D5.1: Report on the testing methodologies and testbed setup

Work package	WP 5
Task	Task 5.1
Due date	M26
Submission date	17/03/2025
Deliverable lead	AUTH
Version	2.0
Authors	See next page
Reviewers	Agapi Mesodiakaki, Konstantinos Ntontin, Luís Santos, Timo Kellermann
Keywords	6G, NTN, Testbed Configuration, Demo Setups, Requirements, KPIs

www.ether-project.eu

Co-funded by the European Union

Grant Agreement No.: 101096526
Call: HORIZON-JU-SNS-2022

Topic: HORIZON-JU-SNS-2022-STREAM-B-01-03
Type of action: HORIZON-JU-RIA

Document revision history

Version	Date	Description of change
v0.1	2025/01/08	TOC release
v0.2	2025/01/31	First internal draft
v0.3	2025/02/14	Second internal draft
v0.4	2025/02/21	Ready for internal review
v1.0	2025/02/28	First internal review
v2.0	2025/03/17	Final version

Author list

Organisation	Author
AVA	Eleftheria Hadjioannou
AUTH	Agapi Mesodiakaki, Marios Gatzianas, Dimitrios Palianopoulos, Charalampos Bratsoudis
CA	Hamada Alshaer, Fadhil Firyaguna
i2CAT	Víctor Montilla, Roger Pueyo Centelles, Joan Adrià Ruiz de Azúa, Timo Kellermann, José Ávila
SIOT	Sergio Aguilar Romero, Shubham Gupta, Diogo Oliveira
LIU	Nikolaos Pappas
NCSRD	Fotis Lazarakis, Antonis Alexandridis, Anastasios Paraskevopoulos
UL	Konstantinos Ntontin, Symeon Chatzinotas, Jorge Querol, Turker Yilmaz
NBC	Godfrey Kibalya, Angelos Antonopoulos
OPL	Lechosław Tomaszewski
NETAI	Alexis Duque, Marco Fiore
UBW	Luís Santos

DISCLAIMER

ETHER (*sElf-evolving terrestrial/non-Terrestrial Hybrid nEtwoRks*) project has received funding from the **Smart Networks and Services Joint Undertaking (SNS JU)** under the European Union's **Horizon Europe research and innovation programme** under Grant Agreement No 101096526.

COPYRIGHT NOTICE

Project co-funded by the European Commission in the Horizon Europe Programme		
Nature of the deliverable:	**R**	
Dissemination Level		
PU	*Public, fully open, e.g. web (Deliverables flagged as public will be automatically published in CORDIS project's page)*	✓
SEN	*Sensitive, limited under the conditions of the Grant Agreement*	
Classified R-UE/EU-R	*EU RESTRICTED under the Commission Decision No. 2015/444*	
Classified C-UE/EU-C	*EU CONFIDENTIAL under the Commission Decision No. 2015/444*	
Classified S-UE/EU-S	*EU SECRET under the Commission Decision No. 2015/444*	

* R: Document, report (excluding the periodic and final reports)
DEM: Demonstrator, pilot, prototype, plan designs
DEC: Websites, patents filing, press & media actions, videos, etc.
DATA: Data sets, microdata, etc.
DMP: Data management plan
ETHICS: Deliverables related to ethics issues
SECURITY: Deliverables related to security issues
OTHER: Software, technical diagram, algorithms, models, etc.

TABLE OF CONTENTS

LIST OF FIGURES

Co-funded by the European Union

LIST OF TABLES

ABBREVIATIONS

3GPP	3rd Generation Partnership Project
5G	5th generation (mobile network)
5GC	5G Core Network
5G PPP	5G Infrastructure Public Private Partnership
6G	6th generation (mobile network)
6GIA	6G Industry Association
A2G	Air-to-Ground
ADC	Analog to Digital Converter
AI	Artificial intelligence
AMF	Access and Mobility Management Function
AoI	Age of Information
API	Application Programming Interface
CI/CD	Continuous Integration and Continuous Development
CIoT	Cellular IoT
CN	Core Network
CNF	Containerised Network Function
CPU	Central Processing Unit
CU	Centralized Unit
DAC	Digital to Analog Converter
DC	Direct Current
DevOps	Development Operations
DN	Data Network
DU	Distributed Unit
E2E	End-to-end
EC	European Commission
ECM	Evolved Connection Management
EE	Energy Efficiency
EM	Engineering Model
eNB	eNodeB
FL	Feeder Link
FPGA	Field Programmable Gate Array
GEO	Geostationary Earth Orbit
GIS	Geographic Information System
gNB	gNodeB
GSOA	Global Satellite Operator Association
HAPS	High Altitude Platform System

HDL	Hardware Description Language
HO	Handover
IEEE	Institute of Electrical and Electronics Engineers
IF	Intermediate Frequency
IFEC	In-Flight Entertainment and Connectivity
IoT	Internet of Things
IP	Internet Protocol
ISL	Inter-Satellite Link
ITU	International Telecommunication Union
ITU-R	ITU – radiocommunication sector
ITU-T	ITU – telecommunication standardisation sector
K3s	Kubernetes
Kepler	Kubernetes-based Efficient Power Level Exporter
KPI	Key Performance Indicator
LEO	Low Earth Orbit
LO	Local Oscillator
LTE	Long Term Evolution
LoRa	Long Range
MANO	Management and Network Orchestrator
ML	Machine Learning
MME	Mobility Management Entity
MO	Mobile Originated
MT	Mobile Terminated
NAS	Non-access Stratum
NB-IoT	NarrowBand IoT
NFV	Network Function Virtualisation
NGMN	Next Generation Mobile Networks (Alliance)
NGSO	Non-geostationary orbit
NICT	National Institute of Information and Communications Technology
NTN	Non-Terrestrial Network
OAI	Open Air Interface
OS	Operating System
OSM	Open Source MANO
PER	Packet Error Rate
QoE	Quality of Experience
QoS	Quality of Service
RAN	Radio Access Network
RCP	Required Communication Performance

REST	Representational State Transfer
RF	Radio Frequency
RFi	Radio Frequency interference
RRC	Radio Resource Control
Rx	Receiver
SAIH	Semantic-aware Information Handling
S&F	Store and Forward
SDG	Sustainable development goal
SDN	Software Defined Network
SDR	Software Defined Radio
SIMD	Single Instruction Multiple Data
SnT	Interdisciplinary Centre for Security, Reliability and Trust
SoC	System on Chip
SotA	State-of-the-Art
TLE	Two Line Element
TN	Terrestrial Network
TSDSI	Telecommunications Standards Development Society, India
Tx	Transmitter
UE	User Equipment
UN	United Nations
USRP	Universal Software Radio Peripheral
VAoI	Version Age of Information
VNF	Virtual Network Function
VNFI	Virtual Network Function Infrastructure
xNF	any-type Network Function

Co-funded by the European Union

EXECUTIVE SUMMARY

ETHER aims to architect and evaluate next generation sustainable 6G networks, integrating the terrestrial, aerial and space layers, aiming to support a variety of 6G services. To that end, it focuses on demonstrating the following three targeted use cases through proof-of-concept demos: (1) flexible payload-enabled service provisioning for delay-tolerant IoT applications via LEO satellites, (2) unified RAN for direct handheld device access at the Ka band, and (3) ETHER architecture demonstration for air-space safety-critical operations. The demonstration activities showcase the ETHER solution's capabilities, including energy-efficient service management, seamless handovers, and AI-based resource allocation for air-safety operations.

To that end, this deliverable elaborates on the testing methodology and testbed setup of the ETHER demo activities. In particular, the document 1) provides an overview of a structured step-by-step testing methodology and detailed time plan that will be followed throughout the project, 2) identifies key testing areas and demo objectives, 3) provides a list of test cases/scenarios including the KPIs to be measured and the process that will be followed in each case, 4) describes the test facilities that will be used and the relevant testbed capabilities, detailing the hardware and software components as well as their specifications, and 5) highlights any related risks and mitigation strategies so as to ensure the successful demo execution within the scheduled timeline.

In a nutshell, this deliverable aims to set the foundations of the successful execution of the ETHER demo activities by providing a clear, detailed and feasible testing methodology plan. Next WP5 deliverable D5.2 is expected to shed further light on the integration activities and interfaces of each demo.

1. INTRODUCTION

As detailed in D2.2 [1], ETHER has identified three targeted use cases, namely: i) ETHER flexible payload-enabled service provisioning to semantics aware and delay-tolerant IoT applications, ii) ETHER unified RAN for direct handheld device access at the Ka band, and iii) ETHER architecture demonstration for air-space safety critical operations. To that end, ETHER targets at 3 associated proof-of-concept demonstrators that showcase the capabilities of the unified RAN, the seamless handover policies and the overall ETHER multi-layered architectural solution. In particular, the demo activities include:

- The 1st demo relates to flexible payload-enabled service provisioning from LEO satellites for providing delay-tolerant services to IoT devices on the ground. In this demo, the semantics-aware information handling algorithms will be incorporated and tested to further enhance the energy-efficiency of the considered battery-powered devices. In addition, the ETHER flexible payload functionality will be demonstrated to enable the management of the NB-IoT service over a target region among the different satellites.
- In the 2nd demo, the capabilities of the Unified ETHER RAN will be showcased and tested. To this end, the antenna designs for direct access of handheld devices from LEO satellite together with the advancements in the channel modelling will be incorporated in the space emulation platform of UL. In addition, the energy-efficient vertical handover algorithms will be also incorporated and their performance in the case of a vertical handover between the terrestrial and the space layer will be tested and evaluated.
- In the 3rd demo, the overall ETHER solution will be showcased and evaluated in an air-safety critical operations' use case. In particular, in this demo, the ETHER architectural solution will be tested leveraging an AI-based traffic forecast tool feeding the ETHER decision engine for efficient E2E resource allocation, while ensuring that the required QoS is guaranteed. The ETHER edge orchestrator will also be incorporated and evaluated in terms of its capability to provide zero-touch edge resource orchestration in the aerial and space layers.

To that end, this document aims to provide insights on the demo testing methodology and demo setups. In particular, it outlines the purpose of each demo, the demo blocks, and the specific ETHER technologies involved in each case. The software and hardware components as well as their specifications are also detailed, including a bill of materials for each demo. The KPIs to be tested and the process that will be followed in each demo is also specified. Special focus is given on providing a detailed step-by-step testing methodology composed of a number of intermediate tests following a predefined time plan to guarantee the successful demo completion and execution within the predefined schedule. Finally, this document presents any identified related risks and mitigation plans.

To that end, the structure of this document is as follows:

- Section 1 provides an introduction to the document, presenting the document objective and scope, as well as it structure.
- Sections 2, 3 and 4 present the detailed testing methodology and testbed setup of each one of the demos 1, 2 and 3, respectively.
- Section 5 summarises the main findings of the document, reiterating the importance of a detailed and thorough testing methodology for the successful project demonstration completion. It offers conclusions and references to future related deliverables regarding the demonstrator activities of the project.

2. DEMO 1: FLEXIBLE PAYLOAD-ENABLED SERVICE PROVISIONING TO SEMANTICS AWARE AND DELAY-TOLERANT IOT APPLICATIONS

2.1 USE CASE DESCRIPTION

Use Case 1 considers the provision of global Cellular IoT (CIoT) service to UEs that run delay-tolerant IoT applications, such as remote environmental monitoring, agricultural sensing, mining, wildlife conservation and asset tracking. A sparse Low Earth Orbit (LEO) constellation of a few tens of satellites offers ubiquitous 3GPP-compliant non-terrestrial NB-IoT coverage, leveraging Store & Forward (S&F) capabilities.

The **Flexible Payload**, developed in T3.2 (Objective. 3, Innovation Technology T-4), lays the technical foundation for Use Case 1. It provides the **regenerative satellite payload** platform with the capacity to deploy a CIoT service –as well as other services and protocols– dynamically, as the satellite flies over a target region, at a given moment in time. The operations are controlled by a ground-based **Management and Network Orchestration (MANO)** entity, which instructs the satellites to activate and deactivate different services in a coordinated manner and assists in the exchange of status and context with the satellites to come (T-7).

A **S&F-based NB-IoT** service leveraging the **horizontal handover policies** and enhancements from T3.4 will be deployed on the Flexible Payload, as well as other IoT and non-IoT applications (T-6). This service provides global CIoT coverage to IoT devices on ground that can benefit from ubiquitous connectivity for delay-tolerant applications. By using **semantic agents** and employing **timeliness metrics**, these IoT devices can reduce the amount of transmitted data through the *IoT device — satellite — ground segment — IoT cloud* chain, effectively improving energy efficiency while preserving the conveyed information (T-5).

Figure 2-1 offers an overall picture of the scenario considered for Use Case 1. It shows a sparse constellation of LEO satellites (top) enabled by the Flexible Payload that provide discontinuous coverage to IoT devices scattered all over the globe (bottom). Orchestration of services running on the satellite is performed on the ground (bottom right), by the Network Function Virtualisation (NFV) Management and Network Orchestrator (MANO), during the Feeder Link (FL) availability windows.

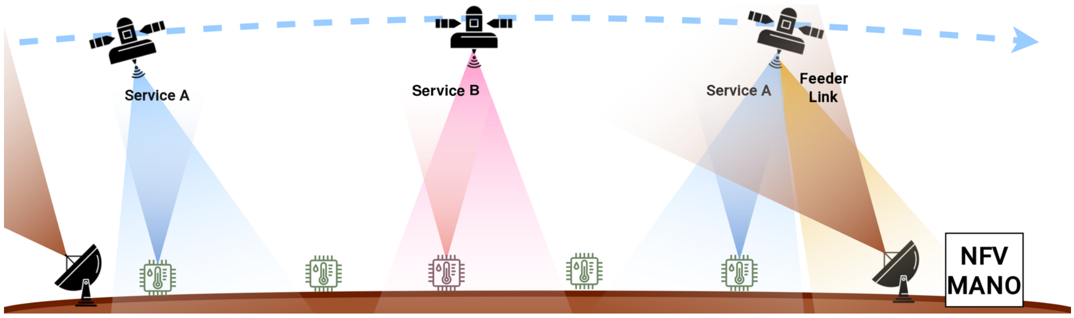

Figure 2-1: Use Case 1 scenario, showing a sparse constellation of Flexible Payload-enabled LEOs.

2.1.1 Flexible Payload and service orchestration

Since Demo 1 aims to focus on the part of satellites that run the services (i.e., the payload), the architecture of a representative node will only include the essential hardware to represent the payload, hence leaving the rest of the typical modules used to control the overall satellite (power management, latitude control, solar panels, on-board computer, etc.) aside.

Use Case 1 is based on a regenerative payload approach, offering the ability to process services' data locally (i.e., in-flight). The hardware used to implement the payloads is fully representative of a real platform: a Field Programmable Gate Array (FPGA) System on Chip (SoC) based on the Xilinx UltraScale+ family, which has been already tested in real missions [2] [3] [4] [5]. The payload management is performed by the flexible payload framework (adapted and integrated in T3.2), which is a customized environment based on a mix of hardware and software resources that aims to provide all the necessary tools to ensure a correct deployment of the onboard services.

At the functional level, the Flexible Payload framework focuses on the scalability of services and resource sharing via network to provide a remote, controlled and intelligent deployment. In this sense, the integration of Kubernetes (K3s) [6] in the payload itself is essential. Payload services can be accessed from the ground segment using Kubernetes. This interacts directly with the NFV MANO and Virtual Network Functions (VNFs) on the satellites via feeder links. As an extension, it is even possible to consider the use of Inter-Satellite Links (ISLs) allowing payloads within a constellation to exchange data without the need of a ground station. However, this is beyond the scope of Demo 1.

The goal of this combined architecture (flexible payload technology using service orchestration) is to contribute to experiment with the following aspects:

1. Edge computing: The base hardware on which the flexible payload runs (FPGA SoC) allows the creation of optimal architectures for the execution of applications. On the one hand, it has logic cells that allow adding new hardware peripherals or replacing existing ones, enabling, among other things, to accelerate software processes and, consequently, reduce computing consumption. In addition, the integrated ARM processor has the NEON extension, which is an advanced Single Instruction Multiple Data (SIMD) architecture that can optimize and accelerate complex mathematical operations, software-based encodings and decodings. The combination of everything allows us to obtain a very balanced platform in terms of consumption and computing power.

2. Repurposing in-orbit infrastructure: The ability to replace a large part of the hardware by reprogramming the logic cells (hardware reconfiguration), together with software virtualization mechanisms [7], allows the creation of fully adaptable payload designs, even when the satellite is already in orbit. With all these mechanisms active, if mission

conditions change, the task of reconditioning the services of a satellite in flight becomes possible.

3. Federated satellite systems: The proposed architecture enables federated satellite systems through a ground-based orchestrator that combines Open Source MANO (OSM) and Kubernetes to control resources and deploy VNFs on satellites. This orchestrator allows for on-orbit repurposing by dynamically scheduling and deploying services only when required. By leveraging cloud-native technologies, the system can efficiently manage distributed space services across multiple satellites, optimizing resource utilization and enabling flexible service provisioning.

4. Reconfiguration of in-orbit infrastructure: The ability to completely reconfigure satellite nodes (payloads) opens the door to making deeper infrastructure changes. Considering a system where the nodes are connected via Ground Station or even via ISLs, the complete reconfiguration of each node allows to deploy systems that comply with the vision proposed in NFV, where each node has functions that can be shared in the network (constellation of satellites) and that are accessible according to premises (workload of each node, geographical layout, etc.).

5. Service continuity: The orchestration system leverages geospatial data to ensure service continuity across the satellite constellation. By utilizing orbit predictions, the orchestrator can proactively deploy VNFs on satellites as they approach target areas. This predictive approach allows for seamless service migration between satellites, ensuring uninterrupted availability as satellites move in and out of coverage zones. The system's ability to anticipate satellite positions and deploy services accordingly significantly enhances the reliability and consistency of service delivery.

6. Unified network management: This approach allows for consistent policy application and streamlined resource allocation across the network. The unified orchestrator can efficiently distribute information throughout the constellation, ensuring that all satellites maintain up-to-date policies and service versions. This network-wide synchronization enables more effective utilization of distributed satellite resources and facilitates seamless service migration between satellites.

2.1.2 Store-and-forward architecture for delay-tolerant IoT applications

ETHER contributes to delay-tolerant IoT applications by providing them with global connectivity, paving the way for adoption of ubiquitous IoT solutions. Because these applications do not rely on low latency, they benefit from low-density LEO satellite constellations that offer more cost-effective IoT services than mega constellations or Geostationary Earth Orbit (GEO) satellites. Furthermore, ETHER's contributions align with current 3rd Generation Partnership Project (3GPP) standardization efforts, aiming to reduce connectivity costs, prevent vendor/operator lock-in, and enable IoT use cases to scale globally.

Delay-tolerant applications such as remote environmental monitoring, agricultural sensing, mining, wildlife conservation and asset tracking greatly benefit from global coverage. Although a single LEO satellite in a polar orbit can provide global coverage using store-and-forward mechanisms, these applications often require a minimum level of service that exceeds the capabilities of a single satellite. Consequently, they can benefit from a sparse constellation.

