Title	**Optimized IIMEO Solutions**
Project Name	IIMEO – Instantaneous Infrastructure Monitoring by Earth Observation
Project Number	101082410
Deliverable number	D4.9
Document Number	
Issue / internal Revision	01 / 11
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

This documents briefly discusses the components developed and integrated into the IIMEO systems right before the final on-site trial of the integrated on-ground- and on-board-system.

The SAR and VIS processing components evolved during the project, particularly with regard to SAR, and their integration was completed at a late stage. As a result, the solutions documented in the previous technical deliverables already include optimizations and reflect the latest implemented processing approaches.

Individual methods and components have been evaluated in earlier deliverables, including D2.5 [1] for algorithms, D3.5 [2] for on-board processing, and D4.5 [3] for on-ground processing. Building on these results, the overall system has been assessed in D4.6 and D4.7 [4] , with a focus on the integration of SAR and VIS processing, the interaction between on-board and on-ground systems, and the user interfaces.

In order to provide a system overview even without access to sensitive IIMEO documentation, this deliverable lists and briefly describes the optimized IIMEO solutions, considering hard- and software and their integration, as they were used for the final on-site trail.

1.1 Structure of the document

The document is structured as follows. Section 2 summarizes the overall IIMEO system architecture and its main components. Section 3 describes the sensor system, including the SAR and VIS sensors used. Section 4 presents the on-board processing solutions, covering both SAR and VIS data processing chains as well as their integration. Section 5 describes the on-ground system, including data transmission, processing components, and user interfaces. The emphasis throughout is placed on those elements that have not been described in detail in publicly available previous deliverables.

2 IIMEO SYSTEM ARCHITECTURE

As already laid out in [5] , the IIMEO system is overall structured as shown in Figure 2-1.

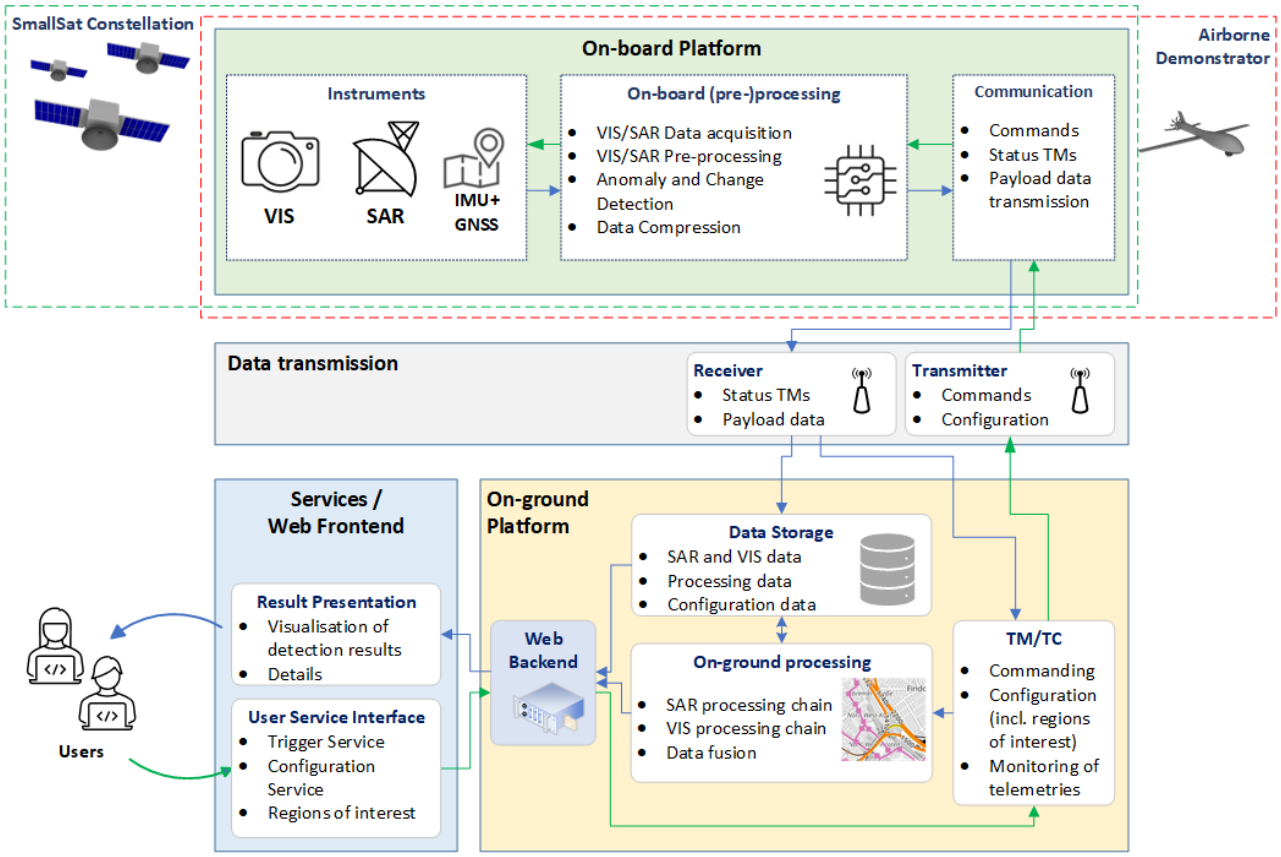

Figure 2-1 System architecture from[5] .

The on-board platform includes the SAR and VIS instruments, the on-board processor with obstacle detection algorithm implementations, and the communication hardware. The on-board processor is used to do most of the processing already before transferring data to ground to avoid data transfer bottlenecks that would be created by transmitting raw SAR or VIS image data.

During the flight campaign, with the on-board system installed on the airborne demonstration platform, the data transmission block represented essentially an IP over LTE connection. LTE connectivity was found to be unreliable in the region where the airborne demonstrator was operated. When the on-board system is operated in the laboratory and not on the demonstrator platform, the network is replaced by a regular IP over Ethernet network.

The on-ground platform complements the processing steps performed in the on-board platform, if required. It includes additional data processing, data storage and a web backend that implements the interface to the user services. The on-ground platform has been implemented as a cloud-based prototype platform, using custom or commercial/open-source tools for data distribution and service provision.

The services/web frontend includes all user related services, mainly services for defining and starting monitoring tasks, and reception and presentation/visualisation of the monitoring results. In particular, a region of interest can be selected, for which the monitoring is performed.

For each of these aspects, solutions were developed during the IIMEO project. From the top to bottom of Figure 2-1, we developed a sensor system discussed in chapter 3, provided on-board processing hard- and software discussed in chapter 4, both being part of the on-board platform in Figure 2-1. The ground system, discussed in chapter 5 covers the services and on-ground platform boxes of Figure 2-1, with the on-ground platform itself being discussed in section 5.1 and the front-end in section 5.3.

3 SENSOR SYSTEM

The ability to monitor ground surfaces with airborne and spaceborne sensors is very important in many applications, e.g. the detection of changes caused by natural disasters. Sensors in space are very accurate and can provide continuous monitoring of large areas. Unfortunately, satellites offer only very limited space and electrical power for the payload, which sets the parameters for the development of a space-based sensor. A cost-effective solution for Earth Observation is the use of electro-optical sensors, which are lightweight and require little power, but can only be used in daylight and good weather conditions. The use of active sensors emitting electromagnetic waves with millimetre or centimetre wavelengths, combined with synthetic aperture radar (SAR), provides high-resolution images in all weather conditions, day and night.

3.1 Synthetic Aperture Radar (SAR)

The SAR system used in IIMEO is the MIRANDA-35 SAR sensor developed at Fraunhofer FHR. It is an advanced, state-of-the-art design that enables high-resolution, real-time imaging in many applications. The use of the Ka-band (35 GHz) makes the sensor sensitive to small structures compared to lower frequency ranges; road textures and obstacles on railway tracks can therefore be detected. An advanced frequency modulated continuous wave (FMCW) generator allows a radar bandwidth of more than 2 GHz, which corresponds to a distance-independent resolution of a few centimetres in both the range and cross-range directions. A very high image contrast is achieved by the high signal to noise ratio of the heterodyne receiver.

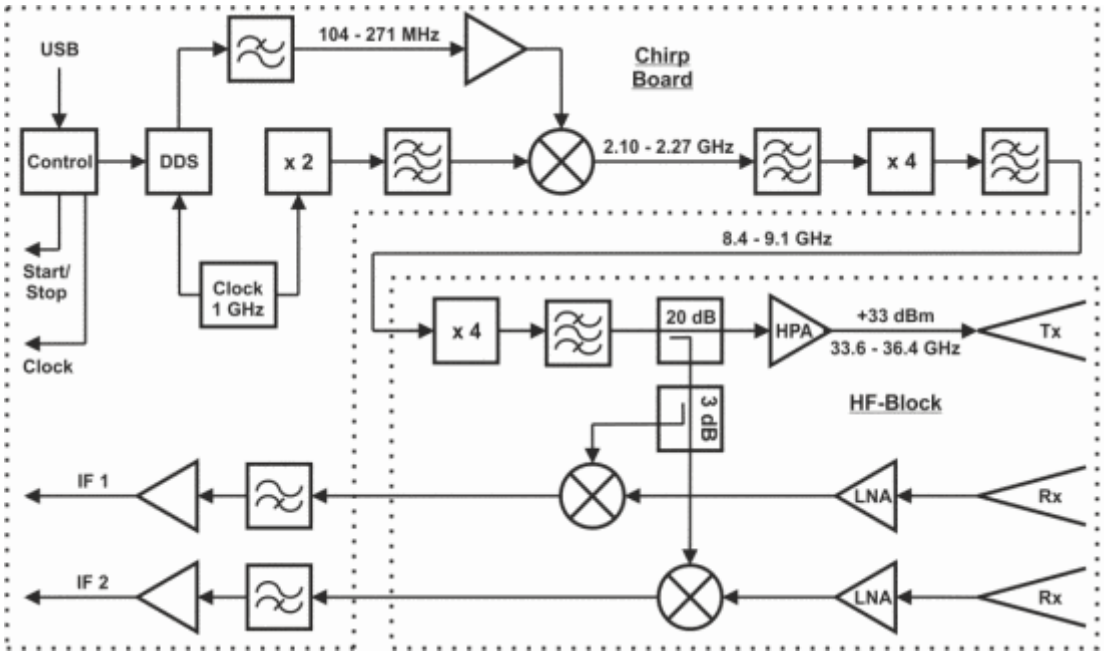

Figure 3-1: Block diagram of the MIRANDA-35 radar frontend

The quality of the radar measurements using an FMCW system depends directly on the frontend components used. The linearity and phase stability of the generated signal (chirp) at the carrier frequency play a decisive role, as it is used both for transmission and down-conversion. The block diagram of the MIRANDA-35 radar frontend is presented in Figure 3-1. The core component of the chirp generator is a Direct Digital Synthesizer (DDS) that generates a signal in the MHz range. The DDS is controlled by an ultra-stable master clock that also drives other frontend components. This technique ensures the coherency within a single chirp as well as between sequenced chirps, enabling high resolution SAR processing. The output spectrum of the chirp generation module is between 8.4 GHz and 9.1 GHz. This signal is then passed to the frequency multiplier-by-four and finally amplified to +33 dBm (2 W). A small portion of the transmitted signal is passed to the dual-channel heterodyne receiver, where it is mixed with the signal from the received radar echo to obtain an intermediate frequency (IF) in the lower MHz region. The IF signal is then digitized, stored and processed in the high-performance backend, which was developed specifically for MIRANDA 35. It enables a quick-look capability by using an advanced real-time, on-board SAR-processor. In small aircraft, the processor must cope

with unstable flight conditions, which requires a highly accurate inertial measurement unit (IMU) and advanced correction algorithms. This issue is probably not applicable in a satellite application.

