



ASPRID

Airport System Protection from Intruding Drones

DEMO Software for Validation

Deliverable ID:	D3.3
Dissemination Level:	PU
Project Acronym:	ASPRID
Grant:	892036
Call:	H2020-SESAR-2019-2
Topic:	Innovation in Airport Operation
Consortium Coordinator:	INTA
Edition date:	04 February 2022
Edition:	01.00.01
Template Edition:	02.00.03

Founding Members





ASPRID

AIRPORT SYSTEM PROTECTION FROM INTRUDING DRONES

This Project Deliverable is part of a project that has received funding from the SESAR Joint Undertaking under grant agreement No 892036 under European Union's Horizon 2020 research and innovation programme.



Abstract

This document provides a technical documentation of the availability of the software developed and adapted for ASPRID validation activities.

The document describes the overall architecture of the validation platform and a guide of the applications that will be used in the gaming exercises.



Table of Contents

Abstract	2
1 Introduction.....	5
1.1 Purpose.....	5
2 Background	6
2.1 ASPRID project	6
3 Validation Platform Architecture.....	9
3.1 Validation Exercise Platform / Tool characteristics.....	9
3.2 Validation Platform Architectural view.....	12
3.3 Validation Platform and ASPRID System mapping and specifications.....	12
4 ASPRID Web App.....	15
4.1 Drone Operator	15
4.2 ASPRID Operator / Viewer	19
4.3 eArts integration	22
4.4 ASPRID Demo APP	23
5 References.....	24

List of Tables

Not Applicable



List of Figures

Figure 1. ASPRID Architecture Concept Scheme	9
Figure 2. Asprid prototype server	10
Figure 3. Asprid Configurator	10
Figure 4. eArts Aerodrome simulator	11
Figure 5. Asprid App	11
Figure 6. Validation platform architecture	12
Figure 7. ASPRID Architecture Validation Mapping	13
Figure 8. Drone Operator Layout	16
Figure 9. Drone Operator sidebar	16
Figure 10. Drone creation	17
Figure 11. Drone Trajectory	18
Figure 12. Drone Start Movement	18
Figure 13. ASPRID Operator Layout	19
Figure 14. Hazard sidebar	20
Figure 15. Countermeasure fired	20
Figure 16. Countermeasure failed	20
Figure 17. Countermeasure selection	21
Figure 18. Countermeasure successful, drone killed	21
Figure 19. eArts ASPRID integration	22
Figure 20. ASPRID Demo App	23



1 Introduction

1.1 Purpose

This document is the deliverable D3.3 “DEMO Software for Validation” and provides the availability note for the software developed in T3.3 within WP3 for ASPRID Project [8]. It is described an overall architecture of the validation platform, and a user guide of the applications that will be used in the gaming exercises.



2 Background

This Section reports the background information for the Validation platform description.

In detail, the background information consists of the following Sub-sections:

- The description of the Validation Platform Architecture in chapter 3.
- A User guide for all the tool used by the experts in chapter 4.

The HMI requirements could be found in deliverables D2.3 [2], D2.4 [3] and D2.5 [4].

2.1 ASPRID project

ASPRID (Airport System Protection from Intruding Drones), is the Exploratory Research response to the request made from SESAR, to cope with the problem of protecting airport operations from drone intrusions under a holistic and operationally oriented approach.

In particular ASPRID project objectives are:

- To get awareness of the critical aircraft and airport operations scenario degradation due to drone intrusions, deriving for them the possible evolutions with related details;
- To define a complete concept of an innovative single platform system architecture supported by procedures and regulation issues to manage both airport intrusion and airport operations efficiently.
- To validate the system architecture according to safety and security performance objectives.

For that purpose, ASPRID consortium determine that the best way to protect airport operations from intruder drones is increasing awareness about such undesired events and setting-up procedures and protocols to manage them aiming at mitigating their effects on the operations.

Moreover, all of this can be achieved by a technological system of systems that Detects, Identifies, and Tracks the threat. Being this system supported by collaborative decision support functionalities in terms of Threat Assessment & Alert, Communications and Procedures & Course of Actions (CoA) definition. Threat neutralization is as a final action that is defined on technological elements too.

ASPRID solution will consider the integration within all airport ecosystem, e.g.: Air Traffic Management Systems (ATM), airport Air Traffic Control (ATC), and U-space systems, fully supporting the implementation of new technologies with existing ones.



All the above require a real interdisciplinary approach on multiple levels to consider and integrate all those elements in an efficient way, including the human factor, present in all phases from alert to final decision making, and to countermeasures activation.

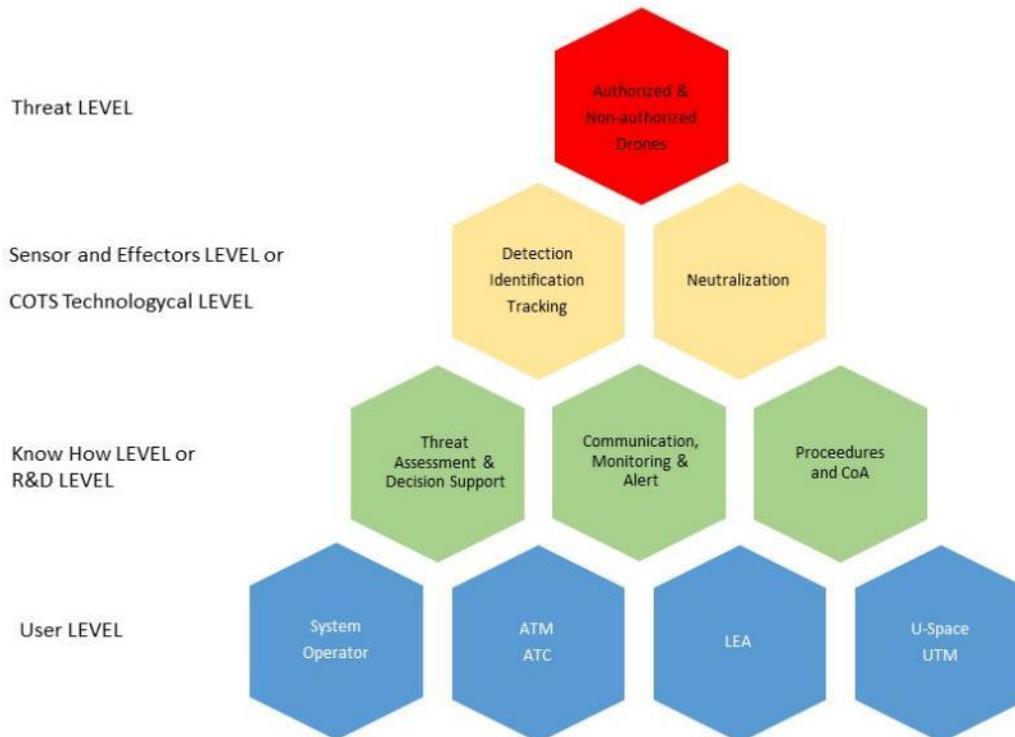


Figure 1 - ASPIRD pyramidal concept: layers and blocks

In this way, the project will address the following mainstreams:

- Reviewing state-of-the-art and assessment of applicable regulations and procedures.
- Identification of the problem (threat, assets to protect & operations) , setting the scenarios:
 - By means of risk and vulnerability assessment
 - Hierarchization & selection
 - Deriving decision support system requirements
- Definition of an airport protection measure, from drone system a collaborative decision support architecture on two levels:
 - Sensor and Effectors
 - Know How



- Perform validation activities:
 - Requirements integration
 - Sensitivity studies
 - Human Man Interface (HMI) based gaming exercises.