However, Feeder Link connectivity between satellites in a low-density LEO constellation and ground stations is only available at a limited number of locations, making continuous connections not feasible. This situation requires the adoption of regenerative payloads that support store-and-forward mechanisms to provide connectivity, particularly in remote or underserved regions where ground station placement is not practical and therefore continuous end-to-end connectivity is not possible.

To enable seamless connectivity within multi-satellite environments under store-and-forward operation, dissemination of User Equipment (UE) context is essential. ETHER advances this area by providing an architecture and mechanisms that efficiently disseminate UE context among sparse satellite constellations, thereby increasing scalability and reducing overall delays. This approach ensures that IoT devices can maintain service connectivity with multiple satellites, despite the inherent limitations of discontinuous feeder links.

ETHER builds on previous work [8], which introduced a 3GPP-compliant architecture implementing store-and-forward via proxy functions distributed between the satellite payload and the ground network. This distribution addresses Feeder Link discontinuities by placing certain core network functions on the satellite. The proxies are categorized into three types:

- Authentication Proxy, which facilitates attach/registration procedures across visibility windows by buffering authentication data when the feeder link is unavailable and forwarding it upon reconnection.
- User Data proxy, which buffers Mobile-Originated (MO) and Mobile-Terminated (MT) traffic to enable discontinuous transmission.
- User Context proxy, which disseminates UE context across multiple satellites, vital for multi-satellite scenarios.

Although Kellermann et al. validated the Authentication and User Data Proxies, the UE context transfer for multi-satellite operation was demonstrated only by cloning the entire Mobility Management Entity (MME)/Access and Mobility Management Function (AMF) from a source satellite to a target satellite, which proved inefficient and disruptive. In this context, ETHER incorporates a novel UE context dissemination mechanism as an evolution of the architecture, ensuring that the UE context is more efficiently distributed among satellites to enable seamless handover and maintain service reliability in multi-satellite environments.

Building on this, the work in D3.1 [9] focused on designing, implementing, and validating a novel store-and-forward User Context Proxy. This mechanism enables per-UE context updates between the MME/AMF entities on different satellites, without overwriting the entire in-memory satellite database and disturbing the state of other UEs.

The S&F UE Context Proxy was validated using a laboratory testbed emulating an NTN scenario with two LEO satellites, a ground segment, and multiple NB-IoT UEs [9]. The validation successfully demonstrated UEs' seamless horizontal handover, confirming the architecture's ability to maintain multi-satellite service continuity despite discontinuous Feeder Links. The study further emphasized the importance of effective context dissemination for horizontal LEO-to-LEO handovers, as timely updates across satellite can enhance service reliability, avoid unnecessary UE reattachments, and improve resource efficiency for both UEs and satellites.

Finally, the proposed architecture is particularly well-suited for the integration into Flexible Payload systems. By enabling dynamic dissemination of UE context, payload utilization can be optimized through the allocation or removal of core network functions alongside UE contexts according to demand or availability. This approach ensures seamless service availability for IoT applications across varying constellations sizes.

2.1.3 Semantics-aware information handling solutions

ETHER will contribute to IoT networks through a semantic-aware approach aimed at generating the right amount of data and transmitting the relevant content to the right place at the right time. To achieve this, semantic attributes are employed and optimized as key performance metrics while considering the available resources. The semantics of information refer to the timeliness and importance of data. Leveraging semantics in integrated TN-NTNs will enable the generation and transmission of only a small fraction of data without compromising the conveyed information, leading to improved energy efficiency, lower latency, and reduced storage requirements.

In Use Case 1, by utilizing semantic agents and employing timeliness metrics, such as Age of Information (AoI), and/or joint timing and content-based metrics, such as Version AoI (VAoI), we aim to reduce the amount of communicated data between the IoT device and the satellite network while preserving the conveyed information. The VAoI is a semantic metric in status update systems that measures both the timeliness and relevance of information, reflecting how many versions the receiver lags behind the source as new content or versions are generated. VAoI extends AoI which just captures the freshness of information, the time elapsed since the freshest information has been received, and neglects the content.

The results in the published papers [10] [11] [12], as reflected in [9], showed that minimizing VAoI leads to an optimal scheduling policy, which significantly reduces the number of transmissions from a ground IoT device to a connected satellite while maintaining the same performance in terms of providing the most recent and relevant data within the network. In Demo 1, the semantics-aware information handling approach is applied to the communication of data between the device and the satellite nodes, where the decision to transmit each data sample or remain idle is made by the semantic agent.

2.2 DEMO PURPOSE

Demo 1 is well aligned with Use Case 1. Participating partners bring in their previous experience and current focus on regenerative LEO satellite payloads, especially in the format of CubeSats. The demo drives a shift towards **more flexible, software-defined regenerative satellite payload architecture, capable of dynamically reconfiguring** functionalities as different services are required to be provisioned.

The purpose of the demo is to test and demonstrate the integration of different applications onto the **Flexible Payload** platform (T3.2), and their management. The lifecycle of these applications will be **dynamically orchestrated**, based on their specific service provision requirements. Among them outstands the connectivity provision to IoT devices using **delay-tolerant S&F NB-IoT**, an application that benefits from the **Horizontal Handover** developments (T3.4) that offer seamless connectivity to IoT devices. In turn, these devices generate a stream of data that benefits from **Semantics-Aware Information Handling** solutions (T3.3) to improve the energy efficiency associated to their transmission.

2.2.1 Key performance indicators to be tested

ETHER defined in D2.2 [1] a set of Key Performance Indicators (KPIs), that serve as measurable target values by which it is possible to verify the efficacy of the proposed system architecture. In the case of Demo 1, the targeted key performance indicators of use case 1 are the following:

- 75% higher energy efficiency compared to the State-of-the-Art (SotA)

- 100% network coverage (i.e., global outdoors coverage)
- Ensure correct service deployment by showing the capabilities to execute simultaneous or sequential services.
- Capability to deploy software-based and hardware-based services.

Table 2-1: Demo 1 KPIs.

Identifier	KPI	Description
ETH-KPI-UC1-01	Energy efficiency	>75% compared to SotA
ETH-KPI-UC1-02	Network coverage	100% global network coverage
ETH-KPI-UC1-03	Multiple simultaneous / sequential services	At least 2 services executed simultaneously or sequentially
ETH-KPI-UC1-04	Different service types	Deployment of SW based services, HW-based services or a mix of them

It is important to clarify the definition of these KPIs and how they will be measured during the demo execution. This section will address this clarification; specifically, the section is structured in two blocks, one per KPI, in which the definition of the KPI and the measurement method is addressed.

ETH-KPI-UC1-01 - Energy efficiency

Energy efficiency is a performance indicator regarding the capability of the system to achieve the required functionalities and performance reducing the energy cost associated (e.g., reducing the energy waste). In this demonstration, the efficiency improvement stems from decreasing the amount of data generated at IoT devices and communicated within the system while preserving the conveyed information. For this purpose, the semantic-aware algorithms developed in T3.3 can reduce unnecessary transmissions by optimizing the semantics of information and filtering the exchanged data packets.

Considering this number of transmissions metric, the energy consumption of a system due to the transmission of packets (E_n^{tx}) is defined as follows:

$$\forall n \in \mathbb{N}\ E_n^{tx} = \sum E_i^{tx} \tag{2.1}$$

Where n corresponds to the number of packets transmitted, and E_i^{tx} the energy consumption corresponding to the transmission of packet i (in Joules). Following this definition, the energy consumption corresponding to the transmissions of a packet (E_i^{tx}) is defined as follows:

$$\forall i \leq n\ E_i^{tx} = W_i^{tx} \cdot \frac{L_i}{Rb_i} \tag{2.2}$$

where W_i^{tx} corresponds to the consumption (in Watts) of transmitting the packet i, the L_i represents the length (in bits) of the packet i, and Rb_i the data rate (in bps) of the transceiver to transmit the packet i. Because the demonstration will use the same transceiver for each packet, the transmission consumption and the data rate will remain constant among the packets. So, the length of the packet will directly drive the corresponding packet energy consumption.

Finally, if n_1 represents the number of packets transmitted without applying the semantic-aware algorithms, and n_2 otherwise, the energy saving gain is formulated as follows:

$$\forall n_1, n_2 \in \mathbb{N}\ \Delta E^{tx} = \frac{E_{n_2}^{tx} - E_{n_1}^{tx}}{E_{n_1}^{tx}} \tag{2.3}$$

Considering this definition, the KPI of energy efficiency will be computed from the measurements of the number of transmitted packets in the system. Specifically, the different modules that compose the demonstration will be monitored through application uptime/downtime (when data is transmitted), which ensures the measurement of the number of transmissions. Details of the system to collect the telemetry of the execution of the demonstration (and thus the corresponding KPIs) are presented in the following section.

ETH-KPI-UC1-02 – Network coverage

The coverage metric indicates the percentage of the Earth surface that is being served by the network. It is important to remark that the coverage metric does not require the desired level of service to be concurrently accommodated. If this is the case, another indicator shall be defined, such as service continuity. This demonstration is based on the first use case of ETHER, which is focused on the service of NB-IoT from satellite systems. This service is characterized by accepting large delays in the communications, and thus the satellite constellation is typically conceived as a LEO sparse constellation. This constellation generates a disrupted topology, in which the sensors have discontinuous connectivity with the satellites (and thus the core network).

In this case, the global coverage is achieved because the satellites that conform the constellation such follow orbital trajectories that their ground track enables to map all the Earth surface. This ground track in conjunction with the footprint of the nadir-oriented antenna (for the service link) shall cover all the surface of the Earth over time. Multiple constellations can be defined that satisfy this requirement (e.g., Walker). Among them, the main characteristic is that satellites follow a polar orbit (i.e., inclination close to 90º), which ensures that the ground track leverages the Earth rotation to cover all the surface.

Regarding the demonstration, the validation of this KPI will be performed by analysing the orbit trajectories used to represent the satellites. Specifically, these trajectories are defined from the Two Line Elements (TLE). The validation of this KPI will consist of exploring these TLEs, and computing the propagation of the orbit (by simulation) to validate that the Earth surface is covered.

ETH-KPI-UC1-03 – Multiple simultaneous / sequential services

The flexible payload framework provides an environment to manage service deployment. One of the objectives of the use case is to test at a functional level the basic capabilities of this modular architecture. Given that the framework is constructed by integrating libraries from different locations, it is important to ensure that:

- Services can run simultaneously, without interfering with each other.
- Services can run sequentially and the power on and off mechanisms work properly.

- System resources can be shared correctly even when more than one service is deployed at a time.

This KPI will focus on the virtualization mechanism which provides means to deploy container images of services. It must be verified that the images are recognized by the system and that they can be deployed remotely using an orchestrator. In addition, the capacity for simultaneous deployment of several service images will be evaluated, paying special attention to possible interference between services.

ETH-KPI-UC1-04 – Different service types

Given the disparity of services that operators may require and their nature, one of the objectives set for this use case is to be able to demonstrate that the flexible payload framework is prepared to adapt to different demands. In this sense, this KPI aims to evaluate the ability to deploy different service architectures, differentiating those that can run using only the system's software resources, from those that require specific hardware peripherals which are generally used to accelerate functions. It must be considered that the services tagged as software-based can use without any restriction all the general hardware available by default in the static logic area (ARM processor, DDR memory, sensors or external physical devices). In this sense, function acceleration is intended for hardware-based services, and it will only be performed through dynamic (reconfigurable) logic areas.

To evaluate this KPI at least 3 services architectures will be deployed:

- A software-based NB-IoT service
- A hybrid service (hardware/software-based) that deploys a GNU Radio environment to perform a demodulation.
- A hardware-based RF interference scanner service

2.3 DEMO SETUP

The Demo 1 testbed includes all the elements and technologies necessary to facilitate experimentation of the following aspects:

1. The deployment of local services within the flexible payload framework.
2. The management of services form remote ground stations by means of the K3s and the NFV MANO orchestration.
3. The functionality of services and their interaction with UE via the Service-Link.
4. The deployment of multiple applications with specific containers and their associated services. At least three examples are envisioned:
    a. Horizontal Handovers (5G)
    b. Modulation example via GNU Radio environment
    c. Radio Frequency interference (RFi) scanner

Figure 2-2 includes all these elements and enumerates the main technologies associated to the satellite – ground communication.

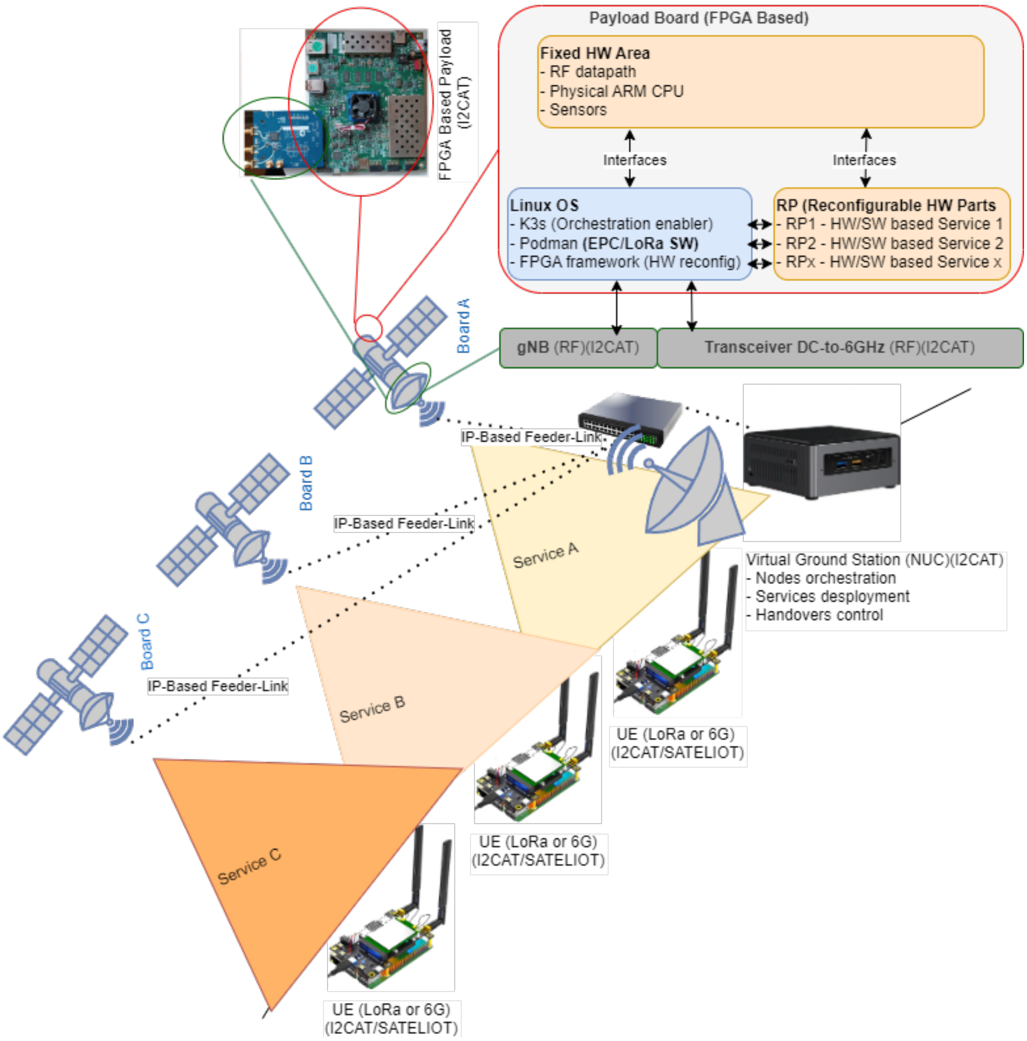

Figure 2-2: Demo setup.

In Demo 1, satellite payloads are represented by the ZCU104 board. At least two payloads are required to be able to evaluate the behaviour of services with communication disruption. Additionally, these payloads can exchange data through the ground station. For correct operation in the laboratory, a virtual ground station is included with the ability to access the payloads commanding through an Internet Protocol (IP) connection (simulating a Radio Frequency (RF) Feeder-Link). On the other hand, the connection of an operator's client is tested using specific User Equipment for the associated services that have been described above. In this way, there are two types of RF links available: a gNodeB (gNB) RF module for 5G connections and a generic RF module for the rest of the services. The generic module is implemented using an Analog Devices transceiver with AD9361 chip.

All components and interfaces are described in more depth in section 2.3.1 and 0 respectively. The list of the main components required for the demo can be found in Table 2-2.

Table 2-2: Demo 1 bill of materials.

Component	Provider	No. of elements
FPGA Board (ZCU104)	I2CAT	2 or More

Component	Provider	No. of elements
RF transceiver (FMCOMMSx)	I2CAT	2 or More
eNode B (eNB)/gNB Amarisoft Workstation	I2CAT/Satelliot	2 or More
UE (RaspberryPi + modem or real one)	I2CAT/Satelliot	2 or More
Virtual Ground Station (NUC)	I2CAT	1
IP Switch	I2CAT	1

2.3.1 Demo Components and Specifications

Demo 1 testbed includes the deployment of at least 2 satellites (payloads) offering services to UEs. When the deployed service is a 3GPP one, the payload connects to an Amarisoft gNB to provide the Radio Access Network (RAN) RF connection to the visible UEs. If non-3GPP IoT applications are deployed, then the payload can use the onboard multi-frequency transceiver (from Direct Current (DC) to 6GHz) to transmit on specific bands. Flexible payload envisions to seamlessly deploy software or hardware-based IoT applications, using the FPGA logic.

The Feeder Link discontinuity of LEO satellites is emulated by interrupting the IP-based Feeder Link connection between the payloads and the Virtual Ground Station using specific software that simulates real orbits' schedules.

Each of the satellites' payload is composed of three main elements: processing platform, RF transceiver and gNB. These are detailed in the following.

Processing platform (ZCU104 with Zynq UltraScale+)

The processing platform is the responsible to run the flexible payload framework. It provides mechanisms to deploy applications both in software and in hardware. When targeting software applications a Quad-ARM64 Cortex-A53 is used as a physical processor. On the top of the software environment there is a custom Linux OS which integrates all the necessary libraries to provide service deployment flexibility and scalability. Among all the included libraries there are four that are considered key to the correct functionality of the environment:

- PODman container framework: allows to virtualize software-based services, applications or libraries in form of image containers that can be deployed in seconds using system commands.
- K3s: A lightweight Kubernetes environment that can run under Linux Operating System (OS) and enables connection to Ground Stations enabling remote control of services.
- FPGA framework: library to access FPGA logic via exchanging bitstreams, allowing to reconfigure FPGA hardware from the Linux OS.
- LibIIO: Industrial libraries to control all the RF data path from the Linux OS, allowing to adapt transmission frequencies to services requirements.

Figure 2-3 depicts the ZCU103 development board on which the Flexible Payload runs.

