															
12. YOLO-Boom															

Integration Points	1. Middleware	2. Sensors	3. COP/IMS	4. OMS	5. OMSDA	6. IWAT	7. Multi-hazard model	8. Risk Assessment Engine	9.Geo XAI	10. Data Aggregator	11. CVDD	12. YOLO-Boom	13. Digital Twin	14. BCA	15. DYMON
13. Digital Twin															
14. BCA															
15. DYMON															

As shown in Table 16, the majority of PLOTO components interface with the Middleware component. This is because, as described in sections 3 and 5.1, most of the interactions among PLOTO components are managed through the MW. Consequently, these interactions are summarized in Table 17 using triplets of components (i.e., Source_Component – MW – Consuming_Component). The detailed descriptions of the integration points listed in Table 17 are provided in the Annex (Section 9).

Table 17: Integration points description summary

Integration Point	Objective	Interface/ Protocol	Status
Sensors – Middleware – COP/IMS (1,2 & 1,3)	To retrieve and store data from sensors to the MW. Sensors' data (as .json files) and Notifications / Alerts (e.g., after data fusion) are sent to COP/IMS.	Apache Kafka, REST API over HTTPS	Complete
Sensors – Middleware – OMS (1,2 & 1,4)	To retrieve and store data from sensors to the MW. Sensors' data (as .json files) are sent to the Improved 3D model.	REST API over HTTPS (data are sent upon request).	Complete
OMSDA – MW – IWAT (1,5 & 1,6)	To upload updated results from the improved 3D model with Data Assimilation to the MW in a format appropriate for forwarding to IWAT (NetCDF, Json).	REST API over HTTPS (data are sent upon request).	Ongoing
OMSDA – MW – RAE (1,6 & 1,8)	To upload updated results from the improved 3D model with Data Assimilation to the MW in field and timeseries format that will be used by the RAE (CSV, Json).	REST API over HTTPS (data are sent upon request).	Ongoing
Multi-Hazard model-MW-RAE (1,7 & 1,8)	To generate flood propagation time-series via GeoTIFFs and to inform the RAE of the final timestep, to enable accurate selection and transmission of impact assessment results and alerts to COP/IMS. (GeoTIFFS, CSV, JSON)	REST API over HTTPS, Apache Kafka.	Ongoing
RAE-MW-IWAT (1,8 & 1,6)	To transfer impact assessment results (e.g., information on the assets at risk and impact of potentially catastrophic events) from the RAE, via MW, to the IWAT system for use in functions like impact assessment display by the user. -Data Formats (fromRAE): .hdf5.	SFTP, Notification sent to Kafka upon update.	Complete
RAE-IWAT (6,8)	Perform Risk Assessment to estimate the assets' damage state, repair cost, and downtime in case of a hazard.	REST API	Complete

Integration Point	Objective	Interface/ Protocol	Status
DA-MW-COP/IMS (1,10 & 1,3)	Receive Raster products and Notifications/Alerts (e.g., after data fusion).	REST API over HTTPS, Apache Kafka	Complete
DA-MW-GeoXAI (1,10 & 1,9)	The module polls the Apache Kafka topic aggregator.notification.satellite_data_uploaded to detect new image uploads. Upon receiving a Kafka message, it extracts the image path and issues a GET request to download the files. These remote sensing images are used to segment floods and predict at pixel level the flooded areas providing the flood boundaries extents. Raster Data Formats: .GeoTIFF Data retrieval (input for the GeoXAI module): geoTIFF	Apache Kafka, REST API over HTTPS	Complete
GeoXAI-MW-IWAT (1,9 & 1,6)	The developed products (satellite images) are sent back to the MW-Geoserver, using a POST request to the corresponding API endpoint. Data (from AI-based decision making): .geoTIFF	REST API over HTTPS	Complete
DA-MW-YOLOBoom (1,10 & 1,12)	The module polls the Apache Kafka topic aggregator.notification.uav.data.updated to detect new image uploads. Upon receiving a Kafka message, it extracts the mission id which indicates the location of the acquired UAS flight images. Then, it creates a GET request to download the files. The downloaded RGB images are used for the detection and cataloguing of construction flaws after the notification.	Apache Kafka , REST API over HTTPS	Complete
YOLOBoom-MW-COP/IMS (1,12 & 1,3)	Data. Notifications / Alerts (e.g., after data fusion). The drone RGB images providing the construction defects are stored into the MW and through the appropriate notification/ alert COP/IMS is informed. Data (from Dynamic (Periodic) IWW monitoring): .GeoJSON, .TIFF, .gpkg.	REST API over HTTPS. Apache Kafka	Ongoing

Integration Point	Objective	Interface/ Protocol	Status
DYMON-MW-COM/IMS (1,15 & 1,3)	To provide data from the flood damage detection tool as GeoJSON files containing flooded assets (lines and shapes).	REST API over HTTPS for data Apache Kafka for Notifications/Alerts	Complete
DA-MW-CVDD (1,10 & 1,11)	The module polls the Apache Kafka topic aggregator.notification.satellite_data_uploaded to detect new image uploads. Upon receiving a Kafka message, it extracts the image path and issues a GET request to download the files. The module provides data for inland water mapping by generating specific raster products that facilitate riverside extraction through the application of edge detection algorithms on WaterMaps on satellite data. In addition, vector products are produced to identify islets and to extract polygons, lines, and points related to the riverside.	REST API over HTTPS, Apache Kafka	Complete
CVDD-MW-COP/IMS (1,11 & 1,3)	The developed products are stored into the MW and through the appropriate notification/ alert COP/IMS is informed	REST API over HTTPS, Apache Kafka	Complete
BCA-MW-IWAT (1,14 & 1,6)	The Business Continuity Module requires the exposure dataset to be able to determine the functionality and usage of the assets at risk. The modulequeries the MW for the date of the exposure dataset and if a newer version is available, it is retrieved. Data and Notification/alerts are exposed to IWAT through MW.	Sftp for data retrieval, Apache Kafka for notifications/alerts.	Complete
DT-MHM-MW (7,13 & 1,7)	The MHM provides the flooded area in Use Case C according to the considered climatic scenario using a specific climatic model. The EUROCORDEX prediction of precipitation are timely downscaled (daily to hourly data), interpolate at existing hydrometry station and send to DT API (via multiple requests). DT API returns the prediction in terms of discharges at three hotspots:	REST API	Complete

Integration Point	Objective	Interface/ Protocol	Status
	Amay, Sauheid and Chaudfontaine. The latter are used as upstream boundary conditions in a hydrodynamic computation that allows to estimate floodplains area and its propagation rate.		

6.3 Integration tests

Table 18 provides a summary of the integration tests for all identified PLOTO integration points. The test identifiers follow this naming convention: COMPONENTX_MW_COMPONENTY_I00x or COMPONENTX_COMPONENTY, where COMPONENTX and COMPONENTY represent the involved components, and I00x is the incremental number assigned to each integration test for a given integration point. Similar to the functional tests described in Section 5, the Objective column offers a concise description of each test and its purpose, while the Execution Method column indicates whether the test is conducted automatically via the CI/CD system or manually.

Table 18: PLOTO integration tests per integration point

Integration point	Test identifier	Objective	Execution Method (CI/CD pipeline, Manual)
Sensors_MW_COP/IMS	Sensors_MW_COP/IMS_I001	Test successful retrieval of sensor data to the COP	Manual
Sensors_MW_OMS	Sensors_MW_OMS_I001	Test successful retrieval of sensor data in real time	Manual
	Sensors_MW_OMS_I002	Test notification and logging in case of data unavailability	Manual
	Sensors_MW_OMS_I003	Retrieve delayed or out-of-sequence sensor data in case real-time was unavailable	Manual
OMSDA_MW_IWAT	OMSDA_MW_IWAT_I001	Test successful retrieval of sensor data in real time	Manual
	OMSDA_MW_IWAT_I002	Test notification and logging in case of sensor data unavailability	Manual
	OMSDA_MW_IWAT_I003	Retrieve delayed or out-of-sequence sensor data in case real-time was unavailable.	Manual

Integration point	Test identifier	Objective	Execution Method (CI/CD pipeline, Manual)
	OMSDA_MW_IWAT_I004	Successful notification of IWAT on new Data available in the MW	Manual
OMSDA_MW_RAE	OMSDA_MW_RAE_I001	Test successful retrieval of OMSDA hazard dataset through MW to RAE	Manual
MHM_MW_RAE	MHM_MW_RAE_I001	Test successful retrieval of MHM hazard dataset through MW to RAE	Manual
RAE_MW_IWAT	RAE_MW_IWAT_I001	Test successful retrieval of RAE risk output data through MW to IWAT	Manual
RAE_IWAT	RAE_IWAT_I001	Successful trigger of RAE, from IWAT	Manual
	RAE_IWAT_I002	Successful RAE results retrieval from IWAT	Manual
DA_MW_COP/IMS	DA_MW_COP/IMS_I001	Successful data upload from DA to MW	Manual
	DA_MW_COP/IMS_I002	Successful notification push for new data uploaded on MW	Manual
DA_MW_GeoXAI	DA_MW_GeoXAI_I001	Test that data are uploaded properly on the MW	Manual
	DA_MW_GeoXAI_I002	Check whether GeoXAI module gets triggered (receives Apache Kafka message) when a new GeoTIFF appears in theMW	CI/CD pipeline
	DA_MW_GeoXAI_I003	Check whether the received Apache Kafka message includes the URL of the corresponding API endpoint	CI/CD pipeline
	DA_MW_GeoXAI_I004	Check whether the API responds to “GET” request	CI/CD pipeline
GeoXAI_MW_IWAT	GeoXAI_MW_IWAT_I001	Check whether the output GeoTIFF image exists	CI/CD pipeline

Integration point	Test identifier	Objective	Execution Method (CI/CD pipeline, Manual)
	GeoXAI_MW_IWAT_I002	Check whether the API responds to "POST" request	CI/CD pipeline
DA_MW_YOLO-Boom	DA_MW_YOLO-Boom_I001	Check whether YOLO-Boom module gets triggered (receives Apache Kafka message) when a new image (e.g., .jpg file) appears in the MW	CI/CD pipeline
	DA_MW_YOLO-Boom_I002	Check whether the received Apache Kafka message includes the URL of the corresponding API endpoint	CI/CD pipeline
	DA_MW_YOLO-Boom_I003	Check whether the API responds to "GET" request	CI/CD pipeline
YOLO-BOOM_MW_COP/IMS	YOLO-Boom_MW_COP/IMS_I001	Check whether the output JSON file exists	Manual
	YOLO-Boom_MW_COP/IMS_I002	Check whether the API responds to "POST" request	Manual
	YOLO-Boom_MW_COP/IMS_I003	Show notification on IMS/COP	Manual
DYMON_MW_COP/IMS	DYMON_MW_COP/IMS_I001	GET - Return 200 OK with JSON response that contains multiple GeoJSON objects for each output type.	CI/CD pipeline
	DYMON_MW_COP/IMS_I002	Show notification on IMS/COP.	Manual
DA_MW_CVDD	DA_MW_CVDD_I001 (STWS,	Check whether CVDD module gets triggered (receives Apache Kafka message) when a new data package (GeoTIF, GPKGs, TXT files) appears in the Middleware	CI/CD pipeline
	DA_MW_CVDD_I002	Check whether the received Apache Kafka message includes the URL	CI/CD pipeline

Integration point	Test identifier	Objective	Execution Method (CI/CD pipeline, Manual)
		of the corresponding API endpoint	
	DA_MW_CVDD_I003	Check whether the API responds to "GET" request	CI/CD pipeline
CVDD_MW_COP/IMS	CVDD_MW_COP/IMS_I001	Check whether the output data package exists	CI/CD pipeline
	CVDD_MW_COP/IMS_I002	Check whether the API responds to "POST" request	CI/CD pipeline
	CVDD_MW_COP/IMS_I003	Show results on IMS/COP UI	Manual
BCA_MW_IWAT	BCA_MW_IWAT_I001	Successful retrieval of exposure dataset from MW	Manual
	BCA_MW_IWAT_I002	Trigger BCA and present the results on IWAT	Manual
DT_MHM_MW	DT_MHM_MW_I001	Check whether the Geoserver in the MW, is reachable for posting the TIFF files generated by MHM	Manual
	DT_MHM_MW_I002	Check whether the FTP server in the MW, is reachable for posting the TIFF files generated by MHM	Manual

6.4 Next integration steps

As highlighted in section 6.1, the next steps towards the final release of the PLOTO integrated system will focus on the finalization of the remaining integration points and corresponding tests, and more importantly on the completion of end-to-end testing workflows that will validate the readiness of the platform for the pilots execution. As described in Table 15, and in collaboration with the activities that will be carried out in the context of *T7.2 - Demonstration implementation,* the plan then is to receive feedback from the initial pilots' execution. This feedback will be used for any required development and testing updates of the PLOTO components and consequently the platform, that will be reflected on its final release (i.e., D7.2).