Artificial intelligence and HMI have been identified as enabling technologies to ensure the efficiency of the ASPRID system.



3 Validation Platform Architecture

The validation platform is a set of tools and Apps connected used to simulate the airport environment and the ASPRID system. The prototypes will replicate the expected behaviour of the architecture proposed in ASPRID deliverable D2.2 [1].

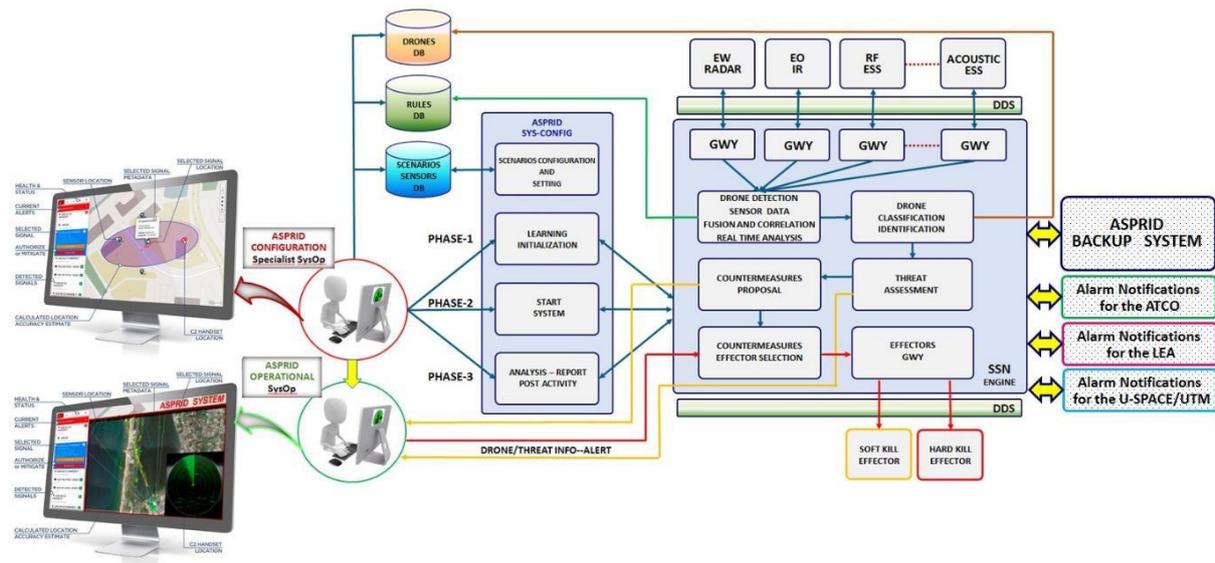


Figure 2. ASPRID Architecture Concept Scheme

3.1 Validation Exercise Platform / Tool characteristics

The validation exercise platform will be composed of the following tools, that will give a simplified¹ version of the ASPRID system (detailed in D2.2) and the environment:

- **Asprid server:** a backend that receives the Asprid configuration, the simulated flight and drone tracks, evaluate the threats and deliver the hazard to the Asprid users.

¹ The term “Simplified” here is twofold: the first meaning is that some parts of the real system are simulated (e.g. the drones, the sensors, the countermeasures), the other meaning is that some algorithms are simpler (for instance the drone detection phase is not implemented with AI, there are no sufficient data, and it is assumed that the drone is detected and classified if is in the sensors’ range).

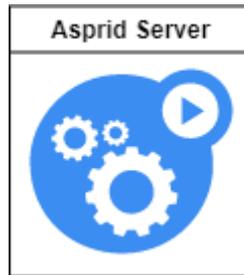


Figure 3. Asprid prototype server

- **Asprid Configurator:** a tool that helps to locate Critical Assets in the designated airport and the Countermeasure units with their related characteristics, together with the location and the range of the sensing elements. This configuration will be stored in the cloud and used by the Asprid prototype system and could also be exported in a standard format.

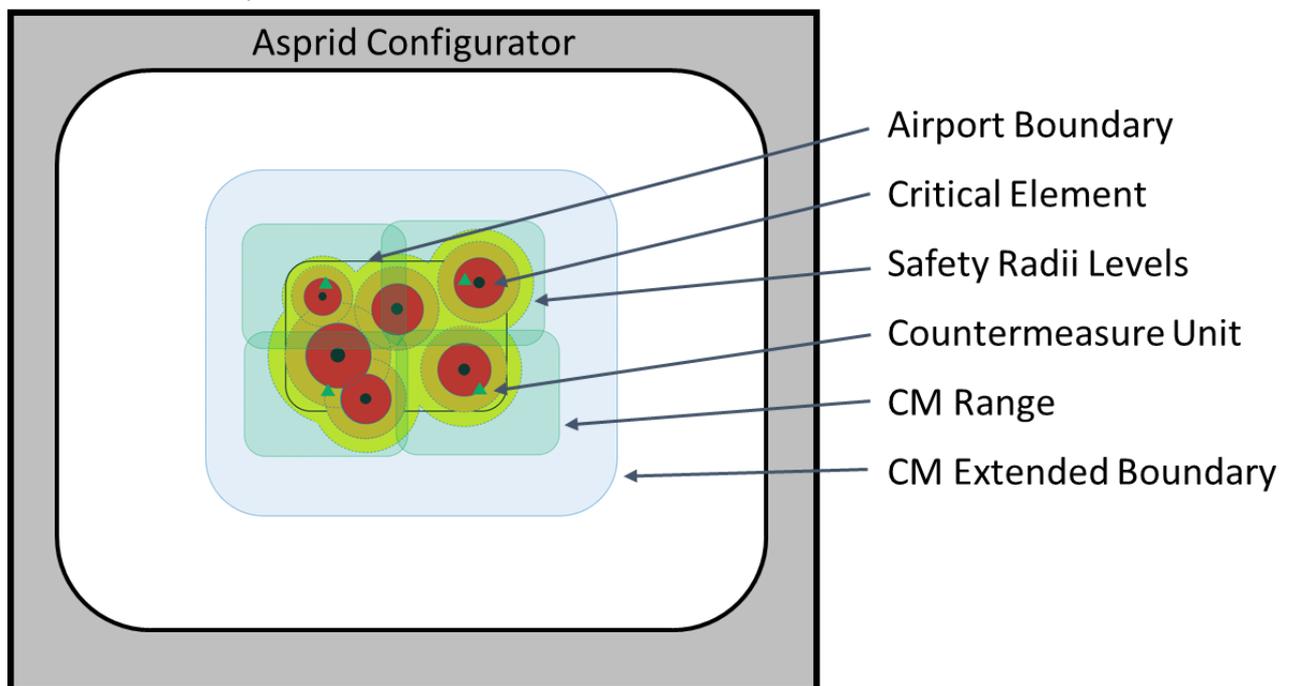


Figure 4. Asprid Configurator

- **eArts Aerodrome/TMA simulator:** an ATCO simulator for airport ground and terminal area operations. It can reproduce different roles (Apron Ground and Tower Controller), the TMA ATCO tactical Controller and the Pseudo Pilot Working Positions. The ATCOs positions could also be configured as Hybrid positions, when the Pseudo Pilot is not requested in the simulation. The server will be responsible to update aircraft tracks and their status, the ATCO CWPs will be used to display aircraft and drone tracks on the map, the type of drone (authorized, menace), and the type of risk/hazard