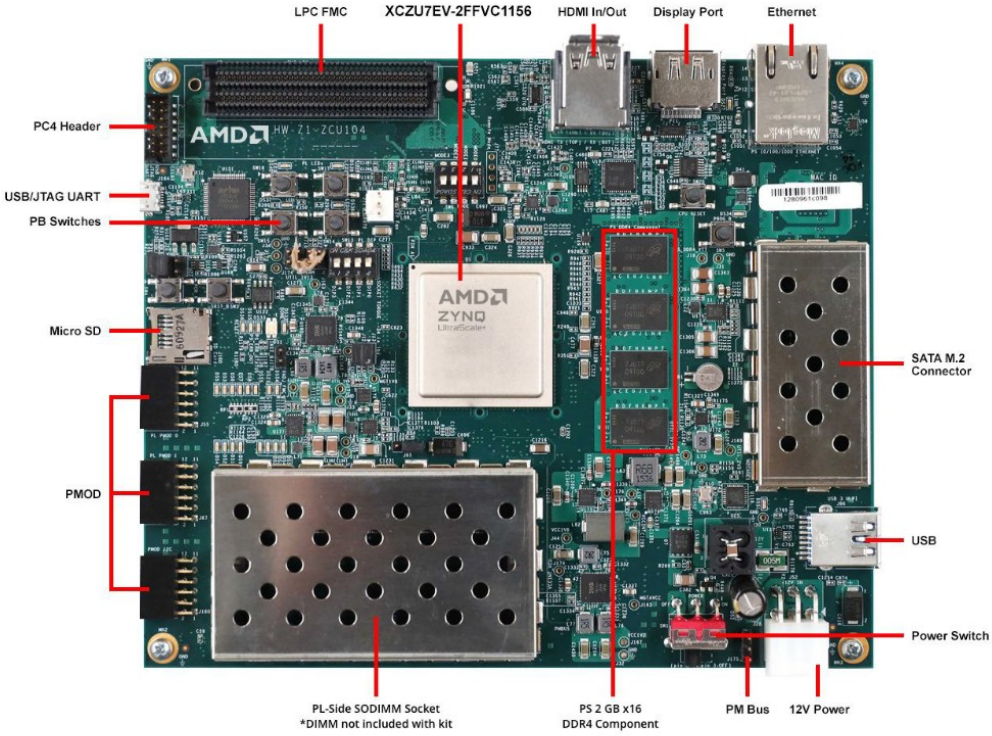

Figure 2-3 - ZCU104 development board and its interfaces.

RF transceiver (Analog Devices AD-936x covering frequencies from DC-to-6GHz).

The AD-936x is an RF transceiver with up to 2 x 2 RF channel (2 Transmitter (Tx) and 2 Receiver (Rx)) with integrated 12-bit Digital to Analog Converters (DACs) and Analog to Digital Converters (ADCs) providing the following features:

- Tx band: 47 MHz to 6.0 GHz
- Rx band: 70 MHz to 6.0 GHz
- Tuneable channel bandwidth: <200 kHz to 56 MHz
- Superior receiver sensitivity with a noise figure of 2 dB at 800 MHz Local Oscillator (LO)
- Rx gain control

Figure 2-4 depicts the FMCOMMS4 board, featuring the AD9361 transceiver chip.

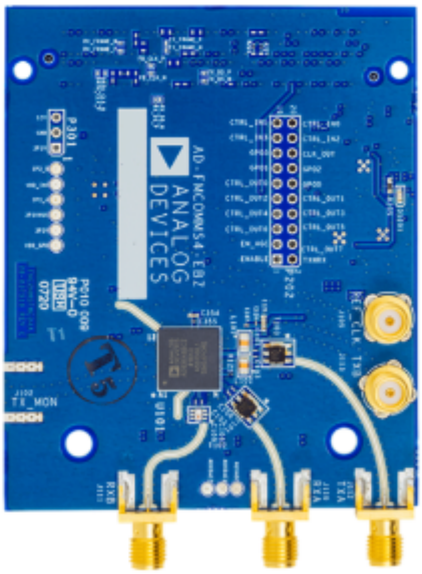

Figure 2-4 – FMCOMMS4 (AD9361 chip) transceiver.

Amarisoft NodeB

The Amarisoft NodeB is a base station platform that supports Long Term Evolution (LTE), 5G, and NB-IoT. It features configurable radio parameters and integrated logging functions for detailed event tracking. Figure 2-5 depicts it, in a workstation computer casing with external RF antennae.

Figure 2-5 - Amarisoft Callbox Classic.

The Amarisoft provides an eNB or a gNB with the following relevant features:

- NB-IoT Rel. 17
- Software Defined Radio (SDR) based RF frontend with a frequency range of 500 MHz to 6.0 GHz
- Support for 200 kHz NB-IoT channels

- Extensive logging of messages at Physical layer and control and user plane layers (e.g., Non-access Stratum (NAS) signalling)

UEs

Demo 1 uses at least two different UEs, each relevant to a different service running onboard the Flexible Payload:

- NB-IoT based UE
- LoRa based UE

Virtual Ground Station

As ISLs are not included in Demo 1, satellites (i.e., payloads) can upload and download data via direct connection to a Virtual Ground Station during their Feeder Link visibility periods.

The ground segment contains a virtualized framework that holds the relevant functions that are tied to the ground. This includes the semantics filter, the store and forward functions and ground-based core network functions.

Virtualized modules are hosted on standard x86_64-based hardware.

2.3.2 Demo Interface Specifications

This Demo testbed proposes a delay-tolerant IoT service by leveraging a 3GPP-compatible S&F-enabled NB-IoT deployment within the satellite system. By integrating a Flexible Payload, which will enable the deployment and orchestration of services across multiple satellites, over a target region. As previously mentioned, the ground-based MANO Orchestrator, which will control the activation and deactivation of services on the satellites, is one of the core components of this testbed; this Orchestrator not only manages infrastructure resources across different domains such as Core Network and Radio Access Network (RAN), but also enables the deployment and management of network services as well.

This Demo scenario also incorporates a low-density constellation of LEO satellites, that provides global coverage to UE devices on the ground. Through NFV and MANO capabilities, coordination of satellite payloads are coordinate the deployment of network services. The Semantic-aware Information Handling (SAIH) filtering will enable optimised data transmission, ensuring that only the most recent content reaches its destination at the right moment, lowering the number of transmissions in this network architecture.

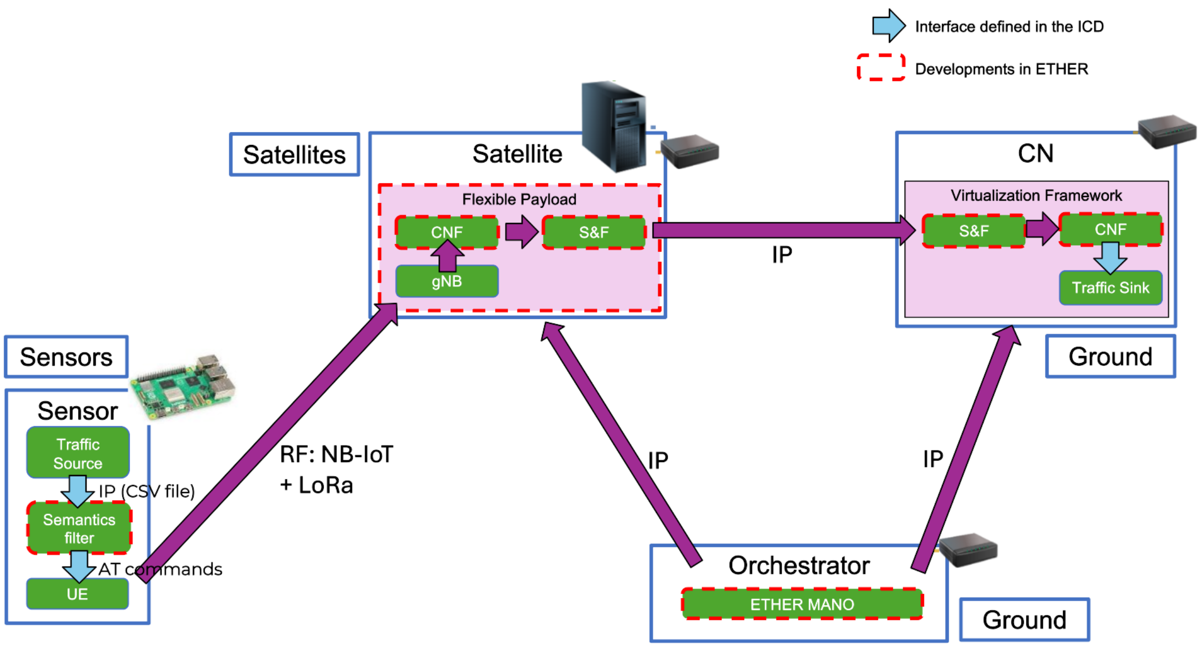

Figure 2-6 - Demo 1 testbed's interfaces.

In Figure 2-6, the Demo 1 testbed is outlined from the point of view of the connections between network components. To clarify how these components communicate externally and internally, below in Table 2-3 and Table 2-4, the interfaces between these Demo 1 testbed components are described, split into Inter-component interfaces (between each component) and Intra-component interfaces (regarding the interfaces between different technologies, services, hardware or software in each component).

Table 2-3: Demo 1 Inter-Component Interfaces.

Termination Point 1	Termination Point 2	Specification	Technology / Protocol
Sensors	Satellite	This interface between the Sensors and the Satellite communicates all relevant UE data towards the Satellite.	NB-IoT/LoRa
Orchestrator	Satellite	This interface allows the orchestrator to control and manage the satellite's operations, including its payload, services, and communication capabilities.	IP
Satellite	Core Network	This interface connects the satellite system to the core network, facilitating data transfer between satellite payloads and terrestrial network elements.	IP
Orchestrator	Core Network	This interface connects the orchestrator to the core network, facilitating the management of the overall network service, including mobility management, resource allocation, and roaming.	IP

Table 2-4: Demo 1 Intra-Component Interfaces.

Demo component	Termination Point 1	Termination Point 2	Specification	Technology / Protocol
Sensors	Traffic Source	Semantics Filter	This interface connects the raw data stream being collected by the sensor to be filtered by the Semantic-aware Information Handling Service.	IP
	Semantics Filter	UE	This interface relays the output of the Semantic-aware Information Handling Service to be transmitted towards the Satellite.	IP
Satellites	gNB	Containerised Network Function (CNF)	This interface connects the onboard gNB to the onboard core network function(s).	IP (S1/N2)
	CNF	Store & Forward	This interface connects the onboard core network functions to the onboard store-and-forward module.	IP
Core Network	Store & Forward	CNF	This interface connects the ground-based store-and-forward module to the ground-based core network functions.	IP
	CNF	Traffic Sink	This interface connects the ground-based core network function to the traffic sink.	IP

2.3.3 Intermediate Demo Components and Specifications

To reach the final Demo 1 testbed, a middle step implementation is done, described in **Error! Reference source not found.**, with a separate environment that connects multiple UE nodes to a local Data Centre. This implementation is twofold: it helps validate the connections between our proposed UE sensor architecture, including the SAIH service, and an entity that is hosted on a local data centre; but also, it validates the usage of the SAIH service for the purposes of energy efficiency and the relevant Demo KPIs.

For this smaller testbed, we containerize our SAIH service into a Docker image on the local server. This Docker image is then deployed through a Kubernetes cluster on the UE device. By using K3s [6], a lighter Kubernetes distribution designed for production workloads in resource-constrained scenarios, we comply with the Demo 1 setup constraints regarding the UEs. After deploying this containerized service on the UE device's Kubernetes cluster, the SAIH is ready to process the sensor data and transmit this data towards the data centre.

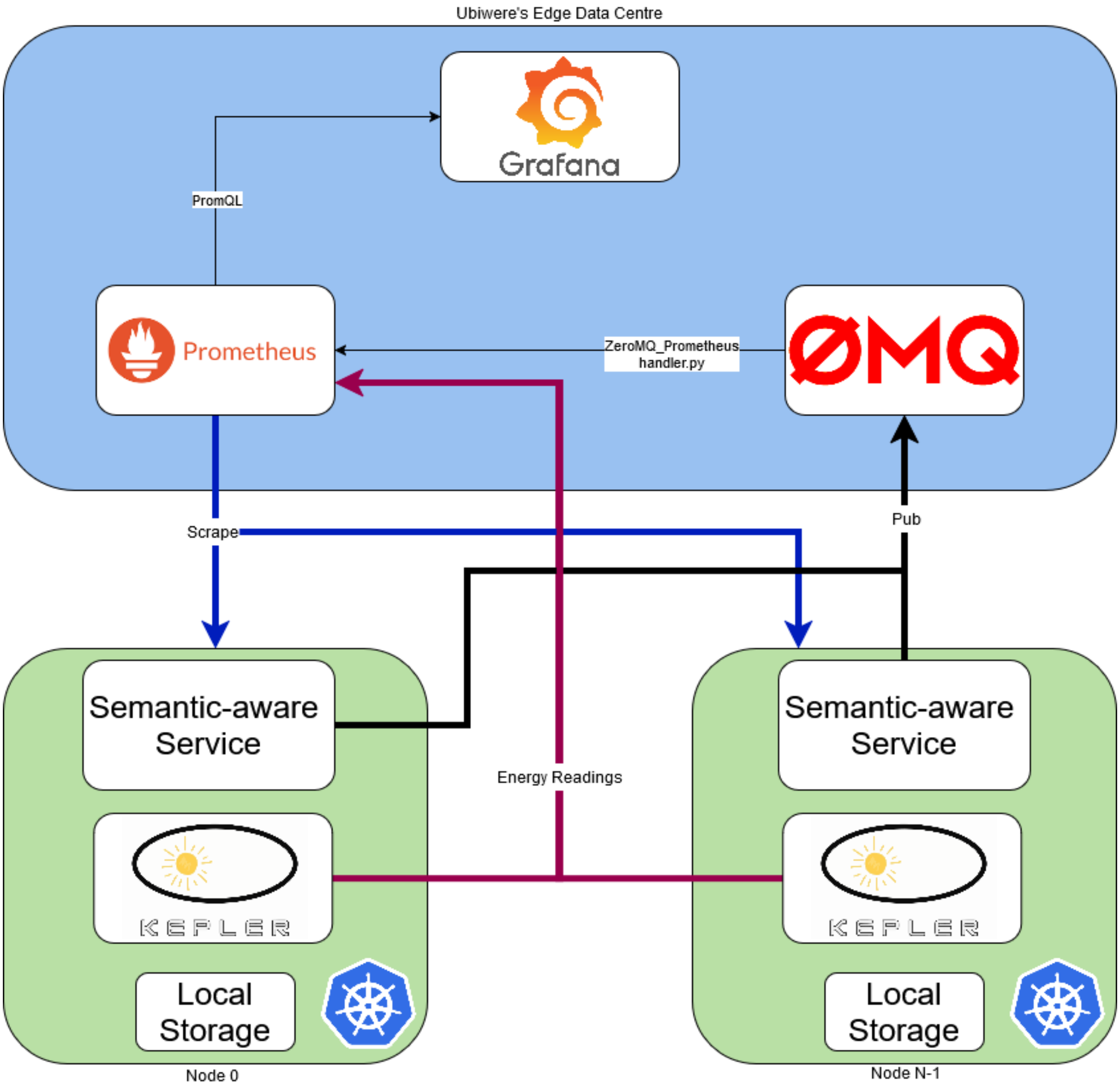

Figure 2-7: Sensor messaging architecture.

This intermediate testbed is implemented using multiple tools that enable the intended purpose of this Demo:

- Prometheus [13] - is a system monitoring tool that collects and stores data such as application performance metrics and system health in a time-series format. Prometheus can also be natively integrated with Kubernetes to automatically discover and scrape metrics from the containerized workloads running on our UEs.

- Kepler (Kubernetes-based Efficient Power Level Exporter) [14] - is a tool that exports a variety of metrics to Prometheus, where the main ones are those related to energy consumption. It does so by reporting on Central Processing Unit (CPU) performance from an application running in a specific Kubernetes cluster, in a Prometheus-friendly format.

- Zero MQ [15] - is an open-source broker-less messaging and communication library, enabling a scalable communication between components and abstracting away the complexity behind these communications through a simple Application Programming Interface (API). In this setup, messaging is done through a publish-subscribe model, in that UEs publish relevant and standardised information (filtered through the SAIH service) towards the subscribed central server.

- Grafana [16] - is a data visualisation and monitoring platform that allows the user to create dashboards that display real-time data from a variety of sources. It integrates with Prometheus to relate information regarding the communications done by ZeroMQ, the energy readings of Kepler, and the application and system information of the UEs.

Following this implementation, the Continuous Integration and Continuous Development (CI/CD) processes of the SAIH service deployment will follow standard Development Operations (DevOps) procedures, using GitLab to host the pipeline files that handle service deployment on the UEs, as well as the remainder of the necessary software tools outlined previously. Logs outlining deployment and update activity of the service will be available for the user on the monitoring service. More detailed, in Figure 2-8, is a flow diagram of how this lifecycle of the service deployment will operate.

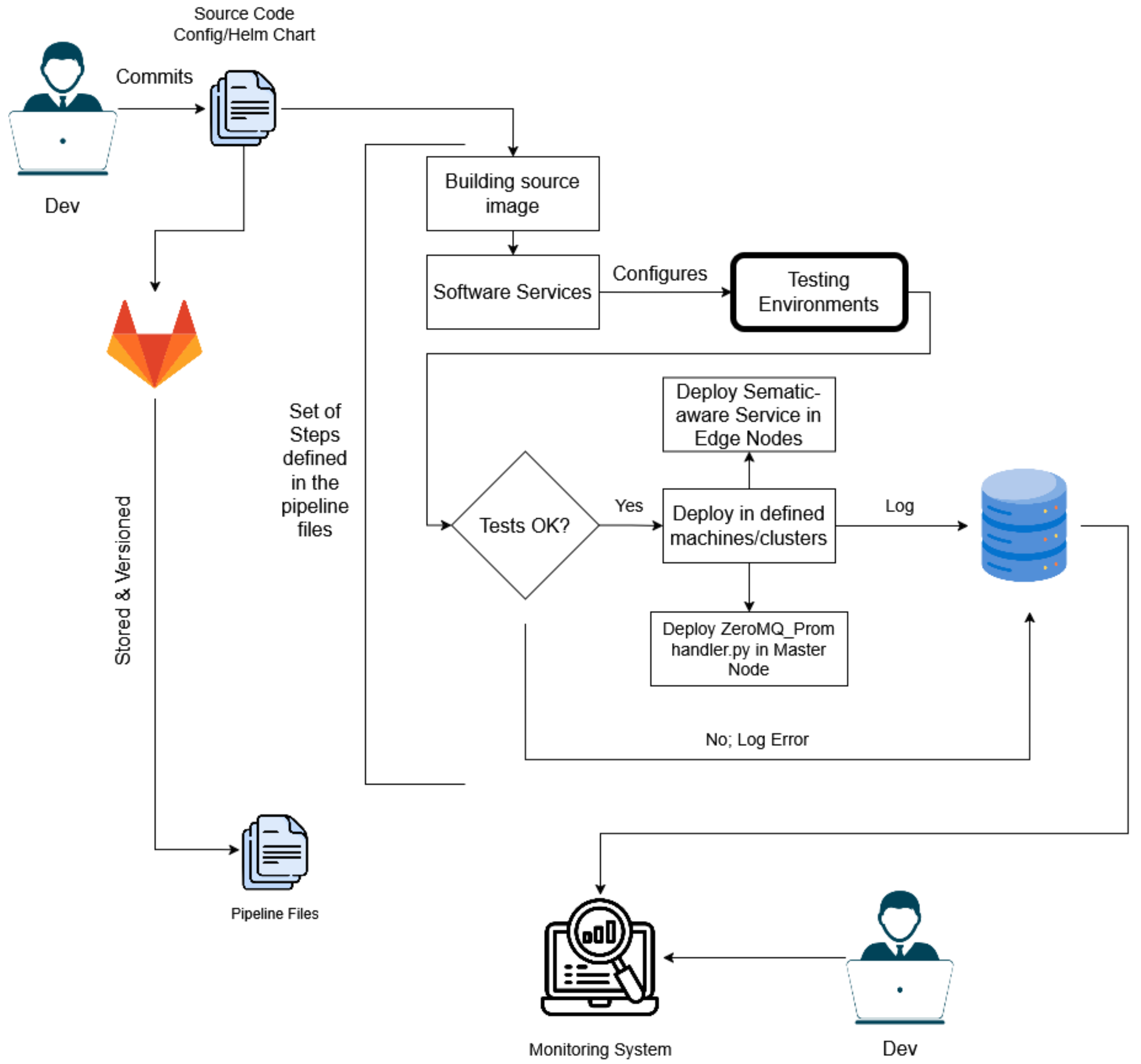

Figure 2-8: CI/CD support to Sensor messaging architecture deployment.