7 Customization and full characterization of the measures to be selected for simulation and implementation

The objective of Task 2.5 *"Customization and full characterization of the measures selected for simulation and implementation"* is to propose, customize, and fully characterize adaptation and mitigation measures for successful large-scale implementation as part of WP7 (T7.2, T7.3). The work, with a specific focus on IWW and connected infrastructure, aims to re-design and optimize pre-identified solutions to enhance their applicability to the PLOTO platform.

This section presents the selected solutions by adapting design parameters to suit specific application needs. Nature-based solutions will be deployed and monitored in real-world scenarios, including ports, railway stations, or riverbanks. These measures will ensure the solutions are practical and scalable.

7.1. Methodology used in customization and full characterization of the measures to be selected for simulation and implementation

Task 2.5, "Customization and full characterization of the measures selected for simulation and implementation," was conducted over months 12-30 of the PLOTO project as part of WP2 "End-User Requirements and Platform Design".

This report aims to present some aspects related to the necessity of mitigation measures in each one of the use cases studied. To achieve this goal, Danubius University's Galati experts team developed and adhered to a structured actions plan, that spans months 12 to 30 and includes the following key activities:

- A1 (M12–M16): Analyse selected solutions related to the multi-hazard model.
- A2 (M17–M20): Identify hazards and assess the specific hazard impact for each PLOTO Use Case.
- A3 (M20–M26): Define parameters.
- Final Report (M26–M30): Integrate findings into work package 7 deliverables.

7.1.1. Selected solutions related to the multi-hazard model

This work builds upon the findings from previous Deliverables developed under WP2, especially *D2.2: "Definition of the requirements use cases, and system specifications"* and *D2.3: "Geographic data and services inventory and definition and interdependencies of the selected measures for improved resilience"*. Significant work was performed in Chapter 3 - PLOTO end-user needs and good practices analysis of adaptation and mitigation measures of D2.1. These results provided a foundation for our approach, guiding the decision to analyse the selected solutions according to the modules.

Climate change is expected to increase the frequency and severity of extreme weather events, leading to higher transportation costs, operational disruptions, and infrastructure damage, as highlighted in Table 19. However, IWW transport emerges as a unique and sustainable alternative to counter these impacts. Compared to other transport modes, IWW transport significantly reduces external costs. By bolstering the resilience of IWW infrastructure, significant strides towards the European Union's sustainable transport goals can be made. This will ensure the continuous operation of IWW under adverse weather conditions and the minimization of long-term climate-related risks, thereby playing a crucial role in achieving the EU's broader objectives for sustainability and resilience.

Table 19: Key findings from D2.1. and effects

Key findings based on the end-user questionnaire applied in D2.1. **Who:**	
Water Levels as the Most Relevant Weather Issue	**Low water levels** affect shipment capacity and routes. **Intense precipitation** impacts loading operations. **High water levels** influence storage capacity.
Forecasting Needs	Advanced forecasting (more than one week) for **low water levels and icing** is crucial for better operational planning.
Information Accuracy and Notification Frequency	Water level information is already accurate; additional hourly notifications are not necessary.
Port-related and Traffic Information Gaps	There is a lack of timely and detailed **port-related and traffic information**, which is critical for smooth operations.

The future of IWW transport depends on its seamless integration into multimodal transport systems and its ability to adapt to the impacts of climate change on navigation. Despite the Danube's historical significance, its usage has declined since 1990. Reversing this trend, requires substantial investments in infrastructure maintenance, development, and modernization to boost cargo volumes and improve efficiency.

Additionally, regional differences in vessel characteristics, port conditions, and waterway features, such as locks, highlight the need for tailored approaches. Cross-border cooperation, exemplified by Hungary's water agreements, plays a pivotal role in mitigating risks and fostering resilient, sustainable water transport systems across the region.

Table 20: Mitigation measures possible solutions

Mitigation Measures possible solutions **Who:**	
Enhanced Forecasting and Early Warning Systems	Implement predictive tools to provide **longer-term forecasts** (one week or more) for critical water levels and icing conditions. AI-based systems can further improve accuracy and lead time for alerts, enabling better logistical planning.
Dynamic Loading and Storage Protocols	Develop adaptable protocols for **loading and storage** that can adjust to water level changes. Automated sensors and real-time monitoring systems can help adjust operations dynamically, especially during periods of intense precipitation or high-water levels.
Improved Digital Information Systems	Address the gap in **port-related and traffic information** by integrating digital platforms that provide real-time updates on port operations, vessel traffic, and other relevant conditions. These platforms should

Mitigation Measures possible solutions **Who:**	
	be accessible to all operators and updated frequently to enhance decision-making.
Infrastructure Improvements	Invest in infrastructure to manage low water levels, such as adjustable docks or improved storage facilities. This will reduce operational disruptions when water levels fluctuate.

Furthermore, effective digital information systems are critical to enhancing the economic competitiveness and sustainability of IWW transport, as can be seen in Table 20. Current meteorological networks often fall short to meet the specific needs of IWW users, necessitating innovative solutions to reduce uncertainty. The integration of automated, AI-based systems can enhance operational efficiency and minimize human errors. While location-related information is less critical, the strong demand for electronic port announcements underscores the urgent need for improved digital communication and data management systems tailored to IWW operations.

7.1.2. Assess the specific hazards impact

This step involves a comprehensive identification of potential hazards relevant to each PLOTO Use Case, followed by an in-depth assessment of their specific impacts. The process begins with analysing various environmental, infrastructural, and operational risks that could affect IWW and associated infrastructure. For each Use Case, hazards are evaluated based on their probability of occurrence.

To re-design and tailor, the selected solutions for application within the PLOTO context, an impact assessment was conducted, as illustrated in Figure 12. Particular attention was given to addressing hazards identified as having medium to high impact, such as seismic, weather, and hydro hazards. These hazards were prioritized due to their potential to significantly impact the resilience and operational efficiency of IWW infrastructure and interconnected systems.

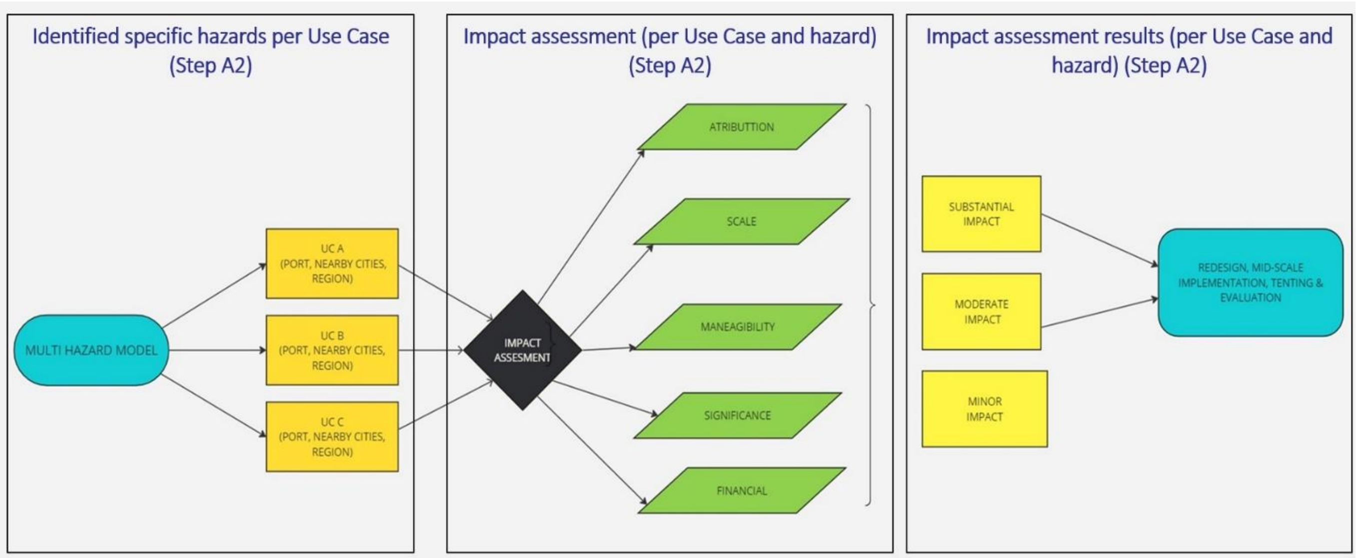


Figure 12: Impact assessment methodology

This approach enables a thorough evaluation of the vulnerabilities within each Use Case and supports the development of tailored adaptation strategies to enhance resilience against the identified risks.

These hazards were analysed using an Excel-based assessment tool, which can be seen in Figure 13, where each use case representative partner provided detailed inputs. This tool allowed for a systematic evaluation of risks, considering expert knowledge specific to each region.

USE CASE A									
		Impact attribution		Impact scale	Impact manageabilit	Impact significance		Financial impact	
		Will the hazard analysed create new impacts or accelerate/exacerbate existing impacts?	If the impact is existing/created or accelerated, how can be ranged?	How individuals and communities will feel the impacts of the hazard analyzed?	How resilient are the end-users? (vulnerable to adverse impacts)	How signifiant are the potential impacts ?	Is mitigation available to manage or reduce the potential impacts?	Potential costs	Potential benefits
Seismic Hazard	Port	exist	low	medium	medium	low	yes	medium	medium
	Nearby cities	exist	medium	medium	medium	low	yes	medium	medium
	Region	exist	low	medium	medium	low	yes	medium	medium
Weather Hazard	Port	exist	high	high	medium	medium	yes	medium	medium
	Nearby cities	exist	high	high	medium	high	yes	high	medium
	Region	exist	medium	high	medium	high	yes	high	medium
Hydro Hazard	Port	exist	low	low	low	low	yes	low	low
	Nearby cities	exist	low	low	low	low	yes	low	low
	Region	exist	low	low	low	low	yes	low	low

Figure 13: Impact assessment tool

A structured template, as presented in Figure 13, was developed to assess the impact of identified hazards for each PLOTO use case. This template provided a standardized methodology for end-users to evaluate hazards based on their potential impacts and the likelihood of occurrence. The template included multiple drop-down lists to ensure consistency and clarity in responses.

End-users were required to fill in the following fields:

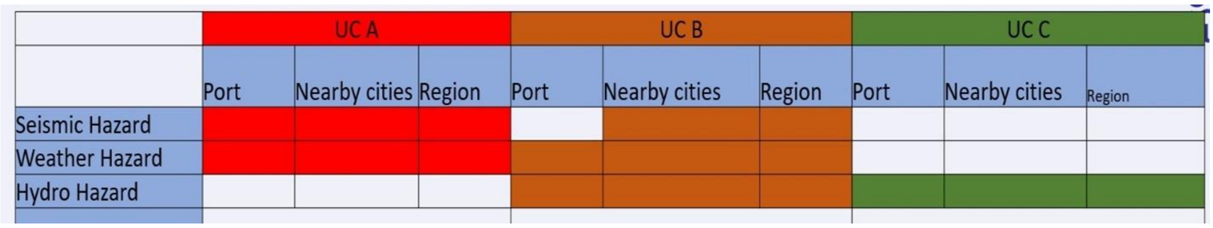
1. **Impact Assessment**:
   - **Low, Medium, High, N/A (not available)**: This drop-down menu allowed end-users to categorize the severity of each hazard's impact on the infrastructure, operations, and systems within the Use Case.
2. **Stage of Hazard Impact**:
   - **Create, Exists, Accelerate, N/A (not available)**: This section focused on how the hazard develops or affects the system. End-users could select whether the hazard **creates** a new risk, **exists** as an ongoing concern, or **accelerates** an already existing issue, helping to determine each hazard's dynamic influence.
3. **Mitigation Capability**:
   - **Yes, No, N/A (not available)**: This field assessed whether effective mitigation measures are in place. End-users indicated whether the area can manage or reduce the hazard's impact.

Mitigation measures are preventive actions designed to reduce the impact of a risk or in some cases eliminate it entirely. The decision to implement these measures depends on the effort and resources required; however, the primary criterion remains the **impact reflected** on the Use Cases and the associated assets. Possible decisions for measures are shown below:

- **No-action**: The possibility to happen and impact is low, no immediate action is taken apart from designing the corrective action.

- **Avoid**: An example would be not to develop a certain action that imposes a risky feature but remains within the task scope.
- **Protection**: Impact and likelihood are moderate to high therefore all actions aiming to minimize impact or remove risk altogether.

