D-Flight GNSS Augmentation ICD and Integration Test Report

DeliverableID	D.5.3
ProjectAcronym	ICARUS
Grant:	894593
Call:	H2020-SESAR-2019-2
Topic:	Topic 31
Consortium coordinator:	E-GEOS SPA
Edition date:	October 2021
Edition:	00.01.05
Template Edition:	02.00.00

Founding Members

Authoring & Approval

Authors of the document

Name/Beneficiary	Position/Title	Date
Orsini Corrado	Technical Manager	01/08/2021

Reviewers internal to the project

Name/Beneficiary	Position/Title	Date
Andrea D'Agostino	GNSS responsible	22/10/2021
All SC members		28/10/2021

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

Name/Beneficiary	Position/Title	Date
Cristina Terpessi	Project Coordinator	28/10/2021
Alberto Mennella	Technical Coordinator	28/10/2021
Corrado Orsini	Technical Coordinator	28/10/2021

Rejected By - Representatives of beneficiaries involved in the project

Name/Beneficiary	Position/Title	Date

Document History

Edition	Date	Status	Author	Justification
01	01/09/2021	TOC review	Corrado Orsini	Review of the structure of the document
02	1/10/2021	Insertion of: Abstract, Purpose of the document, Inventory of GNSS interfaces	Corrado Orsini	First release of the document with Abstract, Purpose of the document, Inventory of interfaces filled
03	19/10/2021	Content update: Inventory of GNSS interfaces		Third release of the document with the Inventory of GNSS Interface update (description of Real-time monitoring services, Real-

				time integrity services, Post-processing monitoring services)
04	26/10/2021	Content update: Inventory of GNSS interfaces		Fourth release of the document with a detailed description of USSP Connection interfaces
05	28/10/2021	Revision	Peter Hullah	Latest revision

Copyright Statement

ICARUS

INTEGRATED COMMON ALTITUDE REFERENCE SYSTEM FOR U-SPACE

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 specifies all the interfaces regarding the GNSS microservice. It describes the input and output interfaces that allow the necessary data to be retrieved to be upgraded by the outcomes of the Telespazio computing unit calculations.

Moreover, the document describes the data flows regarding the GNSS microservice through a series of use cases representative of its operations.

Finally, it contains the test to be performed in order to test and verify the communication between the GNSS microservice and the external sources of data (EDAS and Ground Reference Stations) and between the GNSS microservice and the other ICARUS microservices that need its computation.

Table of Contents

List of Tables

List of Figures

1 Introduction

1.1 Purpose of the Document

The purpose of the document is to represent, synthesise and describe all the interfaces concerning the GNSS microservice composing ICARUS.

The GNSS microservice has the aim of determining, by accessing the data from the GNSS receiver on board the drone or by accessing the drone telemetry provided by the USSPs, the goodness of the navigation performance, allowing an estimate of the signal performance as well as an estimate of the integrity parameters.

To perform this calculation, the system needs to have access to further input data such as data from the EDAS service and data from the Ground GNSS Reference Stations.

All these input sources require the definition of specific interfaces which are summarised in the following chapters.

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
GA	General Aviation
GAMZ	Geometric Altitude Mandatory Zone
GI	Geo Information
GNSS	Global Navigation Satellite System
GO	Ground Obstacle
GPS	Global Positioning System
HPL	Horizontal Protection Level
ICAO	International Civil Aviation Organisation
IoT	Internet of ThingS
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
TCP	Transmission Control Protocol
UA	Unmanned Aircraft
UAS	Unmanned Aircraft System
UDP	User Datagram Protocol
USSP	U-Space Service Providers (alias UTM service provider)
UTM	Unmanned aircraft system Traffic Management (alias U-space)
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

Table 1-1: Acronyms list

2 Inventory of GNSS interfaces

The Navigation System is a fundamental enabling factor for the proper functioning of USSPs and in particular of D-Flight, ensuring the ability to provide the central system with the precise position of all the UAS flying in the VLL airspace.

The services offered by the platform are grouped into three macro-areas:

• Real-time monitoring services

• Real-time integrity services

2.1 Real-time monitoring service

The Real-time monitoring service makes it possible to assess the quality of GNSS navigation systems in real time. This monitoring is done for each of the available GNSS Ground Reference Stations. In the event of an anomaly in the parameters of a station (high estimated errors, etc.), this implies that all drones present in the surrounding area will have problems in localisation.

The real-time monitoring service uses freely accessible dual frequency GNSS reference stations. Therefore, in the most basic form, it returns the GNSS performance in terms of accuracy and integrity on these stations to the USSP.

For the USSP this means having the operating status of the navigation systems in real time with an update every 15 minutes.

For the drone operator, real-time monitoring provides a tool to improve mission planning and management, displaying the SBAS real-time performances on the reference stations closest to the operator's flight space.

2.2 Real-time integrity service

The Real-time Integrity Monitoring service is able to evaluate the status of the GNSS on board the UAS. In this case it is possible to detect not only GNSS problems, but also any components of local errors or disturbances, in real time. In this case, GNSS receivers should be capable of acquiring and transmitting raw GNSS measurements, before the receiver performs the position calculation. In this way it is possible to detect local events that would otherwise remain undetectable.

The GNSS microservice is set up to qualify the data acquired by the on-board GNSS receiver through EDAS / EGNOS and through ARAIM algorithms making the augmentation available.

Thus, the drone operator is able to evaluate the goodness of the GNSS position during the operational phase of the flight using the protection levels (HPL, VPL) that give a guarantee and a metric on the accuracy of the position data (BVLOS Enabler).

As already stated, the GNSS subsystem provides a centralised means of reliably computing the position of registered UASs, in real-time, and is a key enabler for all the ICARUS services (details regarding architecture, requirements and system composition can also be found in the other project deliverables ([1], [2], [3]).

Through appropriate processing of raw GNSS observables from drones, and support messages and data from external entities, it will provide the PVT solution for each monitored UAS, together with its integrity parameters and a validation mechanism based on the use of data coming from a network of trusted reference stations.

The GNSS service is important for the determination of GNSS-based altimetry as a common reference datum for establish the vertical distance of everything is flying with respect to the ground.

