Title	**IIMEO Methodological Framework**
Project Name	IIMEO – Instantaneous Infrastructure Monitoring by Earth Observation
Project Number	101082410
Deliverable number	D4.8
Document Number	
Issue / internal Revision	01 / 08
Status/Release Date	Released / 29.05.2026
Dissemination Level	Public

Funded by the European Union

PREFACE

In 2022, a European Consortium[1] has been selected by the European Commission to implement the project *"Instantaneous Infrastructure Monitoring by Earth Observation" (IIMEO)*. IIMEO is funded by the European Union under the Horizon Europe programme as an innovation action with €2.8 million and runs until 31 May 2026. It aims to develop and demonstrate key technologies for the global monitoring of critical infrastructures from space in near real time. A pilot application will be the monitoring of railway lines.[2]

"Energy supply, communications, transportation – our globalized society is highly dependent on functioning infrastructures. Typical examples are roads and railway lines, but also water pipelines, data cables and power lines," explains OHB project coordinator Daro Krummrich. "Just how critical these infrastructures are for daily life becomes particularly apparent when disruptions occur. These can be caused by natural disasters, extreme weather events or deliberate manipulation. In order to be able to restore the functionality of critical systems promptly after an incident, it is important to quickly gain an overview of the overall situation. This is why IIMEO is about detecting infrastructure malfunctions automatically, across large areas and in near real time, regardless of local weather and lighting conditions."

Figure 0-1: Schematic of IIMEO's objectives

Infrastructure monitoring is an appropriate use case for satellite-based systems regarding the principles of "NewSpace“: Since global coverage and revisit times of less than one hour are required for infrastructure monitoring, the project partners assume that a suitable constellation in low Earth orbit (500 to 900 kilometers altitude) will consist of at least 24 small satellites.

Synthetic Aperture Radar (SAR) imaging radar instruments are to be used as payloads, which will be supplemented by sensors for the wavelength range of visible light (VIS). This will enable high-resolution images to be generated even at night and under heavy cloud cover.

Another focus of the project is the development of algorithms. Since continuous global monitoring of infrastructure with SAR and VIS sensors produces gigantic amounts of data, it is necessary that these are already processed on board the satellites. This is to avoid the data downlink being a bottleneck in the system. Davide Di Domizio, Research Programme Administrator at the European Health and Digital Executive Agency (HaDEA) and in charge of IIMEO, explains: "In 2022, the Horizon Europe work programme set the ambitious goal of demonstrating the performance of key technologies for future Earth observation systems by 2028. With the development of the planned on-board data processor, IIMEO is well positioned to make an important contribution to this mission."

As the development phase is complete, all relevant key technologies are integrated in an airborne technology demonstrator to verify the suitability of the technical solution before sending it into space as satellite payload. The goal of the flight campaign conducted in 2025 was to demonstrate the end-to-end prototype downstream service, including on-board data processing. The automated detection of obstacles on railway tracks is to serve as an example application. The national company for the management of railway infrastructure in Serbia was won as a cooperation partner and pilot user. Slobodan Rosić, Serbian Railway Infrastructure Risk Manager, points out: "A satellite-based automatic monitoring system makes it possible to collect high-quality information about the condition of the infrastructure in real time without having to interrupt regular traffic and without the need for personnel on site."

---

[1] The project is being coordinated by *OHB Digital Connect GmbH* (OHBDC), a subsidiary of space and technology group OHB SE. *Antwerp Space N.V.* (AWS) brings its expertise to the on-board data processor. The *Institut für angewandte Systemtechnik Bremen GmbH* (ATB) brings its expertise in the implementation of european projects and the definition and management of requirements. The *Fraunhofer Gesellschaft zur Förderung der angewandten Forschung e.V.* (Fraunhofer / FHR) brings ist expertise on SAR-data acquisition and processing. The *Fondazione Brunno Kessler* (FBK) brings its expertise on real-time capable fully automated detection methods based on AI. The *Univerzitet U Nis* (NIS) brings its expertise on railways and fully automated detection methods based on AI.

[2] LinkedIn: https://www.linkedin.com/company/iimeo-europe/

Table of contents

List of figures

List of tables

1 INTRODUCTION

1.1 Purpose of this Document

This deliverable D4.8 provides the IIMEO Methodological Framework developed within Work Package 4 (WP4). It consolidates the methods, decision logic and implementation lessons derived from the integration, verification, experimental evaluation and demonstration activities of the IIMEO project.

The purpose of D4.8 is not to provide another technical specification, prototype description or verification report. Instead, it translates the technical and experimental results of the project into a reusable methodological framework for designing, implementing, validating and reusing IIMEO-type services. The framework explains how user needs, sensor systems, processing chains, platform interfaces, verification logic and service delivery were combined into an end-to-end infrastructure monitoring approach.

The document builds on previous project deliverables as follows. D1.1 provides the requirements and use-case basis, while D1.2 defines the system concept, sensor and algorithm logic, service workflows and KPIs. Within WP4, D4.1 to D4.7 document the specification, implementation, integration, verification and refinement of the IIMEO platform, off-board algorithms, on-board and on-ground system integration, and experimental evaluation. D4.11 provides the public demonstration context for airborne-based near-real-time service operation.

D4.8 uses these inputs at the methodological level. Since this deliverable is public, detailed hardware parameters, source-code details, and sensitive internal system information remain in the underlying technical deliverables. The emphasis is placed on transferable principles, such as requirement capture, functional partitioning between on-board and off-board processing, validation design, service integration, user interaction and evidence-based refinement.

Although the railway pilot provides the empirical anchor of the framework, the document is not limited to railway monitoring. It also identifies which parts of the IIMEO approach can support other critical-infrastructure monitoring services, including services for linear infrastructure and natural-disaster-related applications.

1.2 Document Structure

The document is organised as follows:

Section 2 defines the objectives, scope and evidence base of the framework. It positions D4.8 within the WP4 result chain, explains its methodological role, clarifies its applicability beyond railway monitoring and identifies the project results used as inputs.

Section 3 presents the end-to-end service design methodology. It describes how an operational need is translated into service requirements, sensor and acquisition choices, reference data, processing allocation, platform integration and user-facing service products.

Section 4 describes the validation, performance and scalability methodology. It explains the verification path from module level to system level, the use of reference datasets and test-bench verification, the role of airborne campaign evaluation, and the reasoning needed to extrapolate demonstrator results toward future LEO-based operation.

Section 5 provides guidelines for reuse of IIMEO solutions in other services. It defines how the railway pilot experience can be adapted to other infrastructure domains, including service tailoring, sensor and algorithm adaptation, interoperability, data products and deployment considerations.

The final section summarises the main methodological conclusions and highlights how the IIMEO framework can support future implementation and exploitation of Earth observation services for critical-infrastructure monitoring.

2 FRAMEWORK OBJECTIVES, SCOPE AND EVIDENCE BASE

2.1 Position of D4.8 within WP4 and project results

Deliverable D4.8 is positioned as one of the concluding synthesis deliverables of WP4. It builds on the WP4 result chain, including the specification of the IIMEO platform and infrastructure services, the integration and verification of off-board algorithms, the integration of on-board and on-ground system elements, the experimental evaluation activities and the airborne-based demonstration work.

The role of D4.8 is to consolidate the WP4 implementation, integration and evaluation experience into a reusable methodological asset. It complements D4.1 to D4.7 and D4.11 by abstracting from specific engineering implementations toward repeatable service-building logic for IIMEO-type Earth Observation services.

This methodological logic covers requirement capture, functional allocation between on-board and off-board processing, validation design, interface and platform integration, user-oriented service delivery and evidence-based refinement. In this sense, D4.8 is directly linked to Task 4.4, where the complete pipeline from data source, to platform, to service is integrated, verified and assessed with regard to its transferability to future infrastructure-monitoring services.

The framework therefore does not replace the technical WP4 deliverables. Instead, it uses their results as an evidence base for defining transferable methods and guidelines that can support the design, implementation, verification and adaptation of future services based on the IIMEO approach.

2.2 Applicability of the framework beyond railway monitoring

This section distinguishes between the elements of the IIMEO experience that are intrinsically tied to the railway pilot and those generic enough to guide additional services.

- **Pilot-specific elements**: These include the selected target anomalies (e.g., obstacles and track damages), railway-specific map products, feedback from the Serbian end user (NIS/ISR), and the concrete airborne campaign geometry used for validation.
- **Generic framework elements**: These reusable components include the service design sequence, the KPI logic for performance measurement, the use of reference datasets, the partitioning of on-board and off-board functions, the organization of platform interfaces, and the overall validation methodology.

The generic elements are explicitly connected to broader service classes explored in WP1, such as:

- Pipeline and high-voltage power line inspection.
- Dike, road, and waterway monitoring.
- Natural-disaster-related infrastructure services.

For a detailed strategic plan on adapting these generic technologies, including AI retrievals, multi-instrument data fusion, and on-board processing, to these additional commercial domains (e.g. agriculture, and environmental monitoring), please refer to the ***Roadmap to space Deliverable (D5.10)*** [1] . This roadmap outlines the necessary hardware and software adaptations required to scale the IIMEO solution into a multi-sector, satellite-based 24/7 service.

2.3 Evidence base and project results used as inputs

The methodological framework is based on the IIMEO project result chain. It uses the technical, experimental and organisational results of the project as evidence for deriving general guidance on how IIMEO-type services can be designed, integrated, verified and prepared for future operational use. The framework therefore does not start from an abstract service model. It is derived from the actual development, integration and evaluation path followed in the project.

WP1 provides the initial methodological foundation. D1.1 defines the user needs, the railway pilot use case, additional potential use cases, requirement categories and operational constraints. D1.2 translates this requirement basis into the IIMEO system concept, including the sensor and algorithm logic, the demonstration scenario, the allocation of functions between on-board and on-ground elements, service workflows and KPIs. Together, these deliverables establish the link between operational needs and technical design choices.

WP2 contributes evidence related to sensor preparation, reference-data acquisition, evaluation datasets and prototype algorithms. These results are important for the framework because they show how service requirements are transformed into measurable data needs, training material, test scenarios and algorithmic assumptions. WP3 contributes evidence related to the on-board processor, interfaces and integrated on-board software. This evidence is essential for understanding the role of on-board processing in reducing data-transfer constraints and supporting timely service outputs.

WP4 provides the main implementation, integration and validation evidence for D4.8. D4.1 defines the platform and infrastructure service specification. D4.5 documents the integration and verification of off-board algorithms within the full prototype of the IIMEO platform. D4.6 and D4.7 document the integration of on-board and on-ground system parts, the use of flight-campaign data, the combination of SAR and VIS processing results, the evaluation of user interfaces and the verification of pilot-use-case requirements. D4.11 complements this evidence through the public reporting of airborne-based near-real-time service operation.

D4.8 uses these inputs at the methodological level. Since the deliverable is public, it does not reproduce sensitive technical details, implementation-specific source-code specificities, detailed hardware parameters or internal system information from WP2, WP3, and WP4. Instead, it abstracts the documented evidence into reusable guidance. This guidance covers requirement capture, sensor and processing-chain selection, reference-data design, on-board and off-board partitioning, platform integration, verification logic, user-facing service delivery and evidence-based refinement.

Traceability is a central principle of this evidence base. Each methodological statement in the following chapters should be understood as being linked to concrete project results, verified implementation experience or documented work-package contributions. In this way, D4.8 provides a bridge between the IIMEO technical deliverables and a transferable framework for future critical-infrastructure monitoring services.

3 END-TO-END SERVICE DESIGN METHODOLOGY

This chapter serves as the methodological core of the IIMEO project, outlining the sequential design decisions required to transform an identified operational need into a fully implementable end-to-end service architecture. Rather than treating technical components in isolation, the IIMEO methodology follows an ordered workflow where each phase informs the constraints and capabilities of the next.

A defining characteristic of this methodology is the rejection of the *downstream-only-logic* common in earth observation. In IIMEO, service-specific logic interacts directly with on-board sensing, processing, and data-transfer constraints from the very beginning. The design process follows these five logical stages:

Requirement derivation (from user needs to service logic)

The workflow begins with the pilot user's operational needs (e.g. rapid detection of railway track obstructions), which are translated into service requirements such as weather independence and < 1-hour latency (refer to D1.1 for the full use case specifications).

