ICARUS external I/F test & validation plan

DeliverableID	**D5.4**
ProjectAcronym	**ICARUS**
Grant:	**894593**
Call:	**H2020-SESAR-2019-2**
Topic:	**Topic 31**
Consortium coordinator:	**E-GEOS SPA**
Edition date:	**03 November 2021**
Edition:	**00.01.03**
Template Edition:	**02.00.00**

Founding Members

SESAR JOINT UNDERTAKING

Authoring & Approval

Authors of the document

Name/Beneficiary	Position/Title	Date
Piotr Dybiec/DroneRadar	Project Manager	10/09/2021
Paweł Korzec/DroneRadar	CEO	29.10.2021
Łukasz Górny-Zając/DroneRadar	Chief Architect	29.10.2021
Adam Staszak/DroneRadar	Senior Developer	29.10.2021
Jan Mróz-Jaworski/DroneRadar	System Architect	29.10.2021
Sławomir Kubiak/DroneRadar	Senior Analyst	29.10.2021

Reviewers internal to the project

Name/Beneficiary	Position/Title	Date
Peter Hullah/ Eurocontrol	Steering Committee member	2.11.2021
Alberto Mennella / TopView	Technical Coordinator	3.11.2021
Chiara Sammarco/ EGEOS	Project team member	3.11.2021

Approved for submission to the SJU By – Representatives of beneficiaries involved in the project

Name/Beneficiary	Position/Title	Date
Cristina Terpessi/EGEOS	Project Coordinator	3.11.2021
Corrado Orsini/TPZ	Technical Coordinator	3.11.2021
Alberto Mennella / TopView	Technical Coordinator	3.11.2021
Filippo Tomasello/ EUSC-IT	Consortium Board member	3.11.2021

Rejected By - Representatives of beneficiaries involved in the project

Name/Beneficiary	Position/Title	Date

Document History

Edition	Date	Status	Author	Justification
00.00.01	15.10.2021	Draft	Piotr Dybiec	First content in sections 2, 3 and 4
00.00.02	29.10.2021	Draft	Piotr Dybiec, Adam Staszak, Paweł Korzec,	Final Draft

			Jan Mróz-Jaworski, Łukasz Górny Zając	
00.00.03	03.11.2021	Draft	Document Review	Final version for internal approval
00.01.03	03.11.2021	Draft		Final version

Copyright Statement

ICARUS

INTEGRATED **C**OMMON **A**LTITUDE **R**EFERENCE SYSTEM FOR **U**-**S**PACE

This document is part of a project that has received funding from the SESAR Joint Undertaking under grant agreement No. 894593 of the European Union's Horizon 2020 research and innovation programme.

Abstract

This document contains the inventory of all external interfaces that need to be integrated. It describes the approach to testing and defines the Test Case Descriptor form, which is used to describe test case specifications and document test results. It provides the plan for the integration testing of external systems with the ICARUS CARS platform. Finally, it contains a report on the results of the first tests that the consortium performed.

Table of Contents

List of Tables

List of Figures

1 Introduction

1.1 Purpose of the Document

This document provides guidelines for integration testing of the ICARUS CARS system with external platforms using specified interfaces.

The document starts (Chapter 2) by listing of all identified external interfaces and describing their use. The remaining part describes the approach towards testing the integration of the ICARUS CARS system with external platforms in terms of methodologies and tools to be used (Chapter 3) and the general plan of integration tests to be performed (Chapter 4). The document ends with an initial testing integration report (Chapter 5).

The main purpose is to describe testing activities that would ensure that the ICARUS CARS platform is properly integrated with all identified external systems and thus:

- is supplied with all required information to perform its actions (incoming information flows), and

- can provide the required services to its users (outgoing information flows)

It is not intended to provide a full list of detailed functional tests of the ICARUS CARS platform itself in this document. This topic is in the scope of validation methodology of the ICARUS system, which is provided in the D6.1 deliverable with a defined scope of usability tests, which are also covered in documents D5.1, D5.2 and D5.3.

Reports and results of all above-mentioned tests will be published in two documents: D5.4 (this one, to cover initial integration tests) and D6.3 (to cover remaining integration tests and all functional/usability tests).