2.4 TESTING METHODOLOGY

2.4.1 Intermediate tests and timeplan

Table 2-5: Intermediate test case details.

Test 1.1	**NB-IoT Service with Store and Forward: Satellite and UE**
Phase	M27 (03/2025)
Description	We verify that UEs can successfully connect using the NB-IoT protocol.
Target UCs	Use Case 1
Relevant Requirements	The requirement IDs evaluated with this test case are extracted from D2.2. Specifically, this test aims to validate: • ETH-REQ-UC1-DT-01 (Intermittent – scheduled Contacts) • ETH-REQ-UC1-DT-02 (Intermittent – opportunistic contacts) • ETH-REQ-UC1-DT-03 (Intermittent – predicted contacts) • ETH-REQ-UC1-DT-04 (Congestion and flow Control) • ETH-REQ-UC1-DT-05 (High latency, low data rate) • ETH-REQ-UC1-DT-06 (Connection discontinuity) • ETH-REQ-UC1-DT-07 (Store and forward) • ETH-REQ-UC1-DT-08 (Traffic model MAR) • ETH-REQ-UC1-DT-09 (Mobility management) • ETH-REQ-UC1-DT-10 (Support for different services)
Procedure/ Steps	1. Setup Core and RAN 2. Setup UE 3. Trigger UE registration 4. Transmit data (MO/MT)

Parameters to be tested KPIs/ Or/And Success Criteria	• Verify UE registration via:
	  ◦ UE status
	  ◦ RAN log
	  ◦ Core network function log (MME)
	• Verify data transmission via:
	  ◦ Packet capture at destination

Network Configuration	The RAN and core network components have direct IP connectivity, via virtual and physical interfaces. The service link is RF based, using NB-IoT.
Testbed	The testbed components for this test encompass the sensor (UE), satellite and CN ground. These parts with their respective interfaces are shown in Figure 2-6, with the hardware components shown in Figure 2-2.

Test 1.2	**UE Context Dissemination Among Satellites**

Phase	M28 (04/2025)
Description	Dissemination of UE context among satellites to validate seamless service with more than one satellite.
Target UCs	Use Case 1
Relevant Requirements	The requirement IDs evaluated with this test case are extracted from D2.2. Specifically, this test aims to validate: • ETH-REQ-UC1-DT-01 (Intermittent – scheduled Contacts) • ETH-REQ-UC1-DT-02 (Intermittent – opportunistic contacts) • ETH-REQ-UC1-DT-03 (Intermittent – predicted contacts) • ETH-REQ-UC1-DT-04 (Congestion and flow Control) • ETH-REQ-UC1-DT-05 (High latency, low data rate) • ETH-REQ-UC1-DT-06 (Connection discontinuity) • ETH-REQ-UC1-DT-07 (Store and forward)
Procedure/ Steps	1. Setup two satellites and Core Network (CN)-Ground 2. Adjust Rx/Tx on satellites, so that satellite 1 appears to provide coverage, while satellite 2 does not 3. Setup UE, trigger network registration 4. UE registers with satellite 1, creating UE context 5. UE goes into ECM-Idle 6. Transmit UE context to CN-Ground 7. Transmit UE context to satellite 2 and inject it into the MME 8. Adjust Rx/Tx on satellites, so that satellite 1 is not providing coverage and satellite 2 does provide coverage 9. Trigger data transmission at the UE, so that it sends a service request to the network and transitions from Evolved Connection Management (ECM)-Idle to ECM-Connected (with satellite 2)

Parameters to be tested KPIs/ Or/And Success Criteria	• Validate UE registration status at satellites via:
	
	  • UE status
	  • RAN log (satellite 1 & 2)
	  • Core network function log (satellite 1 & 2)

Network Configuration	The RAN and core network components have direct IP connectivity, via virtual and physical interfaces. The service link is RF based, using NB-IoT.
Testbed	The testbed components for this test encompass the sensor (UE), satellite and CN ground. These parts with their respective interfaces are shown in Figure 2-6, with the hardware components shown in Figure 2-2.

Test 1.3	**Flexible Payload – MANO Service Orchestration**
Phase	M30 (06/2025)
Description	This test registers satellites in the MANO system by initializing K3s on each satellite, registering their clusters as Virtual Network Function Infrastructure (VNFI) in OSM, and adding TLE data to the Geographic Information System (GIS) module. Validation ensures satellites are recognized as VNFI zones for VNF deployment and their orbits are visualized accurately in the GIS system.
Target UCs	Use Case 1

Test 1.3	**Flexible Payload – MANO Service Orchestration**
Relevant Requirements	The requirement IDs evaluated with this test case are extracted from D2.2. Specifically, this test aims to validate: • ETH-REQ-UC1-FP-02 (Payload FPGA resources availability) • ETH-REQ-UC1-FP-03 (Payload FPGA services deployment)
Procedure/ Steps	1. Start the OSM orchestrator on the Virtual Ground Station. 2. Ensure that each satellite has K3s installed and running. 3. Register each satellite's K3s cluster as a VNFI in the OSM orchestrator. 4. Add the satellite's TLE data to the GIS database in OSM. 5. Validate that OSM recognizes each satellite as an available VNFI zone for VNF deployment and that the GIS tool projects orbits and visualizes satellite positions.
Parameters to be tested KPIs/ Or/And Success Criteria	Success criteria are used. It is verified that: • Satellites are correctly registered in OSM as VNFI zones. • The GIS system accurately projects satellite orbits and visualizes their positions.
Network Configuration	There is a Service-Link connection (IP-based) via IP-Switch between the Virtual Ground Station and satellites.
Testbed	The testbed components for this test encompass the satellite and CN ground (Figure 2-2). From the CN ground the OSM orchestrator communicates via IP (Figure 2-6) to the satellite representative node (Figure 2-3) to deploy services on it.

Test 1.4	**Flexible Payload – Multiple Service Deployment**
Phase	M30 (06/2025)
Description	MANO manages the deployment of services in each satellite. Two different approaches for service design have been used: GNU Radio-based service and direct implemented services (with any auxiliar framework in the middle which means that the application is written in C/python or Hardware Description Language (HDL).
Target UCs	Use Case 1
Relevant Requirements	The requirement IDs evaluated with this test case are extracted from D2.2. Specifically, this test aims to validate: • ETH-REQ-UC1-FP-03 (Payload FPGA services deployment) • ETH-REQ-UC1-FP-04 (Payload FPGA resources sharing) • ETH-REQ-UC1-FP-05 (Payload system performance metrics)
Procedure/ Steps	1. Check K3s (SAT) is reachable from Virtual Ground Station using feeder-link (IP-based) connection 2. Check status of already deployed containers (PODs) from MANO

Test 1.4	**Flexible Payload – Multiple Service Deployment**
	3. Stop all services from Virtual Ground Station and deploy at least two per SAT. Services / applications are in container format. 4. In Virtual Ground Station, acquire generated data from services via the feeder-link (IP-based) 5. Check data and metrics coherency
Parameters to be tested KPIs/ Or/And Success Criteria	Success criteria is used. It is verified that MANO can deploy different PODs (containers/services) at the same time and that the services respond as expected and data obtained in Virtual Ground Station is coherent.
Network Configuration	There is a Feeder-Link connection (IP-based) via IP-Switch between Virtual Ground Station and Satellites. Satellites are also offering services via Service-Link (RF-based) via the transceiver interface (libiio library) which is reachable via IP also from the Virtual Ground Station.
Testbed	The testbed components for this test encompass the sensors (UE), the satellite and CN ground. General architecture can be found at Figure 2-2. From the CN ground the OSM orchestrator communicates via IP to the satellite representative node (Figure 2-3) to deploy services on it. Figure 2-6 shows all the interfaces. The satellite connects via RF link or Amarisoft (NB-IoT) to UEs. While services are working, satellite collects data that is shared to ground station (orchestrator) via IP link.

Test 1.5	**Semantic Aware Information Handling Service on UE Sensor**
Phase	M33 (09/2025)
Description	This test serves to validate the Semantic-Aware Information Handling (SAIH) service on the UE devices, for the purposes of reducing the transmissions required when relaying sensor data.
Target UCs	Use Case 1
Relevant Requirements	The requirement IDs evaluated with this test case are extracted from D2.2. Specifically, this test aims to validate: • ETH-REQ-UC1-SE-01 (Sample processing) • ETH-REQ-UC1-SE-02 (Joint sample and transmit) • ETH-REQ-UC1-SE-03 (Support for E2E information handling beyond the sample and transmit) • ETH-REQ-UC1-SE-04 (Content caching)
Procedure/ Steps	1. Setup the Testbed Hardware 2. Setup the UEs 3. Commit SAIH source code configuration and store & version it on a Gitlab 4. Test SAIH service before deployment 5. Log error if it fails and retry 6. Deploy Kepler and SAIH service on UE 7. Deploy ZeroMQ on Master Node

Test 1.5	**Semantic Aware Information Handling Service on UE Sensor**
	8. Connect UE devices to On-premises server 9. Test SAIH using sensor data 10. Monitor and measure results through Prometheus and Grafana

Parameters to be tested KPIs/ Or/And Success Criteria	• Validate the SAIH service integration in Demo 1 via:
	  • Deploying SAIH service on UE device
	  • Testing SAIH service on real sensor data
	  • Validate energy consumption KPI (>75% higher energy efficiency) through reduction of transmissions and Kepler readings

Network Configuration	The UE and server components are connected through IP, through virtual interfaces.
Testbed	The testbed components for this test encompass the sensors (UE) and the satellite. General architecture can be found at Figure 2-2. The diagram related to this intermediate test is shown in Figure 2-7 and their respective interfaces are outlined in Figure 2-6.

Table 2-6: Final test case details.

Test 1.6	**Integrated Flexible Payload-Enabled Service Provisioning to Semantics Aware and Delay-Tolerant IoT Applications**
Phase	M36 (12/2025)
Description	This test represents the full Demo 1 test case. We include test cases ID_00 – ID_04 integrated on the final hardware, in order to provide the validation for Demo 1.
Target UCs	Use Case 1
Relevant Requirements	The requirement IDs evaluated with this test case are extracted from D2.2. Specifically, this test aims to validate: • ETH-REQ-UC1-FP-01 (Payload FPGA resources management) • ETH-REQ-UC1-FP-02 (Payload FPGA resources availability) • ETH-REQ-UC1-FP-03 (Payload FPGA services deployment) • ETH-REQ-UC1-FP-04 (Payload FPGA resources sharing) • ETH-REQ-UC1-FP-05 (Payload system performance metrics) • ETH-REQ-UC1-DT-01 (Intermittent – scheduled Contacts) • ETH-REQ-UC1-DT-02 (Intermittent – opportunistic contacts) • ETH-REQ-UC1-DT-03 (Intermittent – predicted contacts) • ETH-REQ-UC1-DT-04 (Congestion and flow Control) • ETH-REQ-UC1-DT-05 (High latency, low data rate) • ETH-REQ-UC1-DT-06 (Connection discontinuity) • ETH-REQ-UC1-DT-07 (Store and forward)

Test 1.6	**Integrated Flexible Payload-Enabled Service Provisioning to Semantics Aware and Delay-Tolerant IoT Applications**
	• ETH-REQ-UC1-DT-08 (Traffic model MAR) • ETH-REQ-UC1-DT-09 (Mobility management) • ETH-REQ-UC1-DT-10 (Support for different services) • ETH-REQ-UC1-SE-01 (Sample processing) • ETH-REQ-UC1-SE-02 (Joint sample and transmit) • ETH-REQ-UC1-SE-03 (Support for E2E information handling beyond the sample and transmit) • ETH-REQ-UC1-SE-04 (Content caching)
Procedure/ Steps	1. Setup testbed components according to Figure 2-2, Figure 2-7. 2. Initialize UEs, satellites and ground components 3. Register UEs with the network (via satellite 1) 4. Transmit data MO/MT 5. Disseminate UE context 6. Seamlessly connect UEs to satellite 2 7. Transmit data MO/MT
Parameters to be tested KPIs/ Or/And Success Criteria	• Validate the integrated final Demo 1 test case, being Orchestration, Flexible Payload, NB-IoT service, Context Dissemination and SAIH according to the KPIs listed in ID00 – ID04
Network Configuration	See Table 2-3, Table 2-4
Testbed	The testbed components for this test encompass the sensor (UE), satellite and CN ground as well as Orchestration. These parts with their respective interfaces are shown in Figure 2-6 and Figure 2-7. The hardware components are shown in Figure 2-2.

2.5 RISK AND MITIGATION

Sateliot, as a provider of nanosatellite-based services for LEO, has payloads (and their corresponding Engineering Model (EM)) whose hardware supports the requirements of the flexible payload framework. This hardware, based on FPGAs from the Xilinx family, would a priori allow to deploy real satellites in Demo 1. Unfortunately, and after a more in-depth study, it has been concluded that the software system integrated in the EMs is not prepared for a smooth integration of the flexible payload framework. Specifically, the following points have been identified as risky when planning the integration of the flexible payload:

- Use of a custom operating system (Poky Gatesgarth), but released more than 4 years ago. Some libraries are not compatible with some modern applications or do not contain functionalities that have appeared later.

- Use of a Linux kernel (5.4) that is more than 5 years old, incompatible with some required libraries.

- Impossibility of using the eNB/gNB integrated in the EM with external hardware, given that it does not have external interfaces.
- Impossibility of integrating the eNB/gNB into another system with similar hardware, given that the source code is not available.
- Some EM libraries used for operations and telemetry management are not standard, but instead, are part of ad hoc development customized for the EM operating system. It is not on Sateliot's roadmap to update them to improve support for later kernels.
- The energy reduction can be lower than 75% when the KPI on semantics (such as VAoI) is not stringent. However, in relatively strict VAoI requirements, we observe higher energy saving gains compared to non-semantic-aware techniques, which do not take the timeliness of data into account. This depends on the application and the timeliness required for the transfer of its data versions.

To mitigate possible problems and long integration times that could lead to failure in the deployment of demo 1 architecture, it has been agreed to replace the use of a real EM with a representative development kit. This development kit has been used for the complete development of the flexible payload and does not present any hardware or software incompatibility. The fact that it is representative means that it uses the same hardware configuration as a real satellite. Specifically, a Xilinx Zynq UltraScale+ MPSoC FPGA is used as the base platform, which accumulates several successful missions (flight proven).

To overcome the inconvenience of not having an externally connectable eNB/gNB, it has also been decided to use an Amarisoft emulator to demonstrate the connectivity of the UEs to an EPC deployed in the satellite (development kit) using the flexible payload framework. Other IoT service (such as LoRa) do not require an eNB/gNB and can use the standard RF datapath provided by the development kit via an Analog Devices transceiver.

As a summary, the following table shows the previous analysed risks and the mitigation plans:

Table 2-7: Identified demo-related risks and mitigation plans for Demo 1.

ID	Risk	Likelihood (H/M/L)	Impact (H/M/L)	Mitigation Plan
1.1	**Sateliot EM OS integrates non-compatible libraries for the Flexible Payload**	H	L	Replace EM for an Evaluation Board (ZCU104) as a representative Hardware
1.2	**Sateliot EM OS kernel is non-compatible for the Flexible Payload**	H	L	Replace EM for an Evaluation Board (ZCU104) as a representative Hardware
1.3	**Sateliot EM eNB/gNB has not integrable interfaces.**	M	L	Replace EM for an Evaluation Board (ZCU104) as a representative Hardware
1.4	**Sateliot EM eNB/gNB software is not runnable in other Linux environments (source code unavailable).**	H	L	Replace EM for an Evaluation Board (ZCU104) as a representative Hardware

ID	Risk	Likelihood (H/M/L)	Impact (H/M/L)	Mitigation Plan
1.5	**EM proprietary telemetry/operations protocols**	H	L	Replace EM for an Evaluation Board (ZCU104) as a representative Hardware
1.6	**The energy-related KPI target (75 % energy reduction) cannot be met.**	M	M	This can happen when the requirement for the KPI (in our case VAoI) is not stringent, but in cases that we have relatively strict semantics requirements we do not see that.

3. DEMO 2: UNIFIED RAN FOR DIRECT HANDHELD DEVICE ACCESS

3.1 USE CASE DESCRIPTION

The second use case, titled "Unified RAN for direct handheld device access", is illustrated below in Figure 3-1.

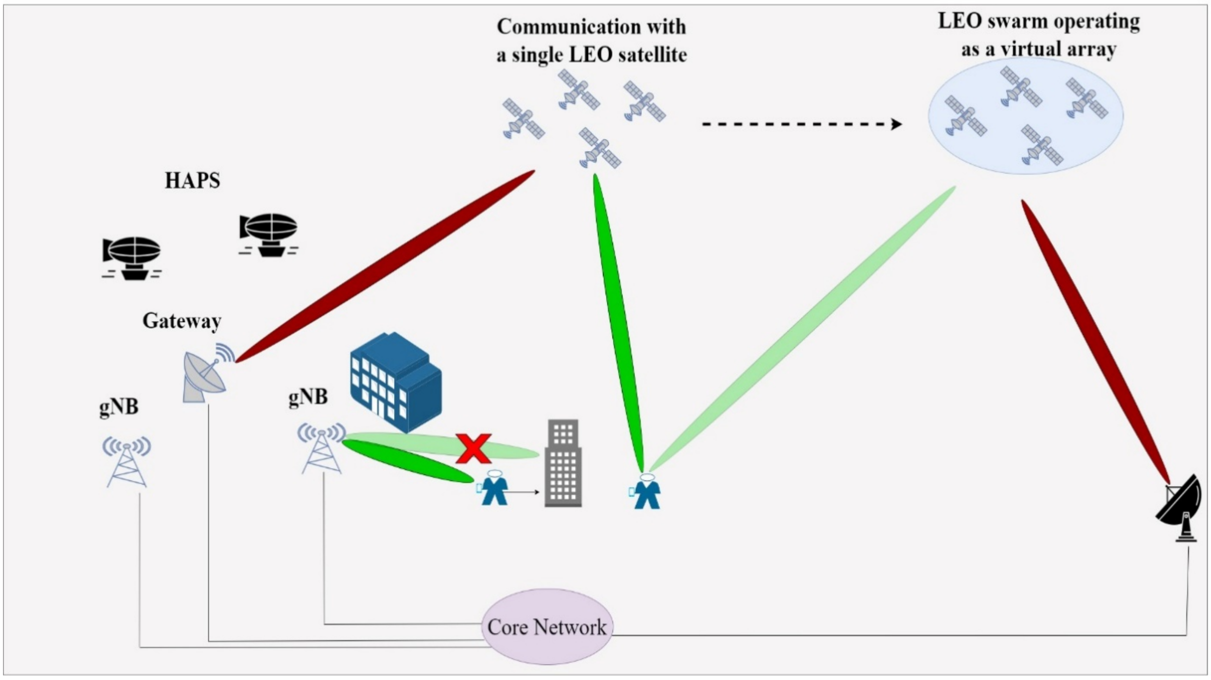

Figure 3-1: Use case 2: Unified RAN for direct handheld device access at the Ka band

We assume that a mobile handheld device is initially connected to a terrestrial site (gNB). As the device is moving, there can be a number of reasons that create the need for the migration of the communication to an NTN gNB, such as a LEO-satellite one. These can for instance be:

1. Deteriorating quality of the signal coming from the terrestrial gNB due to some obstacles or because the user is moving to a remote area, far away from the terrestrial gNB.