This systematic approach ensures that hazard impacts are measured consistently across all Use Cases, providing a clear and actionable dataset for further analysis and the customization of mitigation measures, as can be seen in Figure 14, within the PLOTO project.



	UC A			UC B			UC C		
	Port	Nearby cities	Region	Port	Nearby cities	Region	Port	Nearby cities	Region
Seismic Hazard									
Weather Hazard									
Hydro Hazard									

Figure 14 Impact assessment results

PLOTO Use Case Leaders played a critical role in providing the relevant information for the hazard assessment. After the template was distributed, each Use Case Leader completed the required fields considering the specific conditions, risks, and vulnerabilities in their region or operational area. Their expertise and localized knowledge ensured that the assessment accurately reflected the real-world challenges faced by IWW infrastructure and associated systems.

The information provided by the Use Case Leaders offered valuable insights into the types of hazards prevalent in their respective regions, such as extreme weather events, water level fluctuations, or seismic events. The Use Case Leaders contributed a comprehensive view of each Use Case's risks by categorizing these hazards based on the **impact severity**, **stage of hazard development**, and **existing mitigation capabilities**.

7.2. Specific impacts and mitigation solutions

This section focuses on gathering and analysing detailed information related to specific parameters such as population, critical infrastructure, and assets for hazards assessed as having **substantial** or **moderate** impacts. These parameters are essential to understand the broader implications of hazards on both the human and material resources within each PLOTO use case.

The evaluation includes:

- **Population**: Assessing the potential impact on local populations, including the risk to lives, livelihoods, and communities.
- **Assets**: Identifying critical infrastructure such as ports, waterway facilities, transport networks, and other essential systems that may be affected.
- **Economic and Social Impacts**: Estimating the financial losses and social disruptions caused by these hazards, including damage to infrastructure and long-term economic consequences.

This analysis is critical for prioritizing mitigation efforts, ensuring that resources are directed towards protecting the most vulnerable populations and assets and reinforcing the resilience of key infrastructures within the PLOTO project regions.

7.2.1. Use Case A – Romanian Danube Region

The Danube region in Romania is highly significant in terms of population, assets, and economic and social impacts. Romania has about 30% of its population living within the Danube River Basin, representing nearly 6 million people. Many towns and villages along the Danube are vulnerable to flooding, with cities like **Galați** (around 250,000 inhabitants) and **Brăila** (over 200,000 inhabitants) at heightened risk of economic and social disruption due to weather-related hazards. A large portion of the population in the Danube region relies on agriculture, which is highly sensitive to flooding, drought, and other extreme weather events. In addition, the entire lower Danube is close to the seismogenic Vrancea zone, which has produced significant events in the past 100 years.

The assets in the Romanian cities of Galați, Brăila, and Tulcea are strategically important for regional and national infrastructure, particularly due to their location along the Danube River. However, these cities face significant exposure to seismic and weather-related hazards, which may disrupt operations, damage infrastructure, and impact local economies.

Historical data shows that **major floods along the Danube in 2006** caused an estimated **€400 million** in damages in Romania alone, impacting urban and rural infrastructure. In recent years, low water levels in the Danube have caused significant disruptions. Both Galați and Brăila suffered significant economic losses due to damaged warehouses, storage facilities, and transportation links. For example, during the **2018 drought**, the water levels dropped so much that shipping had to be halted for weeks, leading to substantial economic losses in transport and trade. Droughts in the Danube region can result in major agricultural losses, with **2015** and **2020 droughts** in Romania causing a drop in crop yields by up to **50%**, impacting national food security and rural livelihoods.

Floods have displaced thousands of people in the past. For example, the **2006 flood** saw thousands of Romanians evacuated from towns and villages near the Danube, disrupting lives, education, and health services. Repeated floods have also damaged critical infrastructure such as roads, bridges, and water treatment plants, leading to long-term disruptions in public services and transportation. The social and economic well-being of Danube communities depends on cross-border cooperation, as the Danube flows through multiple countries. Romania collaborates with countries like Hungary, Serbia, and Bulgaria under the framework of the **Danube Commission** and other international agreements to manage water levels, prevent flooding, and ensure navigability.

In the past 100 years, the area has suffered at least 10 seismic events of magnitude higher than M6, with one of the most destructive being the 1977 M7.5 event, which caused the loss of 1,578 lives and 32,900 buildings damaged or destroyed. Although most of the effects were felt in Bucharest, it remains a clear threat for the Danube ports.

An important part of this sector is represented by the mouths of the Danube, where the river outflows in the Black Sea. This area is usually subjected to high navigation traffic, since it represents the main navigation exit from the Danube to the Black Sea, being at the same time the southern gate of the seventh Pan-European transportation corridor that links the Black Sea to the North Sea via the Rhine-Main-Danube navigation system. This navigation sector is subjected to specific conditions that may induce significant navigation hazards. Thus, besides the higher wind conditions, characteristic in this area, the interactions between the sea waves and the currents caused by the river outflow may induce significant enhancements of the wave heights affecting very often the safety navigation.

The Danube region in Romania plays a pivotal role in the country's economy, supporting trade, agriculture, and transport. However, it is highly vulnerable to weather-related hazards such as flooding, droughts, and extreme temperatures. These hazards threaten the region's population and have substantial economic and social impacts, underlining the importance of targeted mitigation measures to protect both people and assets.

The Danube region in Romania is particularly vulnerable to seismic and weather-related hazards that can significantly impact IWW transport, infrastructure, and local communities. Some of the key weather hazards in this region include (Table 21):

Table 21: Cause – Impact – Mitigation measures for PLOTO Use Case A: Romania

Weather-related and seismic hazards	Cause	Impact	Mitigation	Will be tested ?
Flooding	Heavy rainfall, snowmelt, and storms can cause the Danube River and its tributaries to overflow, leading to widespread flooding.	Flooding can disrupt navigation on the river, damage infrastructure (ports, embankments), and threaten communities along the riverbanks. The 2006 flood, for instance, caused severe damage to port infrastructure, residential areas, and agricultural lands. Both Galați and Brăila suffered significant economic losses due to damaged warehouses, storage facilities, and transportation links.	Improved flood forecasting systems, flood defenses, and adaptable infrastructure are critical for mitigating this risk.	YES
Low water levels	Prolonged dry periods, higher temperatures, and decreased precipitation due to climate change.	Low water levels can hinder the navigability of the Danube, reducing shipping capacity, forcing vessels to carry lighter loads, and causing economic losses due to transport delays.	Measures like dredging, water level monitoring, and improving forecasting models to predict low water periods are essential.	YES
Extreme temperatures	Increasing temperatures, especially in summer, can affect river ecosystems, water quality, and cause	During winter, icing on the Danube can block navigation routes, freezing operations in ports and affecting	Enhanced icebreaking operations, as well as designing infrastructure resilient to temperature extremes, are	NO

Weather-related and seismic hazards	Cause	Impact	Mitigation	Will be tested ?
	additional stress on infrastructure.	regional trade and transport.	necessary to minimize the impact.	
Storms & Strong winds	Sudden weather changes and storms are becoming more frequent due to climate change.	Strong winds can cause navigation difficulties for vessels and damage port facilities. Storm surges can exacerbate flooding and erosion along the riverbanks. In Tulcea, where many assets are directly on the river and delta region, high winds can further damage natural levees and protective barriers, increasing the risk of floodwaters reaching populated and economically vital areas.	Advanced warning systems, storm-resistant infrastructure, and emergency response protocols for severe weather conditions can help reduce damage.	PARTIALLY (only impact to port operations and shipping)
Droughts	Extended periods of low rainfall due to climate variability.	Droughts can lower water levels in the Danube, affecting both navigation and water supply for agriculture and industry. It also increases the risk of wildfires in the surrounding areas. Tulcea, with its proximity to the Danube Delta, is particularly vulnerable as drought conditions can lead to ecosystem stress, affecting fish	Drought management plans, water conservation strategies, and improved water usage efficiency in agriculture and industry are essential. Additionally, collaborative water management efforts with upstream and downstream regions can help	PARTIALLY (only impact on navigation)

Weather-related and seismic hazards	Cause	Impact	Mitigation	Will be tested ?
		stocks, biodiversity, and water quality.	manage water levels during prolonged dry periods.	
Extreme waves and complex coastal processes at the mouths of the Danube	Wave-current interactions	The interactions between the sea waves and the currents induced by the river outflow may significantly enhance the wave heights, which can often affect safety navigation.	Developing a complex wave modeling system, defining the most dangerous patterns. This system can also produce and significantly enhance the wave heights, which can often affect and forecast products concerning the environmental matrix associated with the navigation conditions in this coastal area.	NO
Earthquake	Seismic motion	Damages to port infrastructure due to seismic motion	Retrofitting, insurance, and business continuity planning to redistribute operations	YES

The Danube region faces a combination of weather-related and seismic hazards that pose significant risks to infrastructure, navigation, and ecosystems. Proactive mitigation, adaptive planning, and targeted testing are essential to enhance resilience and ensure operational continuity.

These hazards in the Danube region require a multi-faceted approach to adaptation and mitigation, focusing on real-time monitoring, infrastructure reinforcement, and cross-border cooperation to manage risks effectively.

7.2.2. Use Case B – Budapest Port

Port of Budapest is an intermodal hub, where several transportation modes (waterway, railway, and road) are connected, and the Liszt Ferenc international airport is located nearby (app. 15 km away). Within the Port area, multiple types of companies (e.g., shipping company, logistics providers, warehouse operator, transport operator, concrete factory, medical laboratory, retailers with pick-up point, grain storehouse) co-operate establishing logistics chains. One of the main activities of BSZL Zrt. is the letting of property, within the frameworks of which, approximately 24 ha free space, more than

157 000 square meters of covered storage space and more than 10 000 square meters of office space is used by more than 70 lessees operating within the Port. The location and layout properties of the Freeport have made it possible to build an internationally significant port and logistics centre. The container terminal operating within the Freeport - which is the only container terminal with water-side loading capacities in the country - provides an opportunity to join the expectedly prospering container traffic of the Danube. The Port has a humongous development area in the immediate vicinity of the city centre, in one of the most promising areas of Budapest expected to undergo significant developments; this development area offers further development opportunities in addition to the port-logistics functions.

Table 22: Cause – Impact – Mitigation measures for PLOTO Use Case B: Hungary

Weather-related and seismic hazards	Cause	Impact	Mitigation	Will be tested?
Flooding	Heavy rainfall, snowmelt, and storms can cause the Danube River and its tributaries to overflow, leading to widespread flooding.	It blocks vessel traffic. High risk of flooding in spring and at the beginning of summer. Flooding can stop navigation on the Danube, endanger the port infrastructure (railway, cranes). The 2013 flood was the most dangeres in Hungary.	Improved flood forecasting systems, flood defenses, and adaptable infrastructure like mobile dyke are critical for mitigating this risk.	YES
Low water levels	Prolonged dry periods, higher temperatures, and decreased precipitation due to climate change.	Low water levels can hinder the navigability in some sections of the Danube, reducing loading capacity, it	The regulation of Hungarian section of the Danube is essential.	YES

Weather-related and seismic hazards	Cause	Impact	Mitigation	Will be tested?
		blocks vessel traffic and causing economic losses.		
Extreme cold and ice situation	Increasing temperatures, especially in winter, can cause traffic block on the Danube.	During winter, icing on the Danube can block navigation routes, freezing operations in ports, and causing economic losses.	Enhanced icebreaking operations, as well as designing infrastructure resilient to temperature extremes, are necessary to minimize the impact.	NO
Extreme heat	Increase in maximum summer temperatures can cause additional stress on infrastructure.	The extreme heat in summer can have an impact on pavement deterioration and on risk of fires around or within the Port area. It affects railway traffic and shipping.	Reduced speed at the railway traffic is required.	PARTIALLY (Only effects on port equipment and operations)
Storms & Strong winds	Sudden weather changes and storms are becoming more frequent due to climate change.	Strong wind can affect docking and un/loading processes. It may cause service interruptions on adjacent	Advanced warning systems help reduce damage of crains and warehouses.	YES

Weather-related and seismic hazards	Cause	Impact	Mitigation	Will be tested?
		infrastructures (e.g., power supply).		
Earthquake	Seismic motion	Damages to port infrastructure due to seismic motion	Retrofitting, insurance, and business continuity planning to redistribute operations	YES

7.2.3. Use Case C – Wallonia

Use Case C covers a region extending from Liège (Belgium) to the Belgian-Dutch border, with the river Meuse crossing the area from south to north. While the river is navigable further upstream, it is not in the Use Case area. Therefore, a man-made channel (Albert canal), running parallel to the river, serves as waterway in the Use Case area. The Meuse continues northward into the Netherlands, while the Albert Canal heads towards the sea harbor of Antwerp, which makes it critical for commercial shipping in the region.