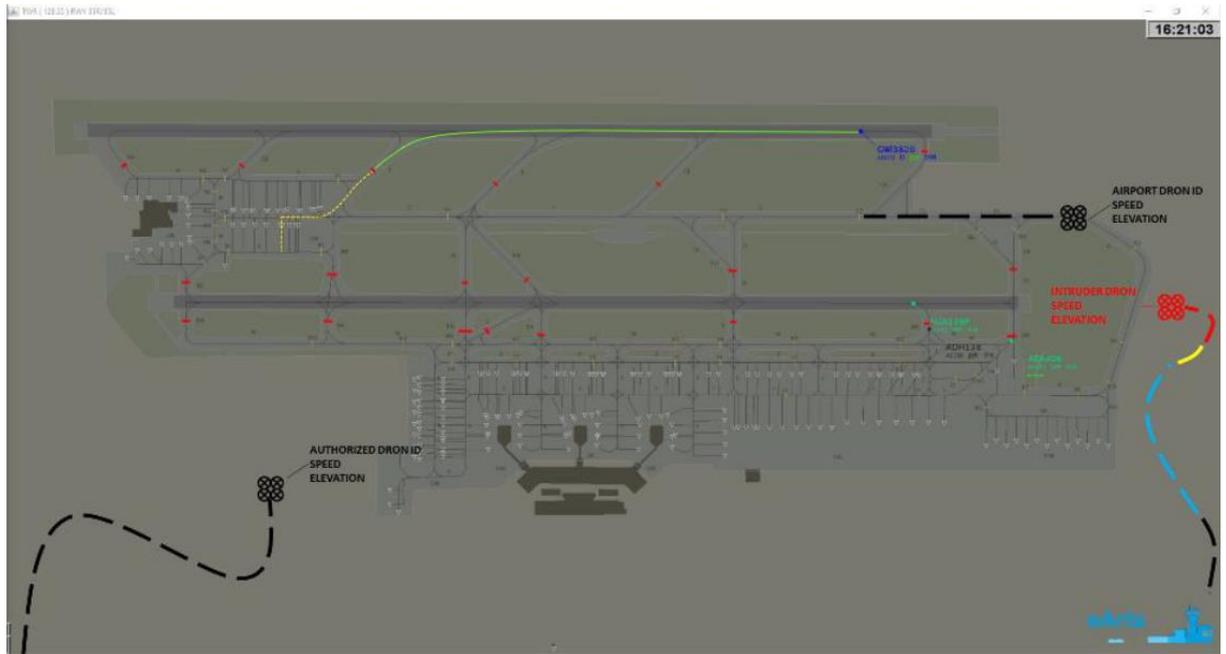


Figure 5. eArts Aerodrome simulator

- **Asprid Web App:** a frontend App that could represent different stakeholder roles (LEA, Ground Handler, Drone Operator) and display relevant information on an airport map. Some roles could interact with the displayed elements and perform some actions (i.e. the drone operator could change the path of the drone, the Asprid Operator could activate a specific countermeasure action)

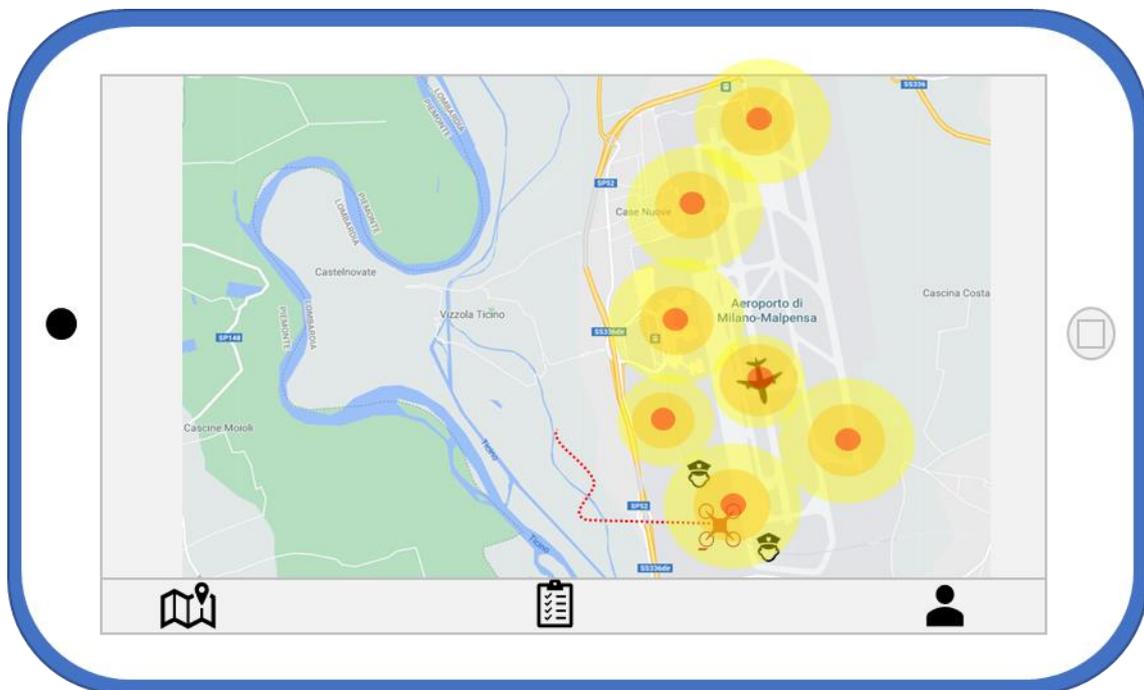


Figure 6. Asprid App



3.2 Validation Platform Architectural view

The following diagram represents how the tools are connected.