2. Saturated traffic conditions of the terrestrial gNB.

3. Low traffic conditions of the terrestrial gNB, for instance at night. In this case, the terrestrial gNB can be switched off, for the reduction of the energy consumption, and have its traffic migrated to an NTN gNB.

Based on reported measurements from the handheld device about the signal strength from other terrestrial sites and from non-terrestrial platforms, a handover process will be triggered through intelligent algorithms (described in ETHER D3.1 [9] and D3.2) that target the maximisation of the energy efficiency subject to constraints related to time availability, flow conservation, power, and capacity.

3.2 DEMO PURPOSE

ETHER Demo 2 well aligns with the ETHER Use Case 2. Its main aim is to showcase that by leveraging intelligent decision algorithm that take into account various parameters, such signal level strength, traffic load in gNBs, and time availability of NTN candidate nodes for the handover, a handover decision from a terrestrial gNB to a LEO-satellite based one can be initiated that is notably more energy efficient than state-of-the-art algorithms that primarily consider the signal strength.

Furthermore, the particular handover process is going to consider existing 3GPP standardized interfaces and protocols. Hence, it will be 3GPP compliant, which is essential for the ETHER vision of unifying the terrestrial and non-terrestrial works under a common interface/protocol framework. Hence, the impact of such a demo activity will be profound in the community.

3.2.1 Key performance indicators to be tested

The targeted key performance indicators of use case 2 are the following:

- 100% global outdoor coverage;
- 99.99999% service continuity (By service continuity we mean the seamless migration of services when switching across different radio access technologies and networks, while the minimum level of service is maintained. Such a seamless migration of services can be achieved by the joint scheduling of communication and computing resources, which would make the process transparent to the users [17]. To achieve the envisioned high value of service continuity, intelligent algorithms will be leveraged that proactively decide about the joint scheduling of communication and computing resources for a particular service, as the network evolves);
- 99.99999% service reliability, i.e., the percentage of time that the desired level of service is maintained;
- 70% more energy-efficient vertical handover (switching from TN to NTN) compared to the SotA.

Table 3-1: Demo 2 KPIs.

Identifier	Requirement	Description
ETH-KPI-UC2-01	Coverage	Provide 100% global outdoor coverage
ETH-KPI-UC2-02	Service continuity	99.99999%
ETH-KPI-UC2-03	Service reliability	99.99999%
ETH-KPI-UC2-04	Energy efficient handover	70% more efficient than SotA

3.3 DEMO SETUP

This demo aims to explore the feasibility and performance of 5G handover mechanisms, particularly within a hybrid Terrestrial and Non-Terrestrial Network (TN-NTN) environment. The demo considers F1-based CU/DU split gNBs, with multiple in-space and on-ground DUs, a single on-ground gNB-CU with 5G Core Network (5GC) connectivity, and a single on-ground UE, as depicted in Figure 3-2.

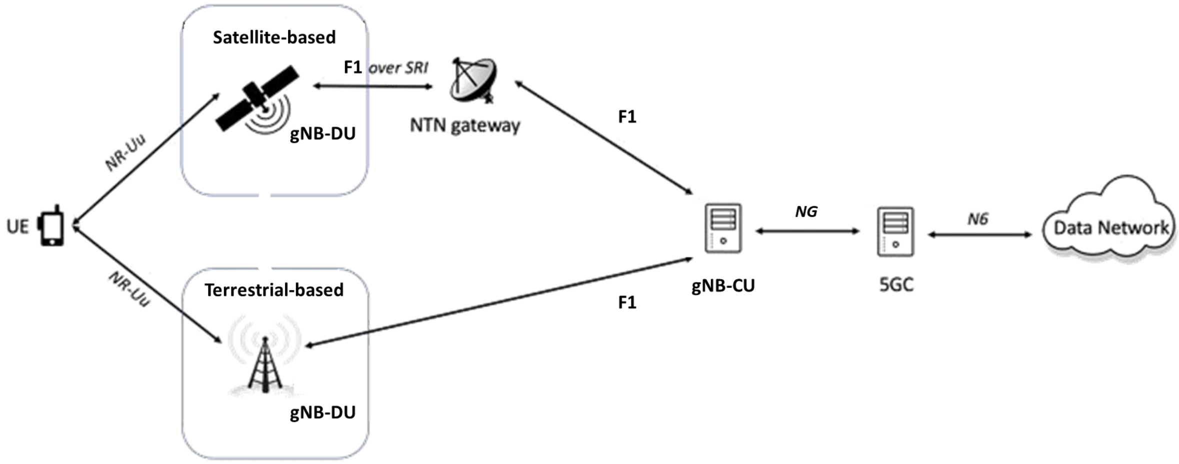

Figure 3-2: High-level system diagram.

In order to showcase the energy efficiency gains of the ETHER handover algorithms, a low traffic scenario is selected for this demonstration. In this setup, the UE is located in a remote area and receives both the terrestrial and satellite-based gNB-DU signals with sufficient quality (higher SINR than a threshold value based on the required UE service type). Although the signal received by the terrestrial gNB-DU is higher than the one from the satellite, the proposed handover algorithm performs an energy-efficient handover to save energy by switching off the terrestrial gNB-DU. The proposed solution is to be compared, in terms of energy efficiency, to the default user association criterion where the UE is connected to the gNB from which it receives the highest signal quality.

To that end, the evaluated KPIs include the seamless service provision for the UE, without disruptions during the whole demo duration including the HO process. This will be showcased by guaranteeing that the signal quality of the UE, as well as its bit/packet error rate, will not deteriorate more than a threshold (set by the particular service that needs to be supported) during the whole demo duration. In order to demonstrate the targeted KPI values of coverage, service continuity and service reliability, extensive evaluation will be realized through simulations in a variety of setups and scenarios under the framework of WP3. Regarding energy efficiency (EE), it will be measured by employing power models developed within WP4. Specifically, the following equation will be used, i.e., the data rate of the UE divided by the power consumption of the BSs of the network.

$$\mathrm{EE} = \frac{Data\ rate}{Power\ Consumption}\left[\frac{bps}{W}\right] = \left[\frac{bits}{J}\right] \quad (3.1)$$

The power consumption model of the gNBs consists of an idle power part and a load-dependent one. Given that the satellite is always active, the deactivation of the terrestrial gNB is expected to provide important energy savings during low traffic hours. Intuitively, this is expected by considering that the total power consumption of commercial LEO gNBs, such as the ones of Starlink, is in the order of 150 W [18], whereas the total power consumption of macro gNBs is in the order of KW [19]. The amount of energy saving will be calculated offline using the power models described in D2.3 [20] and D2.4 [21]. Finally, the energy efficiency gain will be calculated based on the following formula:

$$\text{EE Gain} = \frac{System\ EE\ (terrestrial\ DU\ off) - System\ EE\ (terrestrial\ DU\ on)}{System\ EE\ (terrestrial\ DU\ on)} \quad (3.2)$$

3.3.1 Demo Components and Specifications

This demonstration integrates several in-lab-ready components: SDR-based OpenAirInterface for 5G Radio Access Network (RAN) featuring F1-based intra-gNB-CU handover, Free5GC for 5GC, and SnT's 6GSPACE Lab multi-orbit channel emulation. **Error! Reference source not found.** shows an example of gNB and UE connectivity via SnT 6GSPACE Lab channel emulator used as a reference starting point to the demo. The channel emulator has the following key features: multiorbital and customizable configurations, support for up to 8 independent channels, delay variations up to 330 ms with 1 ns resolution, and Doppler and Doppler rate simulation (±34 ppm / 0.5ppm/s).

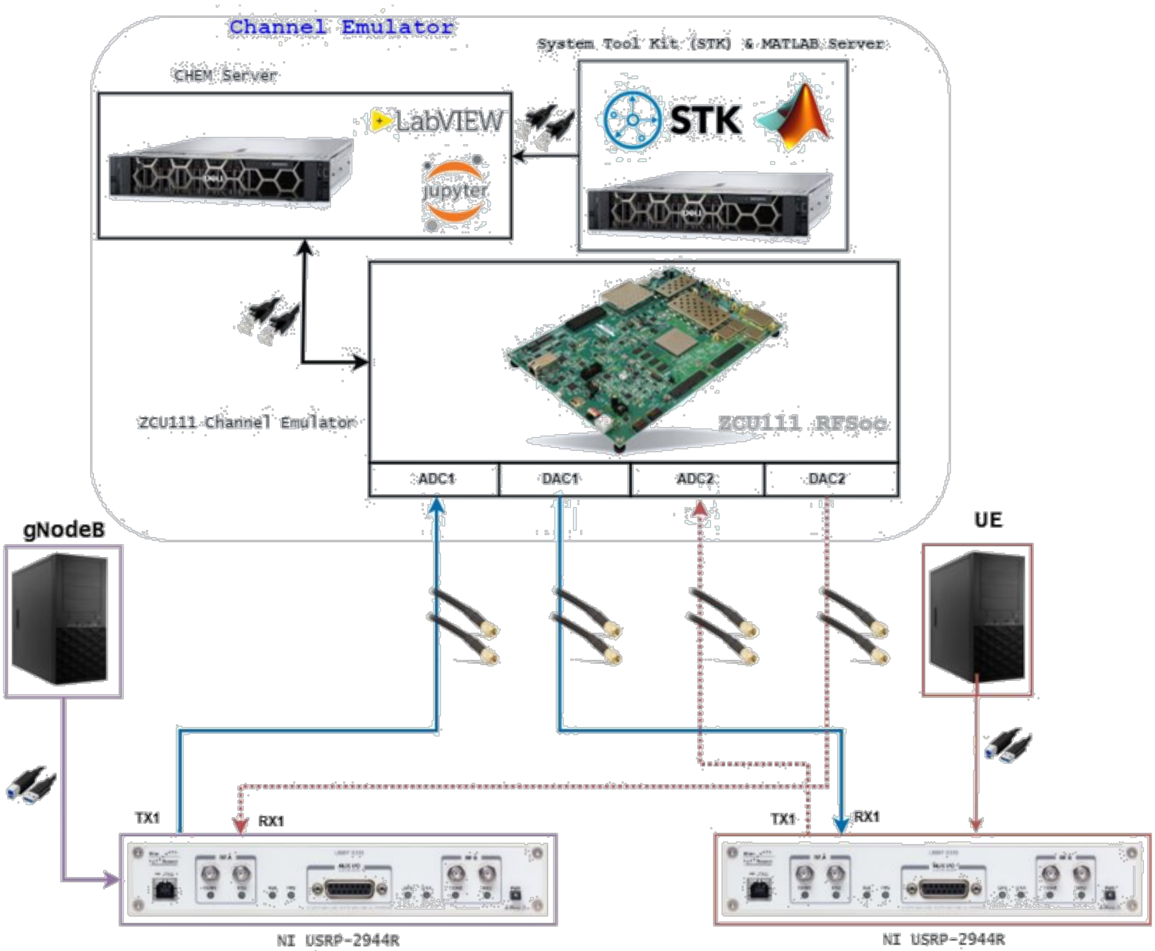

Figure 3-3: Example of gNB and UE connectivity via SnT 6GSPACE Lab channel emulator.

The only option supported in handover mechanisms by OpenAirInterface is the F1-based intra-gNB-CU handover. However, this type of handover already features the same basic steps as any other handover (see **Error! Reference source not found.**):

- Establishing a new radio link before releasing the old one

- Ensuring minimal packet loss during transition
- Utilizing Radio Resource Control (RRC) procedures for seamless mobility

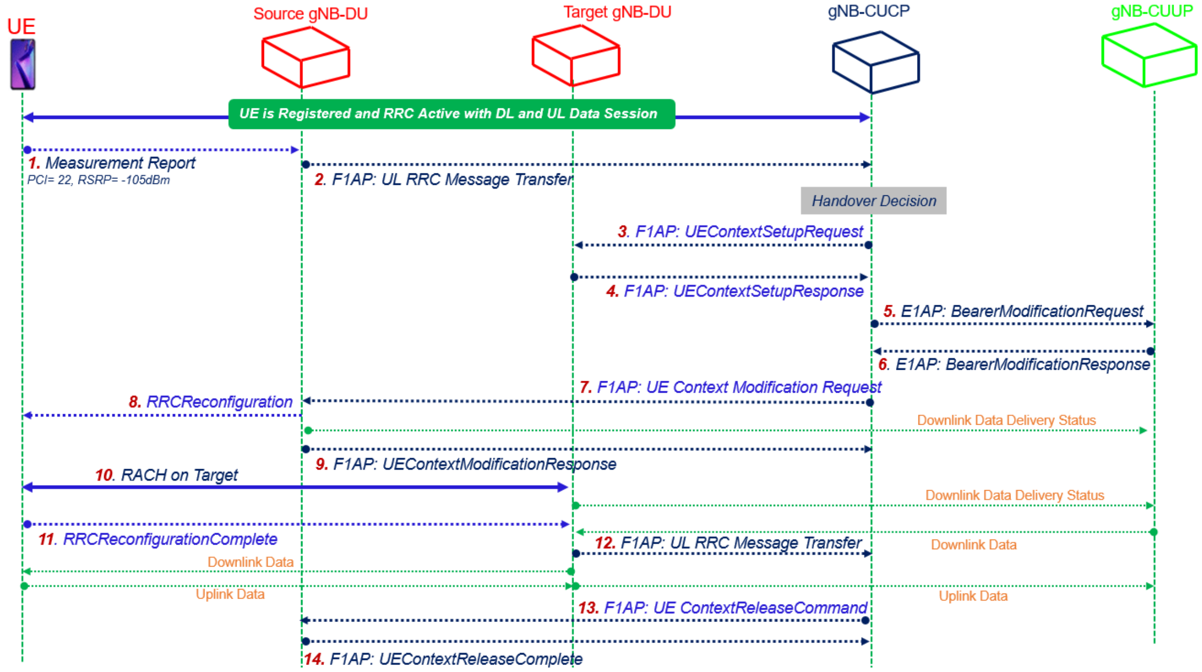

Figure 3-4: F1-based Intra-gNB-CU handover procedure.

Error! Reference source not found. depicts the demo setup comprising of the following key components:

a. **Data Network (DN)**

- The Data Network (DN) is the endpoint where user data is ultimately transmitted.
- Connected to the 5G Core Network (5GC) via the N6 interface.

b. **5G Core Network (5GC)**

- The 5GC is the main control unit responsible for managing network connectivity, authentication, and mobility.
- It connects to the gNB-CU (Centralized Unit) using the NG interface.

c. **gNB-CU (Centralized Unit)**

- The gNB-CU is responsible for controlling multiple Distributed Units (DUs) and managing the RAN (Radio Access Network).
- It connects to two gNB-DUs (Distributed Units) via the F1 interface:
  - Terrestrial-based gNB-DU (bottom)
  - Satellite-based gNB-DU (top)

d. gNB-DU (Distributed Units)

- These two distributed units (DUs) handle the transmission of signals to and from the User Equipment (UE) via the NR-Uu interface.
- Two types of DUs are illustrated in this setup:
  - **Terrestrial-based gNB-DU:** Connected to the CU via F1 and directly communicating with UE.
  - **Satellite-based gNB-DU:** Connected via F1 through an NTN Gateway and handling transmissions in a non-terrestrial network scenario.

e. RF/IF Interface

- The RF/IF (Radio Frequency/Intermediate Frequency) interface is used for real-time signal processing and transmission.
- A hardware module (Universal Software Radio Peripheral (USRP) or FPGA-based device) is utilized to handle RF signals and convert them for digital processing.

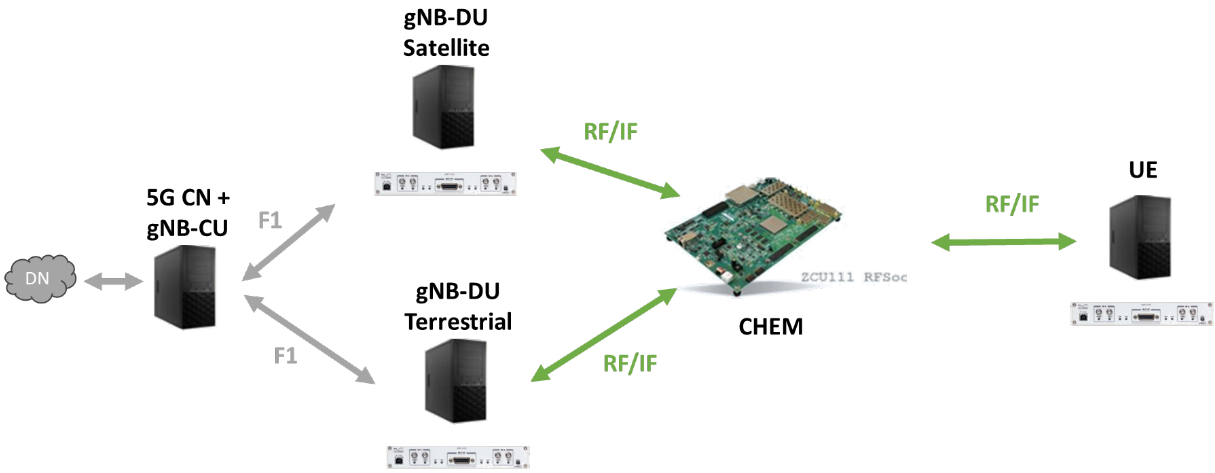

Figure 3-5: Demo setup.

Table 3-2: Demo 2 Bill of materials.

Component	Provider	No. of elements	Availability?
Name of the Component	*Partner that provides it*	*No. of components needed*	*Yes or No*
Workstation (w/ Intel Core i9)	University of Luxembourg	4	Yes
Software Defined Radios (USRP B210 or X310)	University of Luxembourg	3	Yes
Zynq UltraScale+ RFSoC ZCU111 Evaluation Kit	University of Luxembourg	1	Yes

3.4 TESTING METHODOLOGY

3.4.1 Intermediate tests and timeplan

Table 3-3: Intermediate test case details.