In the IIMEO project the MIRANDA-35 system was used as a flight demonstrator. The frontend and backend were modified so that they could be installed on the aircraft used for the flight trials and met the project requirements. For real-time data processing reasons, the resolution was limited to 15 cm x 15 cm, which requires a radar bandwidth of 1000 MHz. The real-time SAR image stream (Level-1) including geodata is made available via USB interface for further processing steps.

In the state of development used in the flight trials, the radar frontend has a volume of about 20 x 40 x 30 cm3 and a mass of about 5 kg. A power of 2 W is available at the output of the amplifier. The slotted wave guide antennas have an opening angle of 3.3° in azimuth and 15° in elevation resulting in an antenna gain of about 27 dBi.

3.2 RGB Cameras (VIS)

The IIMEO VIS sensor system uses synchronized, high-resolution nadir and oblique PhaseOne cameras, integrated with GNSS/INS and ROS 2, to acquire geo-referenced optical imagery of railway infrastructure that supports obstacle detection and fusion with SAR data.

Table 3-1: Cameras in the IIMEO sensor system

	Oblique Camera	Nadir Camera
Type	PhaseOne iXM-50	PhaseOne iXM-100
Resolution	8280 x 6208 Pixel	11664 x 8750 Pixel
Pixel Pitch	5.3 µm	3.76 µm
Sensor Area	43.9 mm x 32.9 mm	43.9 mm x 32.9 mm
Focal Length	80 mm	35 mm
Angle of View	30.4° (long); 23° (short)	63° (long); 49.4° (short)

In addition to the SAR, there are two cameras mounted in the IIMEO sensory system, one obliquely looking camera and a nadir camera. The nadir camera is mounted in a separate wing pod. Its field of view does not overlap with the SAR, however, it could be used to acquire nadir images of railway tracks, avoiding occlusions by buildings or vegetation.

The more interesting camera is the obliquely looking camera. It was mounted in the same wing pod as the SAR sensor and aligned to look approximately in the same direction as the SAR. Hence the pair of the oblique camera and the SAR would be what we speculated to be the ideal setup on a satellite, providing data from the same location captured at the same time, allowing to fuse SAR and VIS data from the same overflight. Some details of the cameras are shown in Table 3-1.

Because of the oblique viewing direction at 50 degrees depression angle, the ground sampling distance varies inside a single image between approximately 10 cm and 40 cm. The final depression angle was concluded after multiple iterations of changing the depression angle to draw a good compromise between the ground sampling distance in the far-away regions of the VIS images and the quality of railway track images in SAR images.

Both cameras are triggered using the pulse-per-second signal acquired via GNSS. Originally, we planned to acquire images at 0.1 Hz, but increased that frequency to 0.5 Hz. This allows both experiments with temporally dense data and to drive the on-board system at the original frequency by dropping images. Acquiring data with at least 0.1 Hz makes sure images with significant overlap, more than one third, are acquired. We use the information from overlapping images to estimate the altitude above ground.

For time synchronization between the navigation system and the camera, the pulse-per-second signal is used to maintain the time delta between two image acquisitions. The absolute synchronization of the camera with the localization system is done by computing the rate of turn of the plane from the image data and matching it to the change of orientation as recorded by the GNSS+IMU system. Computing the clock synchronization in this manner avoids the need to account for all the delays in the startup sequence of the sensor system.

For camera calibration, we use a planar grid of circles to determine the intrinsic camera parameters, i.e. focal length, location of the camera centre, and image distortion. For the cameras' poses with respect to the flight platform, we neglect errors in the positioning and take the position from construction and estimate only the relative orientation by aligning the axes of turns observed during flight by both the camera and the GNSS+IMU combination.

4 ON-BOARD PROCESSING

The onboard system is structured by using the “Robot Operating System” ROS2, as described in [5] . ROS is a set of libraries and tools to build robotics applications, which typically include our technical use cases, e.g., running and querying sensors, running processing chains of multiple, possible concurrent processing steps, communicating the processing results as well as collecting telemetry and logging information.

Each data processing program runs as an independent node. Data exchange between nodes follows the ROS2 publish–subscribe pattern. Nodes publish their outputs on named topics, and subsequent nodes subscribe to those topics to receive inputs. Publishers and subscribers do not communicate directly and do not need to know each other. The ROS2 middleware manages data transfer, including communication across different computers over a network.

Sensor nodes publish messages when new data is available, which triggers further processing. Processing nodes receive messages through subscriptions, potentially buffer them until execution conditions are met, and then publish results to their output topics. Message delivery and, in most instances also buffering, are handled by ROS2 rather than by the node implementations themselves.

There is one exception form communicating using the publish-subscribe pattern. The rate we acquire localization data at is much greater than the image acquisition rate. Because it is not known exactly of which time the plane's localization is needed, localization data is not published—most of the published messages would be ignored anyway. Instead, the communication in this case happens via a service which can be queried with a timestamp on demand.

Since ROS2 handles message passing over computer boundaries via a network, we can run assign nodes to different computers. We make use of this by running the VIS image acquisition nodes and the early processing steps including geo-referencing on an image data acquisition computer, the SAR acquisition and image formation on a SAR data acquisition computer and the main parts of the processing, including the obstacle detection in both SAR and VIS images on a dedicated on-board processing unit equipped with accelerators for neural network inference.

4.1 On-Board Processing Hardware

The computing power required to execute the IIMEO software is composed of three computers. Two of those are dedicated sensor payload preprocessing computers, described in their current configuration in deliverable D3.1 [6] . No optimization effort has been spent on these two computers as there are no important gains to be expected at this stage. The third computer (Unibap ix10-100) is a dedicated hardware accelerator for the neural networks performing the anomaly detection tasks for both the SAR and VIS sensor pathways. The current hardware configuration of this computer has changed compared to the state reported previously in deliverables D3.1 [6] and D3.2 [7] . An additional Myriad VPU has been added to the system via the USB interface. Both Myriad VPUs service the VIS sensor pathway neural network. This is because the design for this neural network is much more costly compared to the network dedicated to SAR processing. It is likely that there is a lot of optimization work that could be done on the designs of these networks, but this falls out of the scope of the IIMEO project due to programmatic constraints. Therefore, it should not be taken as a rule that processing for the VIS sensor pathway is always going to be much more demanding than the processing for the SAR sensor pathway. The dual VPU hardware accelerators are more effective at processing the VIS sensor data with the neural network, as can be expected. Originally, the SAR neural network was intended to run on the VPU, but it is so much less costly to run that this is currently operated via the CPU.

4.2 On-Board Processing Software

The on-board processing software is split into two almost independent processing chains, one processing SAR data and the other processing VIS image data. We present the IIMEO solutions for the former in section 4.2.1 and for the latter in section 4.2.2.

4.2.1 SAR Data Processing

The SAR image processing is structured using a collection of ROS2 nodes. The first part is devoted to the generation of sensed SAR images, while the second part performs obstacle detection by comparing received images against reference ones from the database.

4.2.1.1 Data acquisition and geo-referencing

4.2.1.1.1 iimeo_sar_processor_lib

This library is set up as a ROS2 project and contains two packages:

1. iimeo_sar_processor: The main SAR processor ROS node library
2. interfaces: Defines the message format for processed tile (SAR_IMAGE_TILE) images and IMU data (SAR_IMU)

4.2.1.1.2 iimeo_sar_processor_main

This is the repository of the actual FHR SAR processing ROS2 node. It contains three packages:

1. iimeo_sar_processor: The main SAR processor node
2. iimeo_sar_detection: An optional node that can be used for debugging the ROS2 output of the iimeo_sar_processor node. It receives the tiles from the iimeo_sar_processor and stores them as an image.
3. interfaces: Defines the message format for processed tile (SAR_IMAGE_TILE) images and IMU data (SAR_IMU)

4.2.1.2 Obstacle detection

4.2.1.2.1 sar_database

The SAR database is provided as a ROS2 service which receives information from the image recorded on the current processed tile and provides the corresponding reference image for comparison and inference of presence of obstacles. The call for the query is performed directly from the change detection node.

4.2.1.2.2 sar_change_detection

The SAR change detection node encompasses the core processing tools to perform obstacle presence inference. The node provides inference procedure from a convolutional AutoEncoder (implemented in Keras) trained on project data. Further processing is done to transform the result of the neural network prediction step into a text message to be sent through the channel for the fusion node.

4.2.2 VIS Data Processing

The VIS image processing is structured using a collection of ROS2 nodes forming a graph of processing steps, which is almost a chain but not quite. ROS2's rendering of that graph – with minor modifications to make it fit on a page – is shown in Figure 4-1. We can mentally sort the nodes into four groups, the data acquisition nodes in subsection 4.2.2.1, the geo-referencing nodes in subsection 4.2.2.2, the obstacle detection nodes in subsection 4.2.2.3, and finally the fusion nodes, which consider in the following section 4.2.3. The processing steps corresponding to the nodes are described in [2] .