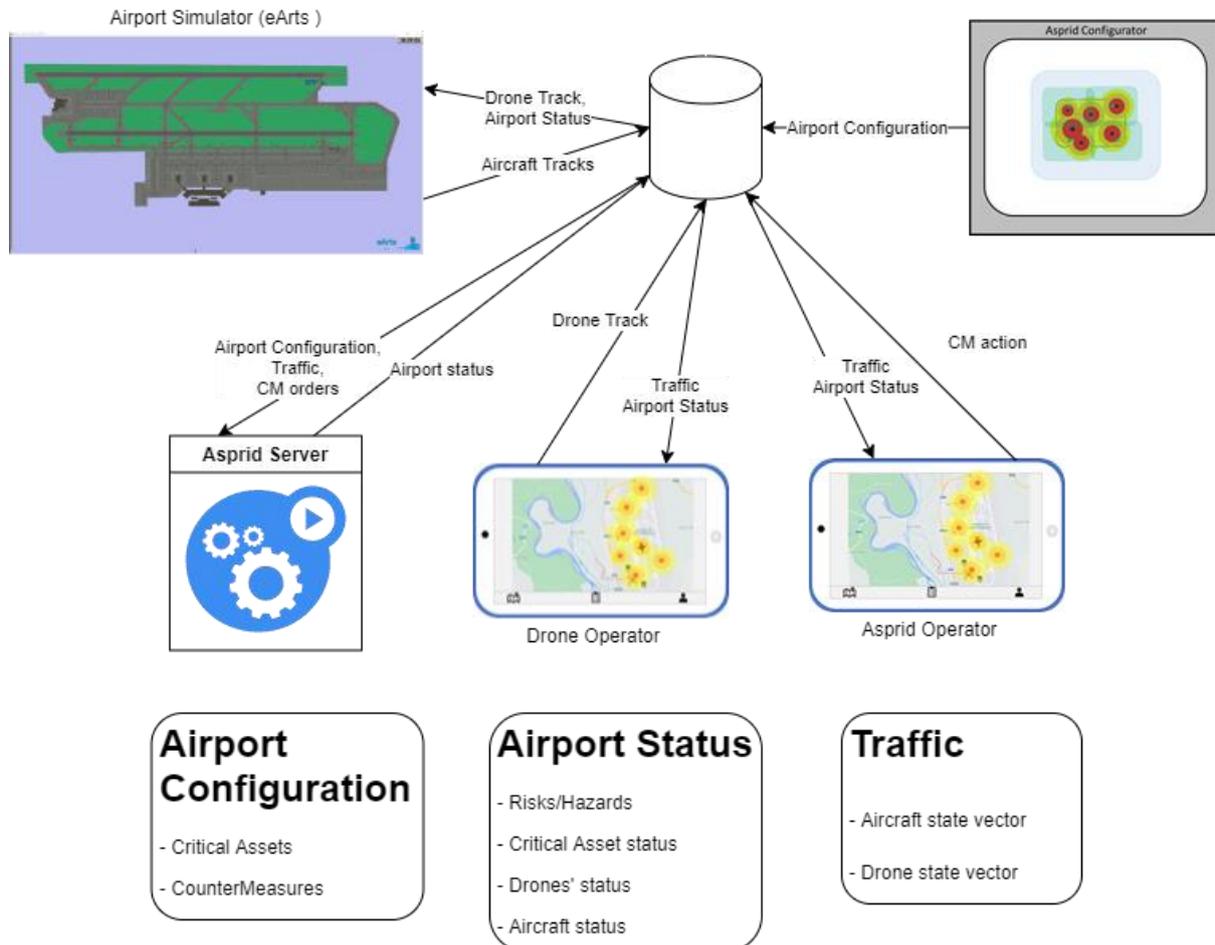


Figure 7. Validation platform architecture

The applications are connected through a Real Time Cloud Storage (FireStore), that stores all the information regarding the airport configurations, the airport status, and the traffic, and will notify the applications with a Publish/Subscriber pattern.

3.3 Validation Platform and ASPRID System mapping and specifications

In the following figure we try to map each component of the validation infrastructure, with its corresponding part in the proposed ASPRID system

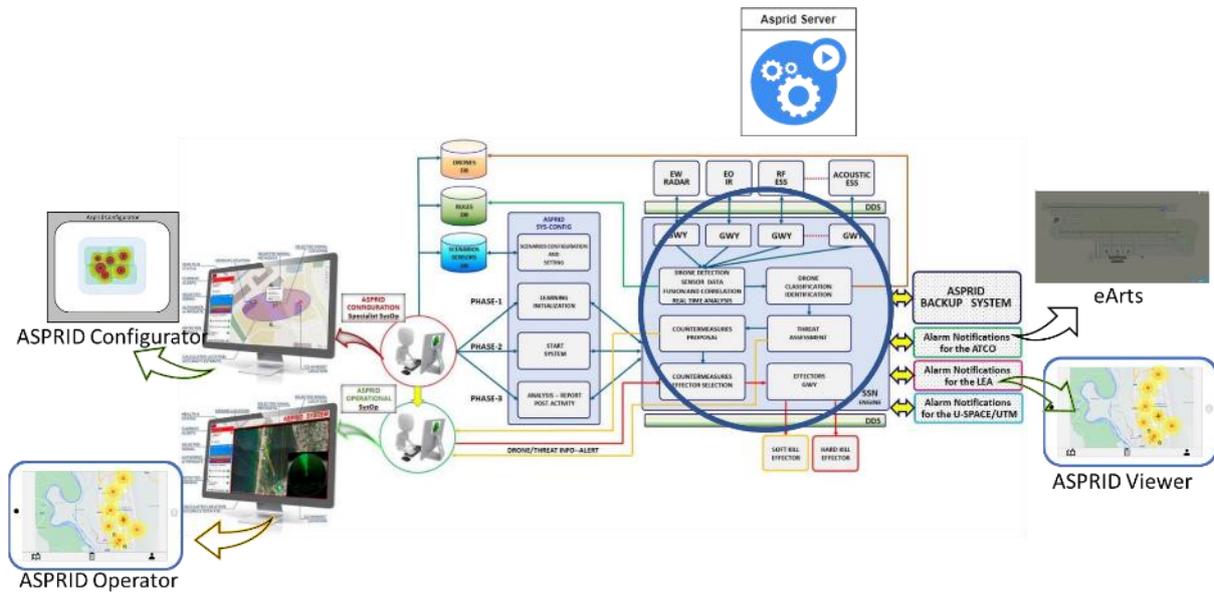


Figure 8. ASPRID Architecture Validation Mapping

The **ASPRID Configurator** has been developed with the requirements specified in D2.3 with frontend application in TypeScript and the backend in NodeJS with TypeScript. The frontend and backend communicate with a REST API protocol, and the backend is connected with the FireStore database to store the ASPRID airport configuration data.

The **eArts** Airport simulator, is a distributed simulator developed in Java/Kotlin and connected with the FireStore database to save the aircraft and vehicle positions and to read drone positions and hazard notifications.

The **ASPRID Operator/Viewer** is the frontend application for the ASPRID users and has been developed with the requirements specified in D2.3, D2.4 and D2.5 using the Flutter framework. It is connected with FireStore database to store the ASPRID orders (countermeasure activation) and to read planes, vehicles and drone positions.

The **Drone Operator** is the frontend application for the drone pseudopilot. It has been developed with Flutter framework and is connected with FireStore to store the Drone orders.

The **ASPRID Server** is responsible of

- update the drone movements,
- generate the hazard alarms (drones close to a critical asset)
- simulate the activation of a countermeasure

It has been developed taking into consideration the requirements specified in D2.4 and D2.5, using deterministic algorithms without the use of AI. It has been developed in NodeJS with TypeScript and is connected to the FireStore database to read aircraft/vehicle positions, the ASPRID orders and the Drone Operator orders. It updates drone position, detect a hazard risk with its classification, and will



determine the effect of an activated countermeasure. This information is saved in the Firestore database.

It will also store all the exercise data in Mongo DB database, for post analysis purposes.