Test 2.1	**F1HO_Inter-2DU_Sim**
Phase	Completed
Description	This is the initial handover test that will run entirely on software to check whether the handover algorithm is implemented correctly within the OpenAirInterface5G RAN and CN software stacks.
Target UCs	UC2: ETHER Unified RAN for Direct Handheld Device Access
Relevant Requirements	ETH-REQ-UC2-FN-01 (Migrate TN to NTN) ETH-REQ-UC2-FN-02 (Vertical handover) ETH-REQ-UC2-NF-01 (Vertical handover)
Procedure/ Steps	**a.** Start the 5GC **b.** Start the gNB-Central Unit **c.** Start the gNB-Distributed Unit 0 simulator **d.** Start the UE simulator and observe the connection on 5GC, gNB-CU, gNB-DU0 and UE logs **e.** Start the gNB-Distributed Unit 1 simulator **f.** Execute the handover and observe the changes in connection on 5GC, gNB-CU, gNB-DU0, UE and gNB-DU1 logs
Parameters to be tested KPIs/ Or/And Success Criteria	Success Criteria: When the handover is triggered, connection at the initial gNB-Distributed Unit stops, and connection at the alternate gNB-Distributed Unit begins.
Network Configuration	Private 5G network that follows the setup given in **Error! Reference source not found.**.
Testbed	This intermediate testbed will involve the simulated versions of the two gNB-DUs and the UE, according to **Error! Reference source not found.**, without the involvement of the channel emulator, for checking the correct implementation of the handover algorithm within the OpenAirInterface5G RAN and CN software stacks

Test 2.2	**F1HO_Inter-2DU_SDR**
Phase	M27 (03/2025)
Description	This is the second step in handover test campaign, which introduces real 5G NR waveform instead of all-software TRX chain.
Target UCs	UC2: ETHER Unified RAN for Direct Handheld Device Access

Test 2.2	**F1HO_Inter-2DU_SDR**
Relevant Requirements	ETH-REQ-UC2-FN-01 (Migrate TN to NTN) ETH-REQ-UC2-FN-02 (Vertical handover) ETH-REQ-UC2-NF-01 (Vertical handover)
Procedure/ Steps	**a.** Start the 5GC **b.** Start the gNB-Central Unit **c.** Start the gNB-Distributed Unit 0 SDR **d.** Start the UE SDR and observe the connection on 5GC, gNB-CU, gNB-DU0 and UE logs **e.** Start the gNB-Distributed Unit 1 SDR **f.** Execute the handover and observe the changes in connection on 5GC, gNB-CU, gNB-DU0, UE and gNB-DU1 logs
Parameters to be tested KPIs/ Or/And Success Criteria	Success Criteria: When the handover is triggered, connection at the initial gNB-Distributed Unit stops, and connection at the alternate gNB-Distributed Unit begins.
Network Configuration	Private 5G network that follows the setup given in **Error! Reference source not found.**.
Testbed	This intermediate testbed will involve the SDR versions of the two gNB-DUs and the UE, according to **Error! Reference source not found.**, without the involvement of the channel emulator, introducing real 5G NR waveform instead of all-software TRX chain

Test 2.3	**F1HO_Inter-2DU_ChEm**
Phase	M28 (04/2025)
Description	This is the final step in handover test campaign, where the NTN channel is emulated instead of simulated over the OpenAirInterface.
Target UCs	UC2: ETHER Unified RAN for Direct Handheld Device Access
Relevant Requirements	ETH-REQ-UC2-FN-01 (Migrate TN to NTN) ETH-REQ-UC2-FN-02 (Vertical handover) ETH-REQ-UC2-NF-01 (Vertical handover)
Procedure/ Steps	**a.** Start the 5GC **b.** Start the gNB-Central Unit **c.** Start the gNB-Distributed Unit 0 SDR **d.** Start the Channel Emulator **e.** Start the UE SDR and observe the connection on 5GC, gNB-CU, gNB-DU0 and UE logs **f.** Start the gNB-Distributed Unit 1 SDR **g.** Execute the handover and observe the changes in connection on 5GC, gNB-CU, gNB-DU0, UE and gNB-DU1 logs
Parameters to be tested KPIs/	Success Criteria: When the handover is triggered, connection at the initial gNB-Distributed Unit stops, and connection at the alternate gNB-Distributed Unit begins.

Test 2.3	**F1HO_Inter-2DU_ChEm**
Or/And Success Criteria	
Network Configuration	Private 5G network that follows the setup given in **Error! Reference source not found.**.
Testbed	This intermediate testbed will incorporate to the T2.2 version the channel emulator, according to **Error! Reference source not found.**

Test 2.4	**Integration of the HO algorithm with the emulator**
Phase	M32 (08/2025)
Description	Interfacing of the AUTH algorithms with the testbed
Target UCs	UC2: ETHER Unified RAN for Direct Handheld Device Access
Relevant Requirements	ETH-REQ-UC2-FN-01 (Migrate TN to NTN) ETH-REQ-UC2-FN-02 (Vertical handover) ETH-REQ-UC2-NF-01 (Vertical handover)
Procedure/ Steps	First, the data traffic and signal quality will be provided as inputs from the OpenAirInterface platform to the algorithm. Based on these inputs, the algorithm will determine whether a handover (HO) is necessary and, if so, decide where the handover should occur. The decision will be output in a format such as JSON or a similar script format.
Parameters to be tested KPIs/ Or/And Success Criteria	Success Criteria: When the handover is triggered, connection at the initial gNB-Distributed Unit stops, and connection at the alternate gNB-Distributed Unit begins based on the energy-efficient ETHER vertical HO algorithm.
Network Configuration	Private 5G network that follows the setup given in **Error! Reference source not found.**.
Testbed	This intermediate will involve the integration of the OpenAirInterface protocol stack with the intelligent handover algorithm provided by AUTH, which will be tested on the demo testbed setup of **Error! Reference source not found.**

Table 3-4: Final test case details.

Test 2.5	**Final Demo 2 demonstration**
Phase	M36 (12/2025)
Description	Final Demo 2 execution
Target UCs	UC2: ETHER Unified RAN for Direct Handheld Device Access
Relevant Requirements	ETH-REQ-UC2-FN-01 (Migrate TN to NTN) ETH-REQ-UC2-FN-02 (Vertical handover) ETH-REQ-UC2-NF-01 (Vertical handover)

Test 2.5	**Final Demo 2 demonstration**
Procedure/ Steps	**a.** Start the 5GC **b.** Start the gNB-Central Unit **c.** Start the gNB-Distributed Unit 0 SDR **d.** Start the Channel Emulator **e.** Start the UE SDR and observe the connection on 5GC, gNB-CU, gNB-DU0 and UE logs **f.** Start the gNB-Distributed Unit 1 SDR **g.** AUTH's algorithm will begin receiving real time information from OAI (for instance SINR), using an interfacing technology such as exposing TCP/UDP sockets, or by using ZeroMQ or other types of APIs, such as Representational State Transfer (REST)/gRPC. **h.** When the algorithm determines that a HO should take place, it will instruct OAI through the interface used in Step g) to perform it. **i.** Execute the handover and observe the changes in connection on 5GC, gNB-CU, gNB-DU0, UE and gNB-DU1 logs
Parameters to be tested KPIs/ Or/And Success Criteria	Success Criteria: When the handover is triggered, connection at the initial gNB-Distributed Unit stops, and connection at the alternate gNB-Distributed Unit begins based on the energy-efficient ETHER vertical HO algorithm.
Network Configuration	Private 5G network that follows the setup given in **Error! Reference source not found.**.
Testbed	This involves the final ETHER Demo 2 execution related to the testbed setup of **Error! Reference source not found.**.

3.5 RISK AND MITIGATION

The identified related risks and mitigation plans are summarized in Table 3-5.

1. Initially, we chose to implement the N2-based handover of Open Air Interface, that was supposed to have been concluded in the 3rd quarter of 2024. However, the process for its finalization has stopped. This creates a big risk for Demo 2, since it is not certain that the N2-handover functionality would be concluded by the completion of ETHER, let alone before.

2. Due to this and as a mitigation measure, we took the decision among the consortium to proceed with the F1-based handover that involves the CU-DU functional split, as it was presented, which is already embedded in Open Air Interface. This results in a low impact since in Demo 2 the prime aim is to show the basic vertical handover functionality between a terrestrial- and a LEO satellite-based gNB. This is not affected by whether a functional CU-DU split is considered or not. At this point we would like to mention that provided that the N2-handover functionality in the Open Air Interface is completed quite in advance before the end of 2025, that ETHER concludes, our aim is to also demonstrate the N2-based handover and compare with the F1-based one.

3. Interfacing means should be developed for the intelligent handover decision algorithms, developed by AUTH, to take as inputs the measurements from the Open Air Interface testbed of UL, such as the ones related to the signal strength, traffic conditions of the GNB-Dus, and time position of the LEO satellite, which would determine its time

availability. Subsequently, after the handover decision is taken by leveraging those algorithms, the handover decision, which would be the output of those algorithms should be fed back to the Open Air Interface testbed of UL and executed in an automated fashion. However, to the best of our knowledge so far, the Open Air Interface platform currently allows only a manual execution of a handover process between two gNBs. Hence, our aim is to examine whether we can automate the process. If we realize that due to limitations in the functionality of the platform this is not realizable, we will be performing the handover process manually, based on the decision engine of AUTH. The impact from this, if it materializes, is medium since on one hand we will still demonstrate the intelligent energy efficient handover, but, on the other hand, not in an automated fashion, which would go against the zero-touch orchestration that is brought by ETHER as an important innovation. However, regarding this we need to take into account that the underlying problem would be current limitations of the available hardware, not the capabilities of our developed innovations. In the future, we would likely expect advanced capabilities of the Open Air Interface platform that would potentially allow automated decisions.

Table 3-5: Identified demo-related risks and mitigation plans for Demo 2.

ID	Risk	Likelihood (H/M/L)	Impact (H/M/L)	Mitigation Plan
2.1	N2-based handover of Open Air Interface is not ready on time.	Materialized	L	Proceed with the F1-based handover that involves the CU-DU functional split, as it was presented, which is already embedded in Open Air Interface
2.2	Not able to automate the vertical handover process in the Open Air Interface testbed, based on the handover decisions	M	M	We will leverage the already embedded manual handover functionality of the Open Air Interface platform, to perform the handover, based on the handover decisions

4. DEMO 3: AIR-SPACE SAFETY CRITICAL OPERATIONS

4.1 USE CASE DESCRIPTION

Aircrafts require different communications technologies along their travel journeys from one airport to another, as shown in Figure 4-1. In this use case, the vision of ETHER hybrid multi-layered network will be demonstrated, comprising of terrestrial, aerial (HAPS), and space-based components in serving aircraft departing from one airport and landing in another airport, possibly passing through the oceanic airspace. The operational requirements for the various technical options of aircraft communication systems in the various phases of flight are assessed against a set of values for some parameters. These are denoted by the term “Required Communication Performance (RCP) type”, which is quantified in terms of communication transmission delay, data rate, continuity, availability, integrity. These will be primarily used to evaluate the RCP in the provision of air traffic services.

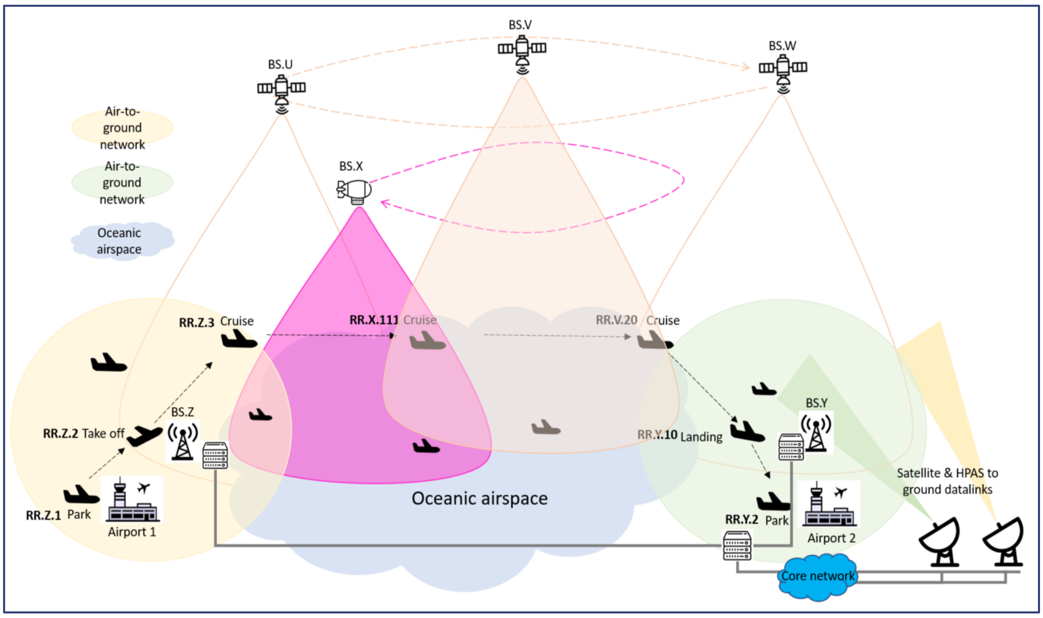

Figure 4-1: Airspace safety critical operations.

This intended use case aims to provide a seamless high-resilient aeronautical data network for safety-critical services to support the following objectives:

- Meet the RCP of the different aircraft flight phases;
- Provide guaranteed E2E aircraft communication services subject to optimal network performance and efficient resource allocation;
- Provision resources and/or migrate service data in network edges that supports advanced avionics services.

4.2 DEMO PURPOSE

The main purpose of the Demo is to develop an integrated 3D aeronautical data network that can support aircraft critical airspace communications. The results shall be driven towards developing SDN-enabled multi-layer communication architecture that can support the existing and emerging avionic services and applications, and, particularly, the required performance for aircraft communications.

4.2.1 Key performance indicators to be tested

Aircrafts are offered different services during their flight phases, which also differ according to the type of aircraft, whether crewed or single pilot. Standard data communication, standard surveillance communication, and strict data communication services require different communication performance, which is measured against multiple metrics [22]. To that end, in this demo we focus on evaluating the following metrics:

- 100% global outdoor **network coverage**, similar to Demo 1 and 2.
- 99.99999% **service continuity.** By service continuity we mean the seamless service migration of the aircraft when switching across different radio access technologies and networks, while the minimum level of service is maintained. Efficient traffic forecasting mechanisms shall be applied simultaneously with efficient resource allocation algorithms to proactively decide about the joint scheduling of communication and computing resources for a particular service during the aircraft flight time.
- 99.99999% **service reliability**, i.e., the percentage of time that the desired level of service of the aircraft is maintained.
- Performance **integrity** (**I**) is defined as the probability of not having undetected errors related to the avionics, navigation, and communication systems. In this demo, performance integrity is directly linked to the Packet Error Rate (PER), with a target range set between 10^{-4} and 10^{-6}.
- > 80% more energy efficient resource allocation than SotA. This will be measured by employing energy-efficient resource allocation algorithms (developed within WP4) which will optimize the BS selection and traffic routing of the aircraft as well as efficiently place any required any-type Network Functions (xNFs), while ensuring that the required QoS of the aircraft is guaranteed. The EE gains will be measured by Eq. (4.1) while comparing the proposed approach with the default resource allocation criteria, i.e., user association based on the highest received signal strength, lowest delay route selection and location-aware xNF placement subject to the power, capacity and QoS requirement constraints.

$$\text{EE Gain} = \frac{System\ EE\ (proposed\ \ alloc.) - System\ EE\ (SotA\ alloc.)}{System\ EE\ (SotA\ alloc.)} \tag{4.1}$$

The majority of the above KPIs are related to the seamless service provision for the aircraft, without disruptions during the service connectivity, while maintaining seamless handover and data migration. To this end, in order to demonstrate the targeted KPI values of coverage, service continuity and service reliability, extensive evaluation has been realized through simulations in a variety of setups and scenarios under the framework of WP4 and reported in D4.1 [23] and D4.2. In the aforementioned evaluation, the desired QoS satisfaction is always met, since it is one of the constraints of the problem under study. Regarding energy efficiency (EE), it will be measured by employing power models developed within WP4, as reported in D4.1 [23].

Table 4-1: Demo 3 KPIs.

Identifier	KPI	Description
ETH-KPI-UC3-01	Network coverage	100% global outdoor network coverage
ETH-KPI-UC3-02	Service continuity, reliability	99.99999% service continuity and service reliability
ETH-KPI-UC3-05	Performance integrity	PER with a target range set between 10^{-4} and 10^{-6}
ETH-KPI-UC3-06	Resource allocation	> 80% more energy efficient resource allocation than SotA

4.3 DEMO SETUP

Demo 3 functional architecture setup is depicted in Figure 4-2. It has been developed to evaluate aircraft communication services connectivity supported with edge computing, 3D network orchestration, efficient traffic routing and xNF placement schemes that are introduced in the WP4 deliverable D4.1 [23]. Thus, the demo testbed is being developed to evaluate the following developed capabilities contributed by CA, NETAI, AUTH and NBC:

- **CA:** 3D Network Orchestrator- computes and configures E2E paths between network nodes according to E2E performance metrics leveraging the SDN control plane.
- **NETAI:** AI-based traffic forecasting tools, which predicts 3D network links features (i.e., link signal-to-noise ratio (SNR), delay, data rate, utilization ratio, packet loss rate) and edge resource demands from the aircraft.
- **AUTH:** Joint user association, traffic routing and xNF placement targeting at high energy-efficiency while guaranteeing the QoS of the aircraft.
- **NBC:** Edge computing orchestrator which in combination with the decision engine equipped with efficient computational resource allocation schemes developed by AUTH and the prediction analytics engine developed by NETAI ensures dynamic and proactive edge resource (CPU core & memory) allocation according to real time requirements.

The Demo 3 testbed setup is shown in Figure 4-2. As shown, a Cisco switch interconnects the different components (i.e. PCs 1, 2, 3) in this demo. PC-1 accommodates (hosts) Ryu controller, the traffic forecasting tool developed by NETAI, the decision engine developed by AUTH, and the CA 3D network orchestrator. PC-1 also hosts Mininet environment that allows to develop SDN-enabled network. Nodes function as OpenFlow switches, while the network links are emulated by configuring them with performance metrics that capture terrestrial, aerial and satellite network layers performance, thus, constituting an SDN-enabled three-dimensional (3D) network. PC-3 hosts the Kubernetes Cluster onboarded on the NBC edge orchestrator. The edge orchestrator dynamically manages the edge computation resources required for processing Aircraft offloaded data computation tasks based on the decision engine's rules developed by AUTH. The decision engine receives also input data from the computational resource usage forecasting tool developed by NETAI. PC-2 comprises two laptops and other computing devices, but it is abstracted by a single PC-2, to simplify the architecture illustration. It accommodates an emulated 3D network of software-defined radios (SDRs) nodes that can be configured to run as 4G/5G gNBs with relevant configuration. Hence, it comprises terrestrial network layer configured with two gNBs (Air-to-Ground, A2G), aerial

(HAP) and satellite network layers configured with single gNB, each. Aircraft channel is configured to run as UE capturing mobility and effects on channel performance metric. A media server is also configured on PC-2, which maintains videos with different resolutions. It allows to generate further traffic in the network following video-on-demand streaming requests by aircraft. Noting that the simulation-based 3D emulated network runs on PC-1 and the SDRs-based 3D emulated network runs on PC-2 are managed and controlled by the SDN controller and 3D network Orchestrator.