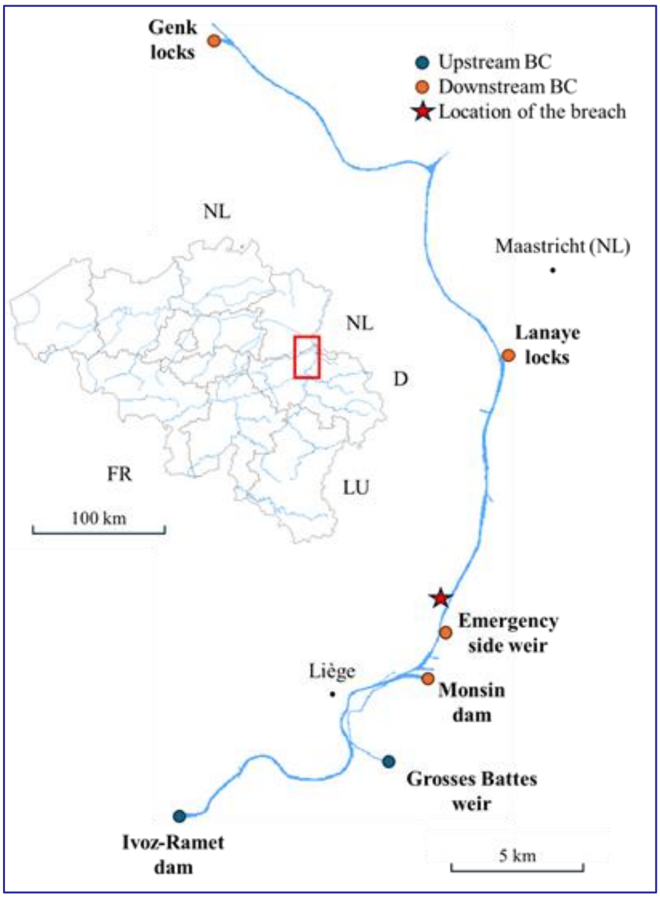


Figure 15: Domain and boundary conditions of the numerical simulations planned for the Use Case C [22]

The main risk considered in Use Case C is flooding, possibly involving breaching of dikes. The floodplains in the Use Case area are densely urbanized, housing not only residential areas but also numerous critical infrastructures such as hospitals, police stations, and water management facilities (e.g., wastewater treatment plants, service reservoirs for the supply of drinking water, etc.). Several settlements, including Lanaye, Lixhe, Nivelle, and Visé, are located between the river Meuse and the Albert Canal, and are particularly exposed as their elevation is below the water level in the Albert Canal. These areas are protected by the dikes of the Albert Canal, but during extreme floods, such as the 2021 European mega-flood, these protective barriers were nearly overtopped. In such an event, breaching of the structure is likely in sections where the dikes are earth-filled or partially reinforced with gabions.

In addition to the settlements located between the river Meuse and the Albert canal, other settlements along the Meuse and the Albert Canal also face significant flood risk. The substantial vulnerability of these regions justifies the consideration of mitigation measures to protect critical infrastructures and reduce the potential damage in the event of a dike failure.

The Liège region is highly vulnerable to flood hazard, which can significantly affect IWWtransportation IWW, infrastructure, and local communities. The root cause of these flood risks is primarily heavy rainfall, exacerbated by the region’s topography, which is characterized by relatively narrow valleys and a steep topography of the tributaries. Unlike other areas of the Meuse basin or European regions, where wider floodplains offer space for flood storage, the limited width of the valley in the Liège region restrict the capacity of floodplains to damp flood waves. Furthermore, the river channels have been heavily modified for navigation, reducing their ability to buffer floodwaters [23].

The impacts of flooding would be far-reaching. Flood events disrupt IWWnavigation, damage critical infrastructure (ports, embankments), and pose a threat to settlements located along the riverbanks. The 1993 Meuse flood severely damaged port facilities and residential areas, while the 2021 flood caused devastating impacts in the Vesdre valley, just upstream of Liège, where over 3,000 homes were affected. Both floods resulted in significant economic losses due to damage to warehouses, storage facilities, and transportation networks. In total, the Walloon region is home to approximately 450 km of IWW infrastructure, vital for international trade. Around one-third of the traffic on these waterways is in transit, with the remainder dedicated to imports and exports.

Apart from the damage to economic assets, flooding in this region poses a threat to human life. In the event of a breach in a dike, a range of critical infrastructure such as residential areas, industry, wastewater treatment plants, gas stations, and facilities dealing with hazardous materials could be inundated. Additionally, emergency services, including police and fire stations, medical facilities, and schools, would be at risk. This area is also home to various sensitive infrastructure elements, including railway networks, highways, and power lines, which could collapse during a flooding event. Such widespread disruption highlights the need for effective flood mitigation strategies.

The mitigation measures proposed within this report include two main approaches.

- First, protection at the household level will be considered by assuming that barriers are implemented for dry-proofing buildings for an outdoor water depth up to 0.5 m.
- Second, the possibility of controlling the Monsin weir (which controls the water level in the Use Case area) to avoid (or delay) dike overtopping and/or reduce breach discharge.

These mitigation measures (Table 23), combined with early warning systems, will be modelled in the What-If scenarios to evaluate their effectiveness in reducing flood impacts. Overall, these measures aim to reduce flood hazard and buildings vulnerability along the Albert Canal and in surrounding urban areas like Oupeye andVisé, hence improving resilience against future flood events.

Table 23: Cause – Impact – Mitigation measures for PLOTO Use Case C: Belgium

Weather hazards	Cause	Impact	Mitigation	Will be tested?
Flooding	Heavy rainfall, snowmelt, and storms can cause the Albert Canal to overflow, leading to widespread flooding.	Flooding can impact more than 4,000 buildings adjacent to the Albert Canal	Improved flood forecasting systems, Controlling the Monsin weir, flood barriers for buildings.	YES

8 Conclusions

Leveraging the outcomes of the components under development (WP3-WP6) and the PLOTO architecture (T2.3), this deliverable details on the ongoing integration efforts aimed at creating the PLOTO integrated platform and laying the groundwork for its instantiations tailored to the requirements of various pilot execution scenarios. To this end, the integration methodology was discussed first. A comprehensive overview of the CI/CD system established to support developers in their development and testing activities and expedite the PLOTO software release is provided. Furthermore, the integration plan throughout the duration of T7.1 is described, followed by the definition of testing activities for the PLOTO components, and the identification and specification of integration points for the first release of the PLOTO integrated system.

The outcomes of T2.5 have been thoroughly detailed, addressing the specific requirements of PLOTO Use Cases A, B, and C. This includes an in-depth analysis of the methodology employed for the customization and full characterization of the corresponding measures, followed by a discussion of the specific impacts and mitigation solutions identified for each use case. The majority of these mitigation solutions is planned to be tested in the context of PLOTO.

D7.2: "The PLOTO Integrated System and Acceptance tests final version" will provide detailed and updated information regarding the final release of the PLOTO integrated platform that will be released to support the final pilot execution scenarios and validations. In this context, as described in Section 6.4, one of the main aspects that will be completed and reported in D7.2 is the validation of the readiness of the PLOTO system through the execution of representative end-to-end testing scenarios which are linked to the pilot execution scenarios.

References

[1] PLOTO Consortium, "D2.2: Definition of the Requirements, Use Cases and System Specifications final version," 2023.

[2] "Github projects," [Online]. Available: https://docs.github.com/en/issues/planning-and-tracking-with-projects/learning-about-projects/about-projects. [Accessed 06 04 2025].

[3] "Slack," [Online]. Available: https://slack.com/. [Accessed 07 04 2025].

[4] PLOTO Consortium, "D6.4: Middleware and DF services 1st release," 2024.

[5] PLOTO Consortium, "D6.5: Middleware and DF services final release," 2024.

[6] "Hetzner cloud," [Online]. Available: https://www.hetzner.com/. [Accessed 14 04 2025].

[7] "Github," [Online]. Available: https://github.com/. [Accessed 14 04 2025].

[8] "Jenkins CI," [Online]. Available: https://www.jenkins.io/. [Accessed 14 04 2025].

[9] "Letsencrypt CA," [Online]. Available: https://letsencrypt.org/. [Accessed 14 04 2025].

[10] "Harbor container image registry," [Online]. Available: https://goharbor.io/. [Accessed 14 04 2025].

[11] "Portainer container management," [Online]. Available: https://www.portainer.io/. [Accessed 14 04 2025].

[12] "OpenVPN," [Online]. Available: https://openvpn.net/. [Accessed 14 04 2025].

[13] "PFSense firewall," [Online]. Available: https://www.pfsense.org/. [Accessed 14 04 2025].

[14] PLOTO Consortium, "D6.1 Integrated HRAP environment first release," 2024.

[15] PLOTO Consortium, "D6.2 Integrated HRAP environment final release," 2024.

[16] PLOTO Consortium, "D4.2: Advanced and Reliable Models for Natural and Man-Made Hazards final version," 2024.

[17] PLOTO Consortium, "D4.4: Multi-Hazard Vulnerability Modules for IWW and connected hinterland infrastructures final version," 2024.

[18] PLOTO Consortium, "D5.2: Dynamic link to hazard and resilience assessment," 2025.

[19] PLOTO Consortium, "D5.1: Assessment along the IWW corridor and a surrounding disaster-affected area, using multi-source remote sensing data," 2024.

[20] L. Isikdogan, "'Seeing Through the Clouds With DeepWaterMap," *IEEE Geoscience and Remote Sensing Letters,* vol. 17, pp. 1662-1666, 2020.

[21] PLOTO Consortium, "D6.3: Business Continuity Models, Adaptation Strategies, Standard Response Procedures," 2025.

[22] V. Schmitz, "Modelling of fluvial dike breaching due to overtopping (Doctoral dissertation, University of Liège)," 2024. [Online]. Available: https://orbi.uliege.be/handle/2268/325201.

[23] V. Kitsikoudis, B. P. J. Becker, Y. Huismans, S. Erpicum, M. Pirotton and B. Dewals, "Discrepancies in flood modelling approaches in transboundary river systems: Legacy of the past or well-grounded choices?," *Water Resources Management,* vol. 34, p. 3465–3478, 2020.

9 Annex: Integration Points Specifications

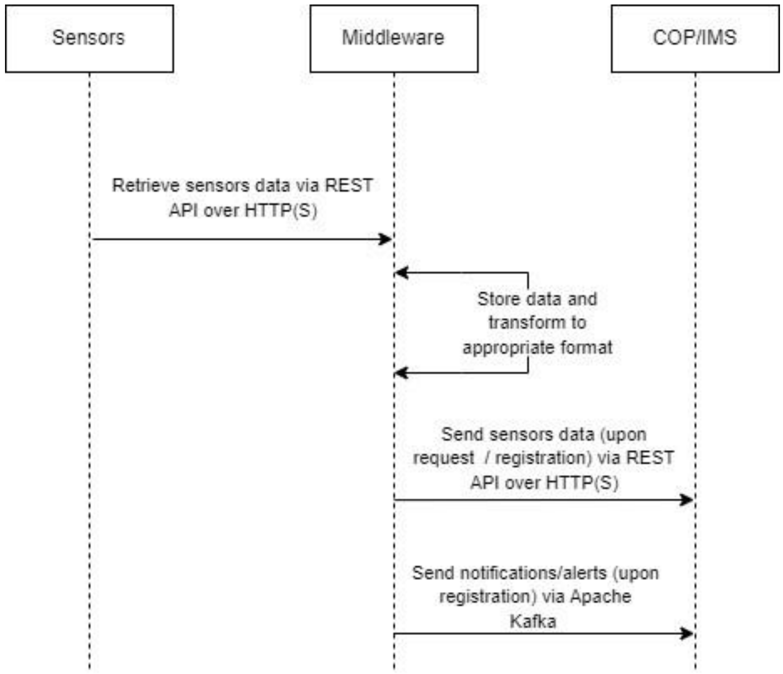
In the following, the detailed description of the integration points identified and summarized in Section 6.2 is provided**.**

9.1. Sensors – Middleware – COP/IMS

Integration Point Description	
Identifier	*(1,2) Sensors – Middleware*
Involved organizations	RISA, End-users
Integration Point Purpose	To retrieve and store data from sensors.
Protocol	The protocol employed for sensors (Use Case A, B and C) is REST API over HTTPS.
Status	*Completed*

Integration Point Description	
Identifier	*(1,3) Middleware – COP/IMS*
Involved organizations	RISA, STWS
Integration Point Purpose	• Sensors' data (as .Json files) is sent to COP/IMS. • Notifications / Alerts (e.g., after data fusion).
Protocol	The protocols employed for: • Sensors data (Use Case A, B and C) are Apache Kafka and REST API over HTTPS (data are sent upon request / registration). • Notifications / alerts produced (e.g., after fusion of data) is Apache Kafka (upon registration).
Status	*Completed*

