Ref. Ares(2026)4107783 - 21/04/2026

SESAR JOINT UNDERTAKING

AWARE Report on Integration of the Improved ASA System with ATC Simulator

Deliverable ID:	**D2.22**
Project acronym:	**AWARE**
Grant:	**101167442**
Call:	**HORIZON-SESAR-2023-DES-ER-02**
Topic:	**HORIZON-SESAR-2023-DES-ER2-WA2-4**
Consortium coordinator:	**FTTS**
Edition date:	**13.03.2026**
Edition:	**00.03**
Status:	**Official**
Classification:	**PU**

Abstract

This document describes the integration of the improved Artificial Situational Awareness (ASA) system with the Air Traffic Control (ATC) System within the AWARE project. Communication between the ASA system and the ATC Simulator is established through a near-real-time, publish-subscribe message-streaming workflow utilizing the NATS project and Google's Protocol Buffers. The document also

addresses the updates that need to be made to the individual systems before the project final validation.

Authoring & approval

Author(s) of the document

Organisation name	Date
TERN	19.12.2025
FTTS	12.03.2026

Reviewed by

Organisation name	Date
Tomislav Radišić / FTTS	03.04.2026.

Approved for submission to the SESAR 3 JU by

Organisation name	Date
Tomislav Radišić / FTTS	03.04.2026.

Rejected by[1]

Organisation name	Date

Document history

Edition	Date	Status	Company Author	Justification
00.01	19.12.2025	Draft	Tern Systems	Creation of document. Draft for sections 1,2,3&4
00.02	16.02.2026	Draft	Tern Systems	Section 4, Section 5
00.03	09.03.2026	Draft	Tern Systems	Section 6

[1] Representatives of the beneficiaries involved in the project.

00.03	12.03.2026	Draft	FTTS	Sections 2, 3 and 5
00.03	13.03.2026	Draft	Tern Systems	Final review and formatting

AWARE

Achieving Human-Machine Collaboration with Artificial Situational Awareness

This document is part of a project that has received funding from the SESAR 3 Joint Undertaking under grant agreement No 101167442 under European Union's Horizon Europe research and innovation programme.

Table of Contents

List of figures

List of tables

1 Executive summary

The AWARE project aims to advance human-machine collaboration in air traffic management through the development of an Artificial Situational Awareness (ASA) system. This system enhances the decision-making capabilities of Air Traffic Control Officers (ATCOs) by providing timely, relevant support using artificial intelligence, enabling them to manage complex air traffic scenarios more effectively.

At the core of the concept is an AI assistant application designed to deliver adaptable and human centric support. The ASA system continuously assesses controller intent, monitors visual attention, and evaluates mental workload in real time. Based on these inputs, it can adjust its support actions to either assist with the task the controller is focusing on or to manage other relevant tasks autonomously. Within this framework, a set of ATCO support tools is being developed and tested. These tools translate the capabilities of the ASA system into practical functions that can assist controllers in their daily operations.

In the simulator setup, the ATCO interacts with the support tools through the Air Situation Display (ASD), which serves as the primary human–machine interface. All operational inputs and visual outputs are handled via the ASD, ensuring that ASA functionality is embedded seamlessly within the controller's familiar working environment. The ASD also acts as the communication gateway between the controller and the ASA system, transmitting ATCO actions and traffic state information to ASA while receiving context-aware recommendations, alerts, and automation outputs in return.

EUROPEAN PARTNERSHIP
Co-funded by the European Union

2 Introduction

2.1 Purpose of the document

The purpose of this document is to describe the work undertaken to integrate the improved ASA system with the ATC Simulator.

2.2 Background

The improved ASA System is the AWARE's solution originally developed in the SESAR project AISA. In AISA, the ASA system was built using the aeronautical data exchange standards AIXM (Aeronautical Information Exchange Model) and FIXM (Flight Information Exchange Model). The previous ASA model was able to create a knowledge graph of the traffic situation. On top of this knowledge graph, a reasoning engine generated situational awareness and proposed monitoring actions, while machine learning modules predicted future traffic states and identified potential conflicts. The proof of concept was validated through human-in-the-loop simulations comparing ATCO situational awareness with and without ASA support. Results showed that ASA achieved situational awareness levels comparable to ATCOs and effectively complemented them by automating certain tasks and proposing actions.

Despite these promising results, several limitations were identified. While the system developed a comprehensive understanding of the traffic situation, it lacked awareness of the ATCO's intent, attention and workload. As a result, its support was sometimes delivered at inappropriate moments, disrupting workflow rather than enhancing it. In addition, the human–machine interface (HMI) did not fully align with operational requirements. ATCOs often lacked information to understand the system's reasoning, which reduced trust and explainability. As ASA was not aware of ATCOs' intent, it could not fully align its support with ATCOs' situational awareness. ATCOs often wanted to choose the level of support they would expect from such a system which was then a limitation of the system. The communication between the Machine and human was in this case one-way: ATCOs could interpret the system's outputs, but the system could not interpret human intent which is an essential requirement for effective human–machine teaming. ATCOs also expressed the need to adjust the level of support provided.

These limitations create the opportunity addressed by AWARE: to improve ASA with awareness of ATCO intent and to deliver support that is explainable and aligned with operational HMI requirements. Also, the ambition is to develop a system capable of assisting ATCOs with intended tasks while autonomously managing high-priority tasks outside their immediate attention. In such a system, ATCO can select their preferred level of support, maintaining situational awareness while mitigating risks such as complacency or the out-of-the-loop effect.