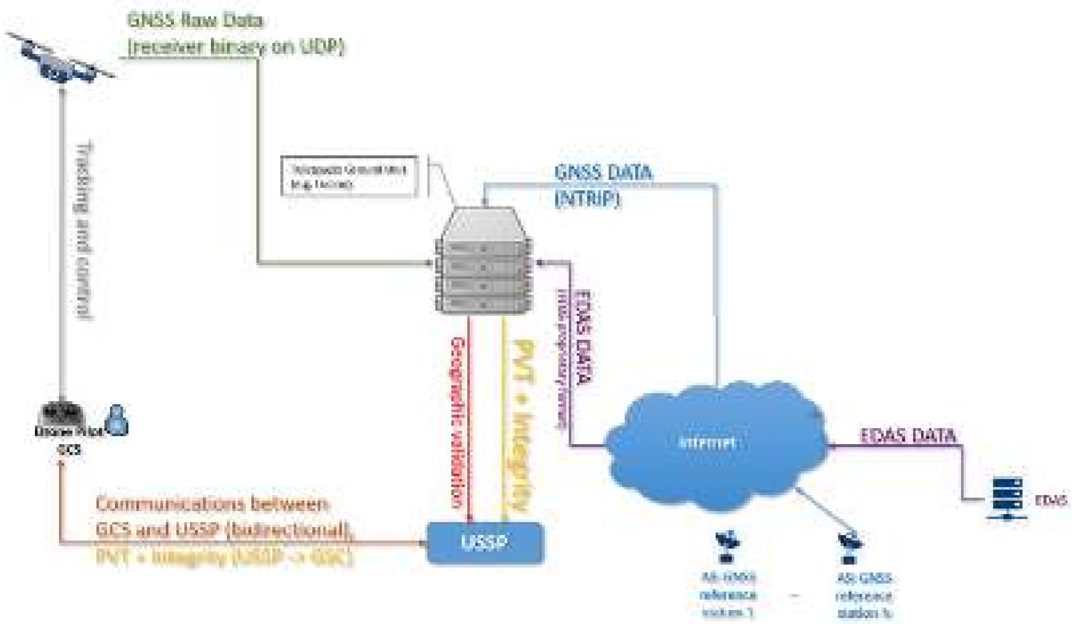

Figure 2-1: High-level architecture of GNSS microservice

GNSS data necessary for the calculation are provided to ICARUS in two ways:

1) via UDP through direct connection with the GNSS receiver on board the drone; in this case it is possible the access to the Raw Data

2) via MQTT through connection with the USSP, in the case of no direct access to raw data, to retrieve only telemetry.

Once the calculation has been performed, the drone position, altitude/height, the signal integrity and monitoring results will be shared with the USSP and with the final user.

The GNSS microservice requires the following external elements:

• real GPS and EGNOS signals acquired via the EDAS terrestrial network;

• connection to the network of IGS stations for receiving raw GNSS data acquired by local receivers;

• connection to additional GNSS stations for monitoring sensitive areas for receiving raw GNSS data acquired by local receivers;

• interfaces with USSPs (D-Flight, Pansa UTM) for the provision of GNSS Performance Monitoring services;

• interfaces with UAS;

• hardware for subsystem deployment and long-term storage of data suitably protected and accessible by accredited users.

A more detailed representation of the units of the subsystem and the data flows can be found in the following context diagrams. The functionalities covered are shown in Figure 2-2 and in Figure 2-3.

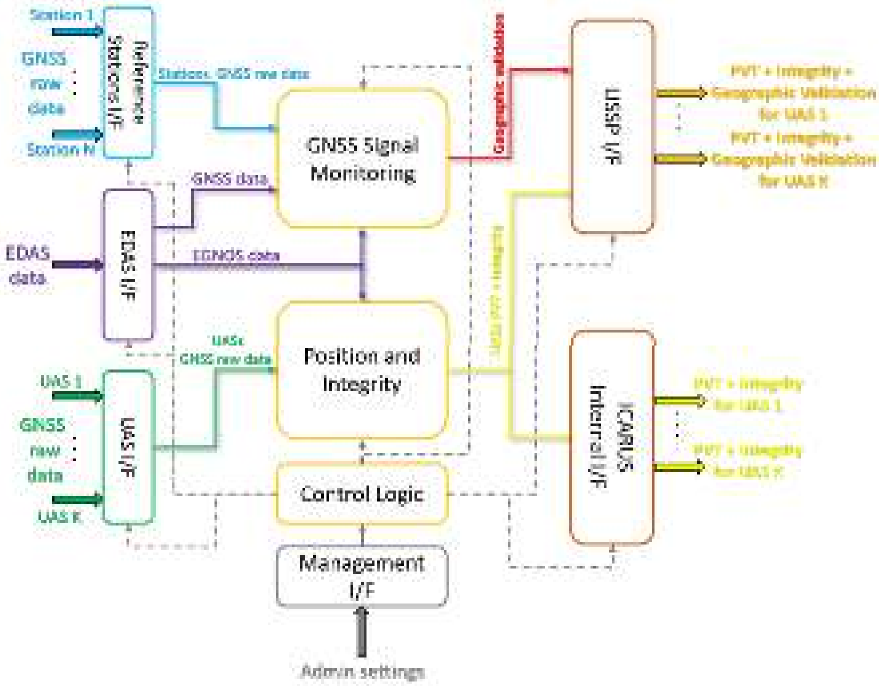

Figure 2-2: context diagram of the GNSS subsystem when direct access to GNSS raw data is foreseen