Sensor and acquisition strategy

Based on these requirements, the system adopts a dual-sensor approach (35-GHz SAR and VIS cameras). The choice of the 35-GHz frequency is a direct response to the need for high-resolution, all-weather monitoring (as detailed in D1.2).

Creation of reference and evaluation material

To ensure the system can be verified, a robust framework for reference data is established. This involves collecting high-fidelity datasets via UAVs and airborne campaigns to train and evaluate detection algorithms before they are committed to hardware (see D2.1 and D2.2) [2] [3] .

Functional partitioning (The *IIMEO challenge*)

A critical design decision involves partitioning tasks between the airborne (flight) segment and the ground segment. To overcome data-transfer constraints, the methodology moves high-bandwidth heavy lifting, such as Level-1 SAR processing and initial anomaly detection, to the airborne segment.

Platform integration and service delivery

The final stage involves the physical and logical integration of these functions into a TRL 6 on-board processing platform. This concludes with the delivery of user-facing service products via a cloud-based interface, providing actionable intelligence rather than raw data.

3.1 Service requirement definition and KPI allocation

The transition from a high-level service concept to a verifiable system is governed by a rigorous process of requirement engineering. Within the IIMEO framework, this process ensures that every technical specification implemented in the airborne or ground segments serves a direct operational need identified by the end-user.

The Methodological Path: From user needs to technical parameters

The IIMEO methodology follows a structured path to convert qualitative user expectations into a verifiable technical architecture. This process begins with the definition of use cases, such as the rapid detection of track obstructions, and the identification of use case requirements. These requirements capture the operational constraints of the railway environment, such as the need for all-weather functionality and minimal latency.

These operational requirements are then translated into quantitative system requirements:

- **Response time:** Derived from the use case requirement for *emergency response*, specifying the total time elapsed between a service request and the delivery of an alert.
- **Spatial accuracy:** Mapping the physical dimensions of rail hazards to the resolution requirements of the 35-GHz SAR sensor (e.g., 50 cm resolution).
- **Functional partitioning:** Defining which requirements must be met by on-board processing versus those handled by the cloud infrastructure to satisfy data-transfer constraints.

Requirement traceability and project alignment

A core principle of the IIMEO design is that requirement definition is not a static documentation task, but a proactive engineering mechanism. To ensure project-wide alignment, the methodology utilizes a Requirement Traceability Matrix. This framework ensures that:

- **Vertical alignment:** Each system requirement is directly traced back to a use case requirement. This prevents a misalignment of scope and ensures that every technical feature provides value to the end-user.
- **Use case validation:** Each use case is linked to specific data products (e.g. Level-2 Change Detection Maps) that act as the tangible output required to satisfy the user's operational scenario.
- **Verification and acceptance:** Every requirement is assigned a formal verification method (e.g. analysis, inspection or test). The primary evidence for these requirements is gathered during the airborne flight campaign activities (as outlined in D4.11) [4] .

The mechanism for system integrity

This structured mapping ensures that the work across different work packages remains synchronized. By linking the high-level use cases from WP1 directly to the on-board and on-ground system design in WP3/WP4 and the evaluation procedure in WP4, the project maintains a closed-loop engineering cycle. This ensures that the final architecture is not just a technical success, but a solution that is fit-for-purpose according to the pilot user’s original operational requirements.

3.2 Sensor system selection and acquisition concept

Persistent monitoring of critical infrastructure needs to be performed day and night, and in all weather conditions. This cannot be achieved by using (passive) visual sensors alone, since they only work during the daytime and in the absence of clouds. However, these limitations can be overcome by additionally employing Synthetic Aperture Radar (SAR) imaging, which actively illuminates the scene with electromagnetic waves at frequencies that can also pass through clouds. Since the synthetic aperture is formed by the movement of the radar system along the flight path, good information on the position and orientation of the radar system over time, and good synchronization of this navigation data, is crucial. The quality of the flight track can then further be improved using autofocus techniques.

Although they look similar at first glance, visual and SAR images possess very different properties:

- SAR images contain amplitude and phase information about the reflected signal, but they lack the natural colour information of visual images.
- The reflectivity and transparency of materials for visual light and for radar signals may differ significantly. Visually transparent material may be impervious to radar, and vice versa.
- Visual images extend perpendicular to the line of sight, while SAR images are formed along the line of sight and the flight path, i.e. in range and azimuth.
- This leads to distortions like foreshortening and layover, resp. shadow in SAR images, which decrease resp. increase with decreasing depression angle. For this reason, a “medium” depression angle is generally chosen as a compromise.
- While the spatial resolution of visual images degrades with range, the spatial resolution of a SAR image is, or better, can be made range independent. This is due to the fact that the range resolution of a SAR image is constant, and its azimuth resolution is determined by the length of the synthetic aperture used for imaging a single point.
- The received power in the radar system decreases with the fourth power of the distance, thus naturally limiting the usable swath of SAR imaging.
- Depending on their velocity with respect to the radar system, moving objects in SAR images will appear defocused and displaced in azimuth, or not appear at all.
- Automatic change detection techniques are employed to discover modifications of the infrastructure. Good change detection with SAR images requires their acquisition geometries to be similar.

To make optimum use of the sensor systems, their information is fused.

3.3 Reference data sets, ground truth and evaluation data

Designing reference data sets is a methodological activity that underpins the credibility of every prototype assessment. Rather than treating data collection as an incidental task, it should be regarded as an engineered process whose outputs must remain valid from early algorithm tuning through final acceptance testing. A well-designed reference corpus ensures that detection performance, georeferencing quality, and service timing can be evaluated under conditions that are representative of later operational deployment on the railway network.

A credible data set must span the full operational envelope. This includes nominal scenes – clear track corridors, benign weather – as well as anomalous conditions such as unexpected track obstructions or extreme environmental states. Anomalous cases are particularly important because they stress the system at the boundaries where failures matter most. The proportion and diversity of anomalies should reflect realistic occurrence rates while still providing enough statistical power to characterise detection and false-alarm behaviour.

Obstacle scenarios should be derived from operational analysis rather than invented ad hoc. Each scenario must specify the obstacle type, size, material properties, position relative to the track, and the operational context in which it appears. Scenarios gain credibility when they are linked to documented operational needs – for example, a fallen tree across the rails detected at a defined braking distance in low-visibility conditions, or a person present within the track clearance envelope. Traceability from each scenario back to a requirement or use-case narrative ensures that evaluation results can be directly interpreted in acceptance terms.

Reference data must cover the variability dimensions that influence sensor and algorithm performance. Key axes include:

- **Aspect angles and scene geometries** – straight track, curves of varying radii, cuttings, embankments, stations, and level crossings.
- **Environmental clutter** – lineside vegetation, ballast reflections, rain, fog, snow, low sun glare, and adjacent infrastructure such as overhead line equipment and signalling structures.
- **Sensor configurations** – different mounting positions on the aircraft, lens focal lengths, radar waveform settings, and multi-sensor fusion combinations.

Systematic variation along these axes prevents overfitting to a narrow subset of conditions and supports robust generalisation claims during verification.

Every reference sample requires structured annotation: target class, bounding geometry, confidence level, annotator identity, and timestamp. Metadata should record sensor parameters, environmental measurements, vehicle speed, track mileage, and scenario identifiers so that results can later be stratified by condition. Data must be partitioned into training, tuning (validation), and test splits before any model development begins. Splits should be scene-disjoint – no track section or pass contributes samples to more than one partition – to avoid information leakage. Versioning and checksums guarantee reproducibility across development iterations.

3.4 Allocation of on-board and off-board processing

The separation of overall system processing logic into on-board and off-board components is straightforward. The user essentially requires an alert of all observed anomalies from orbit. To produce these alerts, the observing satellite must execute the entire computer vision chain on board that leads to anomaly classifications, for all sensors. The user can then ask for refinements or a downlink of higher resolution images for further processing. Alternatively transmitting images for processing on the ground requires a very reliable, highly available, and high bandwidth link. Achieving such a link is likely more difficult than adding modern computing components to perform the computer vision tasks. The more ongoing parallel observation tasks in the system, the more this link would be stressed. Again, it is more convenient to stress on-board computing components since they are always available. On-board computer vision further opens opportunities for downlink optimization, since the whole observable area does not need to be downlinked. This would not be possible with a system that performs processing on the ground, thus further stressing the downlink. In conclusion, it is a natural choice to do computer vision tasks that lead to anomaly detections on board such that the user can be alerted and then follow-up choices can be made such as higher detail focused observations or downlinking images for further processing. This has not always been possible due to the lack of cheap computing power of on-board components, but this is now changing.

3.5 Platform, data management and service integration

The on-ground system is collection of software components which jointly bridge between IIMEO's user and the on-board system, which does the data acquisition and most of the processing. Additionally, some processing steps can be done in the ground system itself. It communicates task data with the on-board system via an encrypted network link, accepts task information via a REST interface, and provides information about the task state and results via the same REST interface. The REST interface is used, in turn, by a mobile app as well as a web interface to display results, i.e. obstacle detections, on a map and to accept tasks from the user.

The individual components are packaged into Docker images, such that they can be deployed in a cloud environment as containers. To communicate data between the components, messages are passed via a message broker. Data products which are large are stored on an S3 storage and only the location of the product is communicated via a message.

The components are structured such that they need not know about the other components in the ground system. We use RabbitMQ to integrate the components using the publish-subscribe pattern. Each component subscribes to an exchange to receive the data to be processed and publishes its results to an exchange with other subscribers. All components run concurrently; work is being done when data is available, yielding a pipeline of processing steps.

The central components of the ground system are the *Web Backend* and the *Payload Data Acquisition Unit*. The *Web Backend* provides the REST interface to accept tasks and communicate results. The *Payload Data Acquisition Unit* talks to the Payload Control Software (PCSW) of the on-board system which handles telemetry and telecommands (TMTC), e.g. accepts tasks from the ground system and communicated (intermediate) results from the on-board system to the ground system. When the *Payload Data Acquisition Unit* receives results from the PCSW, it publishes them on its exchange, and the message broker takes care of delivering the message to the component doing the next processing step. The structure of the minimal ground system without processing components is shown in Figure 3-1.

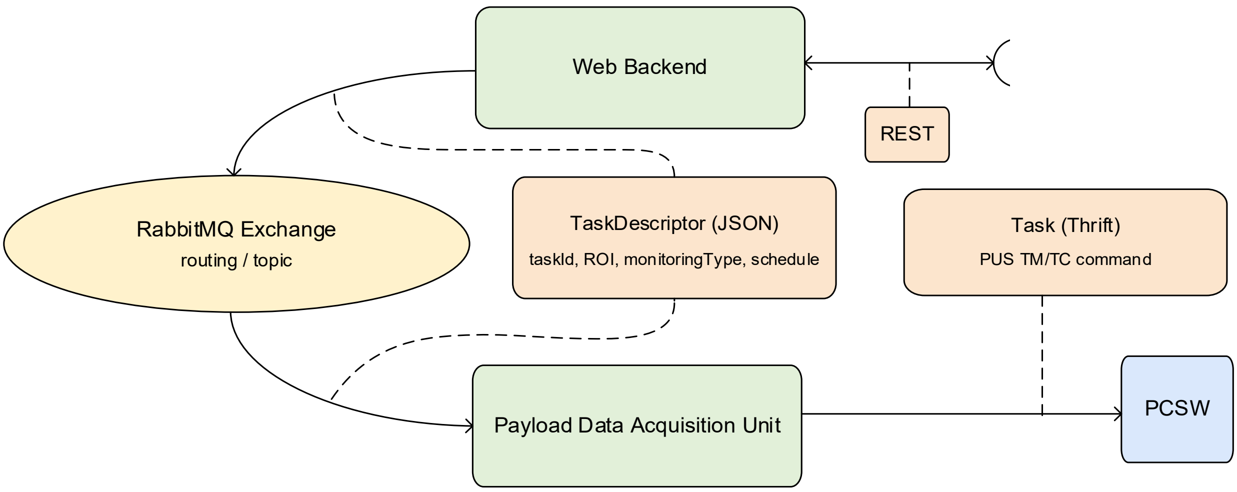

Figure 3-1: Minimal high-level structure of the ground system. Orange: Interfaces; Green: Ground Components; Blue: On-Board Components; Yellow: Message Broker.