Consolidated sequence diagram:



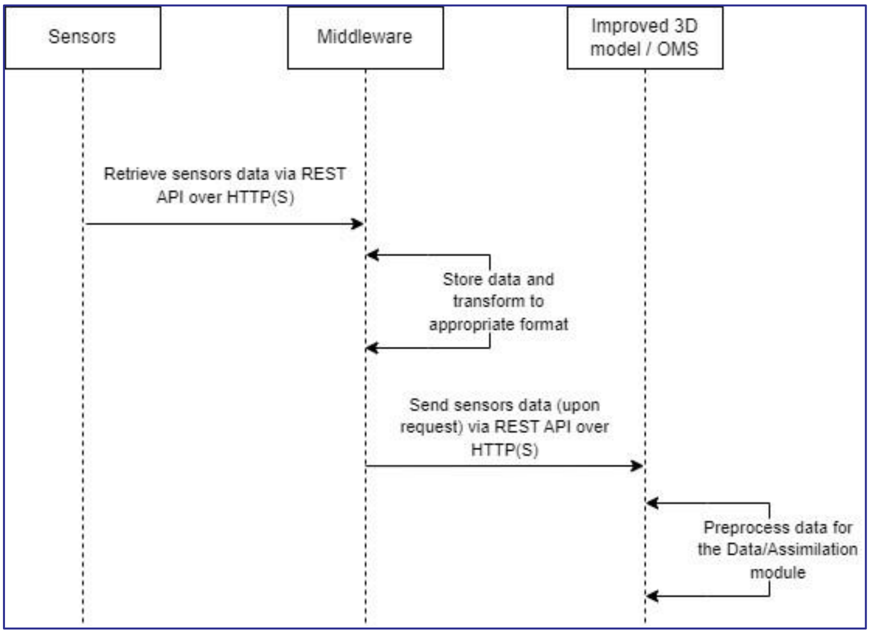
9.2. Sensors – Middleware – OMS

Integration Point Description	
Identifier	*(1,2) Sensors – Middleware*
Involved partners	RISA, End-users
Responsible	RISA
Integration Point Purpose	To retrieve and store data from sensors.
Protocol	The protocol employed for sensors (Use Case A, B and C) is REST API over HTTPS.
Status	*Completed*

Integration Point Description	
Identifier	*(1,4) Middleware – OMS*
Involved partners	RISA, AUTH
Responsible	AUTH
Integration Point Purpose	Sensors' data (as .Json files) is sent to the Improved 3D model.
Protocol	The protocol employed for sensors data (Use Case A, B and C) is REST API over HTTPS (data are sent upon request).

Status	*Completed*

Consolidated sequence diagram:

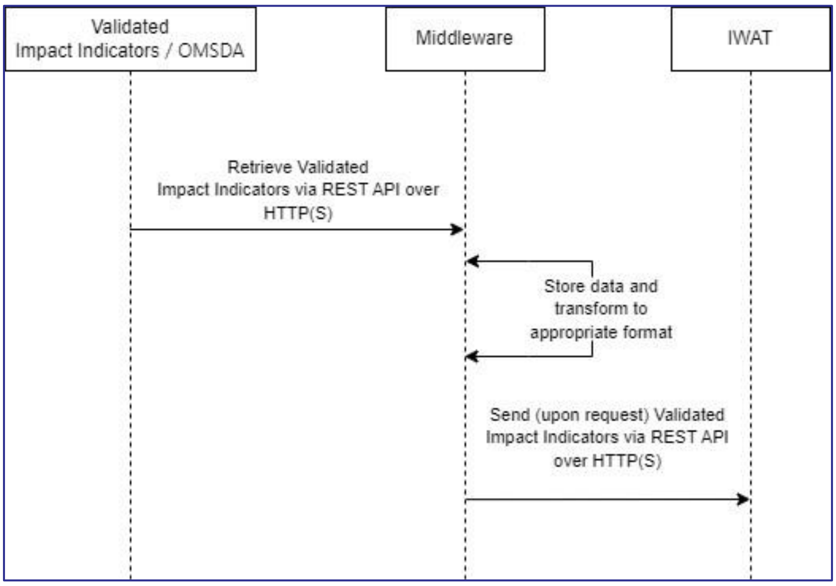


9.3. Validated Impact Indicators (OMSDA) – Middleware – IWAT

Integration Point Description	
Identifier	*(1,5) Validated Impact Indicators – Middleware*
Involved partners	RISA, AUTH
Responsible	AUTH
Integration Point Purpose	To upload updated results from the improved 3D model with Data Assimilation to the MW in a format appropriate for forwarding to IWAT (NetCDF, Json). Upload will typically be triggered on an hourly or 24-hourly (daily) basis, providing nowcasting and next-day forecast meteorological fields appropriate for visualization, as well as pointwise information at selected station locations.
Protocol	The protocol employed for Dynamic Data Formats: .NetCDF, is REST API over HTTPS.
Status	*Ongoing*

Integration Point Description	
Identifier	*(1,6) Middleware – IWAT*
Involved partners	RISA, STWS
Responsible	RISA
Integration Point Purpose	Post Dynamic Data Formats (from Validated Impact Indicators): .NetCDF, .PNG to IWAT
Protocol	The protocol employed for Dynamic Data Formats: .NetCDF, (from Validated Impact Indicators) is REST API over HTTPS (data are sent upon request).
Status	*Ongoing*

Consolidated sequence diagram:



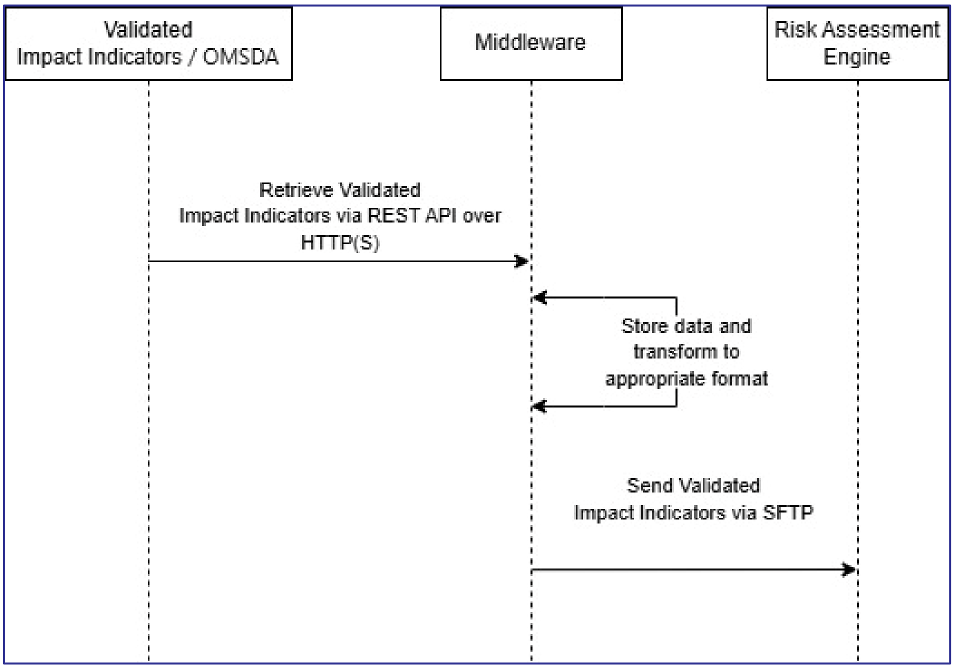
9.4. Validated Impact Indicators (OMSDA) – Middleware – Risk Assessment Engine

Integration Point Description	
Identifier	*(1,5) Validated Impact Indicators – Middleware*
Involved partners	RISA, AUTH
Responsible	AUTH
Integration Point Purpose	To upload updated results from the improved 3D model with Data Assimilation to the MW in field and timeseries format that will be used by the RAE (CSV, Json). Upload will typically be triggered on an hourly or 24-hourly (daily) basis, providing nowcasting and next-day forecast meteorological fields as well as pointwise information at selected station locations.

Protocol	The protocol employed for Dynamic Data Formats: .NetCDF, is REST API over HTTPS.
Status	*Ongoing*

Integration Point Description	
Identifier	*(1,8) Middleware - Risk Assessment Engine*
Involved partners	RISA, SoreCC
Responsible	RISA
Integration Point Purpose	Send Dynamic Data Formats (originating from Validated Impact Indicators): .NetCDF to RAE
Protocol	The protocol employed for aforementioned data is SFTP (data are available).
Status	*Ongoing*

Consolidated sequence diagram:



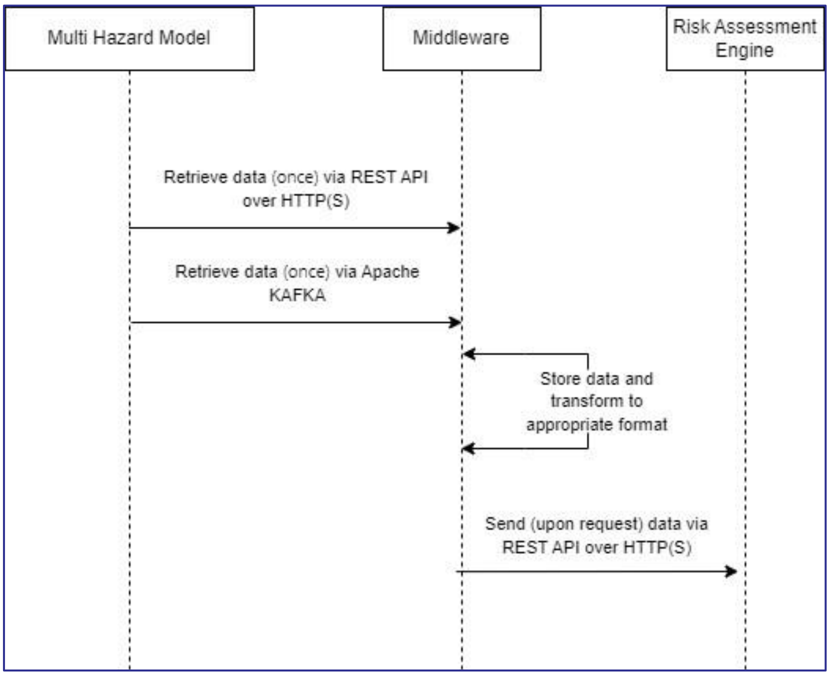
9.5. Multi-hazard model – Middleware – Risk Assessment Engine

Integration Point Description	
Identifier	*(1,7) Multi-hazard model – Middleware*
Involved partners	ULiege, RISA

Responsible	ULiege
Integration Point Purpose	Data Formats: GeoTIFF files The GeoTIFF datasets correspond to water depth maps derived from the Multi-Hazard component's floodplain simulations. Provided as a time series on the Middleware, they capture the dynamics of flood propagation. These outputs are utilized by the Risk Assessment Engine to assess damages and categorize impacted assets by severity: slight, moderate, or severe.
Protocol	The protocols employed for the specific data are Apache Kafka and REST API over HTTPS (data will be sent once).
Status	*Ongoing*

Integration Point Description	
Identifier	*(1,8) Middleware - Risk Assessment Engine*
Involved partners	RISA, SoReCC
Responsible	RISA
Integration Point Purpose	• Data (as .Json files) is sent to COP/IMS. • Notifications / Alerts (e.g., after data fusion). -This integration point facilitates the selection of appropriate impact assessment results. The multi-hazard system determines a flood propagation time-series (where each timeframe is represented by a GeoTIFF file). To ensure alignment, RAE needs to be informed about the last timeframe (i.e., GeoTIFF). This information allows RAE to choose the corresponding impact assessment output dataset.
Protocol	The protocols employed for: • data are Apache Kafka and REST API over HTTPS (data are sent upon request / registration). • Notifications / alerts produced (e.g., after fusion of data) is Apache Kafka (upon registration).
Status	*Ongoing*

Consolidated sequence diagram:



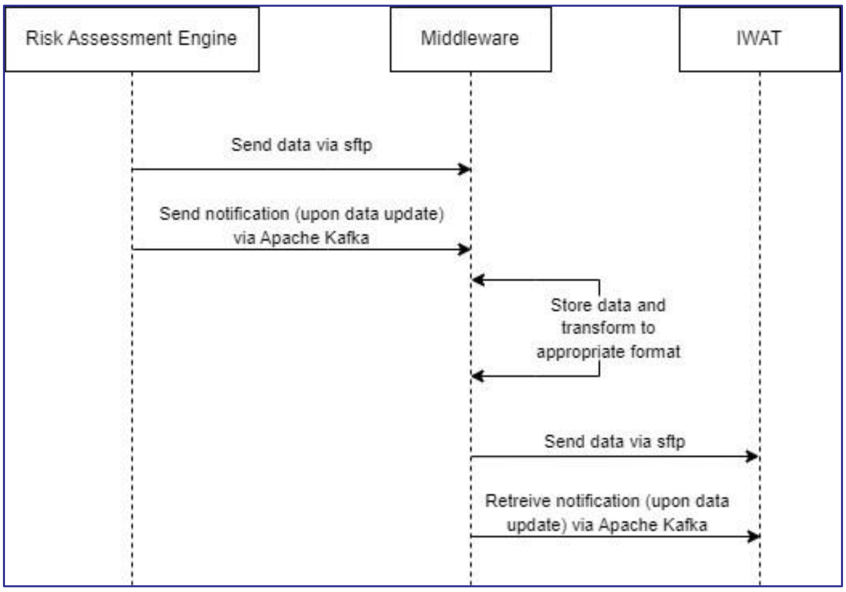
9.6. Risk Assessment Engine - Middleware – IWAT

Integration Point Description	
Identifier	*(1,8) Risk Assessment Engine – Middleware*
Involved partners	SoReCC, RISA
Responsible	RISA
Integration Point Purpose	- To transfer data (from *Risk Assessment Engine*): .hdf5 - To transfer the output of the impact assessment analysis for the examined hazard, generated by the Risk Assessment Engine, to Middleware. Specifically, the impact assessment results contain information on the assets at risk and the impact of the potentially catastrophic events. This integration enables Middleware to receive and store this critical risk data before forwarding it to downstream applications like IWAT.
Protocol	The protocol employed for Data (from *Risk Assessment Engine*) is sftp. A notification is sent to KAFKA upon data update.
Status	*Completed*

Integration Point Description	
Identifier	*(1,6) Middleware – IWAT*

Involved partners	RISA, STWS
Responsible	RISA
Integration Point Purpose	- To transfer data (from *Middleware*): .hdf5 - Middleware delivers hazard risk assessment data to the IWAT system. This data, originating from the Risk Assessment Engine, provides IWAT with the necessary input, such as information on the assets at risk and the impact of the potentially catastrophic events. IWAT will utilize this input for its functions, such as displaying impact assessments to the user.
Protocol	The protocol employed for Data (from Risk Assessment Engine) is sftp. A notification is sent to KAFKA upon data update.
Status	*Completed*

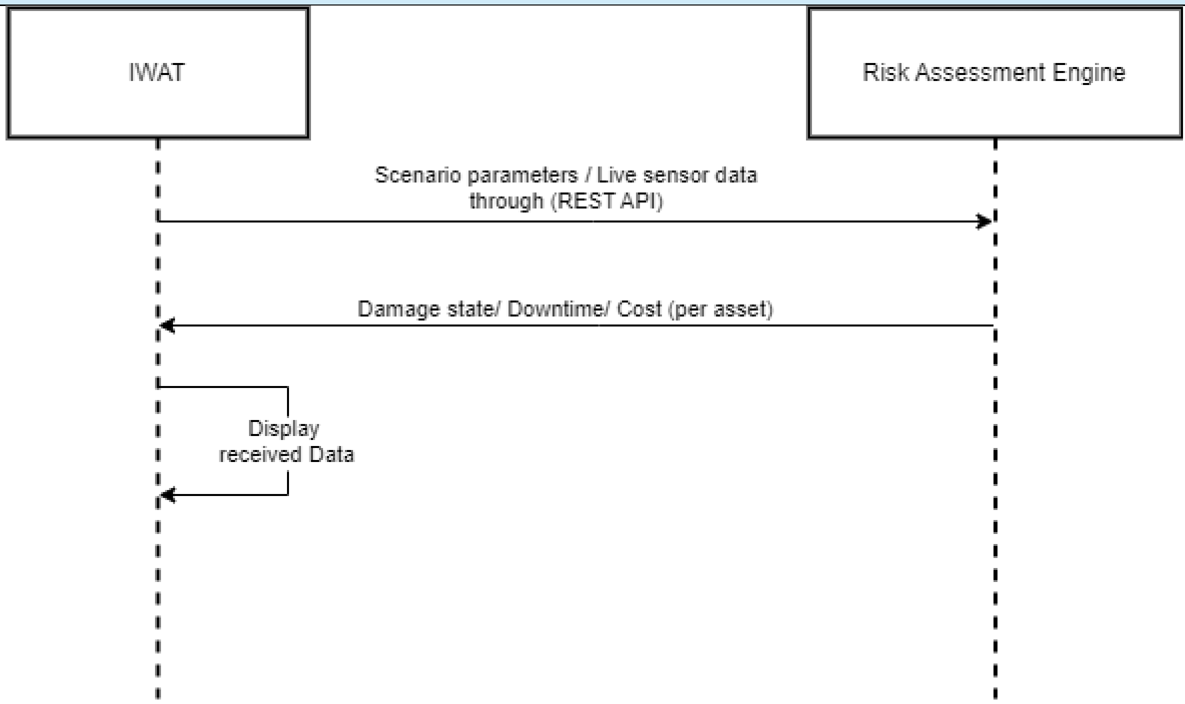
Consolidated sequence diagram:



9.7. Risk Assessment Engine – IWAT

Integration Point Description	
Identifier	*(6,8) Risk Assessment Engine – IWAT*
Involved partners	Sorecc, STWS
Responsible	Sorecc
Integration Point Purpose	Perform Risk Assessment to estimate the assets' damage state, repair cost and downtime in case of a hazard
Protocol	REST API

Status	*Completed*
Sequence diagram	



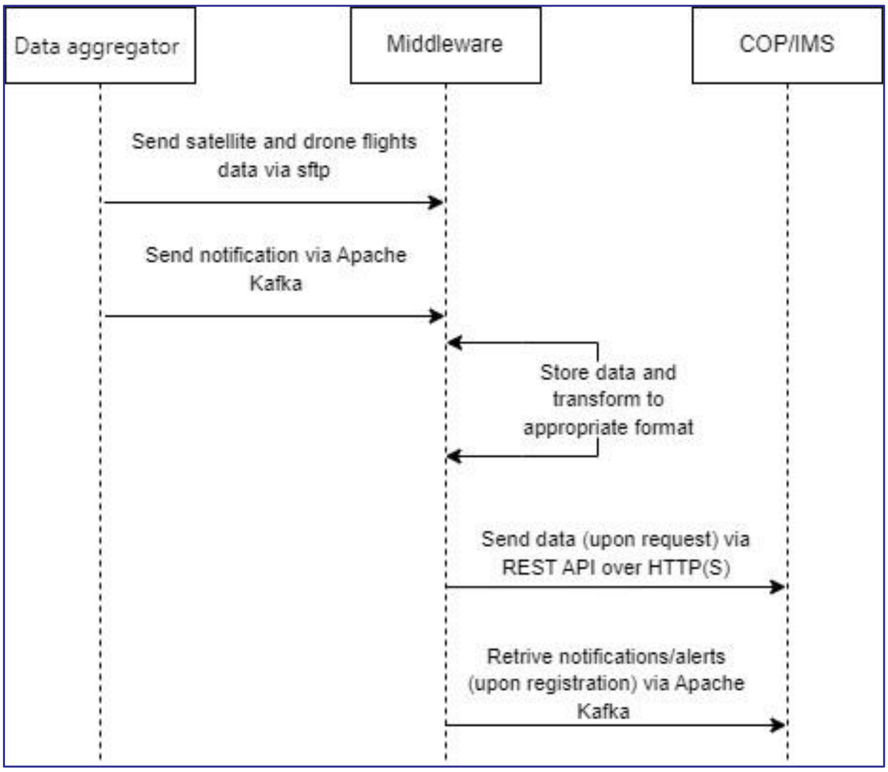
9.8. Data aggregator – Middleware – COP/IMS

Integration Point Description	
Identifier	*(1,10) Data aggregator – Middleware*
Involved partners	STWS, RISA
Responsible	STWS
Integration Point Purpose	Receive Raster products and Notifications/Alerts (e.g., after data fusion). Raster Data Formats: .GeoTIFF, .PNG
Protocol	The protocols employed for the specific data are Apache Kafka (notification) and sftp.
Status	*Completed*

Integration Point Description	
Identifier	*(1,3) Middleware – COP/IMS*
Involved partners	RISA, STWS

Responsible	RISA
Integration Point Purpose	• Raster products. • Notifications / Alerts (e.g., after data fusion).
Protocol	The protocols employed for: • Raster products is REST API over HTTPS (data are sent upon request). • Notifications / alerts produced (e.g., after fusion of data) is Apache Kafka (upon registration).
Status	*Completed*

Consolidated sequence diagram:



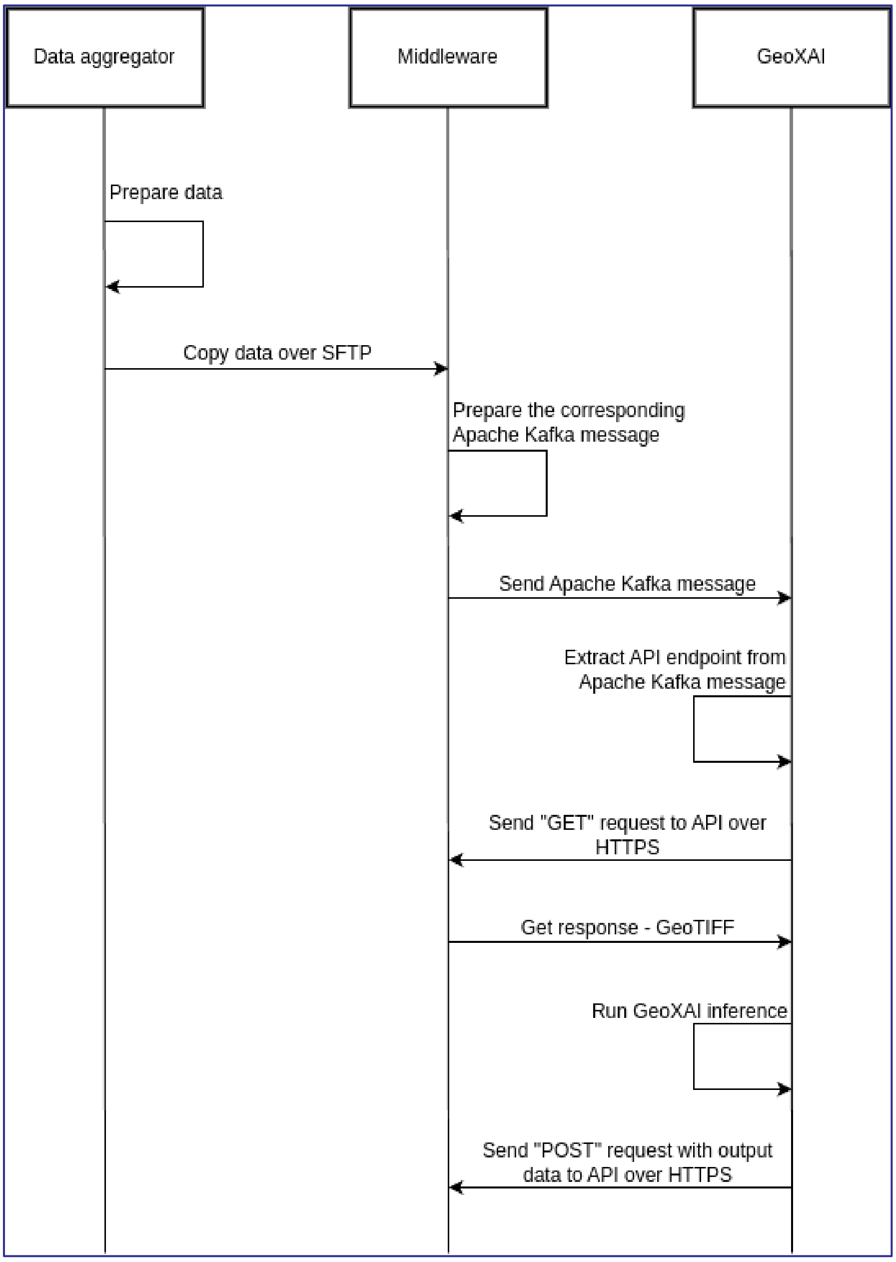
9.9. Data aggregator – Middleware – GeoXAI

Integration Point Description	
Identifier	*(1,10) Data aggregator – Middleware*
Involved partners	STWS, RISA
Responsible	STWS

Integration Point Purpose	Raster Data Formats: .GeoTIFF, .PNG
Protocol	The protocols employed for the specific data are Apache Kafka (notification) and sftp.
Status	*Completed*