AWARE's approach focuses on developing a proof-of-concept AI Assistant Application (AAA) to demonstrate the ASA solution, which supports ATC operators by identifying and supporting them in their goals. To enable effective human-machine collaboration, the system will track ATCOs' attention and HMI interactions. By contextualizing these inputs within the traffic situation, the system will infer ATCOs' intent and determine supportive actions.

The improved ASA will be evaluated under two levels of support:

- Level 2A - Advisory mode: the system provides suggestions only.
- Level 2B - Execution mode: the system both suggests and executes actions on behalf of the ATCO.

The following sections detail the integration of the improved ASA system into the ATC simulator. This process will ensure that ATCOs remain fully in the loop at all times. They will clearly understand the level of support provided, the actions suggested or executed by the system, and retain the ability to override automation when necessary. This design is essential for achieving safe, transparent, and effective human-machine collaboration.

2.3 Structure of the document

The document is structured as follows:

- Section 1 provides the Executive Summary
- Section 2 provides the Introduction
- Section 3 describes the System Overview
- Section 4 describes the Communication between ASA System and ATC Simulator
- Section 5 describes the Integration of ASA in the ATC Simulator Environment
- Section 6 provides the Conclusion and Next Steps

EUROPEAN PARTNERSHIP
Co-funded by the European Union

3 System Overview

3.1 High-Level Concept of Operations

The ATCO support tools represent the primary means through which the ASA system interacts with the Air Situation Display (ASD). While the ASA system functions as the "brain" of the architecture, responsible for analyzing the airspace situation and assessing the controller's intent, the support tools serve as the interface layer that makes this intelligence available to the user.

The ASA system makes decisions based on the controller's intent and the current airspace situation. These decisions are transmitted via the interface to the ATC simulator, where they are either displayed to the user on the ASD as decision-support suggestions or executed directly by the system. This setup ensures that controllers are continuously supported with context-aware, timely information and suggestions/actions relevant to the task that the controller is currently performing.

3.2 Improved ASA System

AWARE's approach focuses on the development of a AAA to demonstrate the improved ASA system solution (Figure 1). The ASA system (represented by green block) uses aeronautical information exchange models to create a knowledge graph representing the traffic situation, converted to the Resource Description Framework. This graph, enhanced with rule-based knowledge and a reasoning engine, builds a semantic understanding of the traffic situation. The blue block represents improved ASA architecture with ability to monitor ATCO gaze and HMI interactions. By contextualizing these inputs within the traffic knowledge graph, the system can extract the ATCO's intent and select appropriate supportive actions. This integration allows the system to align its behavior with the ATCO's current intent, enabling more adaptive, transparent, and cooperative support.

Essential elements of the improved ASA system are:

- Gaze and HMI tracking - monitoring of gaze and HMI interactions to enable inference of ATCO intent.
- ATCO task priority list - a representation of current ATCO tasks, prioritised by ML algorithms.
- Action selection module - a decision-making that determines appropriate system responses according to the predefined level of support.
- ATCO support tools - functionalities that provide advisory or automated assistance to the ATCO.

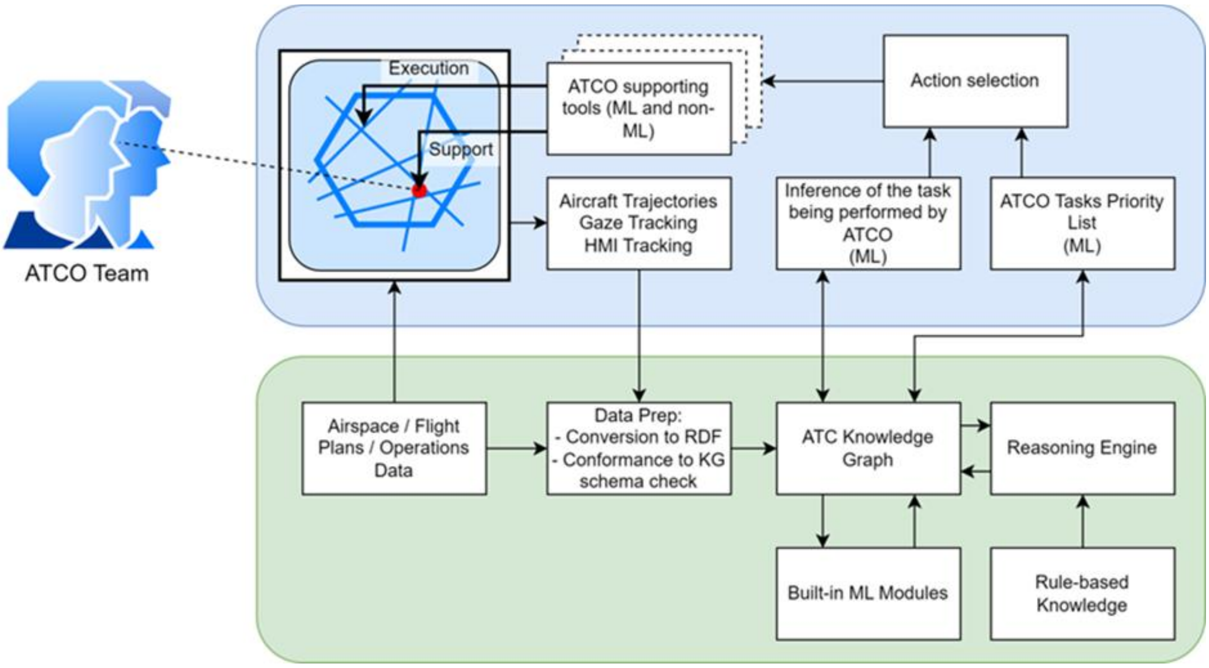

Figure 1.Improved ASA system architecture

3.2.1 Gaze and HMI tracking

The development of a data-driven ATCO intent assessment builds on the synchronized collection of gaze tracking data, HMI interaction logs, and simulator data to infer the ATCO's focus, activity, and underlying intent with respect to the traffic situation. The integration of these data streams enables advanced Attention Tracking and Intent Analysis.