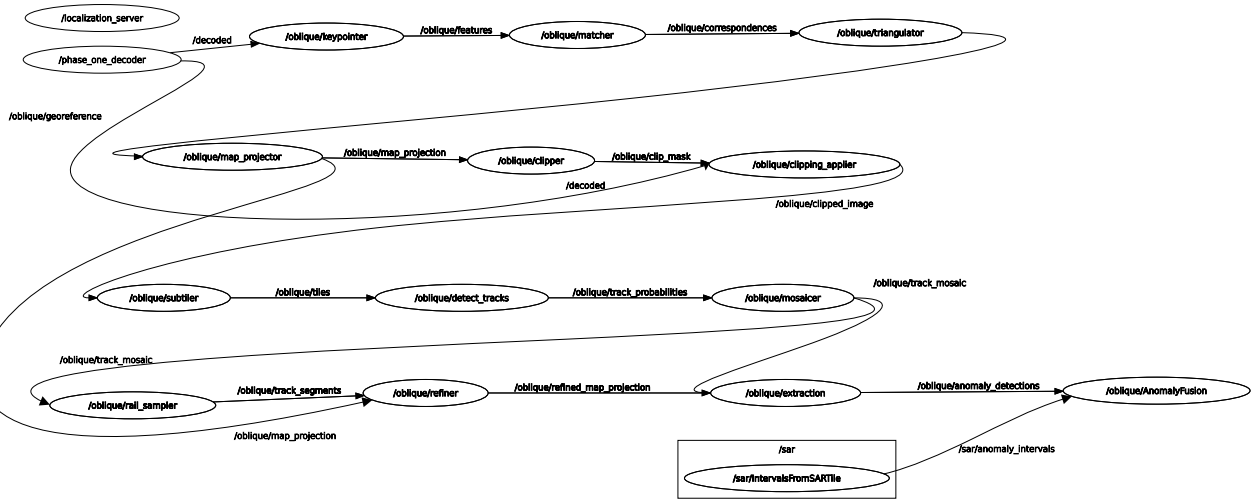

Figure 4-1: On-board system's ROS2 node graph with topics.

4.2.2.1 Data Acquisition

4.2.2.1.1 phase_one_decoder

The phase_one_decoder converts raw images from PhaseOne cameras into formats usable by the VIS processing chain. It subscribes to images encoded in PhaseOne's proprietary IIQ format, decodes them and sends them as numbered images in a standard ROS2 image representation. These decoded images are published as the first processing output of the VIS chain.

4.2.2.1.2 localization_server

The localization_server stores time-stamped position and orientation data from the GNSS/IMU subsystem. It provides the "localization_service", a ROS2 service other nodes can query using a timestamp, the "triangulator" node currently being the only user of that function. In response, it returns the platform pose closest to the requested time, expressed as position in WGS84 coordinates and orientation in an east-north-up frame at that position.

The data flow out of the localization_server does not follow the publish-subscribe pattern, which is why it appears to be disconnected from the other nodes in Figure 4-1, which only shows the data flow organized via the publish-subscribe pattern. The reason is that localization data is acquired with a frequency much greater than the image acquisition frequency and the localization_server cannot know in advance of which points in time the plane's localization will actually be of interest. So, it is queried when localization information is needed by other nodes.

4.2.2.1.3 ohb-ros2-phaseone-acquisition / ohb-phase-one-ros-iiqplayer

The node ohb-ros2-phaseone-acquisition interfaces with a PhaseOne camera, acquires images during flight, and publishes them into the ROS2 system so that the VIS processing chain can treat the camera as a live sensor. The images are published in PhaseOne's proprietary IIQ format, such that the only subscriber is the phase_one_decoder (cf. 4.2.2.1.1).

In addition, the node configures and controls acquisition behaviour via ROS2 parameters (for example camera selection, calibration files, trigger mode, frame rate control, and optional disk storage). It does not subscribe to data topics itself but is the source node driving further processing.

If the ohb-ros2-phaseone-acquisition was used to store images on disk, they can be replayed using the ohb-phase-one-ros-iiqplayer. The player publishes the same messages as the ohb-ros2-phaseone-acquisition node. However, it does not receive data from actual camera hardware but from files on disk instead. For the subscribing node, the stream of messages looks the same as if it came from an actual camera. Using the timestamps of the stored images, the ohb-phase-one-ros-iiqplayer also (almost) replicates the image acquisition timing.

4.2.2.2 Geo-Referencing

4.2.2.2.1 keypointer

The keypointer node extracts image features from decoded VIS images. For each image, it computes keypoints and local descriptors that summarize the image content around those points. The results are published as feature messages and used as input to the matcher, allowing feature extraction to be performed once per image.

4.2.2.2.2 matcher

The matcher establishes correspondences between successive images. It receives feature point messages from the keypointer and buffers features from the previous image. For each new image, it matches features between the current and previous feature sets and determines pairs of corresponding image points. These correspondences are published and used for triangulation and ground plane estimation by the triangulator.

4.2.2.2.3 triangulator

The triangulator computes the 3D-coordinates of physical points from the point correspondences established by the matcher, estimates the ground plane from these triangulations and uses them to compute the map between the ground plane and the image plane. It queries the localization_service for platform poses at the relevant image acquisition times to compute the ground plane in geo-coordinates instead of camera-coordinates, essentially resulting in the geo-reference. It also computes the projections of the image vertices onto that plane. The results are published as IIMEO-specific geo-reference messages.

4.2.2.2.4 map_projector

The map projector converts geo-referencing results into coordinate mappings used by other nodes. It receives ground plane estimates and projected image corners and computes transformations between calibrated image coordinates and ground coordinates in the appropriate UTM zone. The resulting map projection message contains homography data that allows conversion between calibrated image coordinates and geographic positions. Together with the camera calibration characterizing the image distortion, the map_projector's result is used to convert between UTM coordinates and image coordinates.

4.2.2.2.5 clipper

The clipper node generates clipping masks used to restrict further image processing to regions near the infrastructure to be monitored, i.e. railway tracks. It receives the map_projector's output information for an image and loads precomputed railway track polygons from configuration data. These polygons are transformed from geographic coordinates into image coordinates and rasterized into a mask image. Pixels not belonging to areas of interest are marked as invalid, and the mask is published with the same image number as the input image.

4.2.2.2.6 clipping_applier

The clipping_applier applies the mask produced by the clipper to a decoded VIS image. It waits until both the decoded image and the corresponding mask for the same image number are available. For each pixel, it copies image values only where the mask indicates valid data. The result is a clipped image that retains the geometry and metadata of the original image and is published either for transmission to ground of on-ground processing or on-board obstacle detection.

4.2.2.3 Obstacle Detection

4.2.2.3.1 subtiler

The subtiler prepares VIS images for neural network inference to find railway tracks in images. It splits a clipped image into overlapping square sub-images of fixed size. For each tile, it records its position in the original image and whether it contains any non-masked pixels, in order to assemble the railway track detection results from each tile into a single result of the same dimensions as the input image. Tiles containing only masked data are flagged so they can be skipped in later processing. The generated subtiles are subscribed to by the

rail track detector. The subtiling step is necessary to cope with the limit neural network inference hardware on the on-board system.

4.2.2.3.2 detect_tracks

The detect_tracks node performs railway track segmentation on VIS images. It subscribes to tiles created from the clipped images and runs a neural network to compute, for each pixel, the probability that it belongs to a railway track. This processing is skipped entirely for tiles containing invalid pixels only. The output is a probability image tile with the same dimensions as the input tile. The node is made to run on the on-board processing unit using OpenVINO. There is a second version of this node which implements essentially the same function but using PyTorch instead of OpenVINO, in order to be able to run the on-board software on other computers during development, which lack the Intel Myriad OpenVINO-compatible accelerator.

The detect_tracks node uses a convolutional neural network (CNN) trained to for image segmentation of railway tracks. We originally planned to also find obstacles by training an auto-encoder which would fail to reconstruct obstacles in input images making reconstruction failure show up as obstacles. As of the end of the project, that approach does not work satisfactorily. We did go (slightly) beyond our initial railway segmentation approach by separately classifying tracks without sleepers, which are common for tramways or level crossings.

4.2.2.3.3 mosaicer

The mosaicer reconstructs full-size railway track probability images from tile-based processing outputs. It subscribes to track probability image tile results generated by detect_tracks and places them back into their original image locations. Where tiles overlap, it keeps the maximum probability value for each pixel. The resulting image is published for use in rail sampling and anomaly extraction. Tiles overlap to avoid having to use segmentation results from tile edges; those are typically worse than results from inside the tile due to the reduced context around pixels near edges.

4.2.2.3.4 rail_sampler

The rail sampler converts pixel-based track probability images into track centreline representations. It analyses the probability image to find local maxima corresponding to likely rail positions. From these points, it follows the dominant direction of the track to form segments and applies probability and length thresholds. The output consists of track segments in image coordinates. They are subscribed for geo-reference refinement by the refiner and for the actual obstacle extraction.

4.2.2.3.5 refiner

The refiner improves the geo-reference of VIS images using detected railway tracks. It compares projected positions of detected track segments with track geometry from map data and estimates a Euclidean transformation in geographic space that minimizes distances between them. This best Euclidean transformation is then concatenated to the original map between geo-coordinates and (calibrated) image coordinates, resulting in an improved geo-reference at least with respect to the railway track features. This refined geo-reference is then subscribed by the extraction node.

4.2.2.3.6 extraction

The extraction node detects obstacles along railway tracks. It receives track-probability images and corresponding refined geo-references, and it loads railway track map data. For each mapped track segment visible in the image, it traces the segment in image coordinates using the geo-reference. It then computes the obstacle probability for each pixel under the railway track segment from the railway track probability mosaic. Using the geo-reference and the railway track map, it computes the location of each obstacle probability information as the distance from the railway track's curve's start along the curve and compresses it by forming intervals of approximately constant obstacle probability. This interval representation is then independent of a particular image and can be transmitted to the ground system for further processing, however, the more practical setup is to let the AnomalyFusion node fuse the intervals from a single image with the intervals of other images.

4.2.3 Fusion

4.2.3.1.1 AnomalyFusion

The AnomalyFusion node combines probability intervals produced at different times and from different sensors. It receives anomaly data referenced to railway track curves and maintains, for each curve, a set of distance intervals with associated probabilities. When new data arrives, the node updates and merges intervals for the corresponding track. The resulting interval representation is published to the payload control software for transfer to ground.

4.2.3.1.2 IntervalsFromSARTile

IntervalsFromSARTile is the node joining the SAR data processing chain into the VIS data processing chain. It converts SAR tile-level obstacle detection results into track-interval-based representations. It interprets obstacle probabilities associated with SAR tiles and assigns them to distance intervals along railway track curves. The resulting interval data is compatible with the fusion node and the payload control software, allowing SAR-derived information to be combined with VIS-derived anomaly data.

5 ON-GROUND SYSTEM

The overall on-ground system has been composed by three functional blocks: Data transmission, On-ground platform and the Services/Web-Frontend (see Figure 5-1), which are described in the following subsections.