Integration Point Description	
Identifier	*(1,9) Middleware – GeoXAI*
Involved partners	RISA, NTUA
Responsible	NTUA
Integration Point Purpose	Data retrieval (input for the module): geoTIFF To archive and deliver the collected remote sensing imagery for the purpose of segmenting flood events, effectively penetrating cloud cover, and enabling pixel-level predictions of inundated areas.
Protocol	The protocols employed for: • Data (from AI- based decision making) is REST API over HTTPS (data are sent upon request). • Notifications / alerts produced (e.g., after fusion of data) is Apache Kafka (upon registration).
Status	*Completed*

Consolidated Sequence diagram:



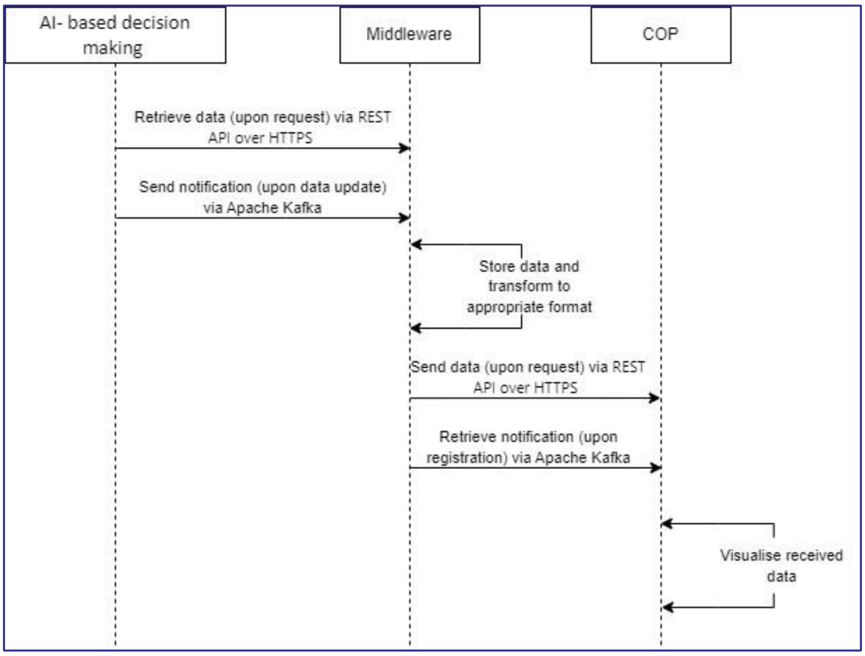
9.10. GeoXAI – Middleware – IWAT

Integration Point Description	
Identifier	*(1,9) GeoXAI – Middleware*
Involved partners	NTUA, RISA
Responsible	NTUA

Integration Point Purpose	Data (from AI- based decision making): .geoTIFF To provide the flood boundaries and extends beyond general flood information for precise geospatial insights.
Protocol	The protocols employed for: • Data (from AI- based decision making) is REST API over HTTPS (data are sent upon request). • Notifications / alerts produced (e.g., after fusion of data) is Apache Kafka (upon registration).
Status	*Completed*

Integration Point Description	
Identifier	*(1,6) Middleware – IWAT*
Involved partners	RISA, STWS
Responsible	RISA
Integration Point Purpose	Data (from AI- based decision making): .geoTIFF Notifications / Alerts
Protocol	The protocols employed for: • Data (from AI- based decision making) is REST API over HTTPS (data are sent upon request). • Notifications / alerts produced (e.g., after fusion of data) is Apache Kafka (upon registration).
Status	*Completed*

Consolidated sequence diagram:



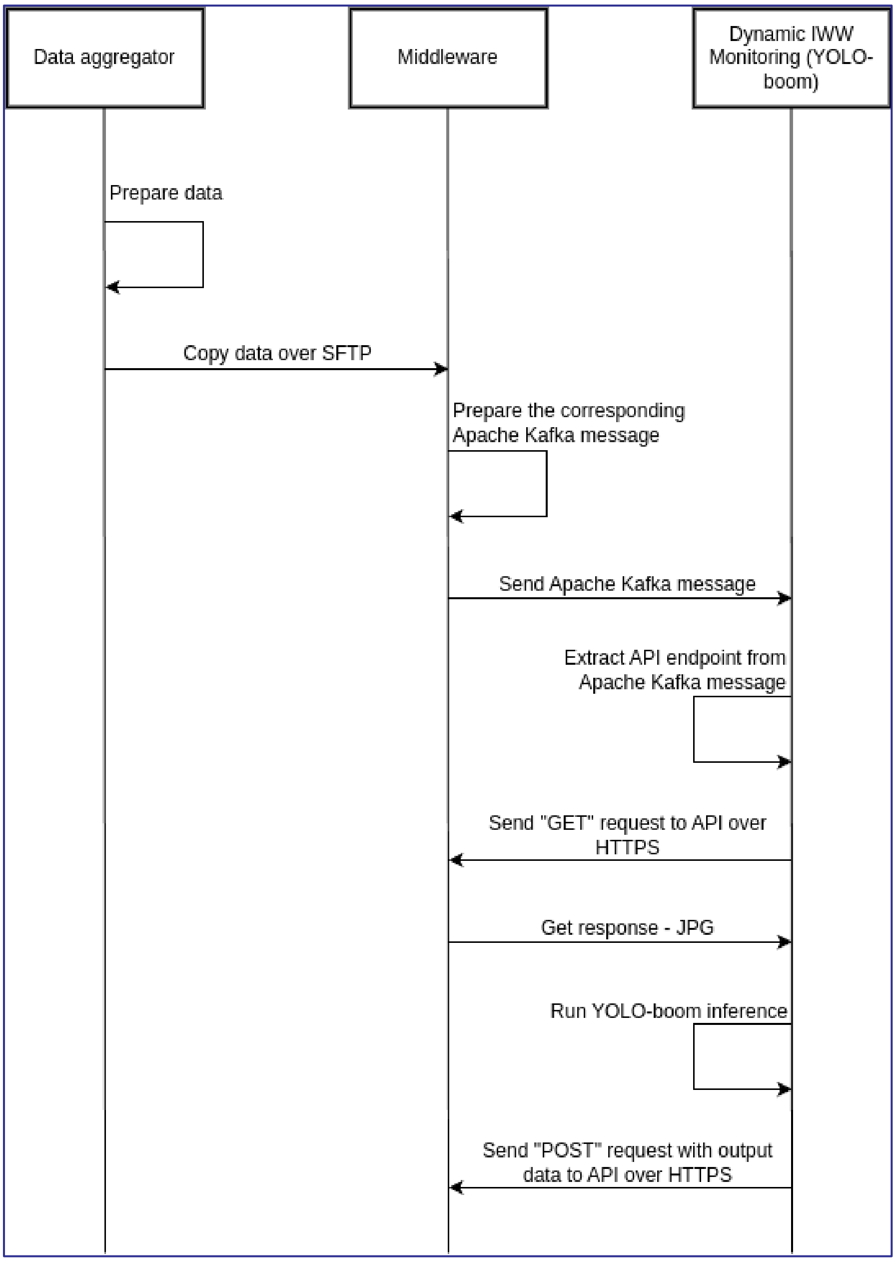
9.11. Data aggregator – Middleware – YOLO-boom

Integration Point Description	
Identifier	*(1,10) Data aggregator – Middleware*
Involved partners	STWS, RISA
Responsible	STWS
Integration Point Purpose	Raster Data Formats: .GeoTIFF, .PNG Notifications/Alert
Protocol	The protocols employed for the specific data are Apache Kafka (notification) and sftp.
Status	*Completed*

Integration Point Description	
Identifier	*(1,15) Middleware – YOLO-Boom*
Involved partners	RISA, NTUA
Responsible	NTUA
Integration Point Purpose	Raster Data Formats: .GeoTIFF, .PNG Notifications/Alert

	The UAS images acquired, and their metadata are utilized, including GPS location, altitude, and camera sensor parameters to detect and catalogue construction defects with precise geospatial information by integrating YOLO v8 and SAM models
Protocol	The protocols employed for: • Data (from Data Aggregator) is REST API over HTTPS (data are sent upon request). • Notifications / alerts produced (e.g., after fusion of data) is Apache Kafka (upon registration).
Status	*Ongoing*

Consolidated sequence diagram:

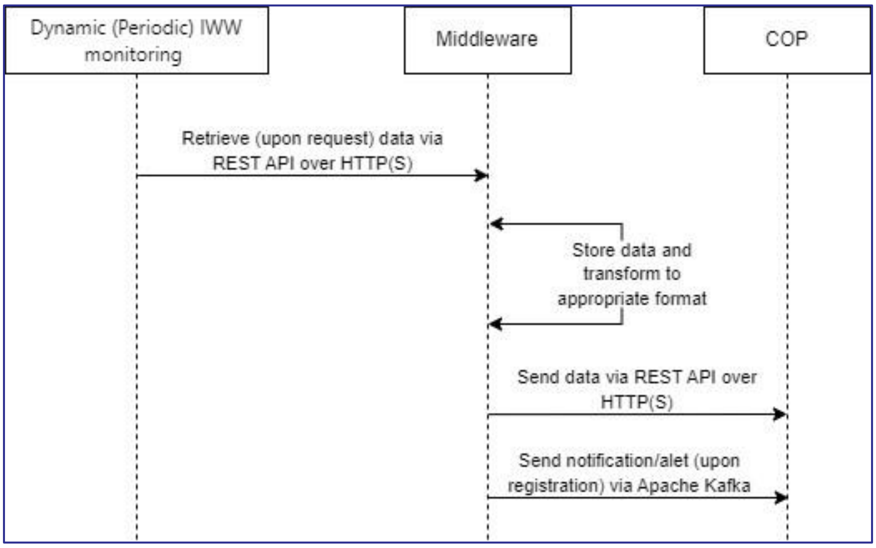


9.12. YOLO-Boom – Middleware – COP/IMS

Integration Point Description	
Identifier	*(1,15) Dynamic (Periodic) IWW monitoring (T5.5) – Middleware*
Involved partners	NTUA, RISA
Responsible	NTUA
Integration Point Purpose	Data (from Dynamic (Periodic) IWW monitoring): .GeoJSON, .TIFF, .gpkg The RGB images with the fused information of the defects are being stored to the MW and through the notification/alert are transferred to the COP/IMS
Protocol	The protocol employed for Data (from Dynamic (Periodic) IWW monitoring) is REST API over HTTPS (data are sent upon request).
Status	*Ongoing*

Integration Point Description	
Identifier	*(1,3) Middleware – COP/IMS*
Involved partners	RISA, STWS
Responsible	RISA
Integration Point Purpose	• Data. • Notifications / Alerts (e.g., after data fusion).
Protocol	The protocols employed for: • Data is REST API over HTTPS (data are sent upon request). • Notifications / alerts produced (e.g., after fusion of data) is Apache Kafka (upon registration).
Status	*Ongoing*

Consolidated sequence diagram:



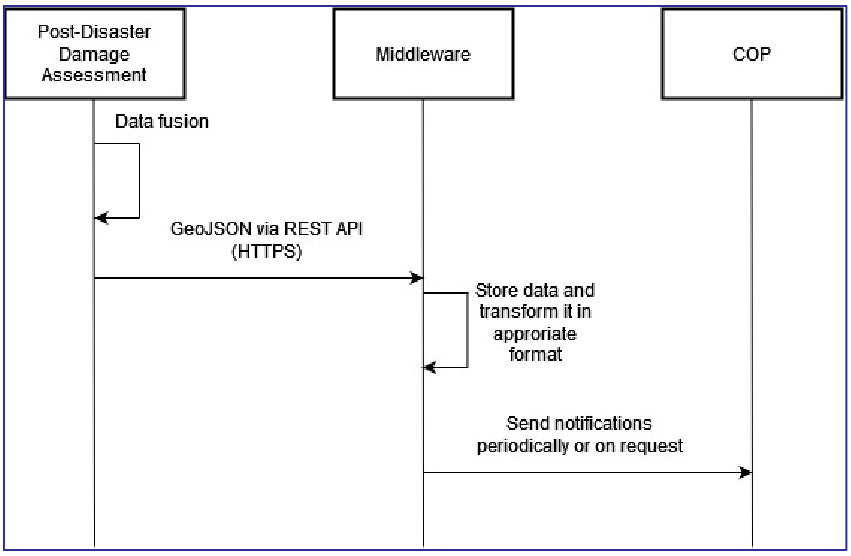
9.13. DYMON – MW – COP/IMS

Integration Point Description	
Identifier	*(1,15) DYMON– Middleware*
Involved partners	UM, RISA
Responsible	UM
Integration Point Purpose	• Data • Notifications / Alerts (e.g., after data fusion). • ShapeLayer, GeoJSON The DYMON module provides results of the damage assessment analysis via the HTTP endpoints in a standardized format (ShapeLayer/GeoJSON).
Protocol	The protocols employed for: • Data is REST API over HTTPS (data are sent upon request). • Notifications / alerts produced (e.g., after fusion of data) is Apache Kafka (upon registration).
Status	*Completed*