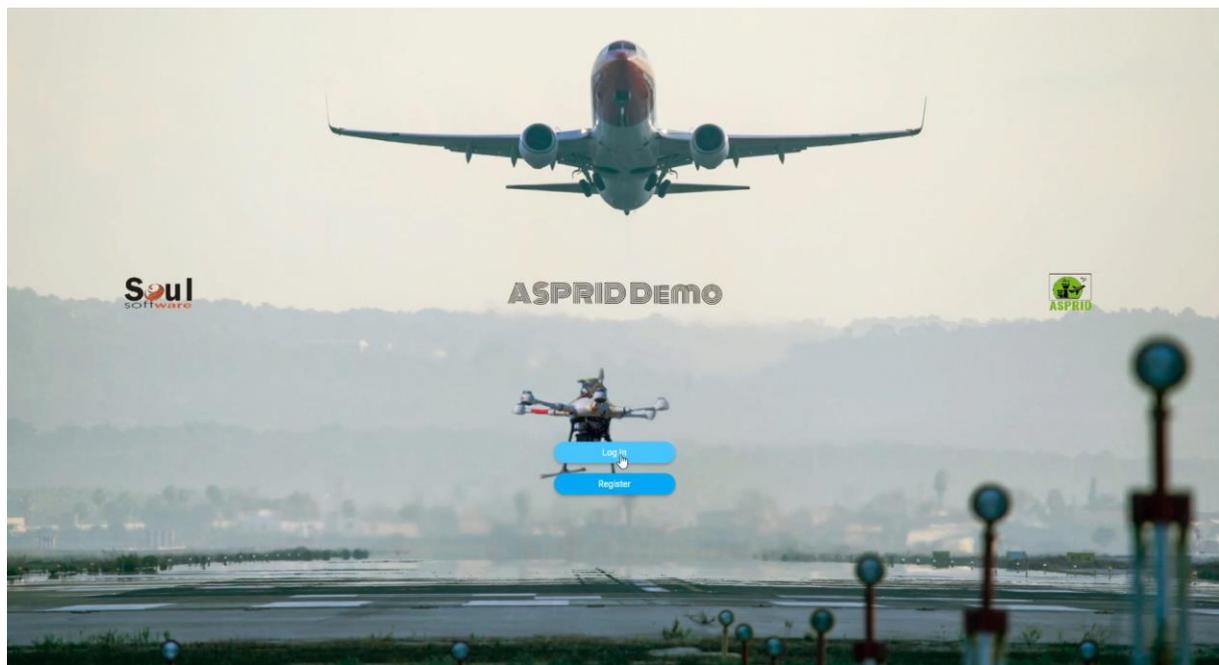


4 ASPRID Web App

This Web Application is used to join the validation exercise with different roles:

Drone Operator	manages one or more drones, and their actions
ASPRID Viewer	shows the airport environment on a Map, with the airplanes, drones and the generated hazards
ASPRID Operator	is identical to the Viewer, but with the ability to activate the Countermeasures

The Web App is accessible through the following link: <https://aspridapp.firebaseio.com/>, only for the duration of T3.4. The first page is the Login/Signup one.



After the Signup phase, the Admin will enable the user and assign the proper role.

Then the user can login to the application and select the proper role and the activated exercise.

4.1 Drone Operator

4.1.1 Layout

The following picture shows the application layout. In the top there is an app bar that displays the title of the application, the selected role, and the status of the exercise. The main part of the layout show a Map centred in the selected airport. On this map it is visible the critical assets and the areas covered

Founding Members





by the sensors. These features could be filtered with the button on the top left. There are also visible the airplanes, the vehicles, and the created drones.

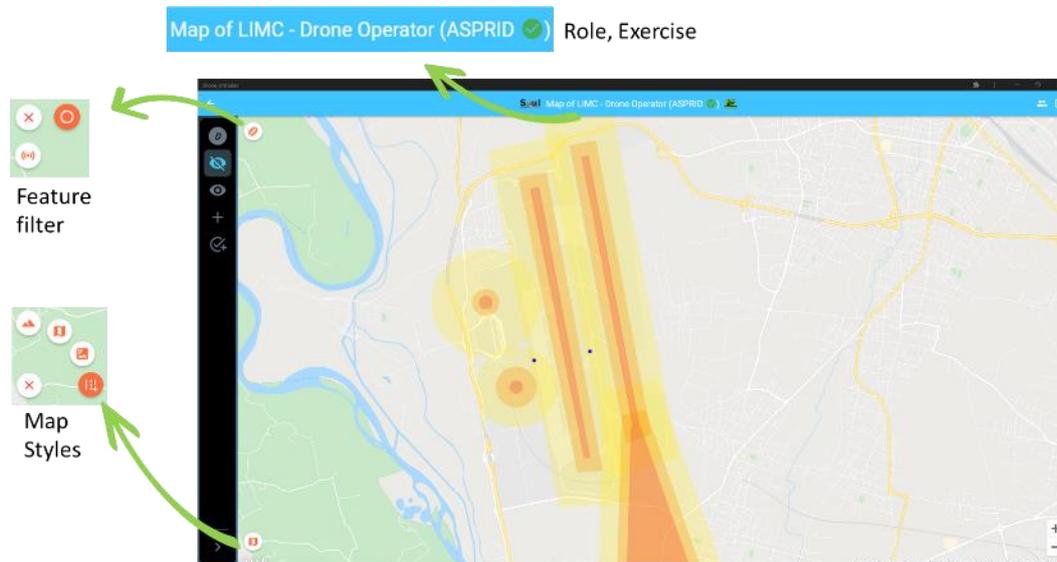


Figure 9. Drone Operator Layout

On the left side there is a sidebar with the actions to manage the drones

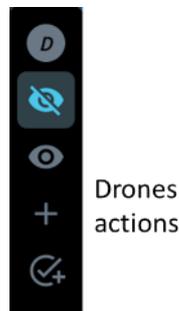


Figure 10. Drone Operator sidebar

4.1.2 Drone Creation

Clicking the plus button on the sidebar, the user can create a new drone, selecting from a list of available types, and placing on the map.

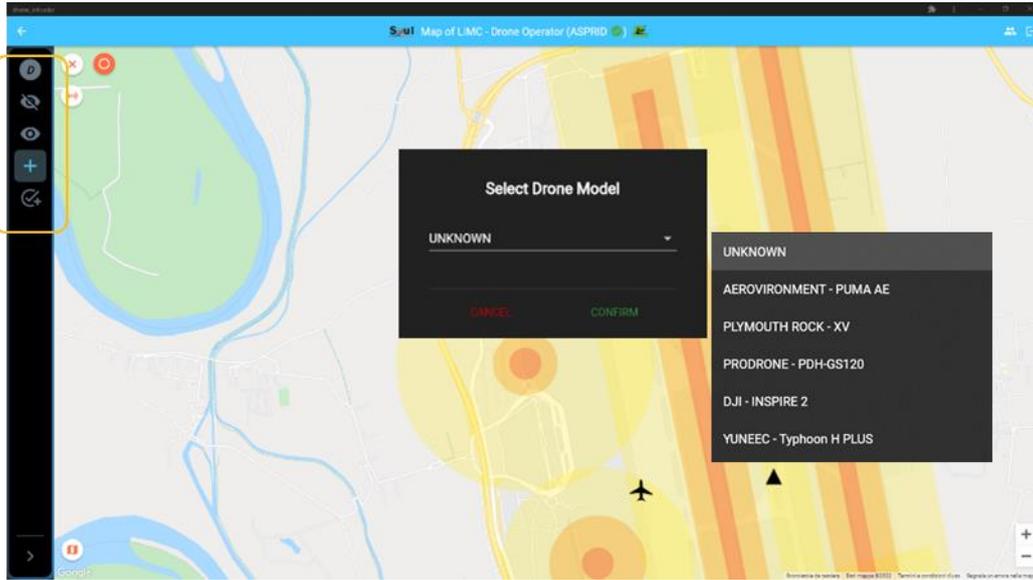


Figure 11. Drone creation

Clicking on the  button it is possible to create an authorized drone, that is highlighted with a green circle .

4.1.3 Drone Trajectory

Once a drone has been created, a trajectory could be added selecting the desired drone and clicking the points on the map.