There are data processing implementations in ground system components which, at first glance, duplicate processing steps on the on-board system. These include the SAR image formation, the railway track detection as well as the VIS image orthorectification. The first two use the ground system's greater processing and memory capacity to provide results which are potentially better than what the on-board system can provide. The VIS orthorectification is not a duplicate. It takes geo-referenced images and resamples them into images whose axes are aligned to the cardinal directions, North-South and West-East, such that the map between geo-coordinates and resampled-image-coordinates is affine. This is necessary to display images in the Web UI. The step is not necessary in the on-board system, which always uses the original image coordinates and the non-linear map between geo-coordinates and image-coordinates computed by the on-board geo-referencing component.

Communication inside the ground system and with the onboard system happens in the context of a task. In the ground system, that context is represented using a *TaskDescriptor*. The task descriptor has a unique task identifier, metadata including progress information, as well as a list of *ProgressDescriptor*s. A *ProductDescriptor* has information on a data product which has been produced in context of the task whose *TaskDescriptor* the *ProductDescriptor* belongs to. Metadata stored in the *ProductDescriptor* includes the timestamp of the product's creation, the identifier of the task the product belongs to, if the product is derived from a single measurement, then the viewing geometry of the corresponding sensor, the data type of the product and the product itself or the reference to the product's location on the S3 storage. When *TaskDescriptor*s and *ProductDescriptor*s are communicated among ground system components, they are represented using JSON text.

For the communication with the on-board system, there is an interface definition for binary data exchange using the library Apache Thrift. That definition has types corresponding to the *TaskDescriptor* along with definitions for a set of remote procedure calls, which are offered by the PCSW. When the Payload Data Acquisition Unit of the ground system wants to communicate with the on-board system, it calls one of the PCSW's procedures. These are designed to mimic PUS and include procedures to start and stop a task, query weather the task has been completed and also a procedure to download product data corresponding to the task.

None of the components explicitly know each other and hence can be easily substituted. If the Web Backend is replaced by a test driver script to exercise the Payload Data Acquisition Unit, the latter component does not learn this. This was very handy during development, because it allowed to write drop-in replacements for individual components and to let the ground system speak to a simulated on-board PCSW.

The on-board system is structured in a very similar manner. There, components are not containerized but are implemented in ROS2 [5] nodes instead, and publish-subscribe is realized by having ROS2 pass messages between the nodes, whose message types are defined using ROS2's interface definition language. In the onboard system, components, i.e. nodes, can also be replaced by a different node providing the same interface. We again use this for test-driver-nodes during development, but also to support different types of hardware; for instance, the obstacle detection on VIS images has one implementation to run specifically on the concrete on-board computer, and a second one using PyTorch [6] to run on essentially any hardware supported by PyTorch.

The decomposition into relatively small components both in the on-ground and in the on-board system turned out to be very beneficial. It will be much easier to incrementally change the nodes for data acquisition at the beginning of the processing chain without the need to touch the rest of the system, than it would be to modify a (hypothetical) large single program to process different inputs.

3.6 User interaction, service products and access methods

In the IIMEO architecture, user interaction is not a finishing layer added after processing, but rather an integral design dimension that shapes the service architecture from the beginning. Our methodology ensures that service outputs and access modes are selected in close cooperation with end-users, guaranteeing that products - such as anomaly intervals, diagnostic maps, alerts, visual overlays, and supporting context information - directly reflect the operational decisions the service is meant to support. By integrating users into the ***early development phases*** *(*as this has been performed in scope of D1.1*)*, we ensure the service is inherently actionable for maintenance and safety workflows.

3.6.1 Core Dimensions of User Interaction and Service Delivery

The project's tasking logic and access methods are designed to accommodate a diverse user base through a tiered interaction model, addressing the following key dimensions:

Web interfaces & mobile views

The service provides a centralised web-based dashboard for strategic oversight, featuring interactive maps and deep-dive analytics. To support field personnel, mobile-responsive views are integrated to ensure critical alerts and simplified situational displays are accessible in diverse operational environments. These two interfaces have proven to complement each other exceptionally well, with the web platform offering comprehensive analytical depth for task planning and configuration, while the mobile view provides streamlined, time-critical access for operational field personnel.

Tasking logic

Interaction is governed by a user-driven tasking model, aligned with the system concept, sensor, and algorithm logic defined in D1.2. Operators define observation parameters (including geographical coverage, revisit frequency, and sensor mode configurations) through a web interface that provides real-time feasibility assessments, ensuring that requested tasks are viable within the constraints of satellite and sensor availability.

Filtering options

To prevent information overload, the platform employs intelligent filtering that translates complex machine-generated data into prioritized, actionable outputs. This allows the system to distinguish between critical anomalies requiring immediate response and routine status updates, tailoring the data stream to the specific needs of the operator.

Alert thresholds

Thresholds are not static system defaults but are defined in close cooperation with end-users to align with the specific risk tolerances of the infrastructure being monitored. By calibrating these parameters to practical operational requirements, the system minimises *alert fatigue* and ensures that reported findings are consistently relevant.

Balance between machine-generated confidence & user-readable reporting

While the platform generates high-fidelity machine-confidence metrics for every detection (i.e. critical for algorithm validation), these are handled as secondary contextual metadata. The primary user interface remains focused on presenting clear, actionable information, allowing users to focus on the operational impact of a detection while retaining the ability to drill down into statistical uncertainty for technical verification when needed.

Ultimately, the usability and long-term acceptance of the IIMEO service rely on an iterative feedback loop with pilot users. This methodology is supported by evidence documented across WP4 deliverables (D4.1-D4.7), which cover the specification, implementation, and verification of the platform, algorithms, and system integration. During the demonstration phase, user input is systematically collected to refine product granularity, presentation standards, and delivery channels. This allows the consortium to continuously adjust visual overlays and alert logic, ensuring they are intuitive for both control room operators and on-site technicians. By treating user interaction as a continuous design variable, IIMEO transforms from a purely technical Earth observation solution into a robust, operationally intuitive monitoring service capable of evolving with the needs of the infrastructure industry.

3.6.2 Methodological UI/UX Guidelines for Infrastructure Monitoring Applications

To translate the automated, event-driven data streaming of the IIMEO ground system into an actionable user experience, the frontend interfaces (Web Application and Mobile App) should adhere to specific UI/UX development guidelines. These guidelines do not operate in isolation, they map directly onto the underlying software components, telemetry structures, and data processing steps (as implemented within WP4):

Geographic Information Systems (GIS) and map-centric design

- **Predefined asset corridors:** Because the application centers heavily on linear infrastructure (e.g., railway lines, power corridors), the map layer should dynamically clip background clutter and highlight the targeted asset geometry to reduce cognitive load.
- **Orthorectified data alignment:** Visual payloads (VIS) should be seamlessly overlayed via affine transformation to align image coordinates with cardinal directions (north-south/west-east orientation) on the map interface, avoiding raw camera distortions for the end-user.
- **Clustering and progressive disclosure:** When displaying large numbers of normal status data or closely grouped anomalies along an infrastructure corridor, the UI should use map-clustering

techniques. Zooming in should progressively reveal discrete anomaly intervals and localized probability data.

Tiered information architecture (Web vs. Mobile responsive views)

- **Web dashboard (strategic oversight):** Optimized for high-density information displays. It should provide side-by-side comparative views (e.g., comparing historical baseline SAR imagery against new change-detection maps) and facilitate complete task parameter planning (geographical coverage, revisit frequencies, sensor configurations).
- **Mobile view (field alerts):** Optimized for low-bandwidth, high-glare outdoor environments. It should suppress complex analytical matrices in favour of simplified, large-format situational displays, location-based navigation to anomalies, and quick status acknowledgement buttons.

Critical alerting and cognitive load management

- **Strict alert categorization:** Anomaly notifications must not flood the user. Detections must be visually categorized using standardized semantic colours mapped directly to calculated anomaly probabilities (e.g., red for high-confidence track obstructions requiring emergency response, yellow for potential maintenance deviations).
- **Actionable metadata displays:** Selecting an anomaly must instantly expose a simplified card containing the essential data derived from the *Product Descriptor* and *Task Descriptor* structures, such as: Unique *Task* and *Result ID*, generation timestamp and processing latency, core anomaly properties (e.g., estimated size or obstacle likelihood classification), or a clear direct call-to-action button.

State synchronization and feedback loops

- **Asynchronous tasking feedback:** Because sensor command tasking and satellite/airborne acquisitions involve unavoidable physical delays, the UI should explicitly reflect the asynchronous lifecycle states of a task (e.g., *Pending, Scheduled, Executing, Processing, Completed*). Users shall see immediate local confirmation that a task request was successfully sent via the ground REST interface.
- **Human-in-the-loop validation:** The UI might feature an integrated mechanism for field personnel or command operators to *confirm* or *reject* an automated detection result. This human feedback loop provides ground truth validation data necessary for iterative refinement of the cloud and edge AI models.

4 VALIDATION, PERFORMANCE AND SCALABILITY METHODOLOGY

4.1 Verification logic from module to system level

For the on-ground modules, all modules receive their data in context of a task, checking of each of the on-ground modules is done using the following pattern:

1) Reception of a Task Descriptor via RabbitMQ
2) Extraction of relevant task and product metadata
3) Loading of input data from S3 object storage
4) Execution of the algorithm as specified in [7]
5) Storage of computed results in S3
6) Creation of Product Descriptors and update of the Task Descriptor
7) Forwarding the updated Task Descriptor to the next module via RabbitMQ

Steps 1-3 and 5-7 are rather generic. They involve calling library functions of RabbitMQ and loading and storing data on the S3 storage. These are checked by code review and/or running the module on a prepared *TaskDescriptor* containing the appropriate *ProductDescriptors* for the module and then checking the output TaskDescriptor with its new ProductDescriptor produced by the module. If that succeeds, the algorithm implemented in the module is considered successfully integrated into the on-ground platform.

Step 4 is algorithm-specific. The outputs of the modules are checked against the outputs of the prototypes developed in the scope of the prototyping work package WP2. Since the implementations in the on-ground system are not exactly the same as the prototypes, we consider it good enough if the outputs are functionally the same but allow for numerical differences.

For the on-board system, the strategy is similar. Individual on-board modules, called nodes in the ROS2 context, need to receive and send messages. These have been tested individually by playing ROS2 messages from recording using a "bag" file. Most nodes offer the possibility to write intermediate results to disk in addition to publishing messages. If possible, the outputs were checked against the outputs of the corresponding prototypes. Otherwise, the outputs were directly checked against the algorithm requirements [3] . For some functional units smaller than nodes are checked by unit tests, which are implemented as small separate programs checking properties of the units under tests. In particular geometry computations used for geo-referencing VIS images as well as dividing images into tiles for railway track detection on the onboard processing unit and the their reassembly into the original image dimensions are checked in this manner, because errors might be subtle and the functionality is used across multiple nodes, e.g. the node assembling tiles into image coordinates is not the node which produced the tiles.

In addition to running on the on-board computers, the on-board system, from VIS data acquisition to results to be transmitted to ground, is also tested in our development environment by using a drop-in replacement of the anomaly detection's railway track detection step using PyTorch [6] instead of the Myriad accelerators.

There is no explicit sequencing of modules. Both the on-board system modules, "nodes", and the on-ground system modules are organized using the publish-subscribe pattern [8] , where each module subscribes to a feed of the data it needs and publishes the data it produces. Whenever data becomes available for a subscription, the module performs its computation and published the result. There is no explicit sequencing to verify. The message dispatch has not been developed in this project. For the on-board system, we rely on ROS2's implementation, which in turn is uses a DDS implementation. For the on-ground system, we use RabbitMQ.

Since we use ROS2 in the onboard system anyway, we use its Interface Definition Language (IDL) to define message types and let ROS2's tooling generate the code for data representation and de-/serialization.