This document is correlated with documents D4.1 and D4.2 as well as D5.1, D5.2 and D5.3. D4.1 covers the design architecture of the system, while D4.2 describes the overall prototype implementation of the new ICARUS CARS platform. The D5.1, D5.2 and D5.3 documents provide more details about the architecture, specific use cases, and integration with the main external systems: DroneRadar's USSP platform, a GI Service Provider and a Weather Service Provider (D5.1), TopView's Cockpit Simulator (D5.2), GNSS supporting systems (ASI GNSS reference stations and EDAS) and D-Flight (D5.3).

In this document, only information required to perform interface-level integration, such as interface specifications and basic structure of information exchanged, is described. If needed, for further information about specific functionalities and details of interfaces usage, the reader is referred to D4.2, D5.1, D5.2 or D5.3.

1.2 Acronyms

Acronym	Meaning
API	Application Programming Interface
AGL	Above Ground Level

ASL (proposition)	Above Surface Level
ARAIM	Advanced RAIM
ATC	Air Traffic Control
ATM	Air Traffic Management
ATZ	Aerodrome Traffic Zone
BKG	Bundesamt für Kartographie und Geodäsie
BNC	BKG NTRIP Client
BVLOS	Beyond Visual Line of Sight
CARS	Common Altitude Reference System
CIS	Common Information Service
CORBA	Common Object Request Broker Architecture
CTR	Control zone
DAA	Detect And Avoid
DEM	Digital Elevation Model
DSM	Digital Surface Model
DTM	Digital Terrain Model
DOP	Dilution Of Precision
DSM	Digital Surface Model
DTM	Digital Terrain Model
EASA	European Union Aviation Safety Agency
EDAS	EGNOS Data Access Service
EGNOS	European Geostationary Navigation Overlay Service
EGNSS	European Global Navigation Satellite System
FLTA	Forward Looking Terrain Avoidance
GA	General Aviation

GAMZ	Geometric Altitude Mandatory Zone
GI	Geo Information
GNSS	Global Navigation Satellite System
GO	Ground Obstacle
GPS	Global Positioning System
HALB	Horizontal Alert Buffer
HPL	Horizontal Protection Level
ICAO	International Civil Aviation Organisation
ISA	International Standard Atmosphere
ISM	Integrity Support Message
ISO	International Organisation for Standardisation
MCMF	Multi-Constellation Multi-Frequency
NTRIP	Networked Transport of RTCM via Internet Protocol
PL	Protection Level
QFE	Query Field Elevation
QNH	Query: Nautical Height
RAIM	Receiver Autonomous Integrity Monitoring
RIMS	Ranging Integrity Monitoring Stations
RGIS	Real Time GIS
RNP	Required Navigation Performance
RTCM	Radio Technical Commission for Maritime Services
SBAS	Satellite-Based Augmentation System
SORA	Specific Operations Risk Assessment
TCU	Telespazio's Computing Unit
TSE	Total System Error
UA	Unmanned Aircraft

UAS	Unmanned Aircraft System
USSP	U-Space Service Providers (alias UTM service provider)
UTM	Unmanned aircraft system Traffic Management (alias U-space)
VALB	Vertical Alert Buffer
VCS	Vertical Conversion System
VALS	Vertical Alert Service
VLL	Very-Low-Level
VLOS	Visual Line Of Sight
VPL	Vertical Protection Level
VPN	Virtual Private Network
WALB	Width Alert Buffer

Table 1-1: Acronyms list

2 Inventory of ICARUS external interfaces

The overall ICARUS architecture is presented in Figure 2-1. It shows the interconnections between the ICARUS CARS platform and external systems. Each arrow represents a specific communication interface. Each interface implements a specific API used by different modules to provide specific functionality. The direction of an arrow indicates whether the interface is incoming or outgoing. As can be seen, each of the systems/platforms can be composed of different modules providing specific functionality.

There are different identified types of external interface used to interconnect the ICARUS CARS system with external platforms. Some of them (VCS.USSP.01) are re-used by different platforms (DroneRadar USSP and D-FLIGHT USSP) and some might have multiple instances (like REF.TCU.01 used to connect multiple GNSS reference stations or UAS.TCU.01 to connect a number of UASes).