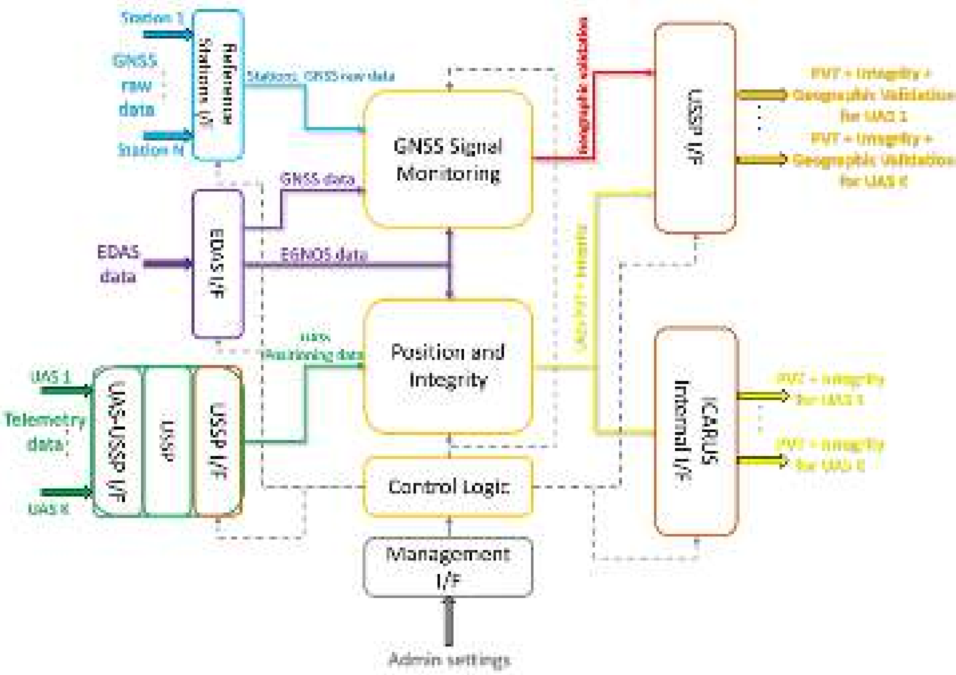

Figure 2-3: context diagram of the GNSS subsystem when no direct access to UAS GNSS raw data is foreseen

The list of the interfaces can be found in Table 2-1: interfaces for incoming data flows (interfaces for incoming data) and Table 2-2 (interfaces for outgoing data).

Interface ID	Source	Destination	Internal / External[1]	Protocol	Data
EDAS.TCU.01	EDAS	TCU	External	EDAS proprietary	SBAS EGNOS messages, RIMS GNSS raw data
REF.TCU.01	IGS EUREF ASI trusted network	TCU	External	NTRIP	GNSS raw data
UAS.TCU.01	UAS	TCU	External	Receiver proprietary format	GNSS raw data
USSP.TCU.01	USSP	TCU	External	JSON on MQTT	Telemetry data
ADM.MAN.01	Administrator (authorised user)	Management unit	Internal	ssh access	Configuration files & settings

Table 2-1: interfaces for incoming data flows

Interface ID	Source	Destination	Internal / External[2]	Protocol	Data
TCU.USSP.01	TCU	USSP	External	JSON on MQTT	Positioning, Integrity, Geographic Validation
TCU.GIS.01	TCU	Geo Information	Internal	JSON on MQTT	Positioning, Integrity

[1] The distinction between "internal" and "external" indicates whether the source component belongs to ICARUS or not

[2] The distinction between "internal" and "external" indicates whether the destination component belongs to ICARUS or not

		ICARUS subsystem			
TCU.VCS.01	TCU	Conversion Service ICARUS subsystem	Internal	JSON on MQTT	Positioning, Integrity
TCU.VALS.01	TCU	Vertical Alert ICARUS subsystem	Internal	JSON on MQTT	Positioning, Integrity

Table 2-2: interfaces for outgoing data flows

Four use cases representative of the GNSS microservice operations and data flows are given below.

The first sequence diagram below demonstrates the ability of the GNSS subsystem to correctly treat EGNOS messages and GNSS raw data to evaluate the GNSS signal performance through:

1. Reception and decoding of raw GNSS data from the reference station network;
2. At the same time, reception and decoding of the EDAS data stream, retrieving the EGNOS SBAS messages and the RIMS GNSS raw data;
3. Processing of the retrieved data to identify and discard possible faulty satellites and to compute the integrity parameters (i.e. Horizontal/Vertical Protection Levels) over a regional grid;
4. Estimation of the integrity parameters corresponding to a given coordinate pair (according to a distance criterion from the nearest grid point computed)