To communicate between the on-board and on-ground systems, we use Apache Thrift, which also generates message serialization and deserialization code for us and also provides the transport as library functions. One of the great appeals of ROS2/DDS [5] [9] , RabbitMQ [10] and Thrift [11] is that they are widely used and already received more testing than we could ever hope to do on a project-specific development.

4.2 Test-bench verification and use of reference datasets

Test-bench verification provides a controlled laboratory environment in which individual functions and processing chains can be exercised independently of the operational platform. Its purpose is to isolate specific modules, replay representative inputs under repeatable conditions, and compare intermediate as well as final outputs against known references. This stepwise approach builds confidence that each component behaves correctly before it is integrated into a larger system and exposed to the cost and complexity of field trials.

The core mechanism of test-bench verification is replay-based testing. Recorded sensor streams – captured during prior measurement campaigns or synthetically generated from validated scene models – are fed into the module under test exactly as if they were arriving in real time. Each replay session is governed by a pre-defined task descriptor that specifies the input data set, the processing parameters, the expected outputs, and the acceptance tolerances.

Reference datasets play a central role in this process. They supply the ground-truth annotations, metadata, and scenario context against which module outputs are assessed. Because the same reference corpus is used across successive development iterations, any regression in performance is immediately visible. Equally, because the datasets are versioned and check summed, results from different test runs remain directly comparable.

During replay the test bench captures not only the final detection or classification result but also intermediate products – such as feature maps, track lists, confidence scores, georeferenced coordinates, and timing stamps. This visibility into the processing pipeline enables engineers to pinpoint where deviations originate, whether in signal conditioning, feature extraction, decision logic, or output formatting. It also confirms that modules exchange the required metadata at each interface, satisfying integration contracts before system-level assembly.

A particularly valuable use of replay-based testing is the direct comparison of on-board and off-board implementations. When identical input streams are processed by the embedded target hardware and by a high-fidelity desktop implementation, the project can confirm functional equivalence – verifying that fixed-point arithmetic, memory constraints, or real-time scheduling have not introduced unacceptable divergence from the reference algorithm. The same paired runs allow precise measurement of execution time on the target platform, supporting timing-margin analysis and identifying bottlenecks that must be resolved before operational deployment.

When discrepancies are observed, the controlled replay environment enables systematic diagnosis: inputs can be shortened, simplified, or perturbed to isolate the triggering condition, turning an observed failure into an understood and correctable defect.

Laboratory evidence, however comprehensive, is not a substitute for field validation. Real-world conditions introduce variability – atmospheric effects, platform dynamics, electromagnetic interference – that no bench environment can fully replicate. Test-bench verification is therefore positioned as the necessary precursor to field trials: it eliminates known defects, confirms baseline performance, and reduces the risk that expensive airborne trials are consumed by issues that could have been detected earlier at far lower cost.

By combining reference datasets, recorded streams, and structured task descriptors within a reproducible test-bench framework, the project establishes a rigorous, stepwise verification path that maximises the diagnostic value of every test and ensures readiness for subsequent operational validation.

4.3 Airborne campaign based experimental evaluation

The airborne campaign-based experimental evaluation represents the decisive system-level validation step in the IIMEO methodology. Module-level tests and replay-based verification provide the necessary evidence that individual components operate correctly, but they cannot fully reproduce the physical, organisational and operational constraints that appear when sensing, processing, communication and service assessment are combined in a real railway environment. For this reason, the airborne campaign is treated in the methodological framework as an end-to-end experiment, not merely as a data-acquisition activity. It connects the railway pilot use case, the airborne demonstrator, the on-board processing concept, the ground platform and the user-facing service into one traceable validation procedure.

In this deliverable, (D4.8), the airborne campaign is presented first as a transferable methodology and then as the concrete application of this methodology in the IIMEO Dornum-Norden flight campaign. This order is important because the role of D4.8 is not to repeat a technical verification report, but to formulate a methodological framework that can be reused for future infrastructure-monitoring services. The campaign is therefore interpreted as a model for planning, executing, documenting and assessing a system-level service experiment in which operational needs, sensor acquisition, processing, communication, visualisation and user assessment are brought together.

Methodology for airborne campaign-based experimental evaluation

The methodological starting point is that an airborne campaign for an IIMEO-type service must be designed as a controlled but realistic system-level experiment. It must be controlled because the service output has to be assessed against known field conditions, predefined objects, documented acquisition geometry and traceable processing configurations. It must be realistic because infrastructure-monitoring services are intended for operational environments where weather, access constraints, communication limitations, background clutter, sensor geometry and user interpretation all influence the final value of the service.

The campaign methodology therefore combines three complementary validation layers. The first layer is scenario-based validation, where the railway pilot use case is translated into observable campaign situations. The second layer is acquisition-based validation, where SAR, VIS and localisation data are collected in a real railway environment under defined flight and sensor conditions. The third layer is service-based validation, where the acquired data are processed, transferred or replayed, fused, visualised and assessed from the perspective of user-oriented railway infrastructure monitoring.

This approach makes the airborne campaign the link between technical verification and operational validation. The individual modules may be verified in laboratory conditions, and recorded data may be replayed in controlled environments, but only the airborne campaign can expose the complete chain to the combined constraints of field execution. The methodological objective is therefore not only to prove that the sensors can acquire data, but to demonstrate that the complete service workflow can proceed from scenario definition to user-facing information.

Translation of railway scenarios into campaign actions

The methodological starting point is the translation of railway pilot scenarios into executable campaign actions. The pilot scenario is based on the detection of obstacles or unexpected objects on or near railway tracks, especially cases that may be associated with infrastructure disruption, weather-related events or blocked track sections. In the campaign, each scenario must be represented through a controlled field configuration. This requires defining the operational meaning of the scenario, selecting a physical or simulated object that can represent it, specifying where the object must be placed relative to the track, and defining the expected service output.

The campaign methodology should use a scenario matrix with three linked dimensions. The first dimension is the railway-operational interpretation, such as a fallen tree, a vehicle-like blockage, displaced natural material, a radar-reflective surface or a radar-attenuating target. The second dimension is the experimental object, such as tree-like vegetation, a vehicle, turf strips, reflective panels, absorber panels or corner reflectors. The third dimension is the expected evidence type, including VIS visibility, SAR backscatter behaviour, georeferencing quality, tile association, anomaly interval generation and fused service output.

This translation step is essential because not every operational railway event can be reproduced directly in a safe and cost-effective field campaign. Events such as landslides, flooding, snow cover or structural damage may require representative objects rather than real hazardous conditions. The methodology therefore accepts simulation objects when they preserve the relevant sensing and processing characteristics needed for validation. The representativeness of each object must be documented, so that later interpretation does not confuse a controlled experimental proxy with the full operational phenomenon.

For future campaigns, the same logic should be applied before field execution starts. Each scenario should have a clear operational meaning, each experimental object should have a documented relation to that operational meaning, and each expected output should be linked to the service logic. In this way, the campaign remains a validation activity and does not become an unstructured collection of airborne data.

Test-site selection and relocation logic

The test-site selection methodology must ensure that the selected railway environment is suitable for repeated airborne acquisition, controlled field operations and user-oriented interpretation. A suitable site does not need to reproduce every feature of the original pilot environment, but it must preserve the essential characteristics required for the experiment. These include standard-gauge railway infrastructure or an equivalent railway geometry, realistic track surroundings, vegetation and built-environment context, level crossings or comparable infrastructure elements, safe access for field operations, and enough spatial extent to support repeated airborne acquisition with and without placed obstacles.

A campaign plan for an IIMEO-type service should also be robust to changes in test-site availability. The service logic should define which characteristics are essential and which are context-specific. Essential characteristics include railway geometry, safe access, the possibility of controlled obstacle placement, acquisition repeatability, availability of baseline and obstacle scenes, and the possibility of pilot-user assessment. Context-specific characteristics include the national approval route, the local infrastructure owner or operator, field access procedures and detailed flight planning constraints.

This distinction is important for future methodological use. If a campaign is relocated, the relocation does not automatically invalidate the validation logic. However, the reasons for the relocation and the suitability of the new site must be explicitly documented. The campaign should show that the practical validation logic remains anchored in the operational needs of railway infrastructure monitoring, even if the physical campaign environment differs from the originally planned pilot location.

Regulatory and safety preparation

Regulatory and safety preparation is an essential part of the validation methodology. It should not be treated as an administrative side process, because it determines what can be safely and legally demonstrated in the field. At methodological level, two approval domains must be addressed.

The first domain is aviation readiness. This includes aircraft operation, flight planning, airspace constraints, crew responsibility, sensor carriage, flight altitude, flight tracks, weather minima, repeated passes, communication procedures and contingency handling. The second domain is railway-site readiness. This includes permission from the infrastructure owner or responsible railway entity, confirmation that no unsafe rail operation will occur during obstacle placement, controlled access for the field team, operational separation from railway movements where needed, and verified removal of all objects after the campaign.

For any comparable future campaign, the minimum documentation should include the responsible aviation operator, the responsible railway infrastructure entity, the approved field-access window, the safety concept for personnel and objects, the communication chain during operations, and the sign-off confirming that the railway site was restored after the experiment.

The level of required documentation depends on the operational context of the selected test site. In more complex operational environments, additional documentation may be required, including coordination with aviation operators, formal safety concepts, defined communication procedures, and verification that the site is restored after the campaign. For an active railway corridor, this may also include more detailed coordination with the railway operator, definition of protected work windows, traffic-management constraints and stronger evidence that obstacle placement cannot interfere with safe railway operation.

Sensor preparation, calibration and campaign readiness

Before the campaign, the airborne demonstrator must pass a readiness sequence that covers both sensor operation and service-chain operation. Sensor readiness includes SAR and VIS configuration, mechanical integration, storage capacity, power and thermal checks, GNSS/INS availability, time synchronisation, data-recording verification and calibration status. For SAR, calibration and geometry checks are essential because obstacle detection and change interpretation depend strongly on viewing conditions, depression angle, swath coverage, platform motion and repeatability of acquisition geometry. For VIS, readiness depends on camera configuration, image timing, navigation-data alignment, georeferencing quality and the ability to support later orthorectification or map-based presentation.

Readiness also includes processing and service preparation. The campaign must confirm that the railway geometry, tile definitions, task descriptors, expected product descriptors and user-facing presentation logic correspond to the selected test site. It is not sufficient to acquire images over a railway line. The system must be prepared to transform observations into service-relevant railway products. Therefore, the go/no-go decision for the campaign should consider whether the acquisition plan, object-placement plan, data-recording plan, processing configuration and user-assessment plan are mutually consistent.

The campaign methodology distinguishes between live airborne execution and controlled replay-based end-to-end testing. During the flight, SAR data, VIS data, localisation data, status information and preview products can be acquired or transferred according to communication availability. However, full sensor data must also be recorded on board and transferred after flight to preserve data completeness. The recorded data can then be replayed through the on-board and ground-side processing chains under controlled conditions. This replay-based continuation of the campaign is not a substitute for the airborne evidence. It is the method that makes the airborne evidence reproducible, diagnosable and suitable for full end-to-end assessment when live communication conditions are not fully stable.

Campaign execution methodology

The campaign execution follows a staged sequence. The first stage is baseline acquisition. The railway section is observed without placed obstacles, creating reference data for the normal scene state. The second stage is obstacle placement. Objects are installed according to the predefined plan, with coordinates, orientation, size, object type and placement condition documented as ground truth. The third stage is airborne acquisition with obstacles in place. The aircraft repeats planned passes over the relevant railway corridor, acquiring SAR, VIS and navigation data under the defined geometry. The fourth stage is field restoration. All obstacles are removed, the railway area is inspected, and completion is documented.

This sequence enables direct comparison between normal and obstacle conditions. It also allows the processing chain to be assessed in relation to known field conditions. Repeated passes are methodologically valuable because they allow robustness to be evaluated against acquisition variability, platform motion and viewing geometry. Any deviation from the plan, including weather changes, incomplete coverage, communication gaps, sensor interruptions, field delays or altered obstacle positions, must be logged because such deviations directly affect interpretation of performance.