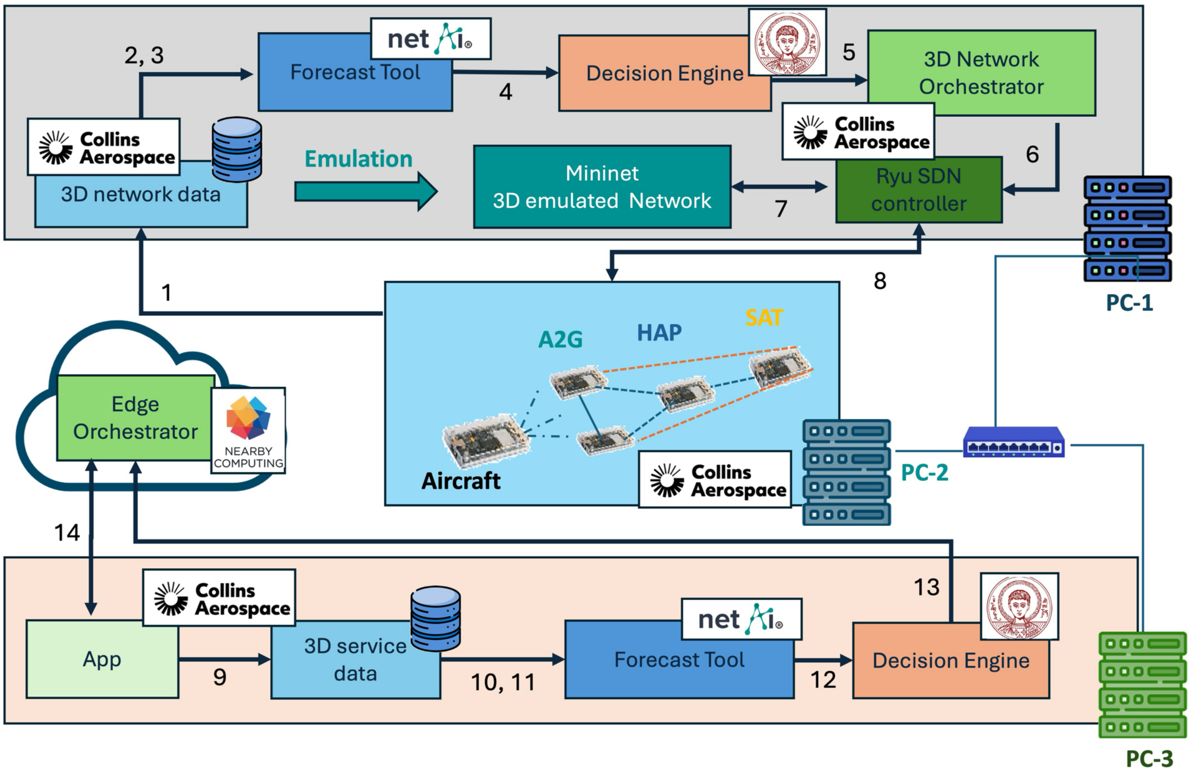

Figure 4-2: Demo 3 Testbed setup.

Demo 3 involves two experimental scenarios, the first is shown at the top part of Figure 4-2, which focuses on the allocation of network resources, i.e., demonstration of aircraft communication services and connectivity provisioning under guaranteed performance. The second scenario, depicted at the lower part of the figure, focuses on the proactive energy-efficient resource allocation. In particular, the detailed interactions among the involved components are given in the following:

1. Network Data is being periodically collected and stored in a database.

2./3. The NETAI forecasting tool is composed of two main blocks, i.e., the block performing the ML training and validation and the block performing the ML inference. Interface 2 fetches network and historical data to the ML training and validation block, while interface 3 fetches live data to the ML inference block. There is also an internal interface between the two forecasting tool blocks (not appearing in the picture) which publishes new training model and network configuration data to the ML inference block.

4. Pushes analytics (forecasts) to the Decision Engine.

5. Pushes decisions from Decision Engine to the 3D Network Orchestrator.

6. The 3D Network Orchestrator sends the actions needed to be performed to the Ryu SDN controller.

7./8. The Ryu SDN controller changes the flow tables of the underlying infrastructure (both the 3D emulated network in Mininet and the hardware in the loop) based on the required actions sent by the 3D Network Orchestrator.

9. App Data is being periodically collected and stored in a database.

10./11. Interface 10 fetches app historical data to the ML training and validation block, while interface 11 fetches live app data to the ML inference block. There is also an internal interface between the two forecasting tool blocks (not appearing in the picture) which publishes new training model and app configuration data to the ML inference block.

12. Pushes analytics (forecasts) to the Decision Engine.

13. Pushes decisions from Decision Engine to the Edge Orchestrator.

14. The Edge Orchestrator applies the computation resources orchestration actions needed to be performed, e.g., scaling up or down the service, according to the App computation resources and performance requirements.

The 3D integrated network infrastructure as targeted by ETHER is envisaged to support critical emerging services such as In-Flight Entertainment and Connectivity (IFEC) services which cannot be easily supported by existing terrestrial networks. These services include collision detection, real-time flight status updates, near-real-time weather forecast, and multimedia entertainment, some of which are associated with strict requirements in terms of latency and throughput. Moreover, these services are requested by airborne users (inflight users and aircrafts) who are characterised by high mobility and transcend a wider geographical area characterised by intermittent coverage, varying channel conditions and fluctuating traffic demand, which renders static orchestration procedures inefficient in terms of resource utilisation and meeting QoS requirements. The dynamic nature of the user traffic and the strict requirements requires intelligent and adaptive orchestration procedures to satisfy the service requirements with minimal resource overheads, given the constrained nature of edge nodes in terms of computational resources and energy capacity. Such a dynamic orchestration procedure should leverage AI/ML analytics for forecasting future resource requirements to permit proactive and dynamic scaling of allocated resources depending on the real-time demand.

In this demo, CA platform generates different volume of synthetic datasets emulating IFEC applications. The generated synthetic datasets are also marked by a business logic emulating the application performance in terms of metrics such as delay, response time, throughput and request blocking rate, among others in terms of the allocated resources. The edge computing resources are proactively scaled up or down according to the forecasted user requirements enabling compliance with the service requirements without resource overprovisioning which would result in increased resource and energy consumption.

4.3.1 Demo Components and Specifications

Error! Reference source not found. summarizes Demo 3 components.

Table 4-2: Demo 3 Bill of materials.

Component	Provider	No. of elements	Availability?
Multicore PCs Desktop	Collins	2	Yes
Switch	Cisco, Collins	1	Yes
eNB/gNB (BladeRF)	Vendor: Nuad Collins	5	Yes
Laptop (PC-3)	Collins	2	Yes
Accessories (RG45 cables and connectors)	Collins	10	Yes

Multicore Desktop PCs

The testbed comprises SDN-enabled 3D emulated network hosted on PC-1, shown in Figure 4-3 and 3D network of SDRs hosted on PC-2. Both networks are emulated to offer terrestrial, aerial (HAPS), and satellite network connectivity services to UE representing an emulated aircraft. They are managed and controlled by SDN Ryu controller and the 3D network orchestrator. PC-1 hosts as well Mininet environment that supports the development of SDN-enabled 3D emulated networks, as shown in Figure 4-3.

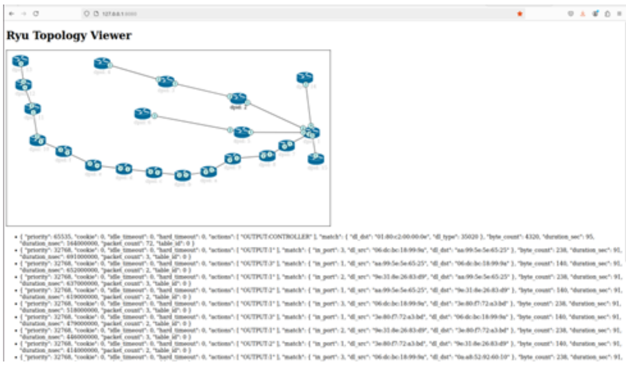

Figure 4-3: SDN-enabled 3D emulated network topology.

In this demo, the 3D simulation communication network data and trajectories of flights from Heathrow to Dublin are considered being stored and maintained on the SDN controller PC (PC-1). The details of the flight journey and trajectory along with the 3D network nodes, including HAP, 5G terrestrial gNBs, and satellites, are depicted in Figure 4-4.

- Dublin (EIDW) to London (EGLL) route
  - **British Airlines** flights: BAW827, BAW829, BAW833, BAW835, BAW837, BAW845
  - **Aer Lingus** flights: EIN152, EIN154, EIN156, EIN158, EIN164, EIN166, EIN168, EIN172, EIN174, EIN178, EIN184
  - All flights from 29/11/2024 to 13/12/2024 (15 days)
  - Real tracklogs extracted from FlightAware (ADS-B source)

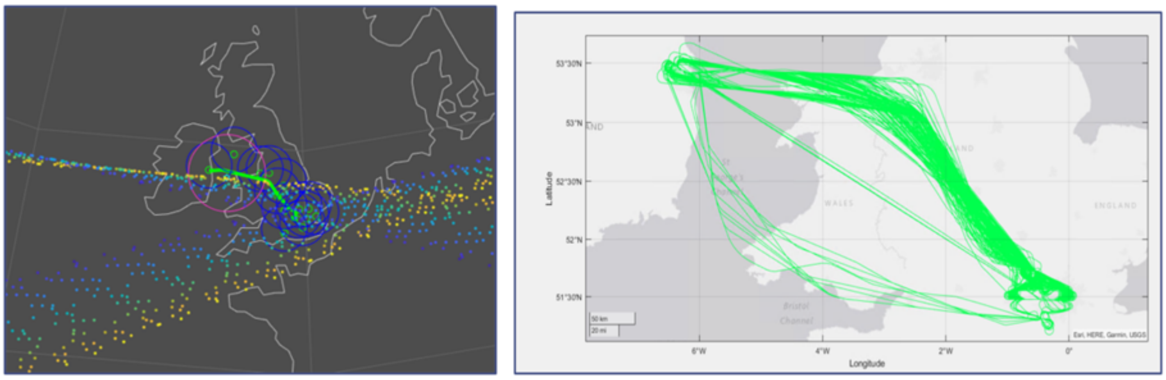

Figure 4-4: IE-UK Flight : Link simulation data collection.

As depicted in Figure 4-4, the A2G stations are located at VDL-2, the ground station sites appear in green circle markers with blue circle coverage, and one HAP station is considered over the Irish Sea (light blue hex marker with magenta circle coverage). The LEO satellites' orbit is chosen with the following configuration:

- Altitude = 550 km
- Inclination = 53 degrees
- Plane shift = 22 degrees
- Longitude shifts = [-25:5:15] degrees

SDRs: eNB/gNB (BladeRF)

Five BladeRFs are procured, of which 4 are being configured on the PC-2 including laptops to enable a 3D network of SDRs that comprise two terrestrial gNBs, one HAP and one satellite, and the fifth is configured as an emulated UE representing the aircraft. A node in this network is a BladeRF SDR, shown in Figure 4-5, which is an off-the-shelf USB 3.0 SDR being configured to run as eNB/gNB, enabling the setup of 5G compliant cell and UE.

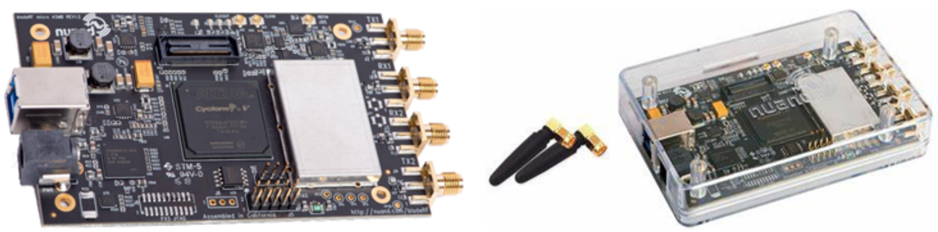

Figure 4-5: BladeRF 2.0 Micro X A4 USB 3.0 Software Defined Radio.

We have tested the suite Radio Access Network (srsRAN) setup and configuration using high performance PCs with Nuand bladeRF SDR as UE and gNB following the configuration shown in Figure 4-6.

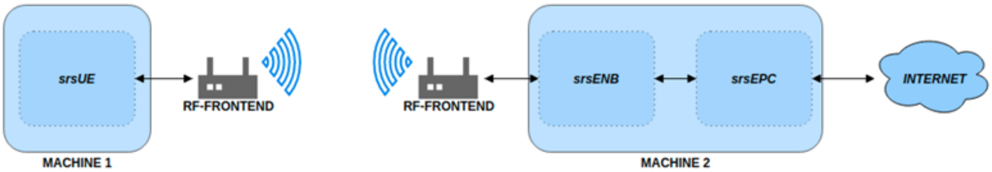

Figure 4-6: srsRAN setup with RF-frontend using bladeRF SDR.

It provides up to 30 MHz I/Q sampling rate which is enough to support the decoding of 20 MHz channel bandwidth. With appropriate srsRAN and bladeRF setup and configurations depicted in Figure 4-8, bidirectional connectivity has been successfully demonstrated between gNB and UE, as shown in Figure 4-7.

SW:
srsRAN UE, gNB, EPC
bladeRF driver
HW:
bladeRF2 xA4,
2 Ubuntu PC,
2 Ubuntu Raspberry Pi 4

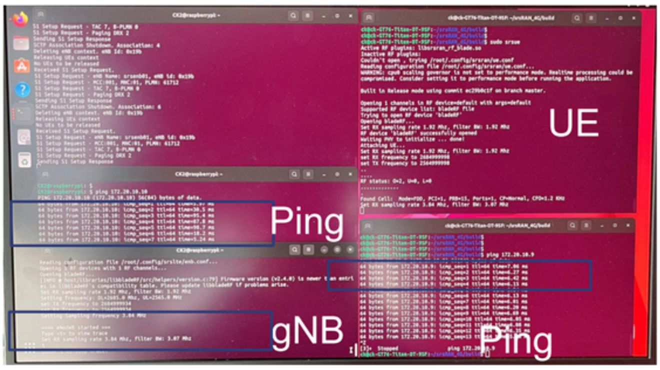

Figure 4-7: UE connection configuration to BladeRF gNB.

Laptops

One of the laptops accommodates the emulated UE SDR aircraft and the other one the gNBs.

gNB.conf

```
[enb]
enb_id = 0x19B
mcc = 001
mnc = 01
mme_addr = 127.0.1.100
gtp_bind_addr = 127.0.1.1
s1c_bind_addr = 127.0.1.1
s1c_bind_port = 0
n_prb = 25   # BW configuration
tm = 1
nof_ports = 1
[rf]
dl_earfcn = 3350
tx_gain = 80
rx_gain = 40
device_name = bladeRF
device_args = auto
time_adv_nsamples = auto  # calibration based on system delay
[log]
all_level = warning
all_hex_limit = 32
filename = /tmp/enb.log
file_max_size = -1
```

UE.conf

```
[rf]
freq_offset = 0  #calibration and synchronization w/ external clock (GPSDO)
tx_gain = 80
#rx_gain = 40
#srate = 11.52e6
#nof_antennas = 1
# For best performance in 2x2 MIMO and >= 15 MHz use the following
    device_args settings:
#    USRP B210: num_recv_frames=64,num_send_frames=64
# For best performance when BW<5 MHz (25 PRB), use the following
    device_args settings:
#    USRP B210: send_frame_size=512,recv_frame_size=512
device_name = bladeRF
device_args = auto
time_adv_nsamples = auto   # calibration based on system delay
#continuous_tx     = auto
[rat.eutra]
dl_earfcn = 3350
# nof_carriers = 1
[rat.nr]
bands = 78
nof_carriers = 1
[usim]
mode = soft
algo = milenage
```

Figure 4-8: gNB and UE configuration files.

Demo 3 has been developed to demonstrate the integration of 3D emulated networks comprised of terrestrial, aerial (HAP) and satellite networks, shown in Figure 4-1, leveraging the capabilities provided by SDN technology. The integrated 3D network is composed of two segments: The first 3D network segment is emulated under Mininet; and the second network segment is comprised of gNBs BladeRF SDR. The first network segment is connected to the second segment, considering the later as hardware-in-the-loop. The 3D network running in

Mininet is emulated with 3D 5G network simulation data, whereas the hardware-in-the-loop network nodes are emulated through setting the right 5G configurations for the parameters in their configuration files corresponding to their terrestrial, aerial (HAP), and satellite network layers.

The aircraft generates communication service requests (e.g., video streaming, communication messages transmission etc.). This demo evaluates the communication services performance of emulated aircraft channels connected to network nodes during their gate-to-gate journey, as shown in Figure 4-1. The 3D emulated network under Mininet environment is integrated with the 3D network of SDRs, shown in Figure 4-9. Aircraft can request on demand media streaming as services.

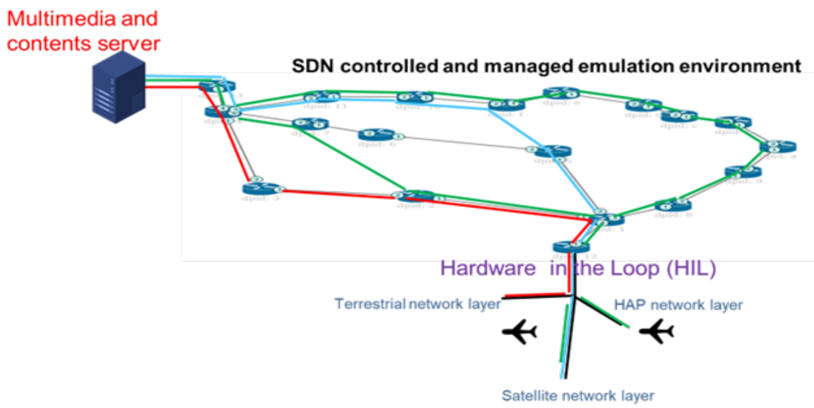

Figure 4-9: SDN-enabled 3D network with hardware-in-the-loop integration.

4.4 TESTING METHODOLOGY

4.4.1 Testing Description and timeplan

Table 4-3: Intermediate test case details.

T3.1	**3D network with hardware-in-the-loop configuration**
Phase	M27 (03/2025)
Description	Check proper emulated 3D network and hardware-in-the-loop operations (CA).
Target UCs	Use Case 3
Relevant Requirements	The requirement IDs evaluated with this test case are extracted from D2.2. Specifically, this test aims to validate: • ETH-REQ-UC3-FN-03 (Channel emulation) • ETH-REQ-UC3-FN-04 (Network resource monitoring)
Procedure/ Steps	1. Check successful aircraft connection to network nodes. 2. Check successful 3D network links emulation and HIL network links metrics measurement.

Parameters to be tested KPIs/ Or/And Success Criteria	• SDR gNBs function according to their network emulation layer, whether terrestrial, aerial, satellite. • Success UE switching between gNBs.
Network Configuration	• BladeRF gNBs channel provide performance metrics of aircraft communications. • Configuring network node and links with telemetry to generate performance metrics data.
Testbed	It involves the SDR network running PC2, as shown in Figure 4-2 and Figure 4-3.