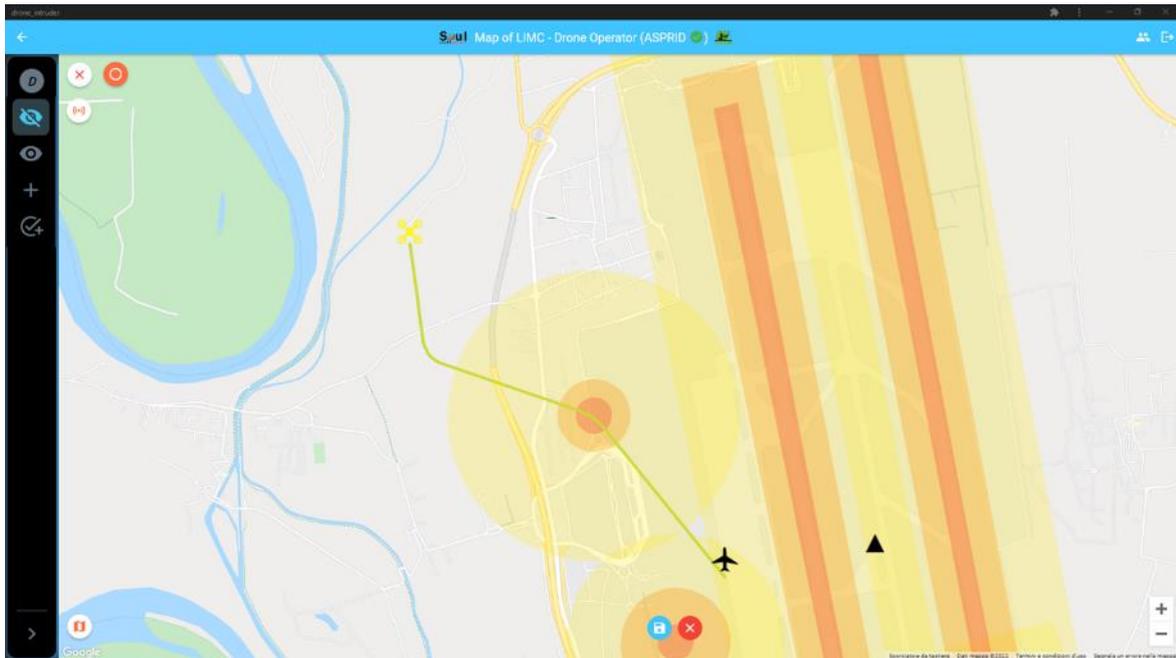


Figure 12. Drone Trajectory

Once a trajectory has been added to a drone, the user can start its movement by selecting it and clicking on the proper button, after set the desired speed.

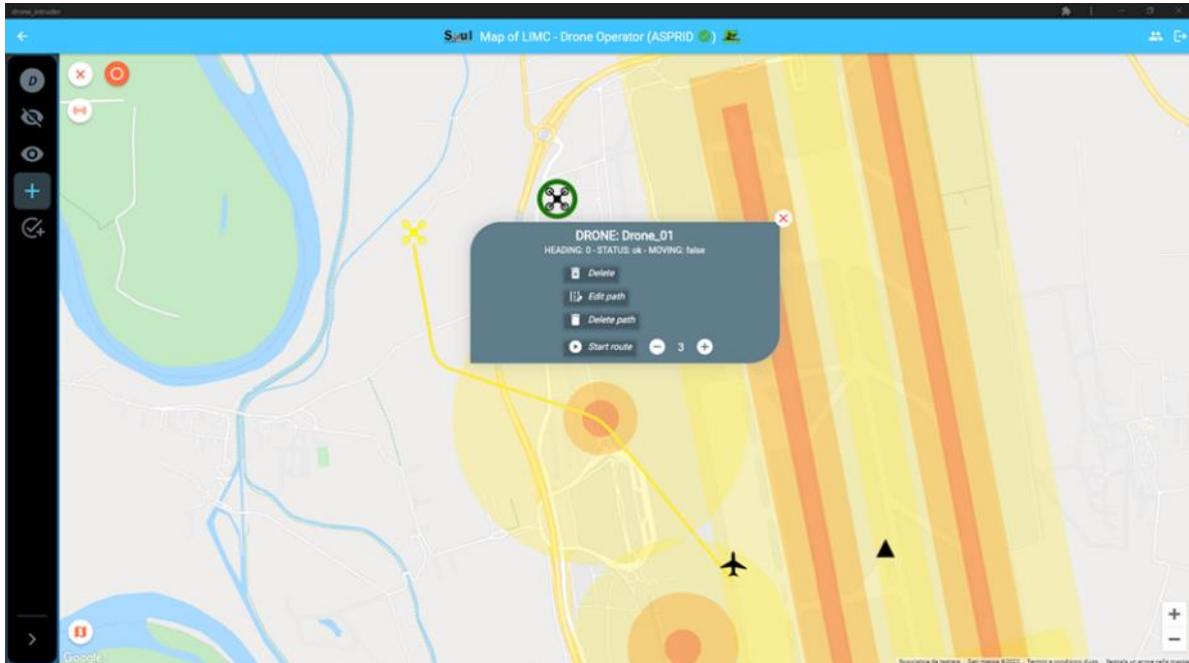


Figure 13. Drone Start Movement

Once the drone has started its movement, the user can put in hovering and then put in motion again.



4.2 ASPRID Operator / Viewer

4.2.1 Layout

The following picture shows the application layout. In the top there is an app bar that displays the title of the application, the selected role, and the status of the exercise. The main part of the layout shows a Map centred in the selected airport. On this map it is visible the critical assets, the areas covered by the sensors and the position of the countermeasure and their ranges. These features could be filtered with the button on the top left. There are also visible the airplanes, the vehicles, and the created drones.

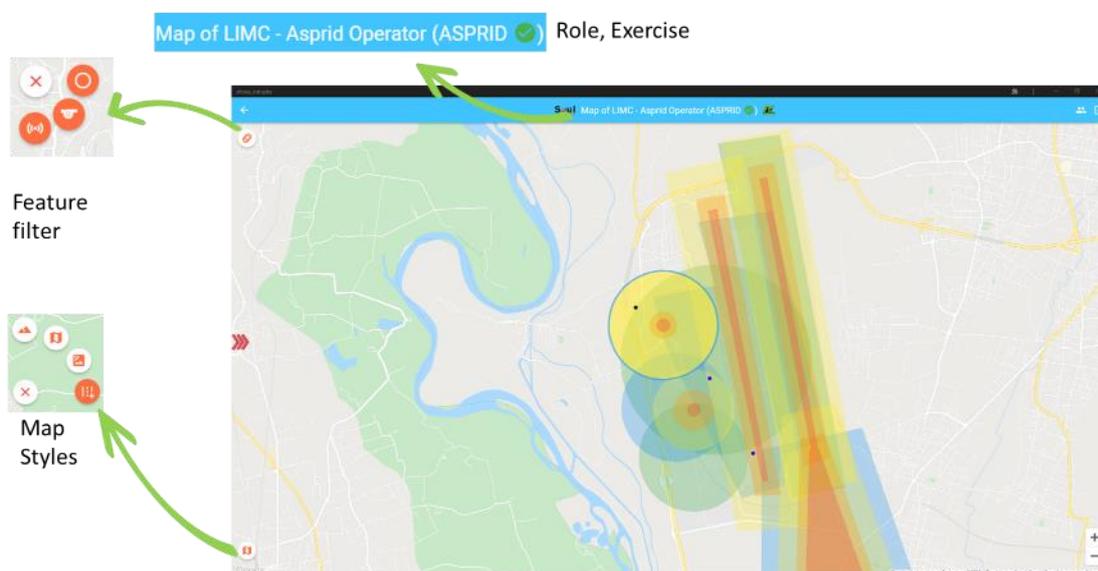


Figure 14. ASPRID Operator Layout

When the ASPRID system detects a hazard, when a malicious drone is close to a Critical Asset, the drone and the associated critical asset will be highlighted on the map. Clicking at the red arrow on the left, a sidebar will open, displaying a list of current hazards. A colour is associated with the risk of the hazard, yellow for low risk, orange for medium, and red for critical.