The airborne phase also tests the practical interaction between sensing and communication. In an IIMEO-type campaign, the demonstrator operates with SAR and VIS sensors and a communication subsystem. Live status information and preview products are useful for confirming that the aircraft is following the intended trajectory and that the system remains operational. At the same time, full data recording on board is necessary to prevent loss of evidence due to limited link availability. The method therefore combines operational realism with evidence preservation: live transfer is assessed where available, while recorded data ensures that post-flight analysis remains complete and traceable.

End-to-end processing and service generation

The campaign is designed to validate the complete service workflow from observation to user interpretation. During acquisition, the airborne segment collects sensor data and navigation data. On-board processing converts high-volume sensor information into compact, service-relevant outputs. In the railway pilot logic, the decisive output is not a raw image but a track-related indication of obstacle or anomaly likelihood. SAR and VIS processing can produce separate intermediate detections, which are then transformed into a common railway-oriented representation and combined through fusion. The resulting output can be represented as anomaly or obstacle intervals along railway-track geometry, with probability information attached to the relevant segments.

Data transfer connects the airborne and ground-side elements. In the operational concept, compact products, telemetry and task-related information are transferred to the ground segment, where the platform ingests, stores, refines and prepares the information for user presentation. Off-board refinement may include additional georeferencing checks, visualisation preparation, fusion review, storage of products and generation of map-based service outputs. The methodological criterion is continuity of the service chain. The campaign must demonstrate that the service can proceed from scenario definition, to acquisition, to processing, to transfer or replay, to off-board handling, to user-facing presentation.

The replay-based part of the campaign is especially important for evidence-based validation. It allows the same recorded airborne data to be processed repeatedly with controlled configurations. This supports diagnosis of missed detections, false alarms, georeferencing deviations, fusion inconsistencies and timing bottlenecks. It also allows comparisons between on-board and off-board implementations or between alternative configurations without requiring additional flights. In the methodological framework, this forms the bridge between airborne realism and reproducible verification.

User-oriented assessment

The pilot user is central to the assessment of operational relevance. The campaign is not successful only because sensor data were collected or because algorithms produced outputs. The decisive question is whether the produced service information is meaningful, interpretable and acceptable for railway-infrastructure decision-making. For this reason, the user-oriented assessment must be connected to the same scenario logic used for campaign planning.

The pilot-user assessment should consider whether the result identifies the correct railway segment, whether the obstacle indication is understandable, whether the probability or alert representation is suitable for operational use, whether the map or mobile display provides enough context, and whether the service output could support a decision such as targeted inspection, traffic interruption assessment or prioritisation of field checks. Feedback should be collected through structured questionnaires, guided review sessions, interviews or comparable documented methods.

The assessment criteria should be framed as methodological assessment criteria linked to the agreed pilot scenario, not as new formal requirements introduced after the campaign. Relevant criteria include successful acquisition of baseline and obstacle scenes, successful association of detections with railway geometry, generation of SAR, VIS or fused obstacle information, availability of results through the platform, usability of web or mobile visualisation, and pilot-user confirmation that the result form is operationally interpretable. Timing should be assessed from the service perspective, meaning the time from acquisition or task execution to availability of user-facing results, while clearly distinguishing live in-flight processing, post-flight transfer and replay-based processing.

Documentation and traceability requirements

Traceability is the main methodological requirement for campaign documentation. Each campaign action must be reconstructable after the flight, so that post-flight analysis can distinguish between field conditions, acquisition effects, processing behaviour, communication limitations and user-interface effects. The minimum documentation set should include the following categories as given in Table 4–1.

Table 4–1: The minimum documentation set for the flight campaign

Documentation category	Minimum information to be recorded
Campaign context	Campaign date, time, location, responsible partners, test-site description, railway section, field access constraints and execution timeline.
Railway and spatial reference	Railway-line geometry, regions of interest, tile definitions, map version, coordinate reference system and any changes made to the spatial input data.
Obstacle ground truth	Obstacle identifier, object type, dimensions, material, operational scenario represented, coordinates, orientation, track relation, tile association, placement time, removal time and responsible field team.
Ground evidence	Photographs before placement, after placement and after removal, together with notes on visibility, alignment and any deviations from the planned layout.
Regulatory and safety evidence	Aviation readiness, railway-site permission, access authorisation, safety briefing, field communication plan, exclusion conditions, restoration confirmation and deviation records.

Flight execution	Planned flight tracks, actual flight tracks, aircraft altitude, speed, attitude, pass direction, acquisition geometry, timing and deviations from the flight plan.
Sensor operation	SAR settings, VIS settings, calibration records, GNSS/INS status, time-synchronisation records, storage status, sensor start and stop times and instrument anomalies.
Environmental conditions	Weather, wind, cloud cover, precipitation, illumination, visibility, ground condition and any environmental factor affecting SAR or VIS interpretation.
Communication performance	Availability of the live link, transferred status data, preview products, interruptions, latency observations and recovery actions.
Processing evidence	On-board processing logs, off-board processing logs, fusion logs, software versions, model versions, configuration files, task identifiers, product identifiers and processing timestamps.
Data management	Raw data inventory, processed product inventory, checksums where applicable, storage location, access rights, replay configuration and product provenance.
User assessment	Web or mobile interface screenshots, user questionnaires, guided review notes, user comments, acceptance observations and interpretation of operational relevance.
Deviations and lessons	All deviations from the plan, their cause, their effect on interpretation, and the corrective or refinement action proposed.

This documentation converts the campaign from a one-time field event into a reusable evidence base. It enables the IIMEO consortium to repeat processing, audit results, explain deviations and derive refinements. It also supports future reuse of the methodology in other infrastructure-monitoring services, because the same traceability categories can be applied to roads, pipelines, waterways, dikes or power-line corridors with domain-specific adaptation.

Applied case: Dornum-Norden flight campaign

The IIMEO airborne campaign was performed in April 2025 in Northern Germany, over a railway environment between Dornum and Norden, with a focused obstacle-placement area near Lütetsburg as shown in Figure 4-1. This represented an adaptation of the original plan, which had foreseen execution in Serbia with the direct support of the ISR. The relocation did not change the methodological objective of the campaign. The railway-operational logic, scenario selection rationale, obstacle interpretation and service-assessment concept remained derived from the railway pilot use case and were developed by partner NIS with support from relevant partners. The practical execution, however, was adapted to the German test environment, including the local railway infrastructure, aviation context, field logistics and approval pathway.

Because of this difference between the planned IIMEO pilot and realised practical execution, the resulting IIMEO methodology combined pilot-user continuity with execution adaptability. The campaign remained anchored in the operational needs of railway infrastructure monitoring, as defined with the ISR, while the practical validation logic was transferred to a German railway environment that preserved the essential characteristics required for the experiment. These included standard-gauge railway infrastructure, realistic track surroundings, vegetation and built-environment context, level crossings or comparable infrastructure elements, safe access for field operations, and enough spatial extent to support repeated airborne acquisition with and without placed obstacles.

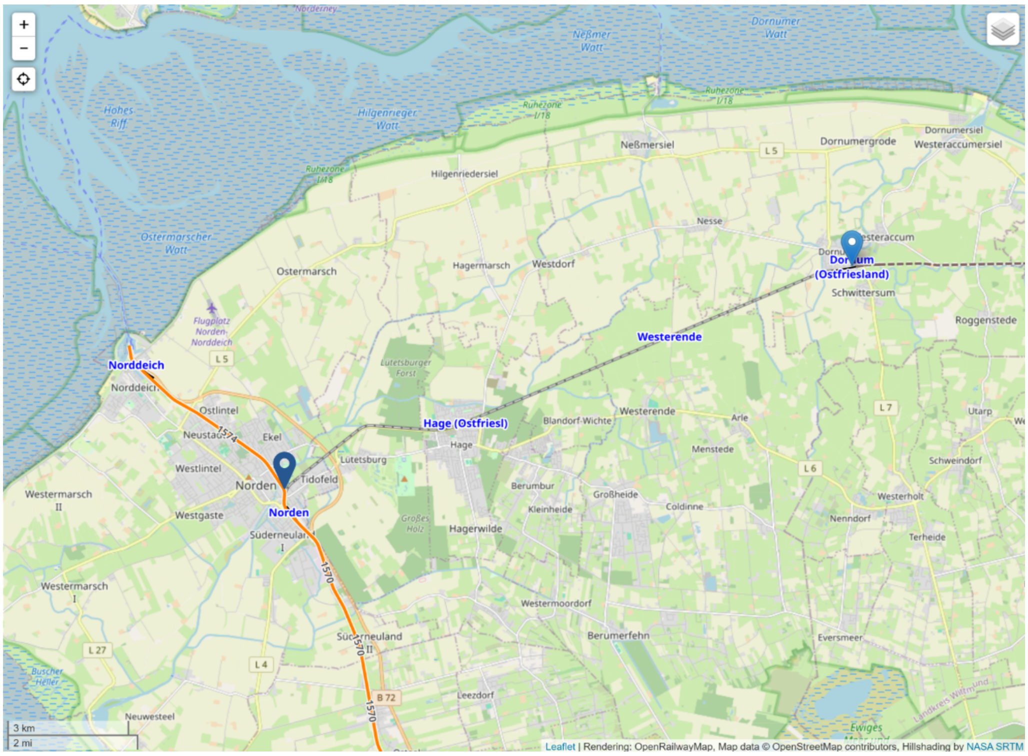

Figure 4-1: Overview of the IIMEO airborne campaign area

The selected German railway environment provided a suitable substitute for the originally planned Serbian railway setting because it allowed repeated airborne acquisition over a real railway corridor with varied surroundings. The line between Dornum and Norden includes standard-gauge railway track and passes through a mixed environment of settlements, agricultural land, vegetation and crossings. Such variability is relevant because infrastructure-monitoring services must not be validated only on idealised track sections. They must also be exposed to realistic background clutter, changing aspect angles, nearby structures and vegetation.

The focused obstacle-placement area near Lütetsburg (Figure 4-2) was selected for logistical and methodological reasons. Obstacle placement required coordinated field work, including safe access, transport of objects, positioning, verification, removal and documentation. The obstacles therefore had to be close enough to be handled within the available campaign time window, but sufficiently separated to avoid ambiguous association between detections and ground-truth objects. The use of predefined tiles or regions of interest, as showed in Figure 4-3, allowed the placed objects to be linked to the spatial logic of the processing chain. This is methodologically important because the IIMEO service output is not simply an image. It is a railway-oriented product that reports obstacle probability or anomaly evidence in relation to track geometry.

Figure 4-2: Overview of obstacle locations near Lütetsburg.

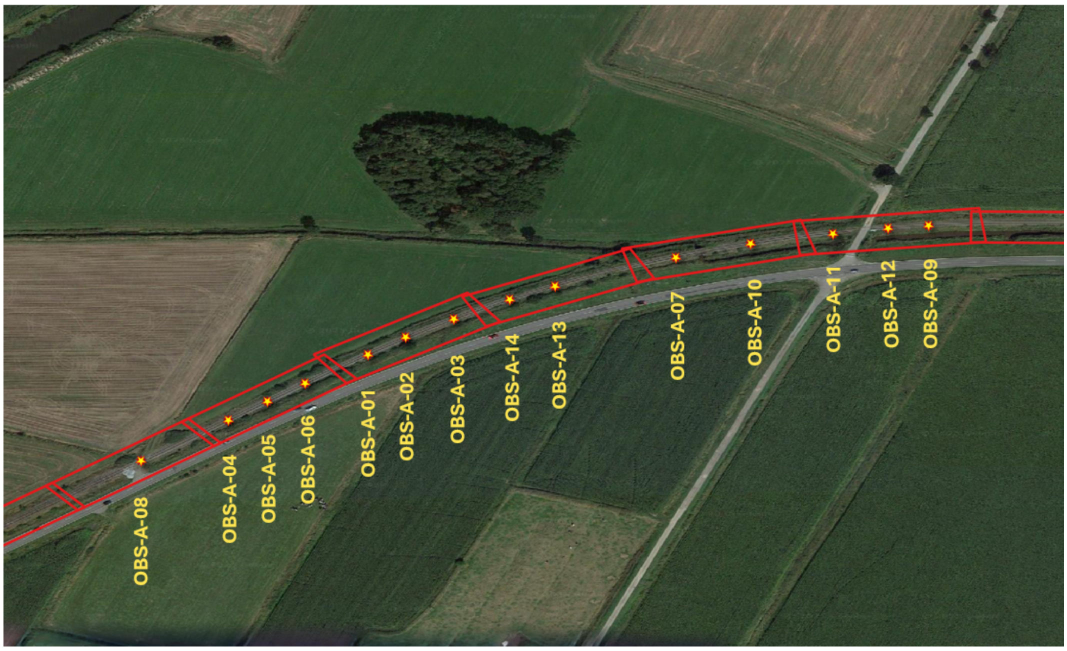

Figure 4-3: Overview of obstacle locations within tiles.