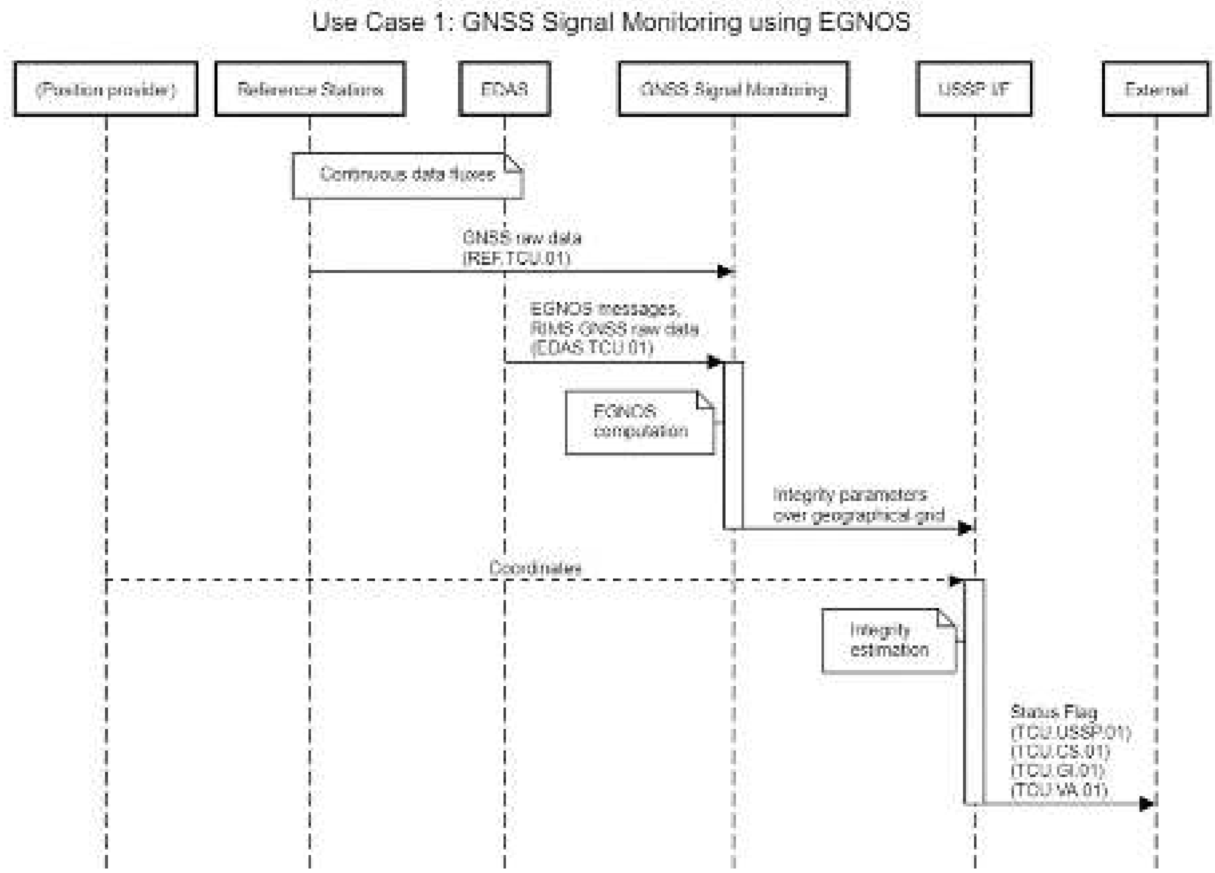

Figure 2-4: Use case 1 sequence diagram

The second sequence diagram demonstrates the ability of the GNSS subsystem to correctly treat stored ARAIM-related data (ISM message) and GNSS raw data to evaluate the GNSS signal performance, through the following steps:

1. Reception and decoding of raw GNSS data from the reference station network;
2. Processing of the retrieved data to identify and discard possible faulty satellites and to compute the integrity parameters (i.e. Horizontal/Vertical Protection Levels) over a regional grid;
3. Estimation of the integrity parameters corresponding to a given coordinate pair (according to a distance criterion from the nearest grid point computed)

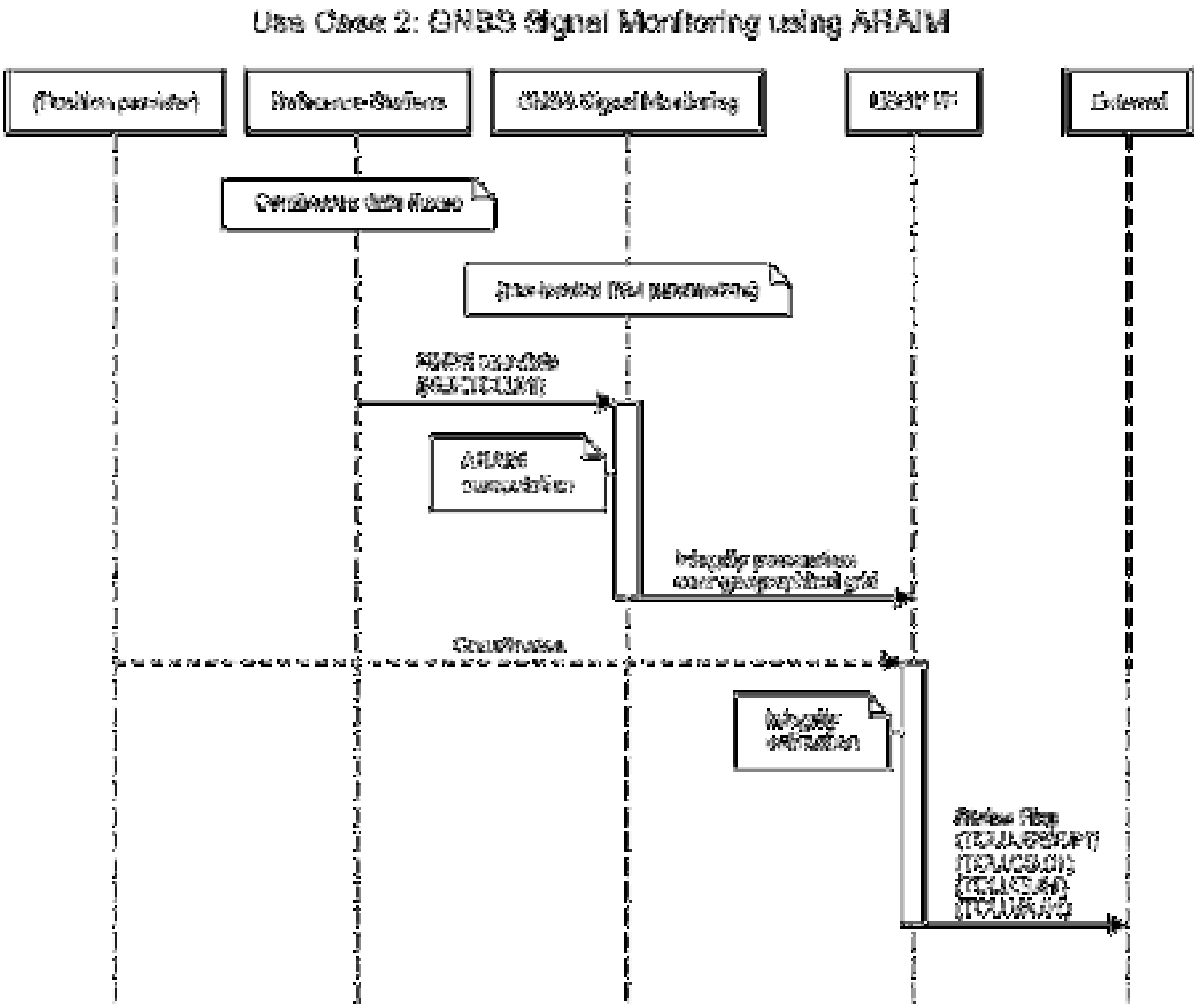

Figure 2-5: Use case 2 sequence diagram

The third sequence diagram demonstrates the ability of the GNSS subsystem to correctly treat EGNOS messages and GNSS raw data from a UAS to compute its position and its integrity parameters through the following steps (in addition and in parallel to what is described in the first sequence diagram):

1. Reception and decoding of the EDAS data stream, retrieving the EGNOS SBAS messages;
2. Reception and decoding of raw GNSS data from the tracked UAS;
3. Processing of the retrieved data to compute the UAS PVT and related integrity parameters (i.e. Horizontal/Vertical Protection Levels).