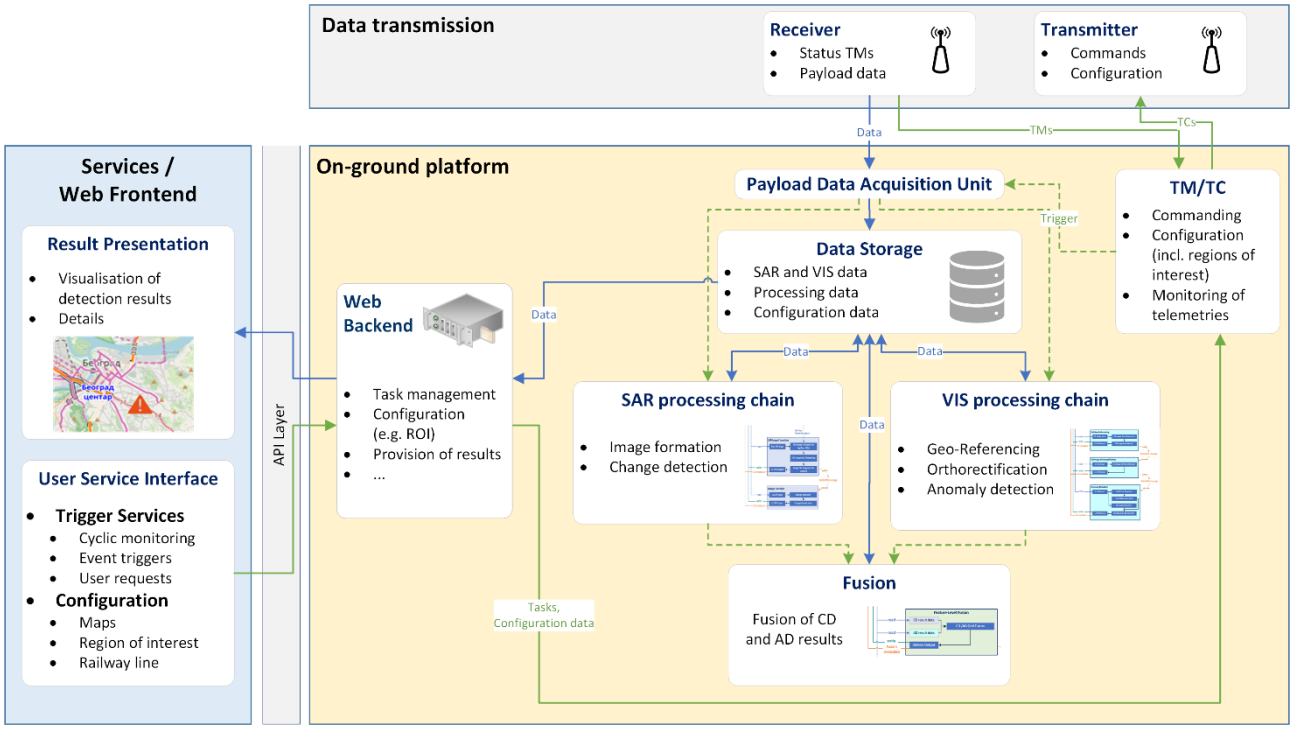

Figure 5-1 High-level architecture of the on-ground system

5.1 On-Ground Platform

The IIMEO platform and the infrastructure services are designed as a cloud-based prototype platform, further on called “on-ground platform”, which provides resources for the supplementary processing and storage of the data received either from the space system or the airborne demonstrator. It also includes components that implement the interfaces to the User Services/Web Frontend on one side and to the on-board platform (via the Data transmission components) on the other side (see yellow box in Figure 5-1).

The on-ground platform is implemented in a modular design and composed by docker containers of the following modules:

- **WebBackend:** This module provides the interface to the user services and the corresponding frontends (*WebUI* and *Mobile App*). It also includes the functionality for configuration and management of tasks and provision of the processing results.
- **TMTC**: This module translates configuration data and tasks into telecommands (TC) for the onboard system (OBS) and processes returning telemetry (TM). It also triggers the *PayloadDataAcquisitionUnit* module to download available data from the OBS.
- **PayloadDataAcquisitionUnit**: This module downloads the data from the OBS and writes it to the *data storage* module. It then triggers the first modules of the SAR and VIS processing chains to start processing.
- **SARImageFormation**: This module is the first part of the SAR processing chain and is intended as a backup solution for the respective on-board image formation processing. It shall process geo-referenced SAR tiles along railway tracks from the acquired SAR raw data, on which the SAR change detection will be processed afterwards.
- **SARChangeDetection:** This module is the second and main part of the SAR processing chain. It analyses the previously processed SAR tiles and returns detected changes on the railway tracks. The detection results are forwarded to the *Fusion* module.

- **VisGeoReferencing**: This module is the first element of the VIS processing chain. It geo-references the acquired VIS raw data with respect to the reference coordinate system. The resulting VIS images are forwarded afterwards to the *VISOrthorectification* module.
- **VISOrthorectification**: This module is the second element of the VIS processing chain. It performs a geometrical correction of the geo-referenced VIS input images for a true-scale and accurate mapping onto the reference map. The resulting VIS images are then forwarded to the final *VISAnomalyDetection* step
- **VISAnomalyDetection:** This module implements the final and main steps of the VIS processing chain. It performs a detection of the railway tracks within the input image data and then analyses these tracks for possible anomalies. The detection results are forwarded to the *Fusion* module.
- **Fusion**: This module loads the results from the SAR and VIS processing and registers them to enable a combined presentation of the detected changes and anomalies. The outcome is provided to the *WebBackend* to finally forward it to the Web Frontend for presentation to the user.
- **Data Storage**: This module provides the required S3 storage functionality for initial data as well as for intermediate and final processing results.

5.2 Communication with Onboard System and Data Transmission

5.2.1 TM/TC

The communication between the on-ground platform and the on-board system is sketched in Figure 5-2.

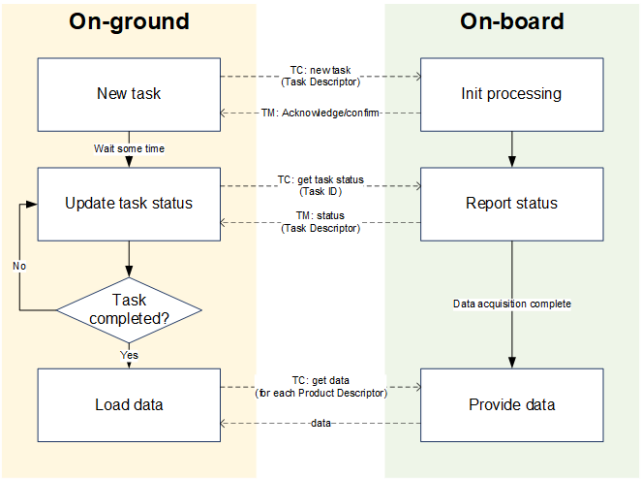

Figure 5-2 TM/TC sequence between on-ground platform and on-board system

The communication is based on Standard telemetries TM(1,1), TM(1,2), TM(1,5), TM(1,6), TM(1,8) for status reporting and mission specific telecommands and telemetries using service type 200, as described in Table 5-1:

Table 5-1 Utilized TMs and TCs

ID	Title	Remark
TM(1,1)	Successful acceptance verification report	-
TM(1,2)	Failed acceptance verification report	-
TM(1,5)	Successful progress of execution verification report	-
TM(1,6)	Failed progress of execution verification report	-
TM(1,8)	Failed completion of execution verification report	-
TC(200,1)	Start monitoring task	• Mission specific service type 200 • Response: TM(1,1) or TM(1,2)

ID	Title	Remark
TC(200,2)	Update task status	• Mission specific service type 200 • Response during running task: TM(1,5) or TM(1,6) • Response after completion: TM(200,7) or TM(1,8)
TM(200,7)	Successful completion of task	• Mission specific service type 200 • Used instead of TM(1,7)
TC(200,10)	Get data	• Mission specific service type 200 Return: data stream (not necessarily a PUS service)

5.2.2 Data Transmission

In the scope of the project's airborne flight campaign, the data transmission component of the ground platform (see right side in Figure 5-3) was set up as an LTE network for communication with the flight demonstrator, which consists of:

- An LTE-Router (LTE-Box, Netgear Nighthawk M1), which is used for both sending TCs to and receiving TMs and data from the on-board system. The data rates for this component are specified with max. 1Gbps for downlink and max. 150 Mbps for uplink.
- A UTM Box (Sophos RED) for providing a secured and encrypted data exchange between the on-board and on-ground networks. The data rate for this unit is specified with max. 1Gbps.

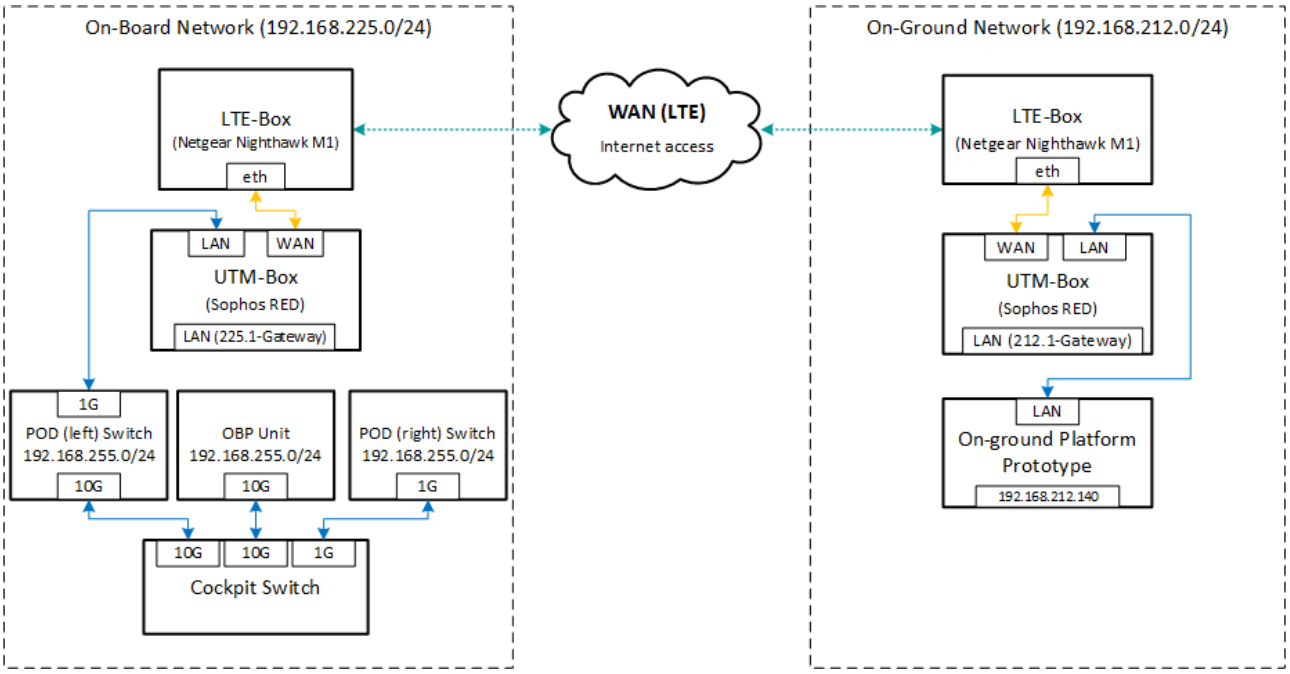

Figure 5-3 Network configuration for airborne system (left side) and ground station (right side)