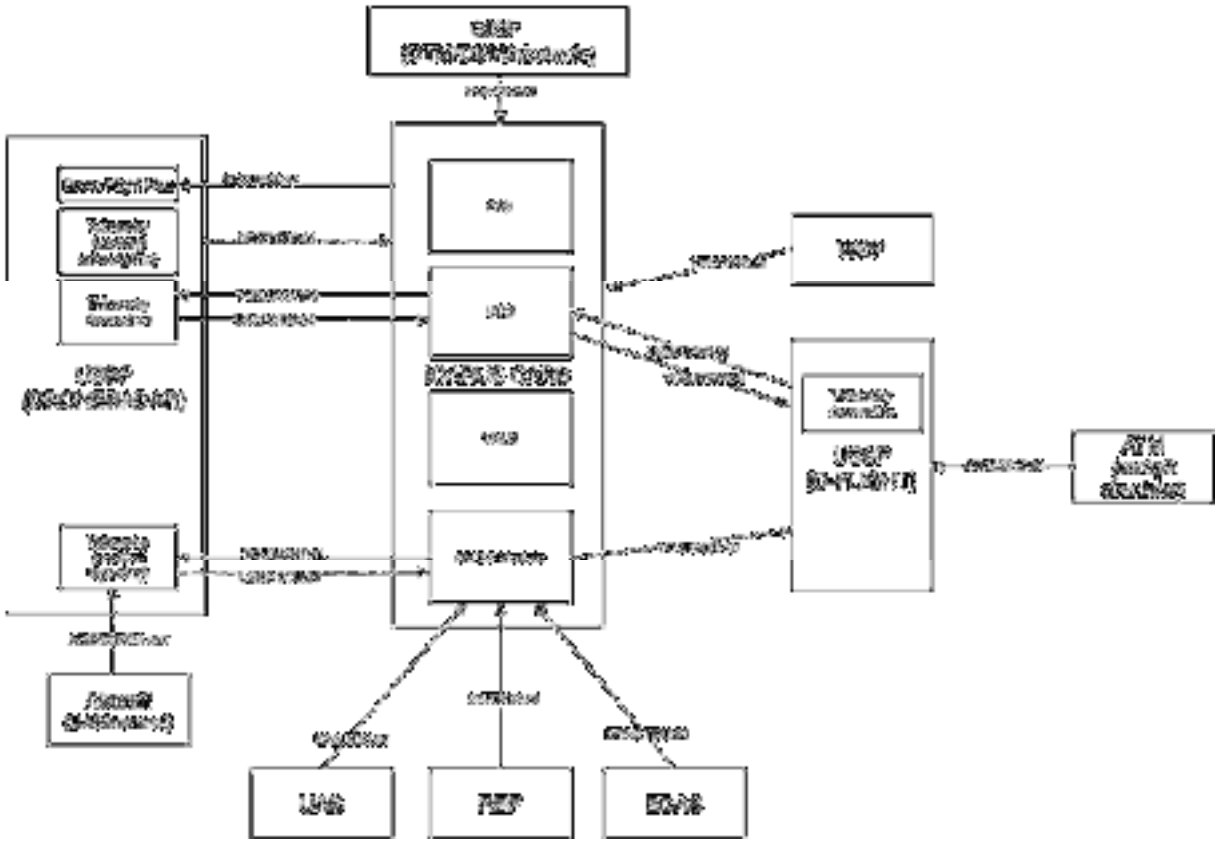

Figure 2-1: Diagram of connections between ICARUS CARS and external systems

The complete list of these external interfaces is provided in Table 2-1 with the following information:

- Names of interconnected modules (external system and components of the ICARUS CARS using this interface);
- References, where the detailed interface specifications and descriptions are provided;
- Main purpose of the interface.

Interface ID	External system	ICARUS CARS component	Reference	Main purpose
GIS.USSP.01	USSP (Drone Flight Plan)	GIS	D5.1, D4.2	To allow a USSP to verify and validate declared flight height against terrain elevation.
USSP.GIS.01	USSP (Telemetry)	GIS	D5.1	This interface is used to provide telemetry information available from the UTM telemetry subscription service.
VCS.USSP.01 / USSP.VCS.01	USSP (Telemetry)	VCS	D5.1, D4.2	To provide access to the VCS service of the ICARUS CARS.
TCU.USSP.01 / USSP.TCU.01	USSP (Telemetry)	GNSS	D5.3, D4.2	To supply the USSP with the external telemetry information, enriched with integrity assurance and geographic validation. To adjust telemetry from the USSP with integrity and geographic validation.
UAS.TCU.01	UAS	GNSS	D5.3, D4.2	To provide raw GNSS data from a UAS to the GNSS module of the ICARUS CARS.
REF.TCU.01	GNSS Reference Station	GNSS	D5.3, D4.2	To provide reference GNSS data from a GNSS Station to the GNSS modules of the ICARUS CARS.
EDAS.TCU.01	EDAS	GNSS	D5.3, D4.2	To provide EDAS data to the GNSS modules of the ICARUS CARS.
WSP.VCS.01	WSP	VCS	D5.1	To supply ICARUS CARS with valid QNH information and readings from weather sensors networks (local pressure and temperature).
GISP.GIS.01	GISP	GIS	D5.1	To supply the ICARUS CARS components with reliable and valid GIS data (DSM and DTM models) as well as additional information (like elevation and undulation parameters).