Integration Point Description	
Identifier	*(1,3) Middleware – COP/IMS*
Involved partners	RISA, STWS

Responsible	RISA
Integration Point Purpose	• Data. • Notifications / Alerts (e.g., after data fusion).
Protocol	The protocols employed for: • Data is REST API over HTTPS (data are sent upon request). • Notifications / alerts produced (e.g., after fusion of data) is Apache Kafka (upon registration).
Status	*Completed*

Consolidated sequence diagram:

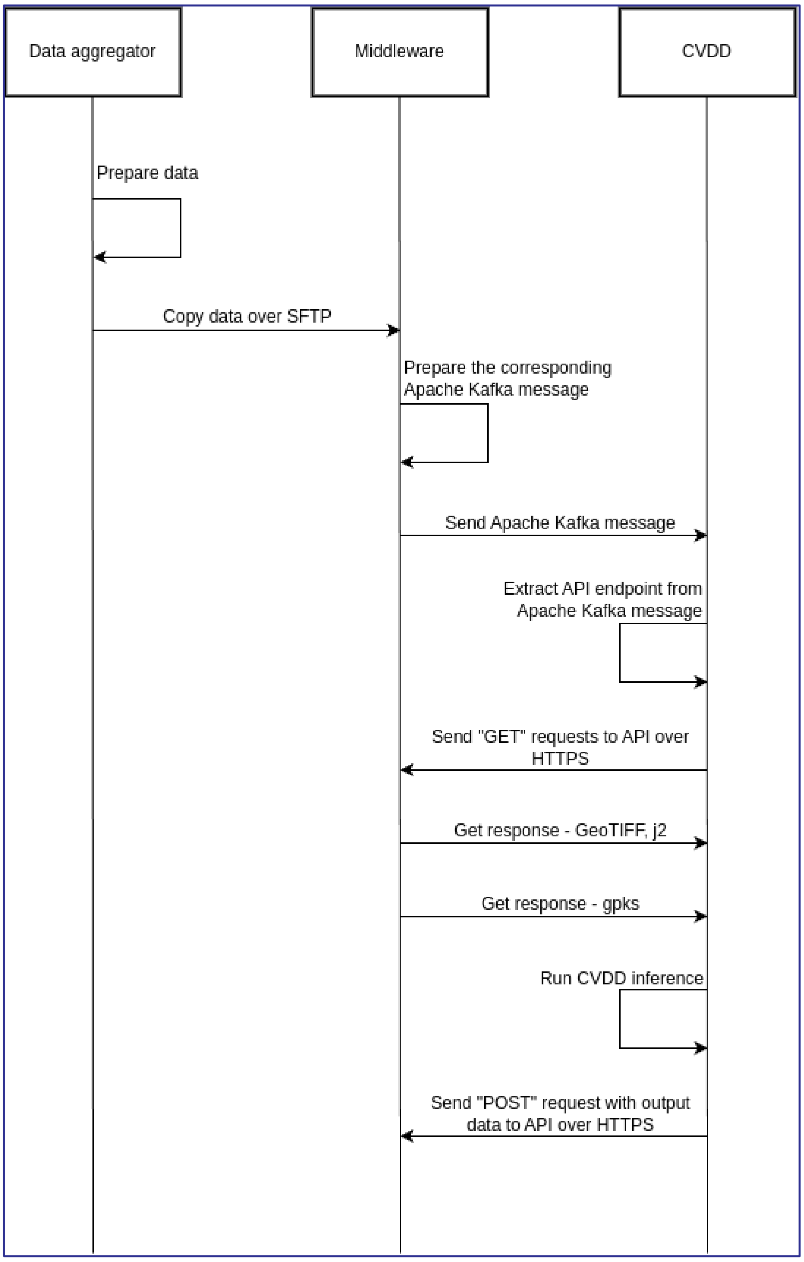


9.14. Data Aggregator – MW - CVDD

Integration Point Description	
Identifier	*(1,10) Data aggregator (T5.2, Ground control station) – Middleware*
Involved partners	STWS, RISA
Responsible	STWS
Integration Point Purpose	Raster Data Formats: .GeoTIFF, .PNG
Protocol	The protocols employed for the specific data are Apache Kafka (notification) and sftp.
Status	*Completed*

Integration Point Description	
Identifier	*(1,11) Middleware - CVDD (T5.3)*
Involved partners	RISA, NTUA
Responsible	RISA
Integration Point Purpose	Notification and data retrieval Data formats: j2, GeoTIFF, GPKGs After the appropriate notification the remote sensing data are being processed to detect the riverside features related to the floods, as well as any new/existing islets.
Protocol	The protocols employed for: • Data (from AI- based decision making) is REST API over HTTPS (data are sent upon request). • Notifications / alerts produced (e.g., after fusion of data) is Apache Kafka (upon registration).
Status	*Ongoing*

Consolidated sequence diagram:



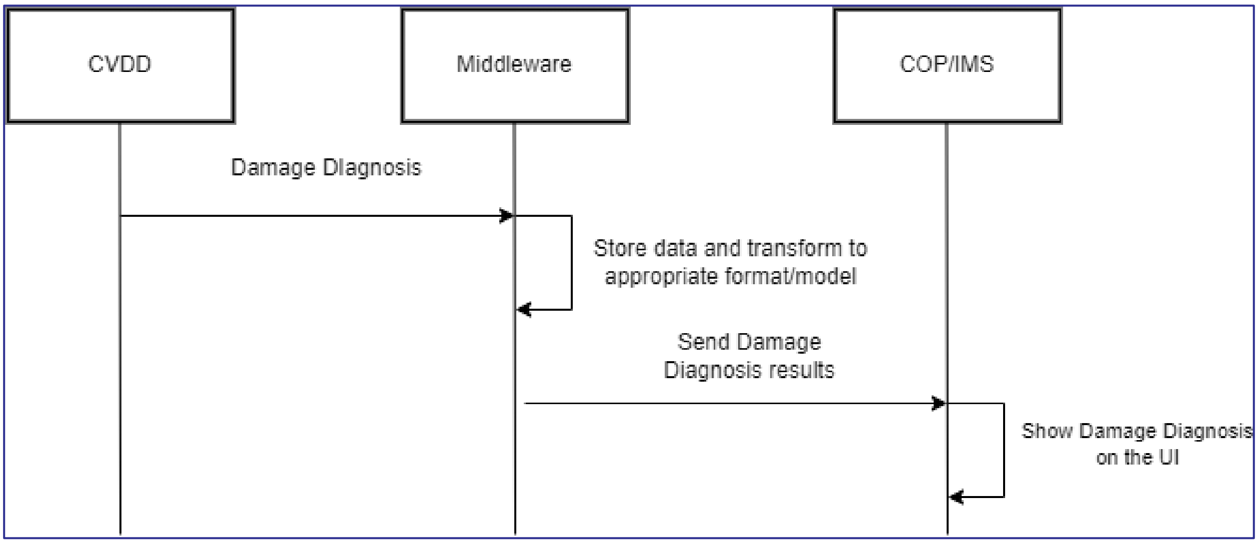
9.15. CVDD – MW – COP/IMS

Integration Point Description	
Identifier	*(1,11) CVDD – MW*
Involved partners	NTUA, RISA
Responsible	NTUA

Integration Point Purpose	Notification and data retrieval Data formats: j2, GeoTIFF, GPKGs, TXTs Raster products that facilitate riverside extraction through the application of edge detection algorithms on WaterMaps on satellite data are stored to the MW. In addition, vector products are produced and stored to identify islets and to extract polygons, lines, and points related to the riverside.
Protocol	The protocols employed for: • Data (from AI- based decision making) is REST API over HTTPS (data are sent upon request). • Notifications / alerts produced (e.g., after fusion of data) is Apache Kafka (upon registration).
Status	Ongoing

Integration Point Description	
Identifier	*(1,3) Middleware – COP/IMS*
Involved partners	RISA, STWS
Responsible	RISA
Integration Point Purpose	• Raster products. • Notifications / Alerts (e.g., after data fusion).
Protocol	The protocols employed for: • Raster products is REST API over HTTPS (data are sent upon request). • Notifications / alerts produced (e.g., after fusion of data) is Apache Kafka (upon registration).
Status	*Ongoing*

Consolidated sequence diagram:



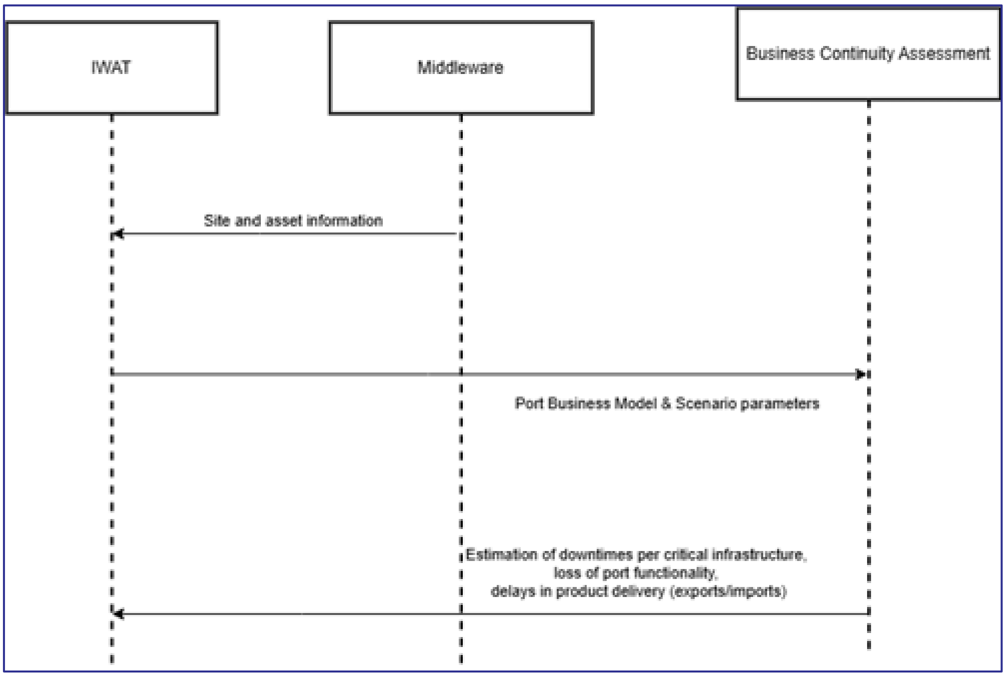
9.16. BCA - MW - IWAT

Integration Point Description	
Identifier	*(1,14) BCA – MW*
Involved partners	SoReCC, RISA
Responsible	SoReCC
Integration Point Purpose	The Business Continuity Module (BCM) requires the exposure dataset to be able to determine the functionality and usage of the assets at risk. The BCM model queries the MW for the date of the exposure dataset and if a newer version is available, it is retrieved.
Protocol	The protocol employed for relevant Data is sftp. REST API to retrieve date of the exposure dataset. Notifications / alerts produced (e.g., after fusion of data) is Apache Kafka (upon data update).
Status	*Completed.*

Integration Point Description	
Identifier	*(1,6) MW – IWAT*
Involved partners	RISA, STWS
Responsible	RISA
Integration Point Purpose	Data Notification/alert

Protocol	The protocol employed for relevant Data is sftp. Notifications / alerts produced (e.g., after fusion of data) is Apache Kafka (upon data update).
Status	*Completed.*

Consolidated sequence diagram:



9.17. Digital Twin – Multi-hazard model - Middleware

Integration Point Description	
Identifier	*(7,13 & 1,7) Digital Twin - Multi-hazard model*
Involved partners	EXUS, ULiege
Responsible	ULiege
Integration Point Purpose	This module provides the flooded area in Case Study C along with its assumed probability of occurrence, based on a predicted precipitation time series. These time series can be sourced from either OMSDA (short-term) or EURO-CORDEX (long-term) to estimate discharge conditions at Amay, Chaudfontaine, and Sauheid via multiple requests to the Digital Twin (DT). The resulting discharge data serve as upstream boundary conditions for a 1D hydrodynamic computation, followed by an interpolation to estimate the 2D floodplain extent and its propagation rate. The flood propagation results are then published as multiple GeoTIFF files on the Middleware, accessible via a GeoServer and an FTP server.
Protocol	REST API
Status	Ongoing

Sequence Diagram