The intent assessment methodology is specifically designed to link gaze tracking and HMI data to the cognitive and operational intent of ATCOs. It combines expert knowledge with data-driven techniques to ensure both interpretability and predictive performance. During data collection, ATCOs annotated recorded data by labeling their intent and strategies. Gaze and HMI data were used to identify behavioural patterns and to apply both rule-based and machine learning methods to predict ATCO intent. This approach ensures that intent assessment remains robust, explainable, and adaptable to dynamic air traffic conditions. By contextualizing gaze and HMI data within the semantic representation of the traffic situation provided by the ASA knowledge graph, the improved system achieves a deeper machine-level understanding of human actions.

The extraction of ATCO intent follows a structured process. First, relevant behavioural features such as gaze allocation, interaction frequency, and spatial attention distribution, are defined and interpreted with support from domain experts to improve operational relevance and explainability. Subsequently, ML models are trained to infer intent from these features and are evaluated against expert-labeled datasets to assess accuracy and operational feasibility. Finally, the inferred ATCO intent is mapped to specific ATCO tasks, enabling the ASA system to recognize which task the ATCO is currently managing and to align its support accordingly. A detailed description of the mapping between gaze and HMI tracking data and ATCO intent is provided in D2.16 [1], and ATCO intent mapping to ATCO tasks is described in D2.17 [2].

3.2.2 ATCO task priority list

Prerequisite for the creation of an ATCO task priority list is an active ATCO task list. The first step in creating it is using current traffic data, generated by the ATC simulator and stored in the knowledge graph, and applying rules written in the Datalog language. Those rules represent how to form new facts from the ones already present in the knowledge graph, with “new facts” representing if a task is active for a single aircraft in the traffic situation. Collecting those new facts produces a list of task instances which are expected to be completed by an ATCO in the specific traffic situation - the active task list.

The active task list changes as traffic evolves and with each ATCO action. Whenever any of the rules are no longer fulfilled, that task is no longer active for a certain aircraft and the instance is removed from the list. Those instances which are present in the list have their features - information which more closely describes each instance - updated for each timestamp. This means that, while the active task list may contain the same task instances for several timestamps in a row, those lists are not identical.

Even though the active task list by itself provides important information (which tasks can/must be accomplished according to traffic situation parameters), it does not communicate the importance of each task instance in the overall situation. For example, while the ATCOs are aware that it is necessary to accomplish standard “procedural” tasks such as assuming and transferring aircraft, those task instances will be completed at different times depending on the existence of an urgent, safety critical task instance. Conversely, while the ATCOs may be aware of the existence of a conflict, depending on the distance between the aircraft they may perform other, simpler tasks before solving it. The prioritization - process of sorting a list according to the importance of each list element - must therefore take into account the values of task features mentioned above to properly organize the active task list.

During prioritization module training, active task lists were compared to tasks performed by ATCOs during data collection experiments. Since it was recognized early on that it would not be enough to identify which instance from the list was performed, each instance was expanded by calculating values of task features. This was important in two ways - it made the learning process much closer to how ATCOs themselves set priorities (i.e. by taking into account other traffic and goals), while also providing the machine learning model with richer contextual information to support more robust generalization.

The prioritization module follows the same approach during operations (inference), taking the active task list as input and using learned “weights” to organize it according to priority. As with similar machine learning approaches, it does not provide a binary decision but instead estimates the probability that an ATCO would choose a specific task instance to perform next. Sorting the instances according to this probability produces a prioritized active task list.

3.2.3 Action selection module

By linking ATCO intent analysis to specific operational tasks and integrating it with the ATCO task priority list, the ASA system provides the necessary inputs for the action selection module. This module determines which actions to suggest or execute based on the selected level of support. The task priority list defines which tasks require attention according to operational urgency and safety relevance, while the inferred ATCO intent reflects what the ATCO is currently focusing on. In many situations, these two perspectives will not coincide, ATCOs may intentionally defer certain tasks despite their priority, making action selection a complex decision-making process that must balance operational demands with human workflow.

In Level 2A (Advisory Mode), the system proposes actions in a way that aligns with the ATCO's current workflow and minimises disruption. In Level 2B (Execution Mode), the system not only suggests actions consistent with the ATCO's ongoing activities but may also autonomously execute tasks identified as time-critical or safety-relevant based on the task priority list. Overall, the action selection logic prioritises alignment with ATCO intent while retaining the capability to intervene when urgent or safety-critical tasks require immediate attention. In both modes, the ATCO remains fully in the loop and can override any system suggestion or automated action. The action selection module is configurable to match the ATCO's preferred level of support, including the number or frequency of suggested actions. ATCOs can adjust the system's sensitivity, for example, choosing to be notified about all tasks from the list or to receive only one recommendation at a time, ensuring adaptive, user-centered support.