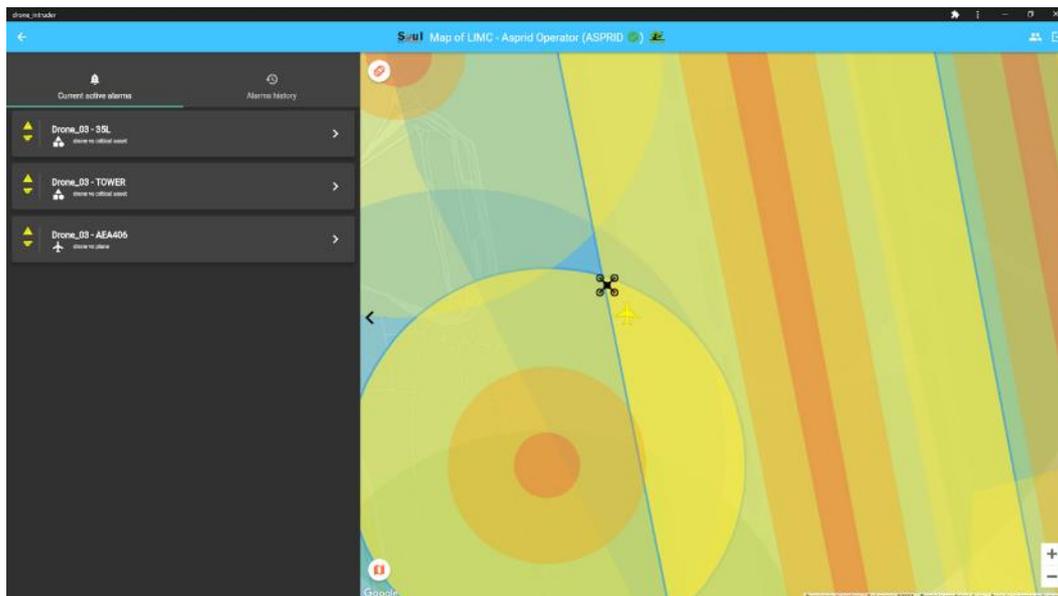


Figure 15. Hazard sidebar

4.2.2 Countermeasure Activation

The ASPRID Operator can activate the countermeasure if all the following conditions are in place:

- The malicious drone has caused a hazard
- The drone is inside the range of one or more countermeasure

In this case, the ASPRID Operator, selecting the drone, will open a dialog box, with a list of the activatable countermeasures.

4.2.3 Countermeasure status

The ASPRID Operator and the ASPRID Viewer, could see the current status of the countermeasure activation, on the sidebar. The countermeasure status could be fired/activated, failed or succeeded. The first two cases are shown with an icon coloured in yellow (fired) and red (failed). If the countermeasure action has been successful, the hazard is deleted, since it is not a risk anymore, and the drone is marked as killed on the map.



Figure 16. Countermeasure fired



Figure 17. Countermeasure failed

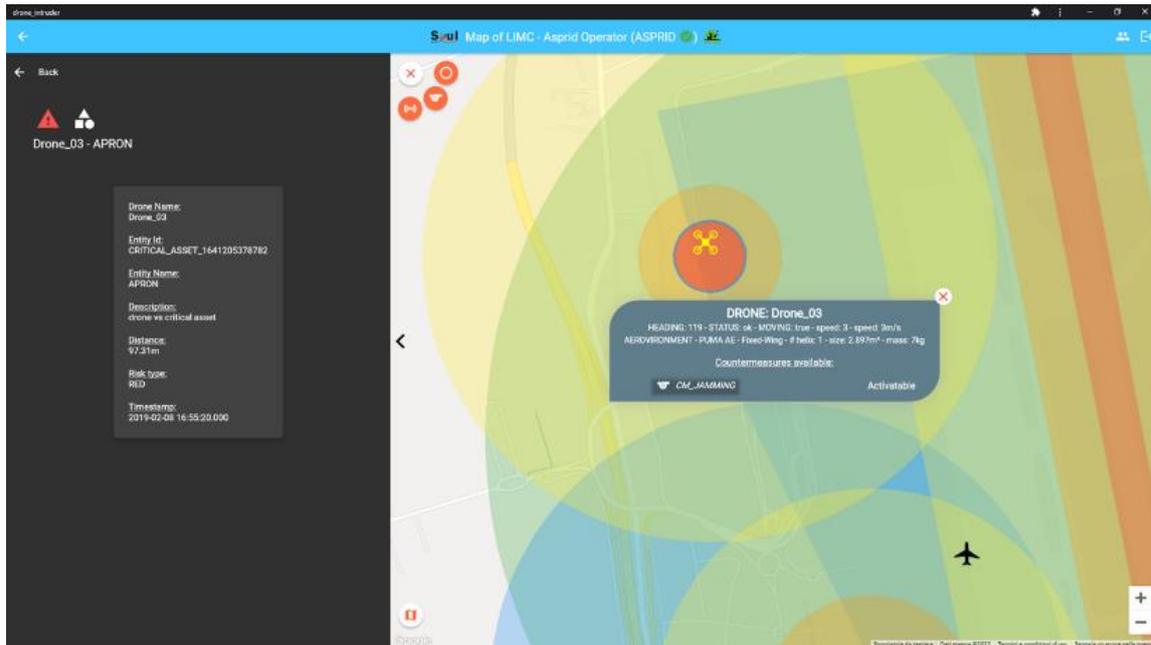


Figure 18. Countermeasure selection

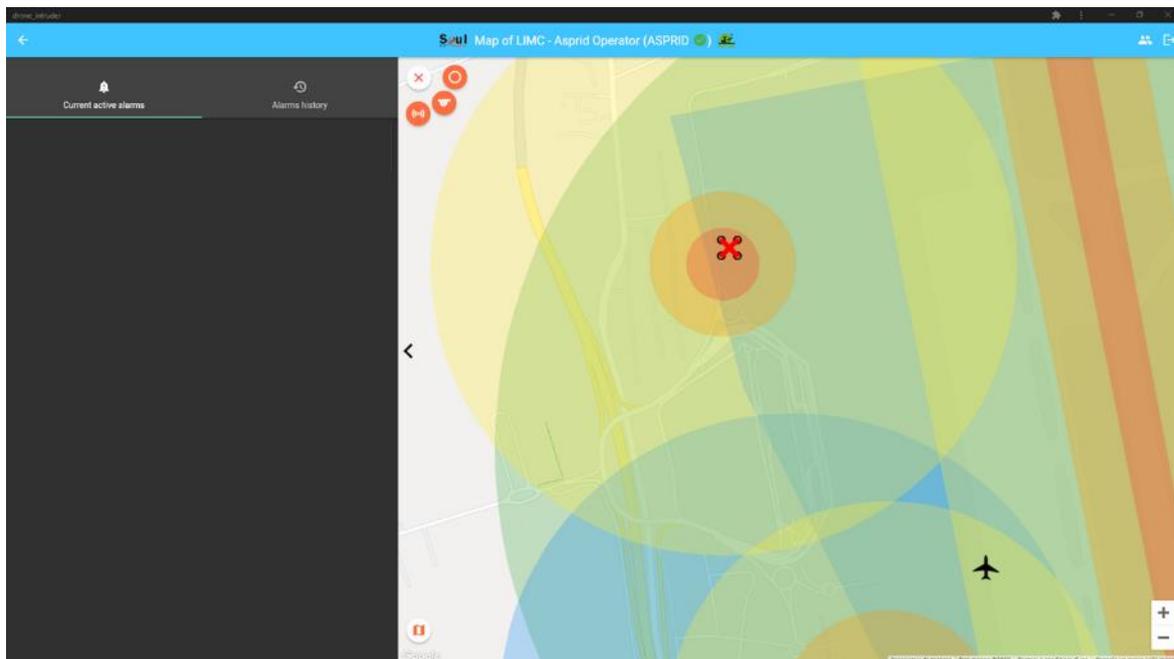


Figure 19. Countermeasure successful, drone killed



4.3 eArts integration

eArts is the aerodrome simulator for the ATC operations. It shows on a map the aerodrome features, runways, taxiways, aprons and the airplanes and the authorized vehicles with their actual position, their orientation/heading, and other relevant information on a Label attached to the symbol. The ATCOs can give clearances to the pilots selecting the appropriate flight.