The obstacles used in the campaign (Figure 4-4) reflected the scenario-translation methodology described above. They included objects and materials representing different railway-operational interpretations and different sensing behaviours. Tree-like vegetation represented fallen-tree or vegetation-related blockage cases. A vehicle represented a vehicle-like blockage. Turf strips represented displaced natural material. Reflective panels represented radar-reflective surfaces. Absorber panels represented radar-attenuating targets. Corner reflectors provided strong radar reference targets. In this way, the campaign did not only test whether objects could be seen, but also whether different types of evidence could be associated with railway geometry and transformed into service-relevant outputs.

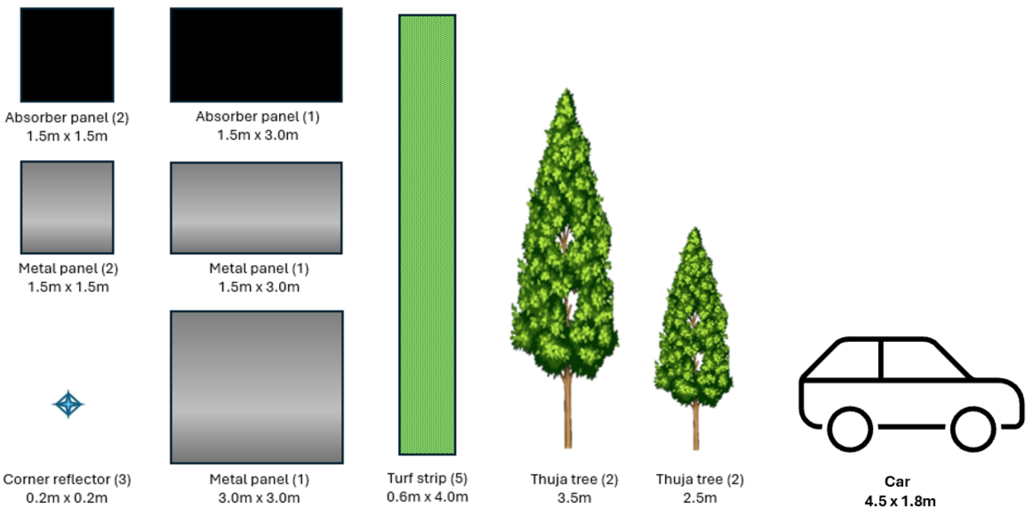

Figure 4-4: List of obstacles placed in the flight campaign.

The approval and safety preparation also followed the methodological logic, but was adapted to the selected German context. In the original Serbian plan, the approval route was connected to the cooperation between NIS and the ISR. After relocation to Germany, the approval logic had to be adapted to the local aviation and railway-infrastructure context. In the present case, documentation was limited to the identification of the responsible railway infrastructure owner, confirmation that no rail operations would take place during the experiment, and agreement on access to and use of the site. The selected railway track was privately operated, in good technical condition, and only used occasionally, mainly during weekends, which simplified the approval and safety coordination.

Before the campaign, the airborne demonstrator was prepared for operation with SAR and VIS sensors, localisation data acquisition and communication functions. The campaign therefore covered both sensor readiness and service-chain readiness. During the airborne phase, the demonstrator operated with SAR and VIS sensors and a communication subsystem. Live status information and preview products were used where available to confirm that the aircraft followed the intended trajectory and that the system remained operational. However, the full data were also recorded on board to preserve evidence completeness.

The campaign execution followed the staged methodology. Baseline scenes were acquired without placed obstacles. The obstacle-placement configuration was then prepared and documented in the focused area near Lütetsburg. Airborne acquisition was performed with the obstacles in place, and the aircraft repeated planned passes over the relevant railway corridor as shown in Figure 4-5, acquiring SAR, VIS and navigation data under the defined geometry. After the field experiment, the obstacles were removed and the railway area was restored.

Figure 4-5: Overview flight tracks.

The flight campaign also demonstrated why a combined live and replay-based methodology is necessary. The airborne phase tested the practical interaction between sensing and communication. In parallel, the original sensor output, including raw SAR radar data and full VIS image sequences, was recorded on on-board storage. After each flight, these recordings were transferred to ground storage to ensure data completeness and to make the data available for subsequent processing and system evaluation. This was especially important because live communication availability was limited and could not be treated as the only evidence path.

The recorded data were later replayed through the on-board and ground-side processing chains under controlled conditions. This allowed the complete on-board processing chain to be exercised as if it were operating in flight, while preserving repeatability and diagnostic control. The replay capability was therefore essential for debugging, validating algorithm behaviour and performing end-to-end integration tests without requiring additional actual flights. In methodological terms, the campaign combined airborne realism with reproducible verification.

The end-to-end service workflow was then assessed from observation to user interpretation. During acquisition, the airborne segment collected sensor data and navigation data. Processing converted the acquired data into service-relevant outputs. SAR and VIS processing produced intermediate detections, which were transformed into a common railway-oriented representation and combined through fusion. The resulting output could be represented as anomaly or obstacle intervals along railway-track geometry, with probability information attached to the relevant segments.

The user-facing service was assessed through the IIMEO platform and its web or mobile presentation logic (Figure 4-6). The service output was not interpreted as a raw sensor image, but as railway-related information intended to support infrastructure-monitoring decisions. The pilot-user assessment therefore considered whether the system identified the correct railway segment, whether the obstacle indication was understandable, whether the probability or alert representation was suitable, and whether the map-based display provided sufficient context for operational interpretation.

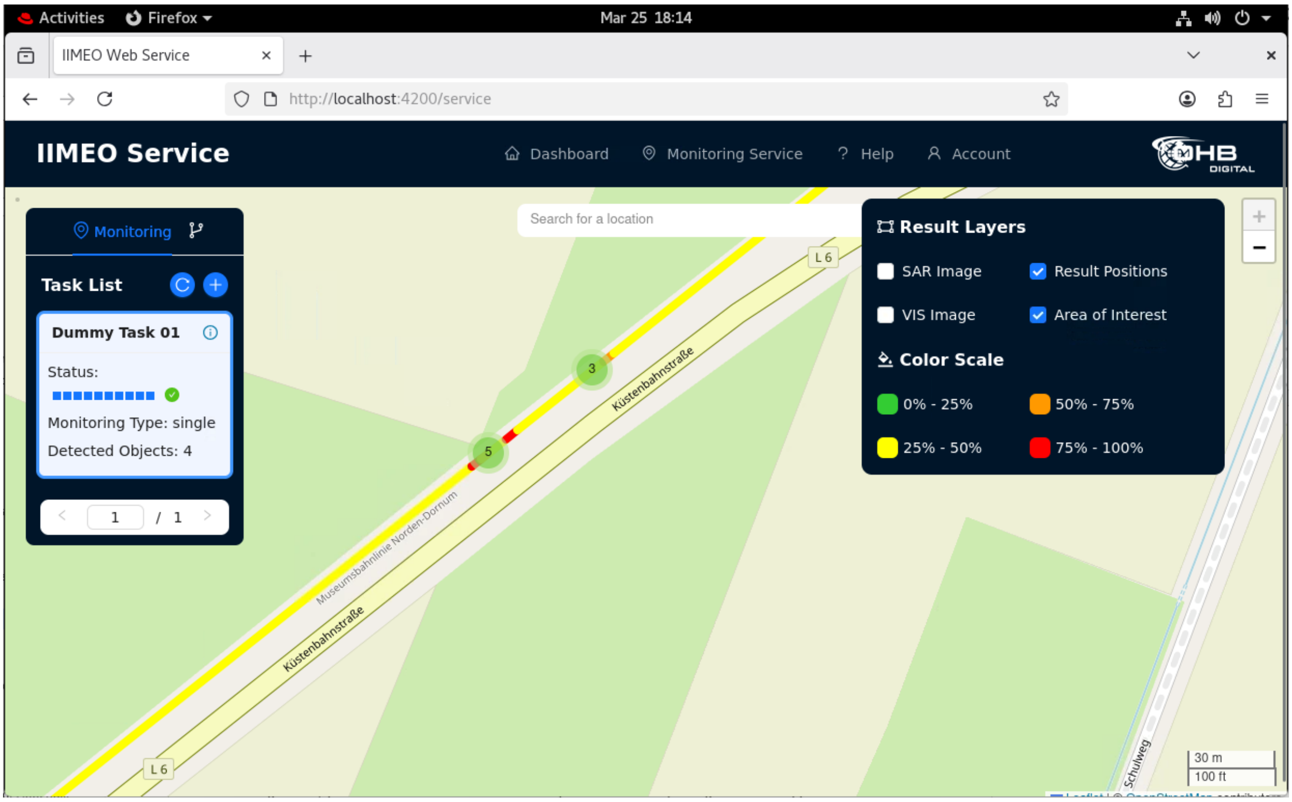

Figure 4-6: Screenshot of the IIMEO Web UI show different obstacle probabilities and indicators.

Methodological conclusions from the airborne campaign

The campaign demonstrates the methodological value of combining controlled scenario design with real operational constraints. Controlled obstacle placement provides known ground truth. Real airborne acquisition introduces platform motion, environmental variability, sensor-geometry constraints and communication limitations that cannot be fully reproduced in the laboratory. Replay-based processing allows the recorded airborne evidence to be used repeatedly for diagnosis and refinement. User-oriented assessment ensures that technical outputs are interpreted in the context of railway operations.

The Dornum-Norden flight campaign also shows that the IIMEO methodology can remain valid when practical execution differs from the original campaign location. The original plan was connected to the Serbian railway pilot environment, but the realised German campaign preserved the essential validation characteristics required for the system-level experiment. This confirms an important methodological principle for future applications: the service logic and validation structure must be stable, while the practical field implementation must remain adaptable to site availability, approval conditions and local infrastructure constraints.

The main methodological conclusion is that an IIMEO-type airborne campaign should be designed as an integrated service experiment. Its value lies not only in the collected SAR and VIS data, but in the traceable connection between user scenario, field configuration, airborne acquisition, processing chain, communication pathway, ground platform, result visualisation and operational assessment. This makes the Dornum-Norden campaign a decisive system-level experiment for the project and a reusable methodological reference for future infrastructure-monitoring services.

4.4 Extrapolation from demonstrator to future LEO operations

There are limits as to what we can learn from the airborne demonstrator about a constellation deployed in space, in particular we do not learn about operational aspects of a space-based constellation, which would ideally be used by multiple users with different infrastructure monitoring use cases. The sensor equipment used on the airborne demonstrator also cannot be transferred to space directly. To acquire the VIS images, the camera needs optics adequate for LEO-like distances of about 500 km instead of the 1 km altitude the demonstrator was flown at. The camera's sensor also has to be changed on one which can operate in space conditions, in particular under exposure to stronger radiation. For SAR, the situation is similar. A system operating in LEO would have to consider atmospheric path delay and distortion and earth rotation. Particularly in the Ka band attenuation along long path through the atmosphere becomes significant, such that the space-based SAR would a have demanding power requirements. In summary, there are a lot of issues to solve for the sensing system, further document in the roadmap to space [1] .