3.2.4 ATCO support tools

Support tools incorporate Datalog rules that operate in conjunction with knowledge graphs to generate decisions supporting the selected operational action. Each tool processes traffic data through a defined set of Datalog-based rules to produce a decision output for the ATCO. Depending on the operational mode, the output may either be presented as an advisory recommendation or executed autonomously. All actions are executed through an independent communication link between the ASA system and the pseudopilot. Within the enhanced ASA system, four support tools have been developed.

(i) Conflict Detection and Resolution (CD&R) tool - identifies conflicts based on trajectory predictions. Once conflict pairs are detected, the tool evaluates a set of potential resolution manoeuvres, including: adjustment from current FL Requested FL, block of climb or descent, direct route to the second or third waypoint, climb or descent by 10 or 20 FL. A prioritisation algorithm is applied to determine the most appropriate resolution. The tool implements a rule-based decision framework that selects the optimal solution according to predefined safety, operational efficiency, and ATCO workload criteria. For example: climb or descent resolutions are considered when the time to conflict is less than 5 minutes, blocked climb or descent measures are considered when the time to Closest Point of Approach (CPA) is less than 3 minutes. Additional evaluation criteria ensure that the selected resolution maximises safety, maintains operational efficiency, and minimises controller workload. Each proposed resolution remains valid for 30 seconds, after which a new set of potential resolutions is recalculated to reflect updated traffic conditions. In advisory mode, the CD&R tool functions primarily as a detection and recommendation system. In execution mode, it acts as a safety net, automatically implementing resolutions when situations become safety-critical.

(ii) Quality of Service (QoS) tool - identifies opportunities to improve flight efficiency by enabling more direct routings and climbs or descents to the requested FL. When a QoS action is triggered, the tool evaluates whether the proposed trajectory is: conflict-free (through integration with the CD&R tool), clear of restricted airspace, operationally compliant with the rules, procedures, practices and LoAs (Letters of Agreement). The QoS tool considers options such as: direct routing to the exit point, and climb or descent to requested FL. It supports ATCOs by identifying suitable opportunities to return aircraft to their planned route following vectoring. This tool helps ATCO identify when each flight can be returned to the route after vectoring. In addition, the tool assesses whether the proposed route modification requires coordination with adjacent units and ensures that such instances are identified before the action is suggested or executed.

(iii) Transfer tool - identifies flights that are eligible for frequency transfer to the next unit. It is based on a set of Datalog rules that verify whether all operational conditions for transfer are satisfied. The tool recommends or automatically executes a transfer only when the flight is free of conflicts and complies with rules, procedures, practices, and LoAs. If any required condition is not fulfilled, the tool withholds the transfer and identifies the need for prior coordination.

(iv) Coordination tool - integrated with all other support tools, as certain actions require prior coordination with adjacent units. It manages both inbound and outbound coordination processes. The tool reduces the need for verbal coordination, particularly when coordination is triggered by a support tool. It automatically prepares and transmits the appropriate OLDI messages to the relevant external units and provides clear feedback to the ATCO. The tool determines which message type is required, identifies the appropriate receiving sector, and pre-populates the coordination message accordingly. While it frequently operates in conjunction with other tools, the Coordination tool also supports the ATCO whenever coordination is required outside the scope of tool-generated actions.

3.3 ATC Simulator

The Air Traffic Management (ATM) system employed in this project is built on a service-oriented architecture, and as such works by passing messages over a network between various services. This design allowed us to incorporate the ASA into the ATM system with minimal changes to the ATM system as the ASA system communicates directly with the Air Situation Display (ASD) of the TERN ATC Simulator.

3.3.1 HMI

The AWARE Human-Machine Interface (HMI) is built on top of the ASD. Customised to support AWARE specific requirements, the ASD, incorporates dedicated UI elements designed for enhanced situational awareness and interaction. The HMI provides an interface to the four ASA tools.

- CD&R tool
- Transfer tool
- QoS tool
- Coordination tool

These tools are described in more detail in the D2.18 Report on ATCO support tools document [3].

3.4 Human Operator (ATCO)

EUROPEAN PARTNERSHIP
Co-funded by the European Union

This section describes the role of the human in the loop and HMI interaction philosophy.

3.4.1 HMI Interaction philosophy

The HMI's design aims to provide a clear and intuitive user experience by making AI suggestions visually distinct. All suggestions received from the ASA tools, follow the same design language: location (track label), glow effect and interaction pattern.

The track label serves as the main interaction interface between the ATCO and flight information on the ASD. Noting clearances, and accessing primary flight information is done through the track label. Therefore, placing AI suggestions directly to the left of the track label ensures that suggestions are always connected to core information, and easily noticeable.

Upon receiving a suggestion from one of the ASA tools, the associated track label and the suggested action are highlighted with the "AI Glow". Multiple states of the track label have been defined, and the corresponding states with the AI Glow in advisory and execution mode. These states are supported by a clearly defined colour system that ensures visual consistency across all tools.