5.3 Infrastructure Services and User Interfaces

The **Services** module with its **Web-Frontend** is the main interface between the end users and the IIMEO on-ground platform (described in section 5.1). As shown in Figure 5-4, it can be divided into the application (the Service UI), which handles the user interaction, the management of tasks and the presentation resp. visualisation of results, and into the data fetcher, which is responsible for the data exchange with the Web Backend on one side (using the RESTful API and data in JSON format) and with the platform's data storage on the other side (using GeoJSON and GeoTIFF formats).

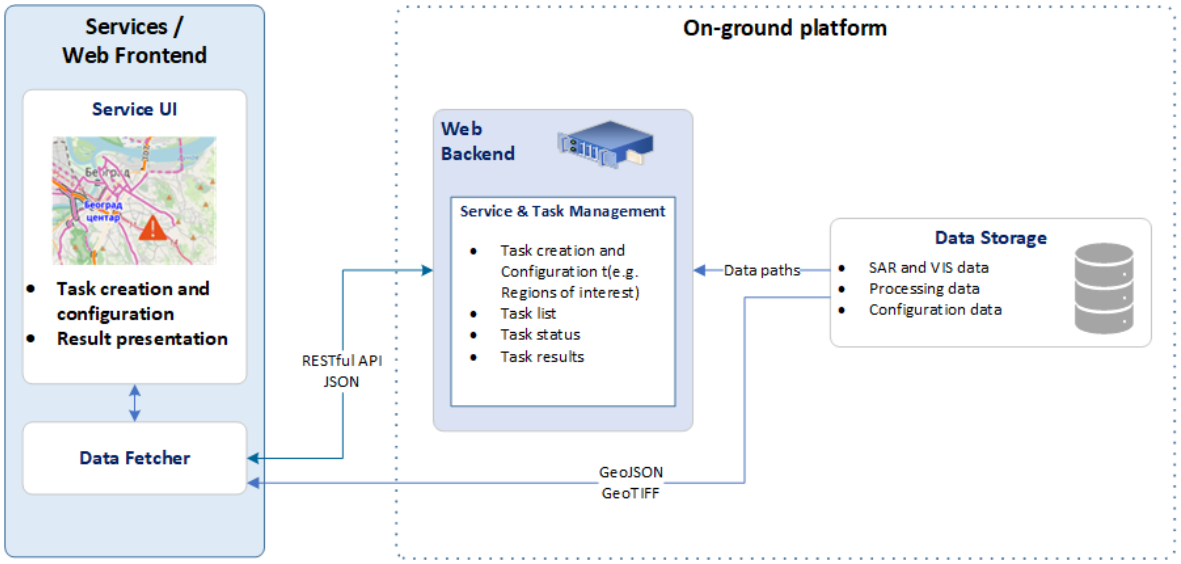

Figure 5-4 Web Frontend Interfaces

The connection between the **Web-Frontend** and the **Web-Backend** (which is part of the on-ground platform) is established via a REST API. The API provides multiple endpoints related to the monitoring tasks, as shown in Figure 5-5:

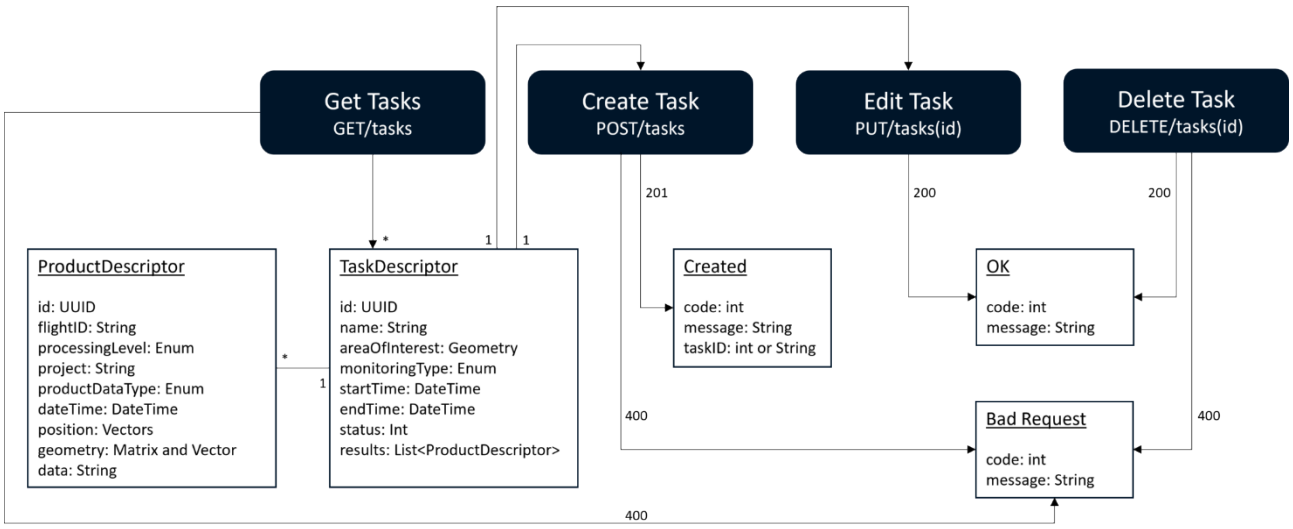

Figure 5-5 REST API interface definition

5.3.1 Web UI

The IIMEO Web UI is the initial user interface for managing monitoring tasks. Its purpose is

- to define the region of interest for the monitoring, the type of monitoring (such as single task, regular/continuous task or event-based task), the monitoring accuracy and the monitoring period,
- to present the monitoring results in an appropriate way to the user.

The Web UI opens with a predefined map section showing all the railway tracks relevant to the user (see Figure 5-6), also showing a control panel with the (initially empty) task list.

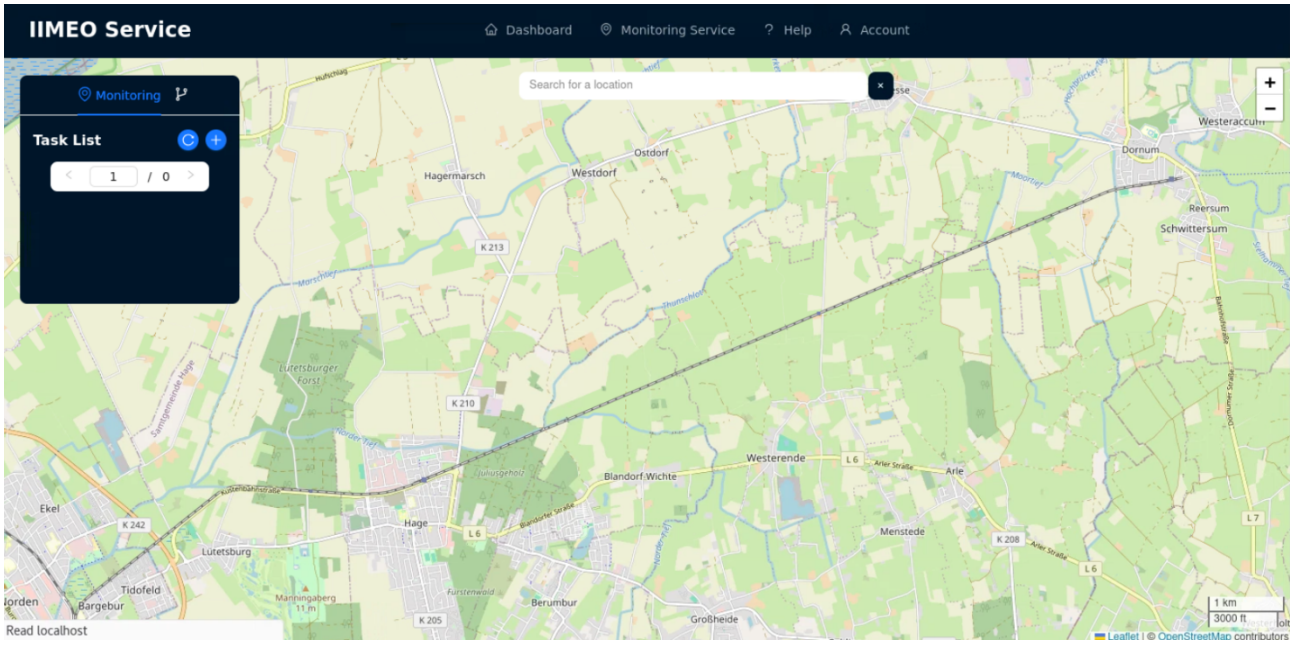

Figure 5-6 Start Screen of the IIMEO Web Service

By pressing the (+) button, the user can define the specific region of interest within the map by setting the positions of a polygon via mouse click (Figure 5-7).

