ENVELOPE

Evaluation and validation of connected mobility in real open systems beyond 5GS

Project deliverable D6.1

ENVELOPE evaluation methodology

HORIZON JU Innovation Actions | 101139048 |
ENVELOPE - HORIZON-JU-SNS-2023

Co-funded by the European Union

6GSNS

Deliverable administrative information

Dissemination level	**PU - Public**
Type of deliverable	R — Document, report
Work package	WP6
Title of the deliverable	D6.1 – ENVELOPE evaluation methodology
Status – version, date	Final – V1, 31/03/2026
Deliverable leader	VICOM
Contractual date of delivery	31/03/2026
Submission date	31/03/2026
Keywords	KPI, KVI, evaluation methodology

List of contributors

Name	Organisation
Gorka Vélez	VICOM
Jon Ander Iñiguez de Gordoa	VICOM
Christos Verikoukis	ATH/ISI
Gergely Kovács	CMS
Ioanna Ioannou	eBOS
Dimitris Tsolkas	FOGUS
Christos Milarokostas	FOGUS
Alex Kakyris	FOGUS
Nikhil Muthakana	HAN
Elvina Gindullina	HPE
Nicola di Pietro	HPE

Davide Montagno Bozzone	HPE
Dimitris Fragkos	ICCS
Pavlos Basaras	ICCS
Stratis Myritzis	ICCS
Apostolos Siokis	IQU
Edoardo Bonetto	LINKS
Jannis Weil	LUH
Amr Rizk	LUH
Nehal Baganal Krishna	LUH
Vasilis Pitsilis	NCSRD
Harilaos Koumaras	NCSRD
Ioannis Kontizas	NCSRD
Dimitris Uzunidis	NCSRD
Gabriele Scivoletto	NXW
Fofy Setaki	OTE
Ján Macháček	Roboauto
Ezio Chiocchetti	TIM
Ramon de Souza Schwartz	TNO
Peter-Paul Schackmann	TNO
Rosario Fria	Teoresi
Valeria Proietti Dante	Teoresi
Rakshith Kusumakar	V-TRON

Quality control

	Name	Organisation	Date
Peer review 1	Gergely Kovács	CMS	19/03/2026
Peer review 2	Konstatinos Katsaros	ICCS	20/03/2026

Version History

Version	Date	Author	Summary of changes
0.1	29/01/2026	Gorka Vélez	Table of contents defined.
0.2	13/03/2026	All authors	All content included for the draft sent to peer review.
0.3	27/03/2026	Gorka Vélez, Jon Ander Iñiguez de Gordoa and Ioanna Ioannou	Changes suggested by peer reviewers.
1.0	31/03/2026	Gorka Vélez	Final modifications after Quality Check & Final Check.

Legal Disclaimer

Table of contents

List of figures

List of tables

Project executive summary

ENVELOPE aims to advance and open the reference 5G advanced architecture and transform it into a vertical-oriented one. It proposes a novel open and easy-to-use 5G-advanced architecture to enable a tighter integration of the network and the service information domains by

- exposing network capabilities to verticals,
- providing vertical information to the network; and
- enabling verticals to dynamically request and modify key network aspects,

all performed in an open, transparent, and easy-to-use, semi-automated way.

ENVELOPE will build APIs that act as an intermediate abstraction layer that translates the complicated 5GS interfaces and services into easy to consume services accessible by the vertical domain. The project will deliver an experimentation framework that will facilitate vertical services in accessing a series of innovations developed in the project, namely: edge computing with service continuity support (federation/migration), zero-touch management, multi-connectivity, dynamic slicing and predictive QoS.

ENVELOPE will deliver 3 large scale Beyond 5G (B5G) trial sites in Italy, Netherlands and Greece supporting novel vertical services, with advanced exposure capabilities and new functionalities tailored to the services' needs. Although focused on the Connected and Automated Mobility (CAM) vertical, the developments resulting from the use cases (UC) will be reusable by any vertical. The ENVELOPE architecture will serve as an envelope that can cover, accommodate, and support any type of vertical services. The applicability of ENVELOPE will be demonstrated and validated via the project CAM UCs and via several 3rd parties that will have the opportunity to conduct funded research and test their innovative solutions over ENVELOPE.

Social Media link:

@envelope-project

For further information please visit www.envelope-project.eu

Deliverable executive summary

The present document constitutes Deliverable D6.1, "ENVELOPE Evaluation Methodology," developed under Task T6.1 to establish the evaluation framework for the ENVELOPE project. The primary objective is to define a unified, comprehensive methodology to test and evaluate both ENVELOPE innovations and Open Call projects.

The evaluation of ENVELOPE innovations is based on a Key Performance Indicator (KPI) evaluation methodology and a Key Value Indicator (KVI) evaluation methodology. The KPI methodology focuses on quantifiable metrics across three core domains: a project-level evaluation of broader milestones and results; a technical performance evaluation that provides experimental assessment of specific innovation targets and critical scalability considerations (with predefined test cases detailed in the annex); and a user acceptance evaluation employing survey-based tools to gauge end-user experience and satisfaction

Complementing the technical metrics, the KVI methodology measures the translation of ENVELOPE’s technological achievements into broader societal, environmental, and economic outcomes. Aligned with the Smart Networks and Services Joint Undertaking (SNS JU) mandate for value-driven evaluation, the KVI framework is executed through quantitative empirical evaluations, structured user surveys, and expert assessments.