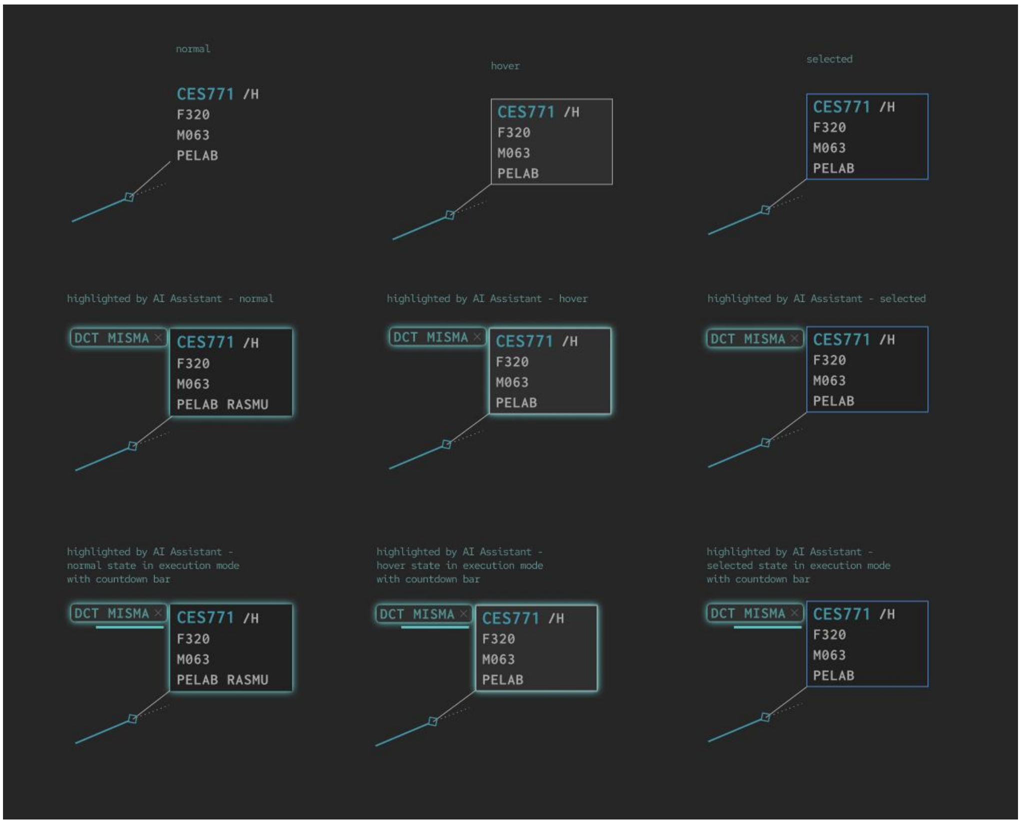

Figure 2.Different states of the track label, and the position of the AI Suggestion button in Figma

EUROPEAN PARTNERSHIP
Co-funded by the European Union

4 Communication Between ASA System and ATC Simulator

When defining the communication mechanisms between the various systems involved in the AWARE project, several guiding principles were adopted. Two of the most important were the preference for open-source technologies wherever feasible, and the establishment of explicit, written specifications for all messages exchanged between systems.

Using established open-source technologies allows the project to leverage well-tested and documented code and systems. A key technology in this regard is the NATS messaging system, which is used to stream messages between the different systems. NATS provides client libraries for most major programming languages, accompanied by extensive documentation and numerous implementation examples. This significantly facilitated the rapid establishment of reliable communication between the project's system components.

In addition, the project makes extensive use of Google's Protocol Buffers as a schema language for defining the structure of messages exchanged between systems. Protocol Buffers provide a set of tools capable of automatically generating code for serializing and deserializing messages between language-specific data structures and a platform-independent wire format. Beyond the technical benefits, the use of clearly defined message schemas has also improved cross-team understanding by providing a precise textual description of the data exchanged between system components.

4.1 Communication Principles

The system has three main components: The TERN ATC Simulator (Polaris + Orion), the ASA and its knowledge graph, and ZHAW's gaze tracking and intent analysis. Messages flow from the TERN ATC Simulator to the other two components, from ZHAW to the ASA, and from the ASA back to the TERN ATC Simulator. The table below summarizes the message topology of the system.

sender	receiver
TERN ATC Simulator	ASA
TERN ATC Simulator	Intent analysis module
Intent analysis module	ASA
ASA	TERN ATC Simulator

Table 1: Summary of message topology

A client–server architecture was adopted for message exchange between the systems. Messages flow through a centralized streaming system in a publish-subscribe workflow based on topics. An alternative peer-to-peer approach based on HTTP was initially evaluated. However, testing revealed that the resulting message latencies did not meet the project's requirements. Consequently, the centralized publish–subscribe model was selected as the preferred solution

Latency is an important consideration in the system design. The faster messages can be delivered from the TERN ATC Simulator to ASA and ZHAW, the sooner these components can analyze system activity and generate recommendations. However, the operational requirement is for near-real-time performance rather than strict real-time processing. Message delivery times on the order of a few hundred milliseconds are therefore acceptable, as such delays are negligible compared with the time required for message analysis or the several seconds that typically elapse between observable updates in the ATM system state. For example, system track updates are generally received only every few seconds.

4.2 Interface Description

4.2.1 Communication Protocols

The messages that the TERN ATC Simulator sends form a fairly continuous stream. In typical operations the system emits up to 100 messages per second, most of which describe what is happening on the ASD screen. This makes it natural to consider a streaming system to transport these messages.

The messages that the ASA sends back to the TERN ATC Simulator could fit into a standard request-response workflow. However, most existing technologies that implement such a flow (at least ones based on HTTP transport) expect responses to follow requests within a handful of seconds at most. Meanwhile, it would be normal within our system for a response to a suggestion not to arrive for up to 30 seconds. We thus decided to stream those messages. Keeping the number of different technologies used minimal also helps with development velocity, as there are fewer parts to be integrated into the various systems.