Nevertheless, the sensing equipment on the airborne platform has been chosen such that the data acquired using that platform is similar to what we expect to become available from satellites in the medium-term, once the issues mentioned above are solved. Hence, most of the data processing for both SAR and VIS, in particular the detection of obstacles on railway tracks and the feature-level fusion of data from SAR and VIS can likely be transferred to work on a spacecraft with little modification. The processing steps following the immediate post-processing of the sensor data are computed by using the Unibap iX10-100 on-board computer based on the Unibap iX10-101, which was selected for, e.g., the NASA HyTI-2 ACMES mission for on-board processing on a cubesat [12] , such that it currently appears that, aside of the immediate post-processing of the sensor data, the current processing steps can be transferred to an actual space-craft with only small modifications.

All the processing involving VIS image data or SAR data is done on the on-board processing unit, such that only the belief about obstacle presence over railway track has to communicated to the ground system. This belief is encoded as intervals of constant probability along curves describing the railway track. The encoding of this interval data is very small, only a couple of bytes per railway track. Thus, regarding the downlink from the on-board platform to the ground system, the issue to be solved before deployment in space is not the data rate but the availability of the data link.

If these technical issues are solved, however, IIMEO from space should yield results which are not worse than what we saw from the airborne platform.

The technical issues aside, there are other hurdles to overcome before we get IIMEO in space [12] . A constellation of multiple satellites with both VIS and SAR for a single use case for a single user will likely not be economically viable. Instead, the railway monitoring use case likely needs to be extended to similar monitoring tasks along linear infrastructure, e.g. roads or powerlines, to support more uses. Also, more than one user needs to be able to task the service and receive results from it, such that multiple tasks for different are executed concurrently by different satellites of the IIMEO constellation. Solving such “constellation sharing” scheduling issues will likely require significant effort before IIMEO can be used as an actual service beyond relatively small experiments.

4.5 Iterative refinement and optimisation workflow

IIMEO is a project during which an experimental combination of VIS, SAR and on-board processing hardware is used to find “things” on linear infrastructure, in the pilot use case obstacles on railway tracks, in real-time. For this, the sensing system was set up, processing algorithms were found and processing software developed, mainly to run on the on-board system, but partially also for the on-ground system. In addition to the sensing and processing, an end-to-end solution was supposed to be developed, involving tasking by the user, communication of task and results between the ground system and the on-board system, as well was presentation of results to the user.

The plan for this was mainly sequential, from specification, to algorithm prototyping and sensing system development, to on-board and on-ground platform development and integration of the prototyped algorithms, to the evaluation of the end-to-end system, with a period to optimize the developments near the end.

During such an effort, a lot of things can go wrong – and did. Moreover, the work packages representing the activities mentioned above have significant overlap because the activities are not entirely different from each other. For instance, one cannot develop the sensing system (work package WP2) without developments of the on-board system (work package WP3), because the on-board system drives the sensing system, which needs intermediate testing during development. Because we had more of these situations, we did algorithm prototyping, on-board system development and sensing system development concurrently, shifting focus as the project progressed.

We also did iterations and refinements early. For instance, we did not immediately proceed with the integration of the overlapping VIS+SAR sensor setup in its initial configuration into the OHB plane. Instead, we refined the viewing geometry until both SAR and VIS images appeared to be good enough for the obstacle detection task. This did cause delays, however, doing this type of refinement later would have resulted in the need to also change the mount for the OHB plane, which we would not have been able to afford both in terms of time and budget.

To account for such situations, we did refinement early for a working sensing system with a partial on-board system; for the processing chains; and also for the algorithms. Some algorithms changed significantly during work packages WP3 and WP4 after the original conclusion of WP2, which was originally concerned with algorithm prototyping. In particular, the SAR obstacle detection and the SAR+VIS fusion changed a lot after experience with actual data from the sensing system.

In the deliverables for the integration of algorithms into the on-board system and the on-ground system, respectively, we documented the then current state of the algorithm, i.e. the D3.5 already has the refined versions of the algorithms.

A little more abstractly, we developed, tested and refined for each “sub-project” within IIMEO before proceeding with activities requiring results from that sub-project.

This approach did come with costs though, for instance, there was no opportunity to evaluate the user interaction with results from “unrefined” components.

5 GUIDELINES FOR REUSE OF IIMEO SOLUTIONS IN OTHER SERVICES

The success of the IIMEO railway pilot serves as more than a singular proof-of-concept: it provides a foundational blueprint for a wide array of infrastructure-monitoring services. This chapter translates the lessons learned during the pilot into a reusable implementation handbook, designed to guide future service providers through the complexities of adapting established EO solutions to new operational domains.

While the IIMEO project has generated a robust general framework, standardizing how we define requirements, select data sources, and validate performance, it is essential to recognize that **reuse is never automatic**. Each new infrastructure class brings unique physical targets, specific anomaly signatures, and distinct integration constraints. Therefore, the goal of this handbook is to define a structured adaptation logic that ensures technical feasibility remains tethered to operational utility.

The logic of structured adaptation

To successfully transition IIMEO solutions to a new service, providers must follow a systematic path of inquiry. This section details a workflow that moves from high-level needs to granular deployment:

- **Operational need & value mapping:** before technical selection begins, the service provider must examine the specific *pain points* of the new infrastructure class. Is the goal preventive maintenance, disaster response, or regulatory compliance?
- **Sensing and processing trade-offs:** we examine how to re-evaluate data selection. While the railway pilot may have relied on specific SAR or multispectral resolutions, a power-line service might prioritize higher temporal frequency over spatial detail.
- **Functional partitioning:** the IIMEO solution allows for the reuse of modular processing blocks (e.g., change detection algorithms). Providers will learn which design dimensions can be transferred with little change and which require bespoke development.
- **Interoperability and deployment:** adaptation is only complete when the data flows seamlessly into the end-user's existing ecosystem. This includes addressing API compatibility, data latency requirements, and cloud-versus-edge processing constraints.

Transferability vs. Customization

The modularity of the IIMEO approach is its greatest strength. By decoupling the sensing logic from the infrastructure-specific analytics, large portions of the project's architecture can be repurposed. For example, the methodology for atmospheric correction of satellite imagery or the data-cleansing pipelines used in WP1 are largely domain-agnostic.

However, the last mile of the service, the definition of what constitutes a critical anomaly and how that information is visualized for a specific stakeholder remains a highly customised task. This chapter provides the diagnostic tools necessary to identify these *transferable constants* vs. *variable parameters*, ensuring that future service providers do not have to reinvent the wheel, but rather, know exactly how to realign it for a new road.

5.1 Service tailoring workflow for additional infrastructures

The Integrated IIMEO approach is designed to be modular, allowing its core logic to be adapted across diverse infrastructure domains. While the railway pilot provides a robust template, the methodology can be generalized to assets such as pipelines, power lines, dikes, roads, waterways, and disaster-prone zones. The transition to a new infrastructure class is not a *one-size-fits-all*-process. It requires a systematic re-parameterization of the underlying service logic.

The following workflow outlines the stepwise process for tailoring IIMEO to a new domain, ensuring that technical sensing capabilities remain strictly aligned with operational needs.

Operational context and value proposition phase

The workflow begins by identifying the operational problem and the stakeholder value proposition. For a new class, such as dike monitoring, the decision context shifts from obstacle detection to structural integrity and flood prevention. Stakeholders must define the economic or safety *pain points*, such as the cost of manual inspections over thousands of kilometres of pipelines or the risk of power outages due to vegetation encroachment.

Target and geometry definition phase

Once the value is established, the service must be re-parameterized for its specific **area-of-interest geometry**. While railways are linear, natural-disaster monitoring may require wide-area polygons. This phase defines:

- **Target objects:** e.g., pylons, embankments, or waterway locks.
- **Target anomalies:** e.g., moisture seeps in dikes, leaks in pipelines, or erosion on road networks.
- **Service mode and timing:** Defining whether the infrastructure requires periodic "health checks" or urgent ad-hoc alerts within the IIMEO < 60-minute target window.

Concept translation and design phase

These inputs are then translated into a technical initial concept. This includes selecting the optimal sensing geometry for the Ka-band SAR to avoid shadows on vertical structures, defining the On-Board Processing logic for specific change detection, and designing user products (e.g. push notifications vs. detailed heat maps). Experience from WP1 indicates that the underlying logic remains consistent across domains - be it tracking fallen trees on tracks or identifying debris in waterways - provided the AI models are trained on domain-specific baseline data.

Validation and documentation phase

The final stage involves establishing validation evidence through the demonstration campaign to verify that the tailored algorithms meet the required detection thresholds. It is a critical recommendation that all assumptions (e.g. atmospheric interference models), constraints (e.g. orbital revisit times), and expected benefits (e.g. 20% reduction in ground patrol costs) are documented and signed off by the end-user before full technical implementation begins. This ensures that the resulting service is not just technically feasible, but operationally indispensable.

5.2 Decision criteria for sensor and algorithm adaptation

This section should summarise the criteria for deciding whether the existing IIMEO sensor chain and algorithms can be reused directly or require adaptation. The text should distinguish between hardware-level reuse, processing-chain reuse, and model-level reuse. Direct reuse may be feasible when the new infrastructure remains a linear feature with similar scale, visibility conditions, and anomaly semantics. Adaptation becomes necessary when the target morphology, background clutter, viewing geometry, revisit need, or available labels differ substantially from the railway case. The section should therefore discuss when to adjust sensing geometry, when to retrain or tune anomaly detectors, when to revise the partitioning between on-board and off-board functions, and when the SAR-VIS fusion logic or map-based reasoning must be reformulated for the new service context.

5.3 Interoperability, data products and deployment considerations

The IIMEO system is divided into three conceptual blocks: on-board data acquisition hardware and software, on-board processing components, and on-ground processing components, which were developed differently and from whom different aspects can be learned.

5.3.1 Data acquisition hard-/software

Data acquisition in IIMEO comprises both hardware and software components. For image data, cameras and SAR are used. For localization, which is needed for geo-referencing and clipping, there are IMU+GNSS unit to obtain the position and orientation of the flight platform.

The data acquisition software consists of small programs, in our case ROS2 [5] nodes, which send the acquired data to the rest of the on-board system. In addition, these nodes also write data to disk for later replay, which greatly simplifies development and testing.

5.3.2 On-Board Processing System

The on-board processing system in IIMEO uses ROS2, a framework implementing most of the cross-cutting functionality needed by the on-board components, e.g. logging, data de-/serialization and message passing, configuration, and node life-cycle management.

We separated the functionality necessary for the on-board processing into relatively small programs, called nodes in ROS2 parlance. These nodes accept and produce data using ROS2’s infrastructure, which makes them easy to test independently using the same interfaces that are also used in the actual system. As a result, if the per-node tests look good, the node will very likely also work when integrated with the rest of the system.

Moreover, it is very easy to swap out a node with a different implementation of the same function. We make use of this when switching from our development environment to the on-board processing unit, which has different hardware for neural network inference acceleration. The actual on-board system therefore just runs a different node for railway track detection, which uses neural network inference, while the rest of the system remains unmodified.

To specify the communication between on-board system components, we did not write separate interface control documents (ICDs). Instead, the interfaces are formally defined using ROS2’s interface definition language, which is easy to read for humans and is also used to generate the data serialization and deserialization code.

5.3.3 On-Ground Processing System

For the on-ground processing system, see also section 3.5, we did not use a single framework implementing all cross-cutting functionality. Instead, we tried to achieve similar design characteristics using multiple off-the-shelf components.

A requirement we had for the on-ground software was that it must be possible to run it in “the cloud.” Practically, this means running on someone else’s computer, which we do not fully control. We handle this in the way that has become typical in recent years, namely by wrapping each on-ground component and its dependencies in a Docker container image. These container images can then be run on essentially any computer running Docker.

To pass data between the different on-ground components, we use a message broker, namely RabbitMQ, to implement the same publish-subscribe pattern that is also used in ROS2. The data passed among the different components is always text, i.e. JSON. We do not pass large binary data blobs between ground system components. Instead, we run an S3 server to store binary data, which is accessible to all on-ground components. Components can then load the data referenced in the JSON-encoded messages they receive via the message broker.

To structure the data passed between ground components, we defined two data structures: the ProductDescriptor, which describes data products, and the TaskDescriptor, which describes tasks for the system to carry out.