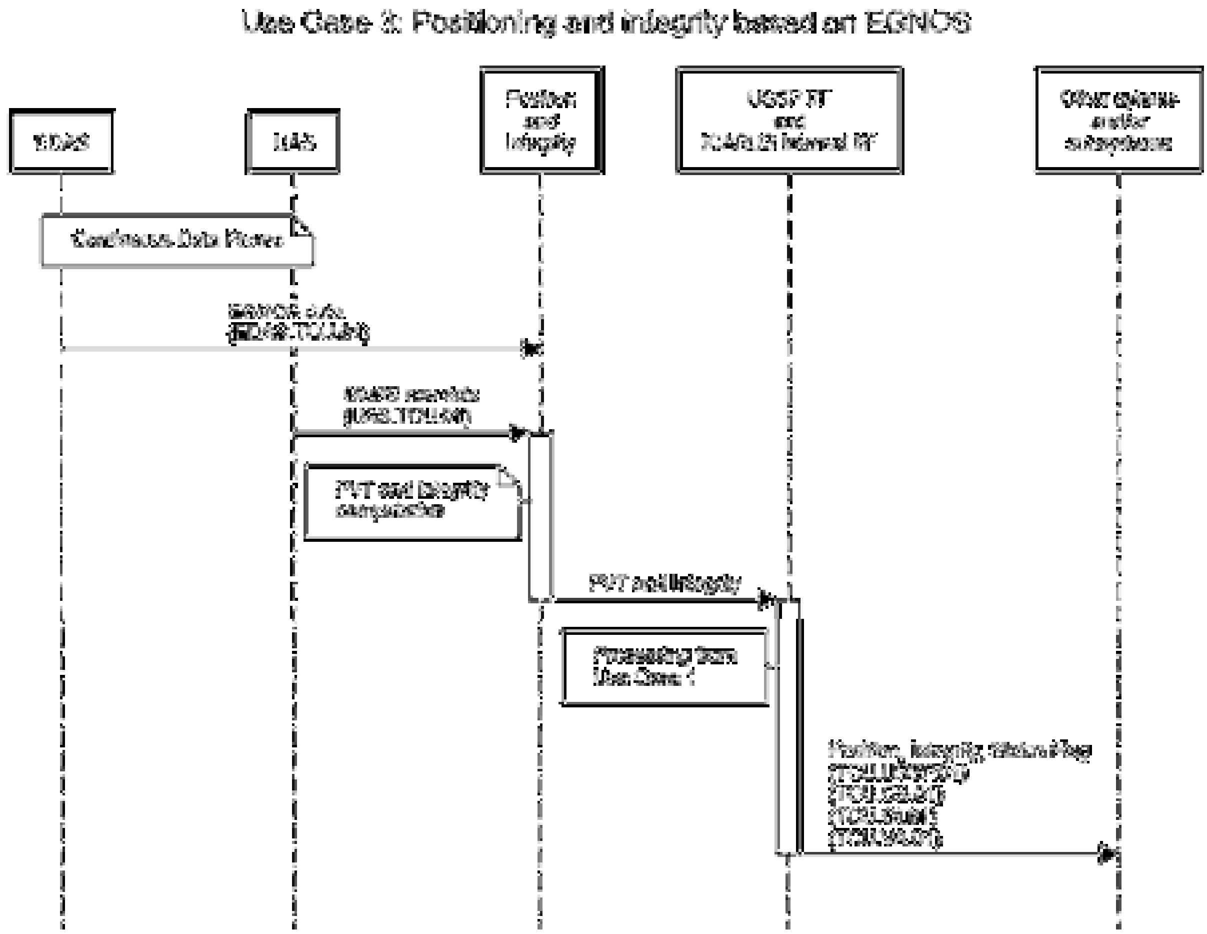

Figure 2-6: Use case 3 sequence diagram

The fourth sequence diagram demonstrates the ability of the GNSS subsystem to correctly treat GNSS raw data from a UAS to compute its position and its integrity parameters using the ARAIM algorithm through the following steps (in addition and in parallel to what is described in the second sequence diagram):

1. Reception and decoding of raw GNSS data from the tracked UAS;
2. Processing of the retrieved data to compute the UAS PVT and related integrity parameters (i.e. Horizontal/Vertical Protection Levels).