Table 2-1: List of ICARUS CARS external interfaces

In addition to external interfaces mentioned in the table, which are directly interconnected to the ICARUS CARS system, there are interfaces that interconnect the external systems themselves, without the direct connection to the ICARUS CARS platform. ATM.USSP.01 is an example of such an external

interface. This interface is required to connect a cockpit simulator to the D-FLIGHT USSP system and it is used to supply telemetry data, which in turn is then passed to/processed by the ICARUS CARS (e.g., VCS service), to the D-FLIGHT USSP.

3 External interface testing approach

Tests are the most important activities during a system's validation process. The main validation process is covered in the D6.1 document. This document provides the approach to testing external interface integration.

During the integration phase we focus on tests that will provide the proof of interoperability between the ICARUS CARS system and external platforms.

Tests can be classified in many dimensions. One of the most basic distinctions is dividing test activities into two generic categories:

- Functional tests
- Non-functional tests

In principle, functional tests cover tests that are aimed at validation, if defined business requirements are met. Sometimes, they are called "black-box tests," because they do not focus on the product's internals, but on the functional aspects of the interface layer. They provide the answer to the question: "What does the product do?". On the other hand, non-functional tests answer the question: "How does the product work?".

Examples of typical functional tests include unit tests, acceptance tests, smoke tests, integration tests, and regression tests. Tests such as performance tests, scalability tests, volume tests, load tests, stress tests, etc. belong to the non-functional category.

Mainly functional tests are proposed in this document. Moreover, this document focuses only on integration aspect and thus contains the integration-related testing information narrowed to testing external interfaces (i.e. with systems external to the ICARUS CARS). All the other testing activities, specifically the non-functional tests, such as performance load tests, are out of the scope of the project. The focus of these integration tests is to verify the functionality of defined APIs, confirming that the communication with the ICARUS CARS system is working according to expectations, and that the necessary information flow is provided.

3.1 Integration test documentation

Two main types of document will be prepared for documenting tests:

- Tests specifications
- Tests reports

Tests specifications contain set of tests to be performed.

Test reports contain reports from the test executions: a list of tests performed and their results, with options remarks and suggestions for improvements/corrections.

As part of the integration testing process, test specifications must be defined. To support this action, a Test Case Descriptor template is introduced (Table 3-1).

Test case name here			
Test case Id	*Add Id number.*		
Test designed by	*Add name of the person that defined the test(s).*		
Test executed by	*Name of the person(s) executing the test(s).*		
Testing date	*When were the tests run?*		
Test purpose	*Provide a description of the purpose of the tests.*		
Test environment	*Describe the setup used for the tests.*		
References	*Insert reference on underlying standards and/or project specification and reference points if any.*		
Validation target/metric	*Define the quantitative or qualitative result(s) expected from the execution of the test case.*		
Tested components	*List the HW and SW components under testing with version info.*		
Tested interfaces	*List the interface(s) and or API(s) that are tested.*		
Other interfaces	*List the interfaces that must be established to execute the test case.*		
Pre-test conditions	*List the pre-requisites that are necessary before the test execution, for example other test-cases that must be completed first. Note that this refers to the test case execution only and not other installation prerequisites that should be part of the deployment descriptors of each component/package.*		
Test tools	*Identify the test tool(s) used for the execution of the test.*		
	# Step	**Description**	**Result**

Integration test steps Definition and results	*1*	*…*	*Result*
	2	*…*	*Result*
	3		*Result*
	n		
Test verdict	PASS/ FAIL/ PENDING	*Describe the status, findings and extensive feedback provided to the next integration cycles. The status of the execution can be:* • *PASS: Results are as expected, and no further actions are necessary* • *FAIL: The targets set for the execution cannot be met due to functional problems that need new/corrected implementations expected in later integration cycles* • *PENDING: The tests cannot be executed due to missing environmental characteristics or other external constraints*	
Change requests		*List here all change requests to the module if any.*	
Defects		*List here all the possible errors reported to the developer. Remember to include id.*	

Table 3-1: Integration test case descriptor template

Each such Test Case Descriptor will cover single test case. It will be named with an appropriate, unique test id and will describe in detail the test purpose, prerequisites, test scenario with execution instructions, expected outcomes, success criteria, results, evaluation, and conclusions.