The NATS messaging system was selected to be the preferred streaming technology. NATS is an open-source system that, at the most basic level, provides ephemeral publish-subscribe streams over TCP/IP based on ad-hoc topics. It promises good performance, which it has delivered in our tests, both over a local network and a VPN between Iceland and Zagreb.

Alternative approaches were also considered, including the use of HTTP as the message transport mechanism. However, this approach encountered recurring issues related to the reuse of HTTP connections, which significantly affected performance. Without reliable connection reuse, the system was unable to achieve the required throughput and latency characteristics. After spending some time investigating and debugging these issues, the team evaluated NATS as an alternative solution. A working implementation meeting the required performance targets was achieved within less than a day, which confirmed its suitability for the project's communication infrastructure.

4.2.2 Message Types and Data Structures

The messages sent around the various pub-sub topics are defined by a Protocol Buffer schema. These are text descriptions of a message that various tools can use to generate code to handle the

EUROPEAN PARTNERSHIP
Co-funded by the European Union

transformation of a programming language's internal data structures into a format that can be passed between computers, and back again.

As a simple example, the following snippet describes a message that communicates the callsign and altitude of a flight:

```
message Track {
    string callsign = 1;
    int32 flight_level = 2;
}
```

The code generated from this description would then be able to convert a data structure that describes such a message to a format suitable for transmission over a network. For example, one option would be to convert it into JSON format:

```
{"callsign": "RYR7654", "flight_level": 300}
```

A complete description of the messages exchanged via the different AWARE components is outside the scope of the current document, but their contents are roughly as follows:

- From TERN ATC Simulator
  - System tracks
  - Flight records
  - The predicted trajectories of each flight
  - Active safety net alerts
  - Information about restricted areas within a sector
  - Telemetry describing user interactions with the ASD
  - The acceptance or rejection of an ASA suggestion
- From the ASA
  - Suggestions of clearances from the various AI tools
- From ZHAW
  - Analysis of ATCO intent

4.2.3 Timing and performance constraints

Update rates vary significantly between the different message types within the AWARE system components. At one end of the spectrum, the TERN ATC Simulator emits a message each time the mouse pointer changes position on the ASD screen, resulting in a nearly continuous stream of messages. Other messages are produced less frequently. For example, new system tracks arrive every three seconds, which also represents the lower bound on the frequency of messages derived from them, such as updated flight records, trajectory predictions, or safety net alerts. Suggestions from the ASA are typically generated at intervals measured in tens of seconds and may extend to several minutes when there is limited activity in the system.

The communication between the components operate with near-real-time latency requirements for message delivery. Delays of a few hundred milliseconds are generally acceptable, as the system operates on time scales comparable to human reaction times. The primary event triggering state updates in the ATM system is the arrival of new system tracks, which occurs approximately every three seconds. Although aircraft move at high speeds, this interval still allows sufficient time for human operators to respond to changes in their movements.

All messages within the AWARE system are timestamped at the moment of creation. The consuming components compare their local time with the message timestamp to estimate their lag relative to the message stream and issue warnings if excessive delays are detected. This mechanism provides an additional safeguard for the timely delivery and processing of information, which is particularly important in safety-critical ATM environments where outdated data could lead to incorrect system behaviour or operator guidance.

5 Integration of ASA in the ATC Simulator Environment

5.1 Integration Testing Approach and Achievements

Communication between the ATC simulator and ASA is implemented through the NATS streaming system, which in turn operates over standard TCP/IP connections. In the weeks leading up to the formal integration at LFV's R&D centre in Malmö in February 2026, this architecture enabled the establishment of a virtual private network (VPN) between TERN and FTTS. This setup allowed the systems to communicate using essentially the same configuration that would later be used during the on-site tests.

Initially, a one-hour integration test session was scheduled per week. In the final weeks prior to the on-site integration, this was increased to two one-hour sessions per week. These remote integration sessions allowed several communication-related issues to be identified and resolved in advance, thereby reducing the amount of troubleshooting required during the on-site activities in Malmö.

This approach proved successful. After setting up the various systems in Malmö, smoke test scenarios could be executed immediately, confirming that communication between all relevant systems operated as intended. In this context, successful communication refers to the timely exchange of messages and the ability of each system to correctly read and write the messages relevant to its operation. As discussed below, several higher-level issues were nevertheless identified during the testing sessions.

The testing activities in Malmö also confirmed that the user interface additions to the ATC simulator, as well as the UI improvements implemented since the first prototype evaluation in November 2025, were effective in reducing or eliminating the issues identified during those earlier sessions. In addition, the sessions provided an opportunity to evaluate the new AI assistance interface alongside the existing UI functionality and to gather valuable feedback from participants.

Following the Malmö integration activities, the practice of conducting twice-weekly remote integration tests between TERN and FTTS has been continued. These sessions are currently focused on diagnosing and resolving the issues identified during the February testing, in preparation for the final validation activities.