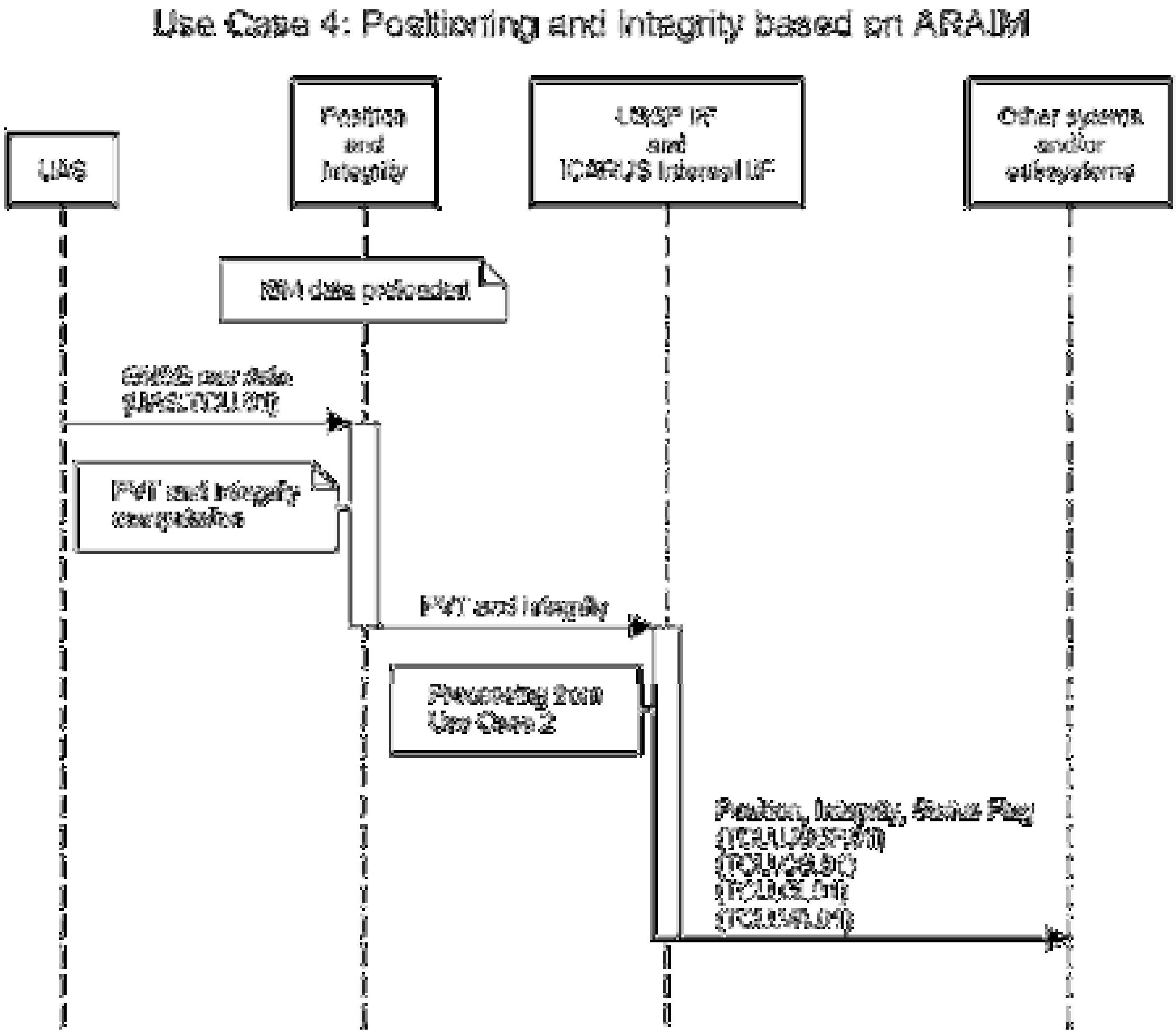

Figure 2-7: Use case 4 sequence diagram

Moreover, common to all the sequence diagrams an administrator has the possibility to set some specific processing options or configurations:

1. The administrator logs on to the maintenance interface through an ssh connection (ADM.MAN.01).
2. The administrator configures one of the following subsystem components by properly editing the configuration files:
    a. Reference station interface: NTRIP client configuration (selection of stations and connection parameters).
    b. EDAS interface: connection parameters (URL and port used).
    c. GNSS signal monitoring: algorithm implemented (ARAIM or EGNOS), observables used.
    d. Position and integrity: algorithm implemented (ARAIM or EGNOS), observables used.

e. Output interfaces: MQTT broker features (address, port, topics).

After the maintenance/reconfiguration activity is completed, the administrator disconnects from the module of interest and from the control logic interface.

2.3 GNSS Microservice Communication

The MQTT protocol, as described in D.4.1, was chosen to guarantee data exchange and communication to the outside (USSP) for the GNSS microservice. MQ Telemetry Transport (MQTT) is a highly reliable messaging protocol since, being based on TCP, it guarantees the delivery of messages even on degraded or slow networks.

Its main features, which make it very suitable for M2M exchange are:

• publish / subscribe mechanism

• very simple

• very low overhead

The GNSS microservice implements the MQTT Mosquitto message broker, an excellent open-source tool that is an autonomous and independent service, which centralises the connections between the most disparate IoT objects (UAS in this case), making M2M communication optimal. Through this broker it is possible to manage all messages received from and transmitted to objects that are part of the network without knowing their number, their characteristics, or the physical network address.

According to this widespread pattern, the *sender of a message* (called *publisher*) turns to the broker to "publish" their message. The *recipients* (*subscriber*) in turn contact the *broker* to "*subscribe*" to receiving messages. The broker forwards every message sent by a publisher to all subscribers interested in that “*type*” (*Topic*) of message. It is easy to imagine, therefore, that it is not necessary for an object to repeatedly query the broker to find out if there are new messages to be read (as in the standard mode): these are automatically notified by the broker if there is a new message to be delivered.

2.3.1 EDAS & Ground Reference Station Data

The exchange of GNSS raw data streams from the EDAS Service and from the Ground Reference Stations, in real time, is implemented through the NTRIP protocol (see Figure 2-1: High-level architecture of GNSS microservice). The main elements that define the actors involved in the distribution of data according to this protocol are:

• NTRIP Caster: this is the element that centralises the GNSS raw data flows from the GNSS receivers;

• NTRIP Server: generally co-located with the GNSS receiver, this module deals with the communication of raw data to the NTRIP Caster;

• NTRIP Client: this is the module that takes care of receiving and decoding GNSS data; the passage of data through this module is preparatory to the processing of raw data.

The EDAS data are retrieved using the EDAS Client, a COTS provided by the European GNSS Agency (GSA) for connection to the EDAS network, that connects to the EDAS server by subscribing to receive raw data catalogued as belonging to the SL1 service (Service Level 1 - RTCM data). Developed entirely in Java, it forwards the raw data received from the EGNOS RIMS via TCP.

2.3.2 Drone GNSS Raw Data

The GNSS raw data, necessary to provide the integrity calculation, are retrieved by the GNSS microservice using the User Datagram Protocol, UDP, see Figure 2-1: High-level architecture of GNSS microservice.

UDP is a protocol that allows connectionless transmission of datagrams in IP-based networks. To reach the desired services on the target hosts, the protocol uses the ports listed as one of the main components in the UDP header. Like many other network protocols, UDP belongs to the Internet protocol family, although it should be classified in the transport layer and therefore as an intermediary instance between the network layer and the application layer.

The UDP protocol is a direct alternative to the extended TCP; the two protocols differ in particular on one point: while transmission via TCP takes place only after the obligatory three-way handshake (mutual authentication between sender and recipient, including creating a connection), UDP forgoes this procedure to reduce the transmission duration to the minimum possible. With UDP, an application can send information very quickly, as there is no need to create a connection with the recipient or wait for a response, so the drone data can be quickly ingested and processed.