T3.2	**3D Emulated Network and HIL Network integration**
Phase	M28 (04/2025)
Description	Enable integrated 3D network with hardware-in-the-loop with control and data plane management via SDN (CA).
Target UCs	Use Case 3
Relevant Requirements	The requirement IDs evaluated with this test case are extracted from D2.2. Specifically, this test aims to validate: • ETH-REQ-UC3-FN-03 (Channel emulation) • ETH-REQ-UC3-FN-04 (Network resource monitoring) • ETH-REQ-UC3-FN-07 (3D unified SDN management)
Procedure/ Steps	1. Check successful connection between emulated network nodes and SDR gNBs. 2. Check successful video traffic streaming to UE. 3. Check successful specific SDN rules on traffic flows control and management.
Parameters to be tested KPIs/ Or/And Success Criteria	• Continuity (delay), Integrity (packet loss rate): Success UE video streaming from the 3D emulated network nodes. • Global service coverage: Success UE switching between gNBs.
Network Configuration	• Setting dynamically the emulated links and nodes with specific metrics. • Configuring network node and links with telemetry to generate performance metrics data.
Testbed	It involves the SDR network running in PC2 as well as the Mininet 3D emulated network and Ryu SDN controller (interfaces 7 and 8) running in PC1, as shown in Figure 4-2.

T3.3	**Testbed Integration with Edge Orchestrator**
Phase	M29 (05/2025)
Description	Enable onboarding of the cloud-native infrastructure (CA) across the different strata to the application orchestrator (NBC) to facilitate dynamic resource and application management.
Target UCs	Use Case 3
Relevant Requirements	The requirement IDs evaluated with this test case are extracted from D2.2. Specifically, this test aims to validate: • ETH-REQ-UC3-NF-04 (3D network programmability) • ETH-REQ-UC3-FN-06 (Service orchestrator)
Procedure/ Steps	1. Check successful deployment of application orchestrator 2. Check success onboarding of the cloud-native infrastructure to the application Orchestrator 3. Check successful establishment of communication between the cloud-native infrastructure and application orchestrator 4. Check successful access to infrastructure metrics
Parameters to be tested KPIs/ Or/And Success Criteria	• The edge orchestrator has connectivity to the cloud-native infrastructure • The orchestrator has visibility to the metrics of the infrastructure and deployed application
Network Configuration	• Network infrastructure provisioned with Kubernetes
Testbed	It involves the Edge orchestrator running in the cloud and the App running in PC3 (interface 14), as shown in Figure 4-2.

T3.4	**Interfacing of Forecasting Tools and Decision Engine**
Phase	M29 (05/2025)
Description	Enable interfacing of forecasting tool output developed by NETAI and the decision engine leveraging efficient resource allocation algorithms developed by AUTH.
Target UCs	Use Case 3
Relevant Requirements	The requirement IDs evaluated with this test case are extracted from D2.2. Specifically, this test aims to validate: • ETH-REQ-UC3-NF-06 (3D network resource optimisation) • ETH-REQ-UC3-NF-05 (3D network connected intelligence) • ETH-REQ-UC3-NF-04 (3D network programmability)

	• ETH-REQ-UC3-FN-04 (Network resource monitoring)
Procedure/ Steps	**1.** Verify successful establishment of communication via REST API between the forecasting tool and the decision engine. **2.** Verify correct execution of resource allocation algorithms via regression testing with reference inputs. **3.** Examine effect of forecasting errors on the quality of produced solution by the resource allocation algorithms and investigate the performance/complexity trade-off via extensive simulations. **4.** Verify proper state representation after execution of the decision engine algorithms.
Parameters to be tested KPIs/ Or/And Success Criteria	• The output of the forecasting tool is successfully taken as an input to the decision engine. • The decision engine returns a resource allocation decision within the specified execution time interval. • Forecast updates are provided frequently enough to ensure decision engine does not operate on “stale” input.
Network Configuration	Indirect only, as the forecasting tool provides the necessary data to the decision engine algorithms to enable network configuration via proactive resource allocation.
Testbed	It involves the Forecast tool and the decision engine (interfaces 4, 12) running in PC1 and PC3, as shown in Figure 4-2.

T3.5	**Interfacing of Decision Engine with 3D network Orchestrator**
Phase	M30 (06/2025)
Description	Enable interfacing of the decision engine leveraging efficient resource allocation algorithms developed by AUTH and the 3D network orchestrator of the Demo 3 platform (CA).
Target UCs	This test corresponds to Use Case 3
Relevant Requirements	The requirement IDs evaluated with this test case are extracted from D2.2. Specifically, this test aims to validate: • ETH-REQ-UC3-NF-06 (3D network resource optimisation) • ETH-REQ-UC3-NF-05 (3D network connected intelligence) • ETH-REQ-UC3-NF-04 (3D network programmability) • ETH-REQ-UC3-FN-06 (Network orchestrator) • ETH-REQ-UC3-FN-07 (3D unified SDN management)
Procedure/ Steps	**1.** Check successful establishment of communication between the 3D network orchestrator and the decision engine.

	2. Verify creation of properly formatted output by the decision engine and its successful communication to the 3D network orchestrator. 3. Examine effect of execution time constraints on the quality of produced solution by the resource allocation algorithms. 4. Verify that consumed and/or newly committed network resources are correctly captured/updated in the state representation of the decision engine.
Parameters to be tested KPIs/ Or/And Success Criteria	• The output of the decision engine is successfully fed to the 3D network orchestrator. • Scalability analysis of proposed resource allocation algorithms w.r.t number of users and network topology.
Network Configuration	Determination of routed paths and selection of service attachment nodes by the decision engine algorithms, with the decision being executed by the 3D network orchestrator.
Testbed	It involves the 3D Network orchestrator and the decision engine (interface 5) both running in PC1, as shown in Figure 4-2.

T3.6	**Interfacing of Decision Engine with the Edge Orchestrator**
Phase	M30 (06/2025)
Description	Enable interfacing of the decision engine leveraging efficient computational resource allocation algorithms developed by AUTH and the edge orchestrator of NBC.
Target UCs	Use Case 3
Relevant Requirements	The requirement IDs evaluated with this test case are extracted from D2.2. Specifically, this test aims to validate: • ETH-REQ-UC3-NF-06 (3D network resource optimisation) • ETH-REQ-UC3-NF-05 (3D network connected intelligence) • ETH-REQ-UC3-NF-04 (3D network programmability) • ETH-REQ-UC3-FN-06 (Service orchestrator)
Procedure/ Steps	1. Check successful connectivity between the orchestrator and the infrastructure running the decision engine or storing the decision engine output. 2. Check successful access of the orchestrator to the decision engine output. 3. Check successful deployment/reconfiguration of computational resources requested by the decision engine. 4. Verify proper state representation of consumed computational resources in the decision engine after the algorithm's execution.

Parameters to be tested KPIs/ Or/And Success Criteria	• The output of the decision engine is successfully fed to the Edge orchestrator. • Scalability analysis of the computational resource allocation algorithms of the decision engine.
Network Configuration	Specified xNFs are deployed into appropriate nodes along the edge/cloud continuum to meet the computational requirements of the network's tasks and offered services.
Testbed	It involves the Edge orchestrator, running on the cloud, and the decision engine running in PC3, as shown in Figure 4-2.

T3.7	**Integrated 3D network orchestration**
Phase	M33 (09/2025)
Description	Enable the network orchestrator to manage network resources and service connectivity through the integrated 3D network with hardware-in-the-loop via unified SDN control and data plane management (CA) leveraging the efficient resource allocation algorithms of AUTH based on traffic forecasting input (NETAI).
Target UCs	Use Case 3
Relevant Requirements	The requirement IDs evaluated with this test case are extracted from D2.2. Specifically, this test aims to validate: • ETH-REQ-UC3-FN-04 (Network resource monitoring) • ETH-REQ-UC3-FN-05 (Multilink functionality) • ETH-REQ-UC3-NF-03 (Handover reliability and delay) • ETH-REQ-UC3-FN-06 (Network orchestrator) • ETH-REQ-UC3-FN-07 (3D unified SDN management)
Procedure/ Steps	1. Check successful collection of network nodes and links metrics data/metrics and their proper processing by the 3D network orchestrator. 2. Check successful computation of network paths by the decision engine and subsequent configuration by the 3D network orchestrator towards meeting a service connectivity requirement based on the link quality and traffic data input from the forecasting tool. 3. Check successful application of specific rules on traffic flow control and management via SDN mechanisms, considering different network orchestration policies and associated decision engine output.
Parameters to be tested KPIs/ Or/And Success Criteria	• The 3D network orchestrator initiates the path computation meeting the target E2E QoS requirements and successfully configures the selected path. • E2E latency and data rate (e.g., the selected path passes through a given set of nodes or avoids a given set of nodes to maximize energy efficiency) will be investigated.

Network Configuration	Network resource (i.e., node/link activation, routing) allocation is performed for the network setup of Figure 4-2 (PC1 and PC2).
Testbed	It includes the network setup of PC1 and PC2, shown in Figure 4-2.

T3.8	**Integrated 3D service orchestration**
Phase	M34 (10/2025)
Description	Enable the edge orchestrator to manage app resources (NBC) of the testbed (CA) leveraging the efficient computational resource allocation algorithms of AUTH based on app forecasting traffic input (NETAI).
Target UCs	Use Case 3
Relevant Requirements	The requirement IDs evaluated with this test case are extracted from D2.2. Specifically, this test aims to validate: • ETH-REQ-UC3-NF-06 (3D network resource optimisation) • ETH-REQ-UC3-NF-05 (3D network connected intelligence) • ETH-REQ-UC3-NF-04 (3D network programmability) • ETH-REQ-UC3-FN-06 (Service orchestrator)
Procedure/ Steps	**1.** Check successful app metrics collection and processing by the Edge orchestrator. **2.** Check successful computation of resource allocation instructions made by the decision engine and associated execution by the Edge orchestrator to meet the requirements specified by the app. **3.** Check successful up/down-scaling of the compute resources to meet the time-varying nature of aggregated service requests and their associated requirements.
Parameters to be tested KPIs/ Or/And Success Criteria	• The Edge orchestrator scales up/down the computational resources based on the decision engine's specified rules so as to meet the app requirements.
Network Configuration	As determined by the decision engine, computational resources necessary to support the xNFs and computational tasks associated to the requested services are dynamically allocated across the network setup of Figure 4-2.
Testbed	It includes PC3 as well as the Edge orchestrator, running on the Cloud, shown in Figure 4-2.

Table 4-4: Final test case details.

T3.9	**Final Demo 3 execution**
Phase	M36 (12/2025)

Description	Demonstrate the performance of two experimental scenarios on aircraft connectivity and computing performance guarantees in 3D networks.
Target UCs	Use Case 3
Relevant Requirements	The requirement Intermediate tests evaluated with this test case are extracted from D2.2. Specifically, this test aims to validate: • ETH-REQ-UC3-NF-04 (3D network programmability) • ETH-REQ-UC3-FN-06 (Network/Service orchestrator) • ETH-REQ-UC3-NF-06 (3D network resource optimisation) • ETH-REQ-UC3-NF-05 (3D network connected intelligence) • ETH-REQ-UC3-FN-04 (Network resource monitoring) • ETH-REQ-UC3-FN-05 (Multilink functionality) • ETH-REQ-UC3-NF-03 (Handover reliability and delay) • ETH-REQ-UC3-FN-07 (3D unified SDN management)
Procedure/ Steps	**1.** Check successful data reception at the output of interfaces **2.** Check successful data transmission through network interfaces **3.** Check successful optimization engines calculation **4.** Check successful computation resources allocation **5.** Check successful network nodes and links metrics collection and processing by the SDN orchestrator
Parameters to be tested KPIs/ Or/And Success Criteria	• Ensure aircraft communication service connectivity continuity. • Network orchestrators ensure E2E connectivity reliability and integrity.
Network Configuration	• Configuring network interfaces with telemetry to generate performance metrics data. • Setting network orchestration policies
Testbed	Figure 4-2 shows a global picture of the connections and architecture of the full testbed involved in this final demo test.

4.5 RISK AND MITIGATION

Use case 3 focuses on delivering solutions to enhance resilient aircraft connectivity, communications, and computing. The table below outlines the identified risks associated with various components and interfaces of the demonstration detailing their likelihood of occurring and potential impact on the demo. To mitigate these risks, we propose effective strategies to minimize their effects and ensure the successful execution of the final demonstrator.

Table 4-5: Identified demo-related risks and mitigation plans for Demo 3.

ID	Risk	Likelihood (H/M/L)	Impact (H/M/L)	Mitigation Plan
3.1	Collins constantly change network proxy configuration, causing network connectivity issues, python libraries incompatibility that would directly affect experimental environment	L	M	Constantly check with the segregate network administrator to exclude the experimental network nodes from update, while updating the network settings with the recent proxy configurations.
3.2	Unavailability of computing resources required by the AI-based traffic forecasting tool for training and validation.	L	L	Adapt the network size accordingly so that the computing requirements of the AI-based traffic forecasting tool can be accommodated
3.3	Not able to automate the App/network orchestration based on the decision engine's actions	L	L	The output of the decision will be manually fed to the Edge/3D network orchestrator, thus still showcasing the core functionality.
3.4	Decision engine resource allocation algorithms unable to provide solution within specified execution time interval dictated by network dynamics	L	M	Lower-complexity suboptimal algorithms will be implemented and their execution environment fine-tuned to guarantee that solution is produced within required time interval.

5. CONCLUSIONS

In conclusion, the ETHER project aims to pave the way for the next generation of sustainable 6G networks by integrating terrestrial, aerial, and space layers to support a diverse range of services. Through the demonstration of key use cases, including flexible payload-enabled service provisioning, unified RAN for direct Ka-band handheld device access, and air-space safety-critical operations, ETHER showcases its potential to revolutionize 6G network capabilities. The proposed solutions emphasize energy-efficient service management, seamless handovers, and AI-based resource allocation for critical applications in air-safety operations.

This deliverable outlines the comprehensive testing methodology and testbed setup that will guide the execution of ETHER's demo activities. By providing a step-by-step testing plan, identifying key objectives and areas of focus, detailing test cases and KPIs, and describing the necessary test facilities and associated risks, this document lays the groundwork for the successful completion of the demonstrations. The upcoming WP5 deliverable D5.2 will build upon this foundation, providing further insights into the integration activities and interfaces for each demo, ensuring continued progress towards the project's goals.

6. REFERENCES

[1] ETHER project, "Deliverable 2.2: Use cases and KPIs," 31 December 2023. [Online].

[2] A. Space, "Commercial Payload (TREVO)," [Online]. Available: https://alen.space/es/products/trevo-subsistema/. [Accessed 2025].

[3] OrbAstro, "Commercial Payload (Guardian - Alpha Spacecraft)," [Online]. Available: https://www.nanosats.eu/sat/orbastro-guardian. [Accessed 2025].

[4] EnduroSat, "Commercial Payload (EnduroSat Contribution To Edge SpAIce)," [Online]. Available: https://indico.cern.ch/event/1469724/attachments/2976104/5240782/KT%20seminar%20-%20EnduroSat%20-%20The%20Edge%20spAIce.pdf. [Accessed 2025].

[5] L. Contreras-Benito, K. Osipova, J. Buitrago-Leiva, G. Gracia-Sola, F. Coppa, P. Climent-Salazar, P. Sopena-Coello, D. Garcín, J. Ramos-Castro and A. Camps, "3Cat-8 Mission: A 6-Unit CubeSat for Ionospheric Multisensing and Technology Demonstration Test-Bed," *Remote Sensing,,* vol. 16(22), no. 4199, 2024.

[6] "K3s: Lightweight Kubernetes," [Online]. Available: https://k3s.io/.

[7] V. Montilla and [et.al], "Deployment of NB-IoT NTN core network functions on Software Defined Radio (SDR) nanosatellites: approach and performance assessment," in *International Astronautical Congress. "74th International Astronautical Congres: 2-6 October, 2023*, Baku, Azerbaijan, 2023.

[8] T. Kellermann, R. P. Centelles, D. Camps-Mur, R. F. M. Guadalupi and A. C. Augé, "Novel Architecture for Cellular IoT in Future Non-Terrestrial Networks: Store and Forward Adaptations for Enabling Discontinuous Feeder Link Operation," *IEEE Access,* vol. 10, no. 10.1109/ACCESS.2022.3184720, pp. 68922-68936, 2022.

[9] ETHER project, "Deliverable 3.1: Initial report on key technological enablers for the seamless and energy-efficient ETHER Network Operation," 18 November 2024. [Online].

[10] E. Delfani and N. Pappas, "Version Age-Optimal Cached Status Updates in a Gossiping Network with Energy Harvesting Sensor," *IEEE Transactions on Communications,* 2024.

[11] E. Delfani and N. Pappas, "Semantics-Aware Status Updates with Energy Harvesting Devices: Query Version Age of Information," in *22nd WiOpt Conference*, Seoul, South Korea, 2024.

[12] E. Delfani and N. Pappas, "Semantics-Aware Updates from Remote IoT Devices to Interconnected LEO Satellites," no. 10.48550/arXiv.2502.07069, 2025.

[13] Prometheus, "Prometheus," [Online]. Available: https://prometheus.io/. [Accessed 18 November 2024].

[14] Kepler, "Kubernetes Efficient Power Level Exporter (Kepler)," [Online]. Available: https://sustainable-computing.io/. [Accessed 28 February 2025].

[15] ZeroMQ, "ZeroMQ: An open-source universal messaging library," [Online]. Available: https://zeromq.org/. [Accessed 28 February 2025].

[16] GrafanaLabs, "Grafana," [Online]. Available: https://grafana.com/. [Accessed 18 November 2024].

[17] Y. Li, J. Huang, Q. Sun, T. Sun and S. Wang, "Cognitive Service Architecture for 6G Core Network," *IEEE Transactions on Industrial Informatics,* vol. 17, no. 10, 2021.

[18] Jackery, "How Many Watts Does a Starlink Use," 10 October 2024. [Online]. Available: https://www.jackery.com/blogs/knowledge/how-many-watts-does-starlink-use#:~:text=How%20Many%20Watts%20Does%20A%20Starlink%20Use%20At%20High%20Performance,watts%20in%20an%20idle%20state. [Accessed 24 February 2025].

[19] O. Arnold, F. Richter, G. Fettweis and O. Blume, "Power consumption modeling of different base station types in heterogeneous cellular networks," in *Future Network & Mobile Summit*, Florence, Italy, 2010.

[20] ETHER project, "Deliverable 2.3: Techno-economic analysis and business plan," 31 July 2024. [Online].

[21] ETHER project, "Deliverable 2.4: Final report on ETHER network architecture, interfaces, and architecture evaluation," 30 August 2024. [Online]. Available: https://www.ether-project.eu/wp-

content/uploads/sites/100/2024/10/ETHER_Deliverable_D2.4_V1.0_final.pdf. [Accessed 28 February 2025].

[22] International Civil Aviation Organization, "Manual on Required Communication Performance (RCP), 1st edition," Montreal, Canada, 2008.

[23] ETHER project, "Deliverable 4.1: Initial report on zero-touch data-driven network and service orchestration in the 3D ETHER architecture," 13 December 2024. [Online].