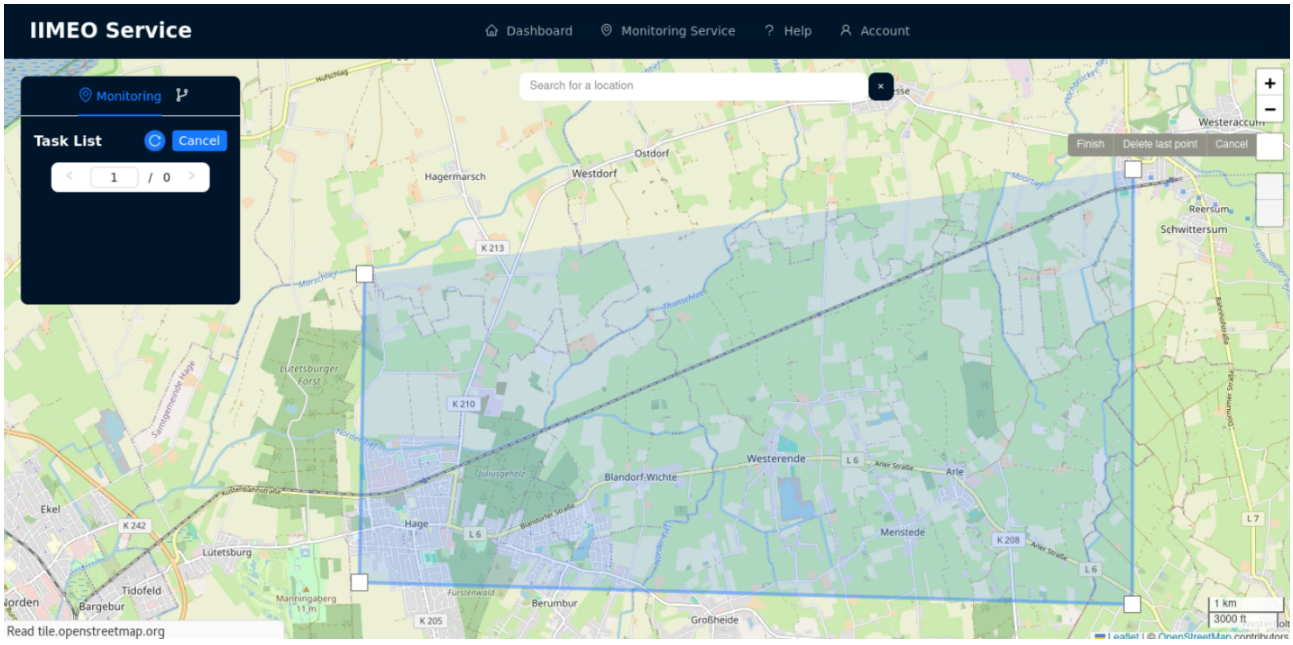

Figure 5-7 Selection of the region of interest

After closing of the polygon, the task configuration dialog opens, providing options to assign a name to the monitoring task and to set the type of the monitoring task, its accuracy as well as the monitoring period and frequency (see Figure 5-8).

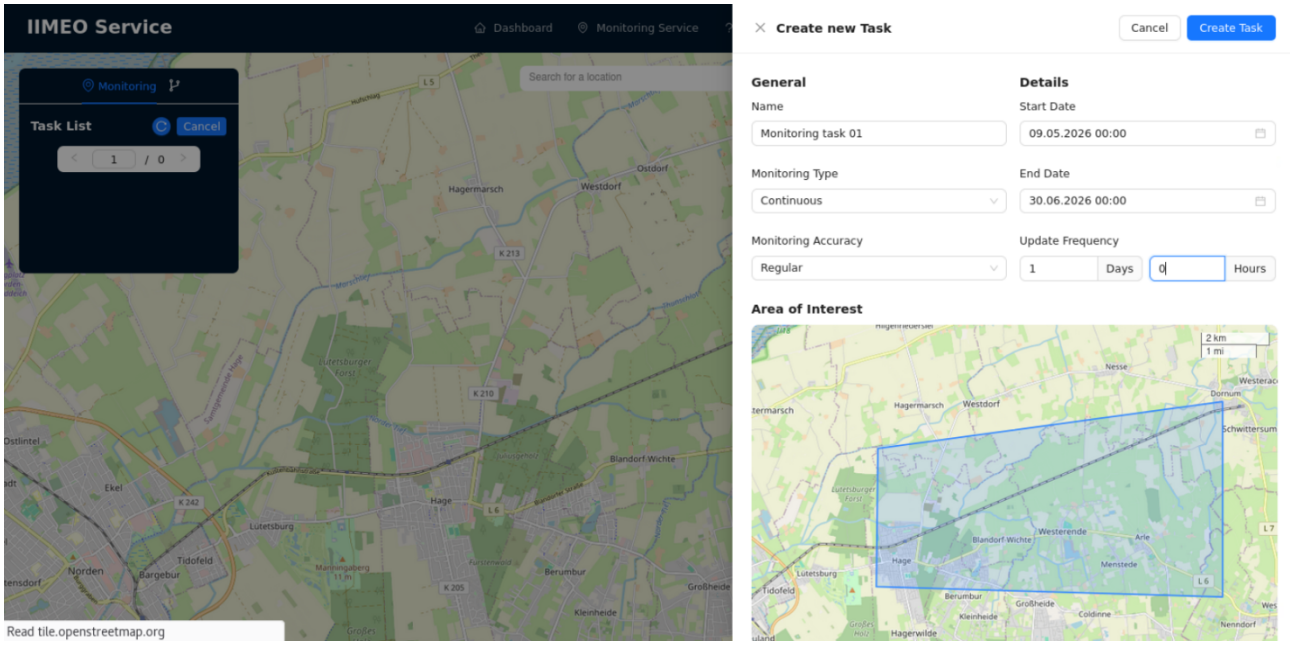

Figure 5-8 Task configuration

With pressing “Create Task”, the commanding of the monitoring task is completed, and the corresponding request is sent to the *WebBackend* in the on-ground platform. Afterwards, the task is shown in the task list in the configuration panel, and the intermediate or final monitoring results are displayed on the map (see Figure 5-9).

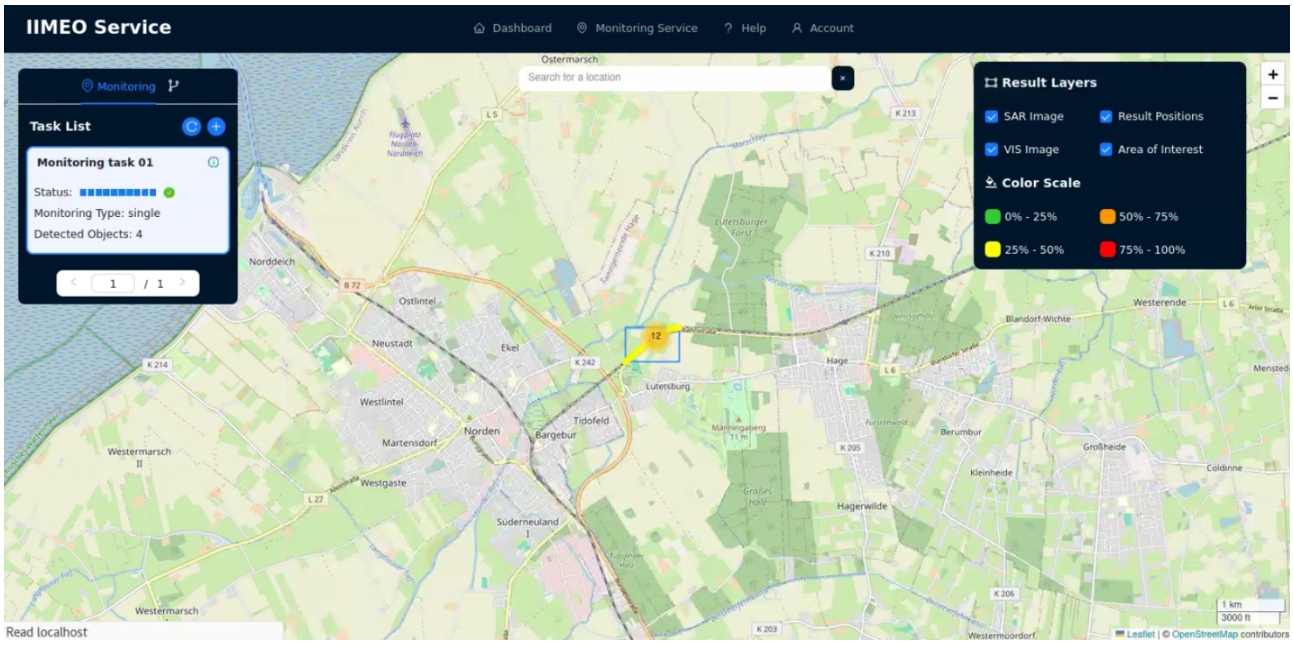

Figure 5-9 Displaying of monitoring results

The probability of identified obstacles on the railway track is visualized by a color-coded overlay of the track, with different colours according to the accuracy of the results, see legend in Figure 5-9 and Figure 5-10. The presentation of the results is getting more detailed when zooming into the scene (see Figure 5-10).

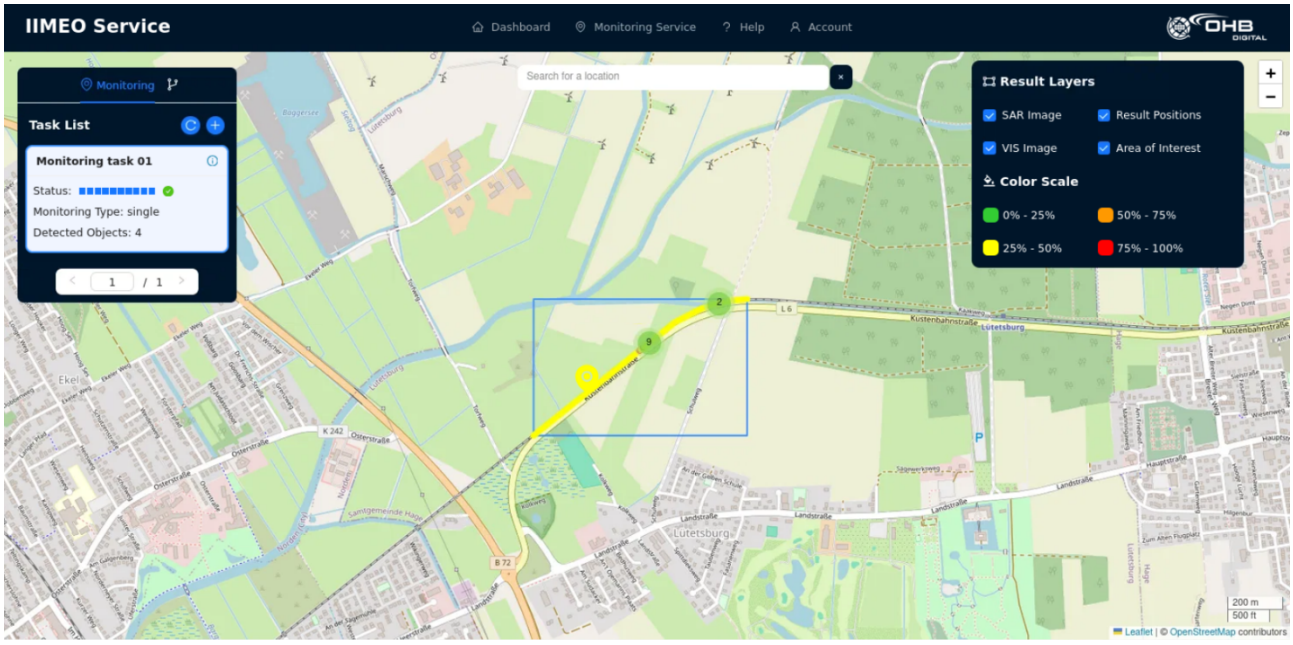

Figure 5-10 Details are shown according to the zoom level

For further inspection of the obstacle data, obstacle indicators are presented in the middle of each interval, showing either the number of identified obstacles within one marker or a detailed view on the detection probabilities within a multi-marker for the interval (see Figure 5-11). Clicking on one of these markers shows the detailed probability for this obstacle.