5.3.4 Communication between the On-Board and On-Ground System

Between the on-board system and the on-ground system there is a network link using IP. In theory, this would allow us to run a ROS2 node in the ground system and receive messages directly from the on-board system over this link. However, the link was established using LTE underneath, which turned out to be unstable, causing significant data loss.

Instead, we use a different approach: a ROS2 node on the on-board system receives and retains the data that is to be sent to the ground at some later point. Towards the ground system, this node exposes an interface different from ROS2's, accepting remote procedure calls (RPCs). Through these RPCs, the node responds with state information of the on-board system or with data resulting from on-board processing. This ground-system-facing facade of the on-board system is implemented using Apache Thrift [11] , which again allows to specify the data types exchanged as well as the RPC interfaces using an interface definition language, which is again easy to ready and used by Thrift to generate code both for accepting RPCs as well as data de-/serialization.

5.3.5 Communication between the ground system and external components

Communication between the ground system and external components uses the same ProductDescriptor and TaskDescriptor structures that are also used internally in the on-ground system. A REST interface is provided to create, delete, and query tasks and data products.

We implemented two clients using this interface: a web-based user interface and a mobile application. In addition, the interface can be used by third parties to develop their own integrations with the IIMEO system.

5.3.6 Recommendations

We have tried both specifying interfaces and data structures in separate documentation, for the Product- and TaskDescriptors, and formalizing them directly in the interface definition languages corresponding to tools we used to implement the interfaces. The latter approach appeared to be superior. During development we made changes to the details of the interface definitions. Having a single source both for the developer's understanding of the interfaces as well as the code using the interfaces removes the burden to synchronize changes. Incompatibilities introduced by modifications were also found very quickly and often automatically, e.g. by error messages from compilers. So if one is in a position that there are no (or not very many) external users of an interface, effort is probably better spent on making it as simple as possible to make changes to interfaces than trying to get the interfaces right on with the first shot.

Using such interfaces as boundaries between different, small programs each implementing different functionality, turned out to be very beneficial. These small components were easy to test and, more importantly, easy to replace. If ROS2 is used as a framework for a system, such as our on-board system, this approach is taken almost automatically, however, it was also not very hard to stick to the approach using a combination of other tools as we did for our on-ground system.

The approach, particularly using ROS2, also had the advantage that it allowed us to keep test cycles short. We did not manage to keep test cycles short for the deployment in the cloud system, such that a lot of manual work was required to get a change made to one of the on-ground components into the cloud infrastructure. This discouraged testing on the cloud infrastructure, which sometimes led to long debugging sessions to find mistakes which were made long before the deployment on the cloud system; those probably would have been found more quickly if the tests were more immediate.

6 CONCLUSIONS

This deliverable has consolidated the methodological knowledge generated during the IIMEO project into a reusable framework for designing, implementing, validating and adapting IIMEO-type Earth observation services for critical-infrastructure monitoring. The framework does not replace the technical specifications, prototype descriptions or verification reports produced during the project. Instead, it extracts the transferable logic behind them and organises it into a coherent service-development methodology. This methodology links user needs, sensor selection, reference-data preparation, functional partitioning, platform integration, validation, user interaction and service reuse into one end-to-end approach.

The railway pilot use case provided the empirical basis for the framework. Its operational problem, namely the timely detection of obstacles or abnormal conditions on railway infrastructure, made it possible to test the complete methodological path from operational need to service output. The pilot context showed that a future infrastructure-monitoring service cannot be designed only from the perspective of sensor capability or algorithm performance. It must begin with the user's operational decision context and then derive the required spatial accuracy, timing, observation geometry, data products, interface logic and validation evidence from that context.

A central conclusion of the project is that IIMEO-type services require early and explicit coupling between service logic and on-board processing constraints. In conventional downstream Earth observation approaches, data acquisition and later processing are often treated as separate phases. In IIMEO, this separation is not sufficient because near-real-time operation and large-area infrastructure monitoring create strong data-volume and latency constraints. The framework therefore treats functional partitioning between on-board and off-board processing as a fundamental service-design decision. Processing steps that reduce data volume and generate compact anomaly information should be placed close to the sensor when this supports timeliness and downlink efficiency, while the ground segment should provide refinement, storage, orchestration, visualisation and user-oriented service delivery.

The project also demonstrated that reference-data design is a methodological activity in its own right. Reference datasets, ground truth, obstacle scenarios and evaluation data must be planned so that they remain useful throughout algorithm development, integration, verification and campaign assessment. This requires controlled normal and anomalous scenes, documented obstacle properties, consistent spatial references, traceable metadata, versioned datasets and clear links between data samples and acceptance logic. Without such traceability, later performance statements cannot be interpreted reliably, and the connection between technical results and user relevance becomes weak.

The airborne campaign-based experimental evaluation was the decisive system-level validation step. It showed how a railway pilot scenario can be translated into campaign actions, including selection of a suitable railway environment, placement of representative obstacles, acquisition of baseline and obstacle scenes, execution of repeated airborne passes, recording of SAR, VIS and navigation data, processing through on-board and ground-side chains, and presentation of user-facing outputs. The relocation of the campaign from the originally planned Serbian environment to Germany also demonstrated the adaptability of the methodology. The railway-operational logic and scenario interpretation developed with the pilot perspective were preserved, while practical execution was adapted to the German test environment.

The airborne campaign further confirmed that field validation and laboratory replay testing are complementary, not interchangeable. Airborne acquisition introduces real platform motion, viewing-geometry constraints, environmental conditions, communication limitations and field logistics. These factors cannot be fully reproduced in a laboratory. At the same time, replay-based testing is necessary because it allows recorded flight data to be processed repeatedly under controlled conditions, supporting debugging, comparison of configurations, timing analysis and reproducibility. The IIMEO methodology therefore combines field realism with controlled repeatability.

The integration experience showed that modularity, stable interfaces and explicit product provenance are decisive for future service scalability. The use of task descriptors, product descriptors, metadata, message-based orchestration, storage back-ends and map-oriented output products allowed the system to connect sensing, processing, fusion, platform services and user interfaces in a traceable manner. This modular approach also supports later adaptation, because individual processing chains, data products or user-facing components can be modified without redesigning the complete system.

The user-oriented assessment confirmed that operational relevance is not equivalent to technical correctness alone. A service output is valuable only if it can be interpreted by the target user and connected to a realistic operational decision. For the railway pilot, this means that obstacle probabilities, intervals, alerts, maps and visual overlays must be understandable in the context of railway inspection, disruption assessment and prioritisation of field response. Pilot-user feedback is therefore not an optional validation add-on. It is part of the evidence that the service concept is fit for purpose.

The methodological framework also clarifies the boundary between airborne demonstrator evidence and future LEO-based operation. The airborne demonstrator provides strong evidence for the processing partitioning logic, multi-sensor concept, service workflow, data-product concept and end-to-end integration approach. However, it does not fully reproduce the orbital environment, constellation scheduling, satellite downlink availability, space-qualified payload constraints or multi-user operational setting. Future space deployment therefore requires analytical extrapolation, additional engineering adaptation and further validation under mission-specific assumptions.

Another key conclusion is that reuse of IIMEO solutions in other infrastructure domains is feasible but not automatic. The railway pilot provides a strong template for linear-infrastructure monitoring, especially where the service requires detection of localised anomalies along predefined corridors. Nevertheless, each new domain, such as pipelines, power lines, roads, waterways, dikes or disaster-affected infrastructure, requires re-parameterisation of the service logic. Target objects, anomaly types, background clutter, required response times, observation geometry, user products and acceptance criteria must be redefined with the relevant stakeholders before technical implementation begins.

The framework therefore recommends a structured adaptation workflow for future services. The process should begin with operational need and value definition, continue with target and anomaly specification, proceed through sensor and processing-chain selection, and then define reference data, validation evidence, user products and deployment interfaces. This workflow prevents premature technology selection and ensures that the final service remains aligned with the user's operational decision environment.

Overall, the IIMEO project has shown that near-real-time infrastructure monitoring by combined SAR and VIS sensing requires a complete service methodology rather than a set of isolated technical components. The decisive contribution of D4.8 is the consolidation of this methodology. It provides a project-derived framework for future services that need to combine high-resolution observation, on-board processing, efficient data transfer, off-board refinement, platform integration and user-oriented service delivery. The resulting framework can support future exploitation of IIMEO results, guide adaptation to additional infrastructure-monitoring domains and provide a methodological basis for the transition from airborne demonstrator evidence toward future operational Earth observation services.

7 REFERENCES

[1] IIMEO Consortium, „IIMEO-OHBDC-D-0042 Roadmap to space and for future Earth observation services," 2026.

[2] IIMEO Consortium, „IIMEO-FHR-D-0007_01_Sensor Systems Reference Data and Evaluation Datasets," 2023.

[3] IIMEO Consortium, „IIMEO-OHBDC-D-0010_03 Specification of On-and Off-Board Algorithm," 2023.

[4] IIMEO Consortium, „IIMEO-NIS-D-0038 Airborne based demonstration of near real-time service operation," 2026.

[5] Open Robotics, „ROS," [Online]. Available: https://www.ros.org/.

[6] The Linux Foundation, „PyTorch," [Online]. Available: https://pytorch.org/.

[7] IIMEO Consortium, „IIMEO-FBK-D-0029_01: Full Prototype and Verification of On-and Off-board algorithm," 2026.

[8] IIMEO Consortium, *IIMEO-ATB-D-0008_01: IIMEO Algorithms Sensor and System Concept,* 2023.

[9] Object Management Group (OMG), „DDS - Data Distribution Service," [Online]. Available: https://www.omg.org/spec/DDS/.

[10] VMware, „RabbitMQ," 2024.

[11] "Apache Thrift," [Online]. Available: https://thrift.apache.org/. [Accessed 28 05 2024].

[12] IIMEO Consortium, *D5.10: Roadmap to Space and for future Earth Observation Services,* 2026.

[13] IIMEO Consortium, *Consortium Agreement, IIMEO-OHBDC-CTR-0002, 01,* 2022.

[14] European Health and Digital Executive Agency (HaDEA), *Instantaeous Infrastructure Monitoring by Earth Obeservation (101082410 - IIMEO), IIMEO-EC-CTR-0001, 01,* 2022.

8 APPENDIX

Appendix A Abbreviations & Nomenclature

Abbreviation	Meaning
AI	Artificial Intelligence
API	Application Programming Interface
ATB	Institut Für Angewandte Systemtechnik Bremen GmbH
AWS	Antwerp Space N.V.
DDS	Data Distribution Service
EC	European Commission
ECSS	European Cooperation for Space Standardization
EO	Earth Observation
FBK	Fondazione Bruno Kessler
FHR	Fraunhofer-Institut für Hochfrequenzphysik und Radartechnik FHR
GIS	Geographic Information System
GNSS	Global Navigation Satellite System
HaDEA	European Health and Digital Executive Agency
HyTI-2	Hyperspectral Thermal Imager 2.0
ICD	Interface Control Document
ID	Identifier
IDL	Interface Definition Language
IIMEO	Instantaneous Infrastructure Monitoring by Earth Observation
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
IP	Internet Protocol
ISR	Infrastructure of Serbian Railways
JSON	JavaScript Object Notation
KPI	Key Performance Indicator
LEO	Low Earth Orbit

Abbreviation	Meaning
LTE	Long Term Evolution
NASA	National Aeronautics and Space Administration
NIS	University of Nis
OHB DC	OHB Digital Connect GmbH
PCSW	Payload Control Software
PU	Public
PUS	Packet Utilisation Standard
REST	Representational State Transfer
ROS2	Robot Operating System 2
RPC	Remote Procedure Call
RTD	Research and Technology Development
S&T	Science and Technology
S3	Simple Storage Service-compatible object storage
SAB	Security Advisory Board
SAR	Synthetic Aperture Radar
SEN	Sensitive
TMTC	Telemetry and Telecommand
TRL	Technology Readiness Level
UAV	Unmanned Aerial Vehicle
UI	User Interface
UX	User Experience
VIS	Visible Light
WP	Work Package