In addition to the typical ATC operations, there are shown also the information distributed by the ASPRID system: the drone positions, their classification, and the detected hazards on an alarm list.

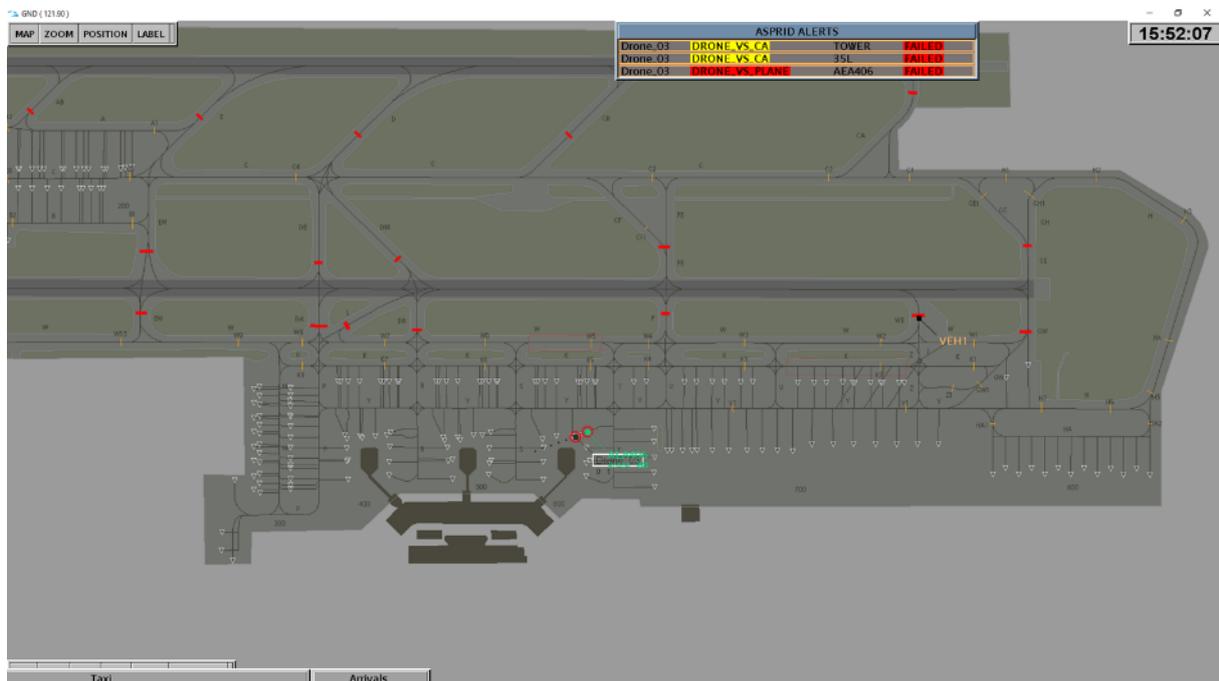


Figure 20. eArts ASPRID integration



4.4 ASPRID Demo APP

The gaming exercises done with the above tools and the experts, will be recorded and could be replayed with the ASPRID Demo app, available at <https://aspriddemo.firebaseio.com/>

This web application does not require a registration, and the user could select the available recorded exercises.

The App is similar to the ASPRID Viewer, with the adding of a Time console to start the replay, and go directly to a particular point of the simulation.

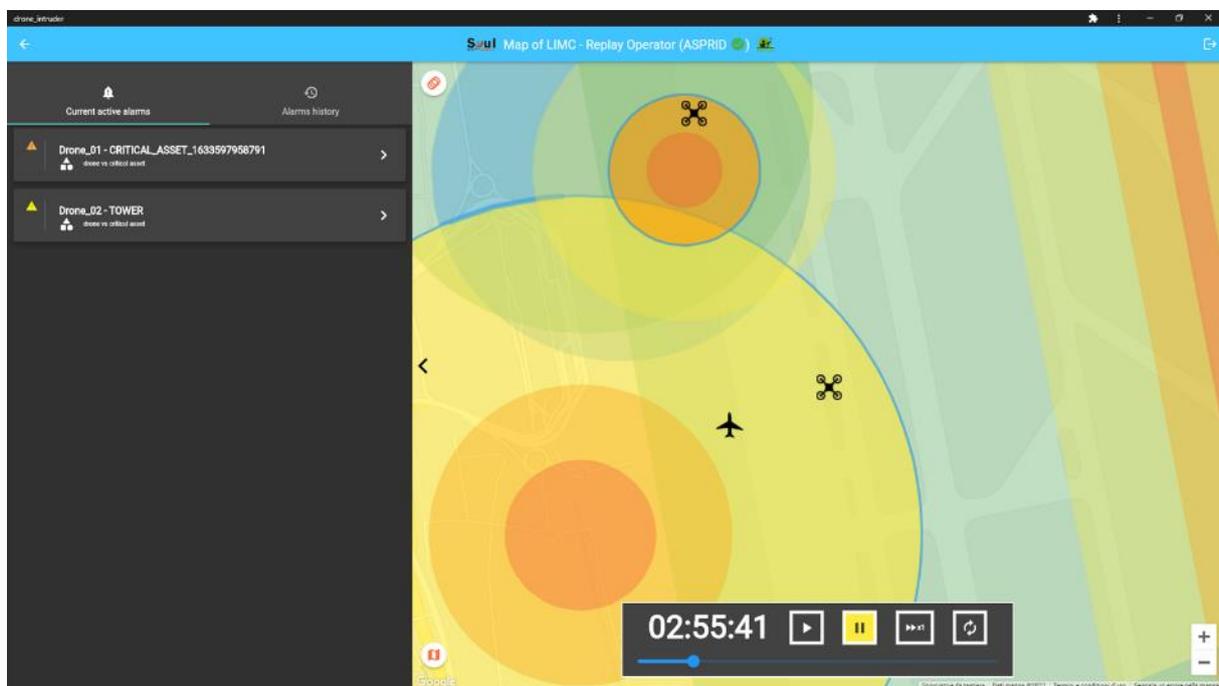


Figure 21. ASPRID Demo App



5 References

- [1] ASPRID.WP2.D2.2.CO.V1.0.FINAL.Operational Concept and Architecture Definition
- [2] ASPRID.WP2.D2.3.CO.V1.0.FINAL.Detection, Identification and Tracking
- [3] ASPRID.WP2.D2.4.CO.V1.0.FINAL.Alert, Procedures and Decision system
- [4] ASPRID.WP2.D2.5.CO.V1.0.FINAL.Response and Neutralization
- [5] ASPRID.WP6.D6.2.CO.V1.0.FINAL.Data Management Plan
- [6] ASPRID Web App: <https://aspridapp.firebaseio.com/>
- [7] ASPRID Demo Software: <https://aspriddemo.web.app/>
- [8] ASPRID Grant Agreement 892036 – Annex 1.