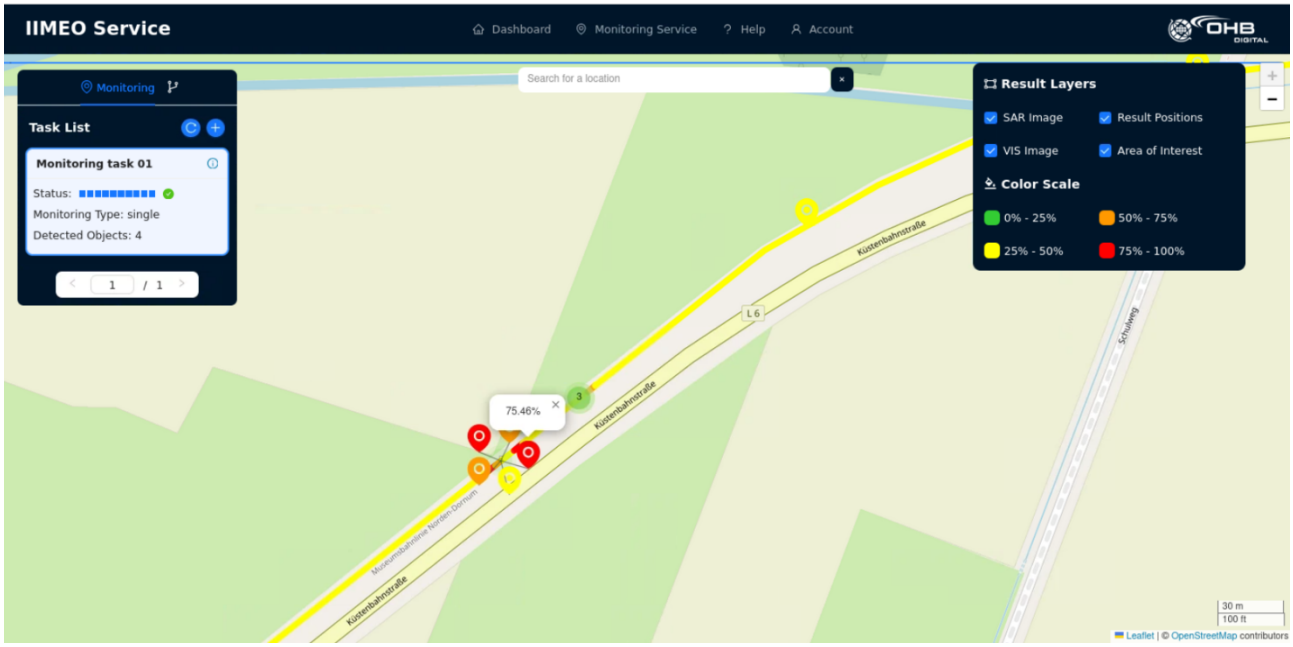

Figure 5-11 Display of obstacle indicators

5.3.2 Mobile App

The IIMEO mobile application serves as a secondary interface for end users to interact with the on-ground platform, providing a convenient and mobile-friendly way to access system updates. Its primary purpose is to deliver real-time information on ongoing tasks and notify users of events requiring their attention. Compared

to the web interface, the mobile application focuses on mobility and timely notifications, offering a streamlined set of interaction features.

The current version of the mobile application addresses the functionalities and user interface requirements, including better data visualization, clearer status indication, and intuitive interaction with map elements. Users can easily explore detected items, with visual cues (e.g., marker color changes) helping differentiate between new and previously viewed information.

Figure 5-2: Map Overview with Detection Markers

The application also provides a detailed event overview once a specific marker is selected on the map. This view presents key information such as detection time, exact location, and processing status. If available, corresponding SAR/VIS images can also be accessed from this interface.

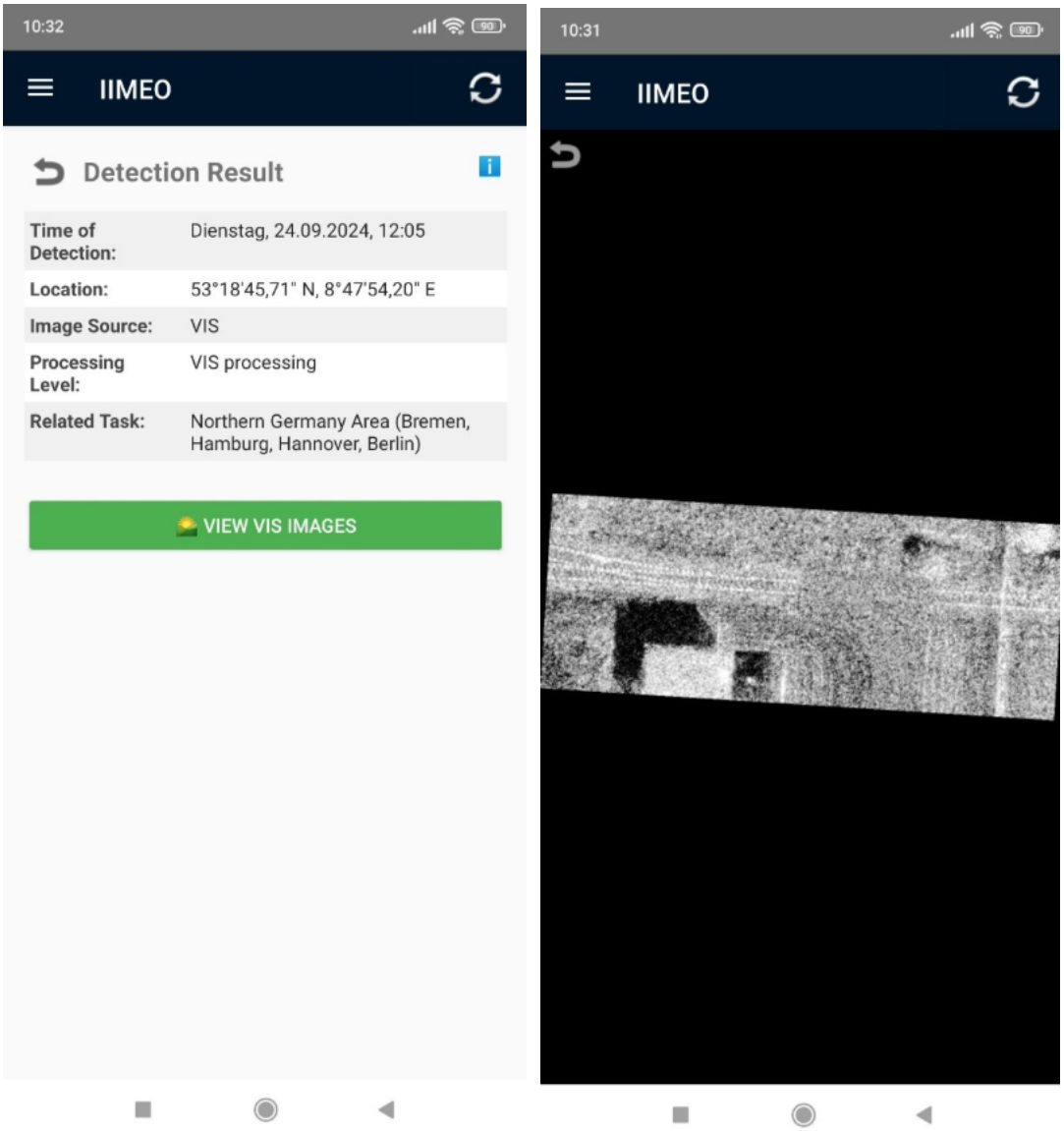

Figure 5-3: Event Details and Detection Image Display

In addition, the mobile app includes a task overview page, where users can browse available tasks and access associated detection results. Selecting a task displays a list of computed detection locations, allowing users to directly navigate to specific points on the map.