2.3.3 USSP Connection

The connection between GNSS microservice and the USSPs is made by MQTT protocol. The *sender of the message* (called *publisher*), in this case the GNSS microservice, turns to the broker to "publish" their message. The *recipients* (*subscriber*), in this case the USSPs (D-flight, Pansa UTM) in turn contact the *broker* to "*subscribe*" to receiving messages. The broker forwards every message sent by the publisher to all subscribers interested in that "*type*" (*Topic*) of message. In this way, all the calculations performed by the microservice are sent to the USSPs.

Figure 2-8 is an example of the connection via MQTT with the USSP to retrieve he necessary data.

Figure 2-8: Example of input drone data topic when no direct access to UAS GNSS raw data is foreseen

The output of the GNSS microservice calculation will be shared with USSP by the MQTT protocol as shown in the example in Figure 2-9.

Figure 2-9: Example output GNSS module calculation to USSP when no direct access to UAS GNSS raw data is foreseen

3 Testing and validation approach

In this chapter a description of the tests will be provided, with the aim of testing the operation of the GNSS microservice with the other components of the system, not only to verify the communication with the Input sources and with the GNSS service clients, but also to validate the correct functioning of the implemented algorithms.

The following tests will be detailed:

1) Test connection and communication with the UAS to receive raw data
2) Test connection and communication with the EDAS to receive data
3) Test connection and communication with the Ground Reference Stations to receive data
4) Test connection and communication with the USSP (D-flight and Pansa UTM) to retrieve data
5) Test connection and communication with the USSP (D-flight and Pansa UTM) to send calculation results
6) Test connection and communication with internal Modules (VCS, GIS, VALS) to send calculation results
7) Test the functioning of the algorithm for integrity estimation (GPS + EGNOS, ARAIM)

1-Connection with UAS	
Test case Id	*GNSS.01*
Test defined by	*Corrado Orsini*
Test executed by	*Andrea D'Agostino*
Testing date	*Validation Activities*
Test purpose	*The test aims to:* *1) check the communication between UAS and the GNSS microservice* *to allow RAW data to be sent* *2) verify the reception of raw data coming from the UAS*
Test environment	*The test environment involves the use of the following components:* *1) UAS sending data remotely from Topview premises* *2) GNSS microservice hosted within the Telespazio facilities for reception*
References	*N.A.*
Validation target/metr	*The expected results are:*

	1) Establish a connection with the UAS *2) correctly receive the raw data with the identification of the UAS*		
Tested components	*List the HW and SW components under testing with version info.*		
	HW-N.A.	*Position and Integrity Module, UDP connection*	
Tested interfaces	*UAS.TCU.01*		
Other interfaces	*N.A*		
Pre-test conditions	*The UAS should be able to send GNSS Raw Data from the on-board receiver*		
Test tools	*N.A.*		
Test repetitions	*<2>*	*N.A.*	
Integration test steps **Definition and results** **Log**	**# Step**	**Description**	**Result**
	1	*Establish a connection via UDP with UAS*	DONE
	2	*UAS send Raw Data*	SKIP
	3	*Raw data are received by the GNSS Position and Integrity Module*	
Test verdict	PASS/ FAIL/ PENDING	*N.A.*	
Change requests		*N.A.*	
Defects		*N.A.*	

2-Connection with EDAS	
Test case Id	*GNSS.02*
Test defined by	*Corrado Orsini*

Test executed by	*Andrea D'Agostino*		
Testing date	*Validation Activities*		
Test purpose	*The test aims to:* *1) check the communication between EDAS and the GNSS microservice to allow sending EGNOS data* *2) verify the reception of EGNOS data coming from the EDAS*		
Test environment	*The test environment involves the use of the following components:* *1) EDAS internet service* *2) GNSS microservice hosted within the Telespazio facilities for reception*		
References	*N.A.*		
Validation target/metr	*The expected results are:* *1) Establish a connection with EDAS service* *2) correctly receive the data*		
Tested components	*List the HW and SW components under testing with version info.*		
	HW-N.A.	*Position and Integrity Module, GNSS Signal Monitoring Module, MQTT connection*	
Tested interfaces	*EDAS.TCU.01*		
Other interfaces	*N.A*		
Pre-test conditions	*The EDAS service should work and an Internet connection should be active*		
Test tools	*N.A.*		
Test repetitions	<2>	*N.A.*	
Integration test steps **Definition and results** **Log**	**# Step**	**Description**	**Result**
	1	*Establish a connection via the MQTT broker with the EDAS*	DONE
	2	*EDAS sends EGNOS Data*	SKIP
	3	*EGNOS data are received by the GNSS Position and Integrity Module and by the Signal Monitoring Module*	

Test verdict	PASS/ FAIL/ PENDING	*N.A.*
Change requests		*N.A.*
Defects		*N.A.*

3-Connection with Ground Reference Stations		
Test case Id	*GNSS.03*	
Test defined by	*Corrado Orsini*	
Test executed by	*Andrea D'Agostino*	
Testing date	*Validation Activities*	
Test purpose	*The test aims to:* *1) check the communication between ground reference stations* *and the GNSS microservice to allow sending raw data* *2) verify the reception of raw data coming ground reference stations*	
Test environment	*The test environment involves the use of the following components:* *1) Ground reference stations data services* *2) GNSS microservice hosted within the Telespazio facilities for reception*	
References	*N.A.*	
Validation target/metr	*The expected results are:* *1) Establish a connection with the ground reference station service* *2) correctly receive the data*	
Tested components	*List the HW and SW components under testing with version info.*	
	HW-N.A.	*GNSS Signal Monitoring Module,* *MQTT connection*

Tested interfaces	*REF.TCU.01*		
Other interfaces	*N.A*		
Pre-test conditions	*The ground reference station service should work and an Internet connection should be active*		
Test tools	*N.A.*		
Test repetitions	<2>	*N.A.*	
Integration test steps Definition and results Log	**# Step**	**Description**	**Result**
	1	*Establish a connection with the ground reference station service via the MQTT broker*	DONE
	2	*The ground reference station service sends raw data*	SKIP
	3	*Raw data are received by the Signal Monitoring Module*	
Test verdict	PASS/ FAIL/ PENDING	*N.A.*	
Change requests		*N.A.*	
Defects		*N.A.*	