5.2 Simulator Scenarios

The scenarios that were used to test the system integration were built on realistic traffic scenarios for a control sector based on the sectors ESMM_2 and ESMM_3, using real flight data collected and provided by Luftfartsverket (LFV). The combined sector called ESMM_23 is an ACC sector controlling flights from F285 to F450. The traffic flow mainly consists of pure enroute traffic, with some vertical movements for flights departing from and arriving at Gothenburg, Malmö and Copenhagen airports. The ESMM_23 controller handles coordination with four external control centers: Denmark ACC, Copenhagen TMA, Gothenburg ACC and Stockholm ACC. Note that Gothenburg ACC was split out from the Malmö control area, in order to add more external coordination cases.

The test scenarios were created to provide medium to very high load on the ATCO with ~30 flights in 20-25 minute scenarios. The ESMM_23 controller has a maximum load of around 20 simultaneously assumed flights, with a lot of crossing traffic and several conflict scenarios. Most of the traffic in the scenarios enter ESMM_23 at some point during the exercise, with the remaining (<20%) flights are pure background (unconcerned) traffic. The scenarios have a maximum of around 25 simultaneously active flights (flights inside or coordinated into the Malmö area of responsibility), which should provide a fair load on the ASA system.

The validation scenarios were designed to cover specific traffic situations that would activate all four ASA support tools. Scenario 1 focuses on analysing ATCO performance under medium traffic load conditions. Scenario 2 is a high traffic load environment, with focus on different conflict resolution cases and strategies (entry and exit conflicts, vertical conflicts, triple conflicts). Scenario 3 examines traffic interactions with Temporarily Reserved Areas (TRAs). Finally, scenario 4 presents a futuristic high-load scenario with traffic levels 25% higher than the most occupied interval. The objective of scenario 4 is to evaluate system performance in such an environment which is beyond current operational realism but illustrates potential future system capabilities. Overall, the scenarios were designed to trigger the ASA system in different traffic scenarios to assess its performance and effectiveness of support provided to ATCOs.

5.3 Limitations

This sectiondescribes the limitations of the ATC Simulator and the ASA System based on the state of those systems during the on-site integration test in February 2026.

5.3.1 ATC Simulator

The following paragraphs list the functional limitations on the ATC Simulator system that were known beforehand or discovered during the February integration workshop. The limitations are split into different high level functionality categories.

Route Adherence Monitoring (RAM)

RAM has not been configured in the system. In practice, when AI assistance is active, there should be little need for ATCO initiated flight vectoring, as such conflict resolutions would be largely replaced by AI suggested re-route or flight level change clearances. This in turn should lead to fewer situations where RAM would be helpful. RAM would thus mainly be helpful in non AI-assisted mode as applied in the baseline scenarios.

Coordination

The TERN ATC Simulator currently only handles basic coordination. Advanced and revised coordination such as RTI, REV etc are currently not supported. This has the following two implications on coordination scenarios:

- Only initial coordination (ACT) can be handled by messaging when running scenarios with no AI support, requiring more verbal coordination than some ATCOs are used to.

- When running with AI support, verbal re-coordination handled by the ATCO without AI involvement can not be integrated into the ASA knowledge graph via the data communication.

Datalink

Speed and heading (vectoring) clearances over datalink (CPDLC) was not supported. The consequence is slightly increased complexity for the ATCO, who needed to remember which types of clearances should be performed over voice for CPDLC equipped flights. It is planned to have CPDLC support for those clearances working during the validations planned in April.

Update Exit Flight Level (XFL)

Due to missing configuration parameters, the ASD did not provide support for setting the exit flight level (XFL). Neither applying a new XFL as part of an ASA re-coordination suggestion (RRV message), nor setting the XFL manually through the UI was supported. Setting the XFL is necessary in order for the ASA transfer tool to work correctly. The ready for transfer check depends on that the ASA knowledge graph has the most up to date information on the negotiated XFL.

Vertical trajectory profile

The Conflict Detection and Resolution tool (CD&R) depends on the ATC Simulator trajectory for conflict detection. This trajectory is a strategic trajectory, always aiming to bring a flight up to the requested flight level (RFL) for the cruise phase. When a flight has a conflict on the RFL level and is therefore cleared to cruise on e.g. RFL +-10FL, the ATC Simulator backend will model the trajectory to descend/climb back to the RFL. This would lead to the CD&R constantly rediscovering the conflict, although it was already resolved.

Initial CFL Set Too Late

The cleared flight level is set in the ATC System backend for inbound flights when the flight passes the AoR boundary. This is too late for our AI assisted setup as the ASA system uses the CFL for conflict calculations even for flights outside the Malmö AoR.

5.3.2 Improved ASA System

Data processing speed

The major limitation of the ASA system was the slow processing of data from the ATC System. The ASA was not able to keep up with the data flow from the ATC System, causing ASA suggestions to lag behind the actual air situation, and effectively missing out on "accepted suggestion" events from the ATC System. The latter would cause e.g. transfer suggestions to be re-sent after the ATCO had already performed the suggested transfer.

Memory exhaustion

The ASA system runs within Docker containers, which are independent subsystems of the host machine with allocated resource limits. Initially, these containers were configured to use 50% of the host system's memory. As a result, the ingested data consumed the available memory before the exercise concluded, disrupting and blocking the ASA's system workflow.

Coordination Tool

The ASA coordination tool was not ready, so the integration of that part with the ATC simulator was not tested during the integration activities.