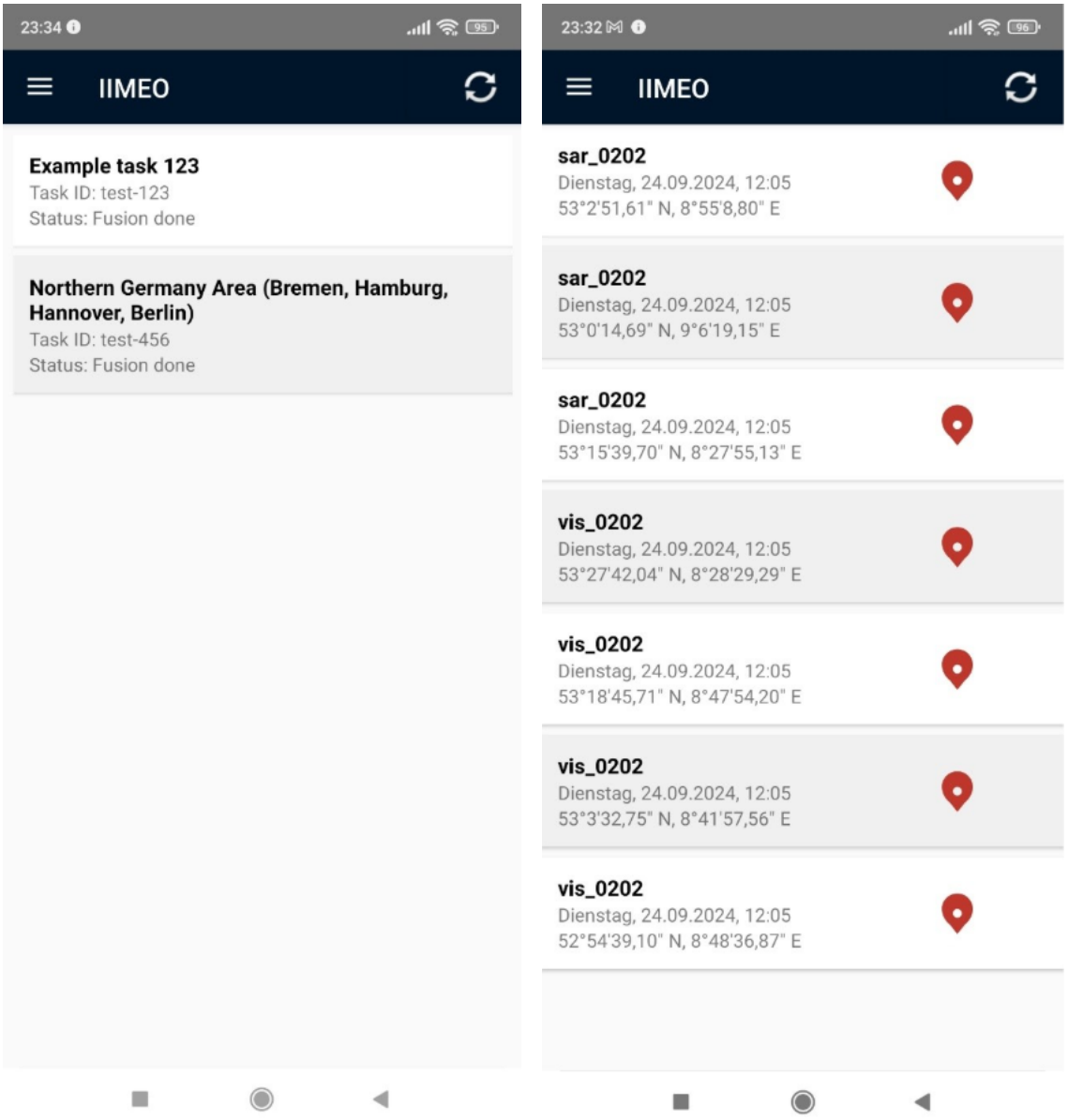

Figure 5-4: Task List and Event Location Overview

To improve usability, the application provides filtering capabilities within the map view. Users can apply task-based filters to display only relevant tasks and their corresponding areas of interest. The map automatically adjusts to focus on selected tasks, and additional task details can be accessed through interactive elements.

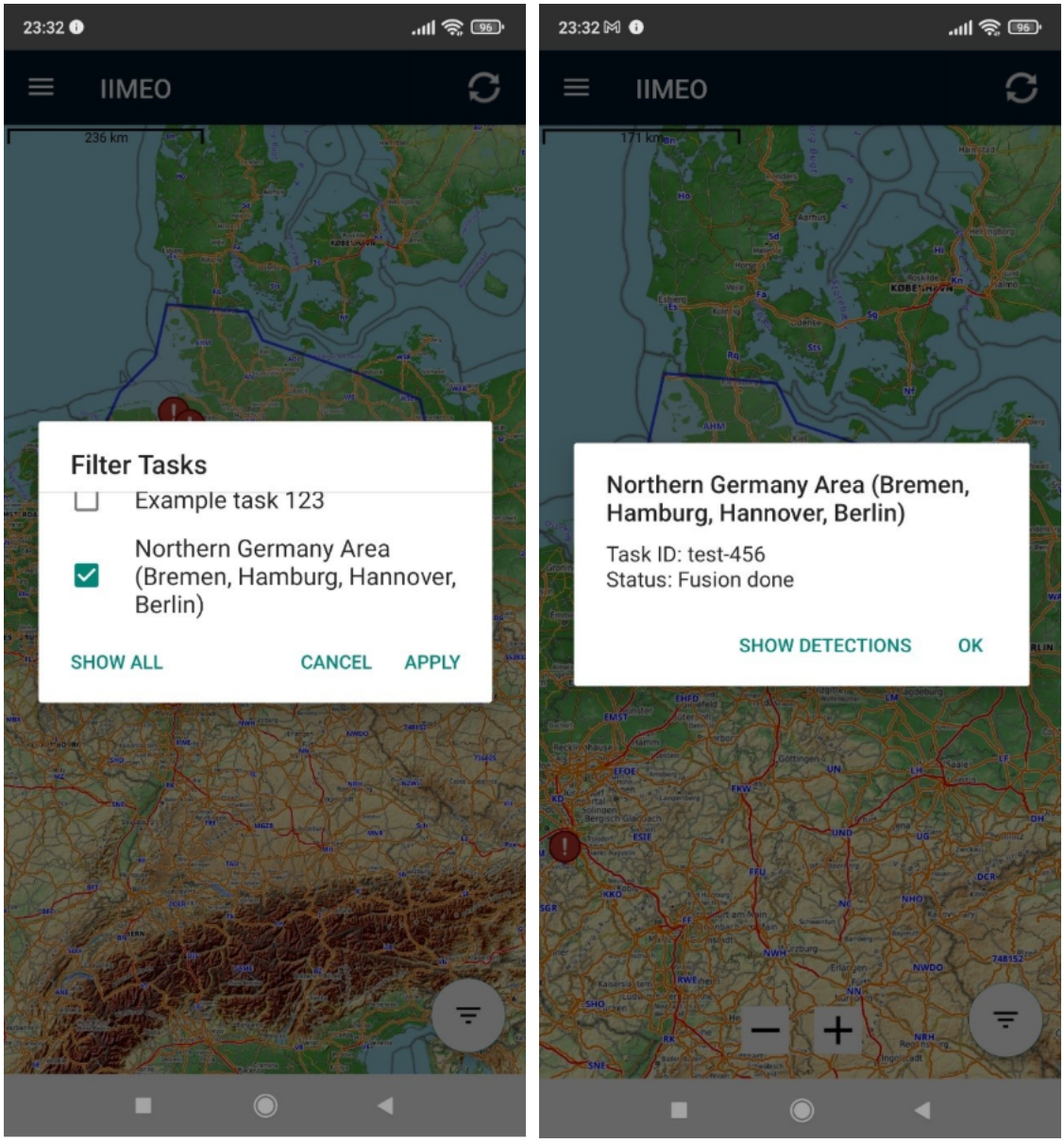

Figure 5-5: Task Filter and Task Details Popup

Furthermore, the application allows users to configure personalized settings. These include selecting different map overlays (e.g., OpenStreetMap or satellite imagery), enabling or disabling additional layers such as railway lines, customizing marker types, and managing notification preferences.

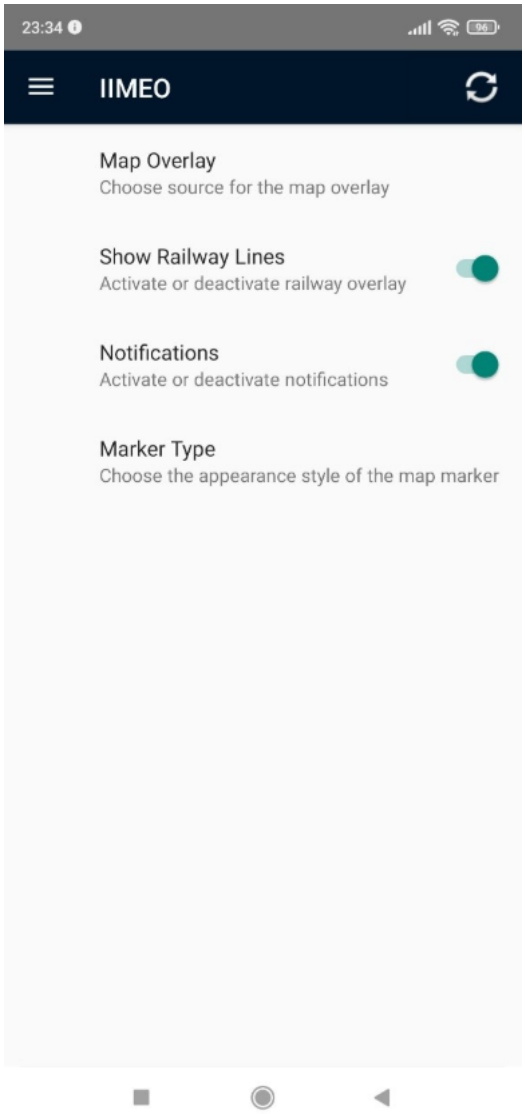

Figure 5-6: App Settings Overview

6 REFERENCES

[1] IIMEO Consortium, „D2.5: Full Prototype and Verification of On- and Off-board algorithm,“ 2025.

[2] IIMEO Consortium, „D3.5: Full Prototype and Verification of integrated on-board algorithms,“ 2025.

[3] IIMEO Consortium, “D4.5: Full Prototype and Verification of integrated Off-Board Algorithms,” 2026.

[4] IIMEO Consortium, “D4.7: System Integration and Experimental Evaluation & Refinement,” 2026.

[5] IIMEO Consortium, „D1.3: IIMEO System Concept,“ 2023.

[6] IIMEO Consortium, „D3.1: Design of On-board hardware,“ 2024.

[7] IIMEO Consortium, „D3.2 Early Prototype of On-board data processor HW,“ 2024.

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

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

[10] IIMEO Consortium, *IIMEO-OHBDC-D-0019, Early Prototype of IIMEO platform, infrastructure services,* 2024.

[11] IIMEO Consortium, *IIMEO Algorithms, Sensor and System Concept,* 2023.

[12] ECSS, *Space engineering - Telemetry and telecommand packet utilization. ECSS-E-ST-70-41C,* 2016.

[13] IIMEO Consortium, *IIMEO-AWS-D-0018, Design of On-board hardware,* 2024.

Appendix A Abbreviations & Nomenclature

Abbreviation	Meaning
ATB	Institut Für Angewandte Systemtechnik Bremen GmbH
AWS	Antwerp Space N.V.
DDS	Direct Digital Synthesizer
EC	European Commission
ECSS	European Cooperation for Space Standardization
EO	Electro-Optical
FBK	Fondazione Bruno Kessler
FHR	Fraunhofer-Institut für Hochfrequenzphysik und Radartechnik FHR
FMCW	Frequency Modulated Continuous Wave
IIMEO	Instantaneous Infrastructure Monitoring by Earth Observation
IMU	Inertial Measurement Unit
LEO	Low Earth Orbit
NIS	University of Nis
OHB DC	OHB Digital Connect GmbH
PU	Public
RES	Classified R-UE/EU-R – EU RESTRICTED under the Commission Decision No2015/444
RTD	Research and Technology Development
S&T	Science and Technology
SAR	Synthetic Aperture Radar
SEN	Sensitive
WP	Work Package