The tests should be designed such that they should be independent of each other – the result of one test should not rely on the result of another test. Also test conditions and test scenarios should ensure that all tests can be repeated, and their results will be consistent and repeatable (to avoid randomicity in test results).

The test scenarios should cover all relevant use cases, should test the quality of information exchanged (its type, format, allowed values, etc) to ensure that the data exchange is reliable and meaningful as well as being resistant to errors.

After test execution, the corresponding test report will be generated. It will contain the list of tests performed with their results.

3.2 Integration testing guidelines

The main purpose of integration testing is to verify the interoperability between the ICARUS CARS system components and external platforms.

When public Application Programming Interfaces (APIs) are tested, this can be done with limited knowledge of the system's internal architecture (grey-box testing). The goal of the project is to design

and build all the project's components with well documented APIs. For the ICARUS CARS integration tests, we assume the availability of well-defined and described APIs.

The output of integration testing, apart from the Pass/Fail result, is expected to provide quantitative and qualitative feedback to the subsequent integration cycles, if necessary.

The integration tests focus on communication and interoperability aspects of the system. During integration tests, external systems/modules/application are tested extensively.

As the interfaces to be tested during this phase are mostly deployed using RESTful APIs, the main tools supporting test execution will be Postman (https://www.postman.com) and cURL (https://curl.se).

4 Integration test plan

The testing activity requires that an appropriate sequence of tasks is performed to properly scope, prepare, execute, and conclude integration tests. The proposed flow of actions is presented in Figure 4-1.

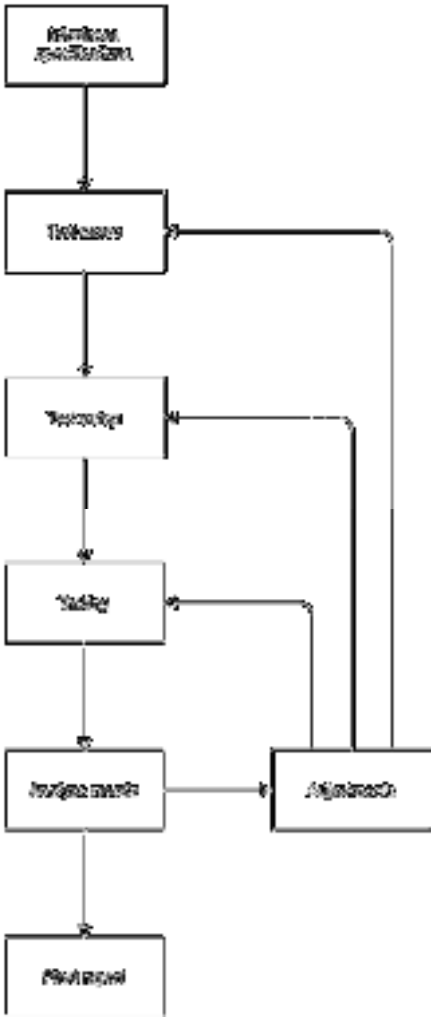

Figure 4-1: Test planning process

To be able to perform any integration tests, the complete inventory of all interfaces to be used and tested first needs to be completed. In addition to the exhaustive list of interfaces, their detailed descriptions and specifications also need to be collected. This is the aim of the “Interface specifications” tasks.

As mentioned above, each of the interfaces might be used differently due to implementation specificity or due to the way the interface is used for a given functionality provided by the platforms integrated. This must be thoroughly captured to recognise the scope of use of each interface implementation. The appropriate set of integration requirements and respective testing procedures can only be defined based on this information. This is the purpose of defining the test-case

specifications phase marked on the above diagram as the “Test cases” step. To better structure this step, the Test Case Descriptor form was proposed in Chapter 3.

Some of the content of this Test Case Descriptor document also provides information required by the next step: “Test setup”. This covers all necessary activities to ensure that all the prerequisites are met and satisfied (systems are up and running, appropriately configured, test data is available, etc). This is a part of the execution test planning.

As soon as the “Test setup” step is accomplished, the test may be executed. This is represented on the diagram by the “Testing” block.