Conflict Detection and Resolution Tool

The CD&R tool was not consistent in suggesting +-20FL climb or descent. Much larger deltas were observed in some cases. Furthermore the tool sometimes suggested flight levels that were not multiples of 10, which are currently never used for ATC clearances. A third issue that was observed was that the tool would sometimes suggest a level where the flight would obviously be in conflict. This issue came from limitations in the trajectory predictor on which the tool relies. The predictor generates trajectories based on the assumption that the aircraft will remain at the requested flight level and initiate descent at the calculated Top of Descent, which is not always the case in actual operations. Additionally, the CDR tool's time-based triggers were not applied consistently.

Quality of Service Tool

The QoS tool was generating recommendations that were operationally ineffective or impractical. For example, it suggested direct routes to aircraft less than one minute from the sector boundary or climbing flights to RFL even after they had begun their initial descent toward destination.

System's adaptability

The improved ASA system did not provide sufficient flexibility in its operational configuration. It functioned exclusively in either suggestion mode or execution mode, with no capability to switch between the two. In addition, ATCOs were unable to choose the degree of system support or control how frequently the system generated outputs.

AI suggestions based on ATCO intent

The integration of the ATCO intent assessment with the ASA tools was not ready as well as the task prioritization.

5.4 Recommendations for further enhancements

5.4.1 Further ASA improvements

The major enhancement of the ASA system considers the final integration of all system components. Although individual components currently function independently, real-time integration is still required. Particularly, the integration of gaze and HMI tracking with the ATCO task priority. This integration would allow the system to determine both the appropriate action and the support level in which the system should function.

Action selection module needs to be clearly defined, aspects that must be addressed are:

1. How many actions can be presented to ATCO simultaneously?
2. What triggers the selection and execution of action?
3. At which level of support should the system be functioning?
4. How action can be selected while ensuring the ATCO remains in the decision loop?

Additionally, ATCO support tools require enhancement in several areas:

1. CD&R tools should incorporate actual trajectory data alongside predicted trajectory data to detect conflicts that arise during ATCO tactical interventions. Reliance on trajectory prediction has resulted in both false positives and false negatives in tool outputs.
2. The Transfer tool should more accurately determine the correct frequency to which a flight should be transferred.
3. The Coordination tool should be fully integrated with the other support tools.
4. The QoS tool must correctly account for the activity status of TRAs. Currently, this information is not processed, and the primary functionality of the QoS tool is only partially validated.
5. All tools should be capable of continuously collecting and storing TERN ATC System data. Currently, memory shortages occasionally lead to the loss of relevant tool outputs.

5.4.2 Further ATC Simulator System improvements

For the ATC Simulator, the limitations were analysed, and the following system improvements are recommended, in order of importance:

1. Update the ASD configuration to allow the user to set the XFL
2. Improve the ATC backend functionality to allow setting the initial CFL, to ensure that the CFL is available before a flight enters Malmö AoR.
3. Add ATC backend functionality to set the XFL based on ASA suggestions for re-coordination, where the new coordination changes the XFL.
4. Add datalink support for Speed and Vectoring clearances applied through the ASD UI.
5. Configure RAM functionality in the ASD.

6. Conclusion and Next Steps

6.1. Summary of Integration Achievements and Limitations

The integration testing between the TERN ATC simulator system and the ASA system successfully confirmed end to end functionality in the QoS, CD&R and Transfer tools. All low level data communication is in place and works correctly. The ASD UI successfully presented all AI suggestions that were received.

The main limitations that have so far been found are in order of importance:

1. Missing ASA Coordination tool
2. Missing ATCO intent assessment integration into AI suggestion prioritization
3. Data processing speed in ASA
4. ATC System trajectory compatibility with CD&R needs
5. CD&R suggestion inconsistencies
6. No XFL change in ASD
7. Initial CFL is set too late in ASD
8. Missing Route Adherence Monitoring (RAM) in the ASD
9. Simulation Engine limitations in OLDI message support

7. References

[1] AWARE consortium, D2.16 AWARE - A Method for ATCO Intent Assessment, 2025

[2] AWARE consortium, D2.17 AWARE - A Module for Mapping ATCO Intent to Tasks, 2026

[3] AWARE consortium, D2.18 AWARE - Report on ATCO Support Tools, 2026

8. List of acronyms

Acronym	Description
AAA	AI Assistant Application
AISA	Artificial Intelligence for Situational Awareness
AI	Artificial Intelligence
AIXM	Aeronautical Information Exchange Model
AoR	Area of Responsibility
ASA	Artificial Situational Awareness
ASD	Air Situation Display
ATC	Air Traffic Control
ATCO	Air Traffic Controller
ATM	Air Traffic Management
CD&R	Conflict Detection and Resolution
CFL	Cleared Flight Level
CPDLC	Controller–Pilot Data Link Communications
CPA	Closest Point of Approach
ESMM	Malmö area sector designator
FIXM	Flight Information Exchange Model
FL	Flight Level
HMI	Human–Machine Interface
HTTP	Hypertext Transfer Protocol
JSON	JavaScript Object Notation
LoA	Letter of Agreement
ML	Machine Learning
NATS	NATS messaging system
OLDI	On-Line Data Interchange
QoS	Quality of Service
RAM	Route Adherence Monitoring
RDF	Resource Description Framework
REV	Revision
RFL	Requested Flight Level

EUROPEAN PARTNERSHIP
Co-funded by the European Union

Acronym	Description
RRV	Referred Revision Proposal
RTI	Request Tactical Instruction
TCP/IP	Transmission Control Protocol / Internet Protocol
TRA	Temporary Reserved Area
UI	User Interface
VPN	Virtual Private Network
XFL	Exit Flight Level

Table 2: List of acronyms