4-Connection with USSP	
Test case Id	*GNSS.04*
Test defined by	*Corrado Orsini*
Test executed by	*Andrea D'Agostino*
Testing date	*Validation Activities*
Test purpose	*The test aims to:* *1) check the communication between the USSP and the GNSS microservice to allow telemetry data to be sent*

	2) verify the reception of telemetry data coming from the USSP		
Test environment	*The test environment involves the use of the following components:* *1) USSP (D-Flight and/or Pansa UTM)* *2) GNSS microservice hosted within the Telespazio facilities for reception*		
References	*N.A.*		
Validation target/metr	*The expected results are:* *1) Establish a connection with the USSP* *2) correctly receive telemetry data*		
Tested components	*List the HW and SW components under testing with version info.*		
	HW-N.A.	*GNSS Signal Monitoring Module, MQTT connection*	
Tested interfaces	*USSP.TCU.01*		
Other interfaces	*N.A*		
Pre-test conditions	*The USSP should be able to subscribe to the MQTT broker Topic and to send telemetry data to external sources*		
Test tools	*N.A.*		
Test repetitions	<2>	*N.A.*	
Integration test steps **Definition and results** **Log**	**# Step**	**Description**	**Result**
	1	*Establish a connection with the USSP via the MQTT broker*	DONE
	2	*The USSP sends telemetry data*	SKIP
	3	*Telemetry data are received by the Signal Monitoring Module*	
Test verdict	PASS/ FAIL/ PENDING	*N.A.*	
Change requests		*N.A.*	
Defects		*N.A.*	

5-Connection with USSP for GNSS results		
Test case Id	*GNSS.05*	
Test defined by	*Corrado Orsini*	
Test executed by	*Andrea D'Agostino*	
Testing date	*Validation Activities*	
Test purpose	*The test aims to:* *1) check the communication between the USSP and the GNSS microservice to allow GNSS computation results to be sent* *2) verify the USSP reception of data results coming from the GNSS microservice*	
Test environment	*The test environment involves the use of the following components:* *1) USSP (D-Flight and/or Pansa UTM)* *2) GNSS microservice hosted within the Telespazio facilities for sending*	
References	*N.A.*	
Validation target/metr	*The expected results are:* *1) Establish a connection with the USSP* *2) Correctly send GNSS results data*	
Tested components	*List the HW and SW components under testing with version info.*	
	HW-N.A.	*GNSS Signal Monitoring Module, Position and Integrity Module* *MQTT connection*
Tested interfaces	*TCU.USSP.01*	
Other interfaces	*N.A*	
Pre-test conditions	*The USSP should be able to subscribe to MQTT broker Topic and to receive GNSS microservice results data*	
Test tools	*N.A.*	
Test repetitions	*<2>*	*N.A.*

	# Step	Description	Result
Integration test steps Definition and results Log	*1*	*Establish a connection with the USSP via the MQTT broker*	DONE
	2	*GNSS microservice send results data*	SKIP
	3	*Computation results data are received by the USSP and shown to the end-user*	
Test verdict	PASS/ FAIL/ PENDING	*N.A.*	
Change requests		*N.A.*	
Defects		*N.A.*	

6-Connection with ICARUS internal modules	
Test case Id	*GNSS.06*
Test defined by	*Corrado Orsini*
Test executed by	*Andrea D'Agostino*
Testing date	*Validation Activities*
Test purpose	*The test aims to:* *1) check the communication between the GNSS microservice and the GIS, VCS,VALS modules to allow GNSS computation results to be sent* *2) verify reception of data results coming from the GNSS microservice*
Test environment	*The test environment involves the use of the following components:* *1) VCS, GIS , VALS modules* *2) GNSS microservice hosted within the Telespazio facilities for sending*
References	*N.A.*

Validation target/metr	*The expected results are:* *1) Establish a connection with the other internal modules* *2) Correctly send GNSS results data*		
Tested components	*List the HW and SW components under testing with version info.*		
	HW-N.A.	*Position and Integrity Module, MQTT connection*	
Tested interfaces	*TCU.GI.01* *TCU.CS.01* *TCU.VA.01*		
Other interfaces	*N.A*		
Pre-test conditions	*The Internal Module should be able to subscribe to the MQTT broker Topic and to receive GNSS microservice results data*		
Test tools	*N.A.*		
Test repetitions	*<2>*	*N.A.*	
Integration test steps Definition and results Log	**# Step**	**Description**	**Result**
	1	*Establish a connection with GIS,VCS,VALS via the MQTT broker*	DONE
	2	*The GNSS microservice sends results data*	SKIP
	3	*Computation results data are received by the GIS,VCS,VALS and verified*	
Test verdict	PASS/ FAIL/ PENDIN	*N.A.*	
Change requests		*N.A.*	
Defects		*N.A.*	

7-Algorithms Validation		
Test case Id	*GNSS.07*	
Test defined by	*Corrado Orsini*	
Test executed by	*Andrea D'Agostino*	
Testing date	*Validation Activities*	
Test purpose	*The test aims to:* *1) verify the correct functioning and validity of the results* *obtained through the GPS+EGNOS and ARAIM algorithms*	
Test environment	*The test environment involves the use of the following component:* *2) GNSS microservice hosted within the Telespazio facilities for sending*	
References	*N.A.*	
Validation target/metr	*The expected results are:* *1) verify the validity of the results obtained using a test case* *already validated to be used as a benchmark*	
Tested components	*List the HW and SW components under testing with version info.*	
	HW-N.A.	*Position and Integrity Module*
Tested interfaces	*N.A.*	
Other interfaces	*N.A*	
Pre-test conditions	*availability of an already validated test case*	
Test tools	*N.A.*	
Test repetitions	*<2>*	*N.A.*

Integration test steps Definition and results Log	# Step	Description	Result
	1	*Use data from the test case*	DONE
	2	*Activate GPS+ EGNOS algorithm and ARAIM algorithm computations*	SKIP

	3	*Compare the results with the benchmark* *verified*	
Test verdict	PASS/ FAIL/ PENDING	*N.A.*	
Change requests		*N.A.*	
Defects		*N.A.*	

Test results will be provided in the second release (issue 2) of this document after the testing and verification activities that are scheduled for between November 2021 and January 2022.

4 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"