The test outcomes can then be compared with the expected results. If any discrepancies, problems or needs for improvement are found, the testing procedures and documentation might be accordingly updated. This is represented by the “Adjustments” feedback task.

After the whole integration testing cycle is finished, the final conclusions will be written in the report, as shown by the “Final report” step.

The entire process of integration testing following the process described above was scheduled as shown in Figure 4-2.

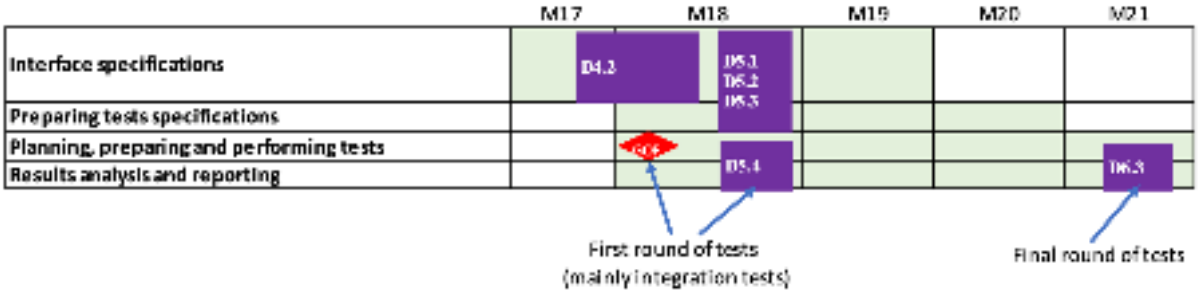

Figure 4-2: External interfaces tests schedule

As the design and build process of the ICARUS CARS system and the customisation of external systems is currently an ongoing process parallel to the external interface testing, it was decided that the testing process will be covered in two deliverables: D5.4 (this document) and D6.3. The current scope corresponds to the tests we have been able to perform and conclude so far, and it is described in Chapter 5. All the remaining tests and retests of tests to be performed will be reported in similar way but in the later deliverable, D6.3.

The mentioned in the time plan, deliverables D4.2, D5.1, D5.2 and D5.3 are crucial from the interface specification perspective. They contain details such as ICDs as a part of the design descriptions of different components of the solution.

5 Integration testing report

As the development of the platform is ongoing and the main testing and validation activities are still to come, this document only presents early feedback from the initial test rounds. The first results and conclusions are collected and summarised in the Table 5-1. The test specifications and test descriptions are provided in D5.1, D5.2 or D5.3 respectively. Only tests that were performed are reported in the table below. All remaining tests are planned to be executed during validation activities (XI.2021-I.2022) and will be reported in the D6.3 document.

Test ID	Test description	Test Result	Comments
USSP.01	*The test aims to:* *1) check the communication between the USSP and the VCS microservice* *2) verify the ability to send the request to the VCS endpoint and receive the response*	PASSED (Partially)	• *The connection between the USSP and the VCS was successfully established.* • *The request to the VCS was sent, although the implementation of the VCS that was tested had very limited performance – it should be improved.* • *The response was received correctly, but the payload contained numerical errors. The algorithm/application needs to be corrected.*
WSP.01	*The test aims to:* *1) check the communication with the Weather Service Provider platform;* *2) check the communication with the weather station sensor.*	PASSED	*Integration tests with the WSP for both QNH queries and weather sensor queries were successful.*
GISP.01	*The test aims to check the communication with the GI Service Provider platform.*	PASSED	*Integration tests were successful – elevation information was provided on request.*

Table 5-1: Summary of integration tests results

6 Applicable and reference documents

[1] ICARUS D3.1, “ICARUS Concept Definition: State-Of-The-Art, Requirements, Gap Analysis”.

[2] ICARUS D4.1, “Design and architecture of the ICARUS system & services”.

[3] ICARUS D4.2, “ICARUS Prototype”

[4] ICARUS D5.1, “UTM Platform Architecture including ICD and Integration Test Report”

[5] ICARUS D5.2, “Cockpit simulator Architecture including ICD and integration test report”

[6] ICARUS D5.3, “D-Flight GNSS Augmentation ICD and Integration Test Report”

[7] ICARUS D6.1, “Validation Scenario Design”

[8] ICARUS D6.3, “Simulation trials Data Analysis & Results”


e-geos


TELESPAZIO


TOP VIEW


dronemaster


EUROCONTROL


EUROUSC
ESPAÑA


EUROUSC


POLITECNICO


SAPIENZA