Furthermore, this deliverable delineates the monitoring and evaluation lifecycle for Open Call projects. It establishes a structured reporting timeline requiring third-party beneficiaries to submit three deliverable reports throughout their project execution, which are subsequently evaluated by the ENVELOPE consortium according to standardized, annexed criteria.

Developed in close collaboration with all Work Package 6 (WP6) partners, this document serves as the foundational baseline for project evaluation. The implementation of the described methodology may lead to further refinements of the methodology which will be reported in subsequent WP6 deliverables.

List of abbreviations and acronyms

Acronym	Meaning
3GPP	3rd Generation Partnership Project
5GS	5G System
AF	Application Function
AI	Artificial Intelligence
ANE	ATSSS Layer – Network Side
API	Application Programming Interface
ATSSS	Access Traffic Steering, Switching, and Splitting
AUE	ATSSS Layer – UE Side
CAM	Connected and Automated Mobility
CCAM	Connected, Cooperative and Automated Mobility
CPU	Central Processing Unit
DoA	Description of Action
EaaS	Experimentation as a Service
EC	European Commission
GDPR	General Data Protection Regulation
GPU	Graphics Processing Unit
ICMP	Internet Control Message Protocol
IQN	In-advance Quality-of-service Notification
KER	Key Exploitable Result
KPI	Key Performance Indicator
KVI	Key Value Indicator
MAE	Mean Absolute Error

MAPE	Mean Absolute Percentage Error
NEF	Network Exposure Function
NWDAF	Network Data Analytics Function
OBU	On Board Unit
OC	Open Call
PCF	Policy Control Function
PDU	Protocol Data Unit
PEOU	Perceived Ease of Use
PLMN	Public Land Mobile Network
pQoS	Predictive Quality of Service
PU	Perceived Usefulness
QoD	Quality on Demand
QoE	Quality of Experience
QoS	Quality of Service
RAN	Radio Access Network
RL	Reinforcement Learning
RMSE	Root Mean Square Error
SNS JU	Smart Networks and Services Joint Undertaking
TAM	Technology Acceptance Model
TRL	Technology Readiness Level
TC	Test Case
TS	Technical Specification
UC	Use Case
UDP	User Datagram Protocol

UE	User Equipment
URSP	UE Route Selection Policy
V2X	Vehicle-to-Everything
WP	Work Package
ZTM	Zero-Touch Management

1 Introduction

1.1 Purpose of the deliverable

The present deliverable, developed as part of Task T6.1, reports on the work carried out within this task for defining the evaluation methodology of ENVELOPE. This includes the project-level Key Performance Indicators (KPIs) evaluation, the technical performance evaluation (including scalability aspects), the user acceptance evaluation and the Key Value Indicators (KVIs) evaluation. In addition, the evaluation process of Open Call projects is described.

1.2 Intended audience

This deliverable is classified as “public” (PU) and is primarily designed to serve as a reference for ENVELOPE Consortium Members throughout the evaluation phases of the project. It is also available to any interested reader who wishes to learn about ENVELOPE’s evaluation methodology.

1.3 Structure of the deliverable and its relationship with other packages/deliverables

This document is structured as follows:

- Section 1 introduces the content of the document.
- Section 2 describes the KPI evaluation methodology, which includes project-level KPI evaluation, technical performance evaluation and user-acceptance evaluation.
- Section 3 describes the KVI evaluation methodology.
- Section 4 explains the evaluation process of Open Call projects.
- Section 5 concludes with key insights.
- Section 6 provides a list of references for the document.
- Annexes detail the test cases and the Open Call milestone evaluation form.

This deliverable serves as a key input for tasks T6.2 – “Large scale trial sites technical evaluations and demonstrations”, T6.3 – “Open call projects on-boarding and technical evaluations and demonstrations” and T6.4 – “Impact assessment and user acceptance”.

2 ENVELOPE's KPI evaluation methodology

Building upon the framework established in Deliverable D2.3 [1], the ENVELOPE project categorizes Key Performance Indicators (KPIs) into two primary classifications:

- **Project-Level KPIs:** Originally outlined in the Description of Action (DoA), these metrics provide a high-level assessment of the overall project outcomes and impact.
- **ENVELOPE Innovation KPIs:** These metrics are designed to evaluate the concrete technological advancements and innovations developed within Work Package 3 (WP3).

The ENVELOPE Innovation KPIs specifically measure four core evaluation targets:

1. The Experimentation-as-a-Service (EaaS) system
2. The ENVELOPE Enablers
3. The Zero-touch management system
4. The Inter-PLMN handover system

To comprehensively measure these targets, the evaluation methodology employs a dual approach: **experimental testing** to quantify technical performance, and **user surveys** to assess the overall user experience.

Consequently, the remainder of this section details the methodology across three distinct KPI evaluation domains (see Figure 1):

- **Project-Level KPI Evaluation:** Assessing the broader project results against the initial DoA objectives.
- **Technical Performance Evaluation:** Detailing the experimental assessment of the specific innovation targets, including critical scalability considerations.
- **User Acceptance Evaluation:** Outlining the survey-based approach used to gauge user experience and satisfaction.

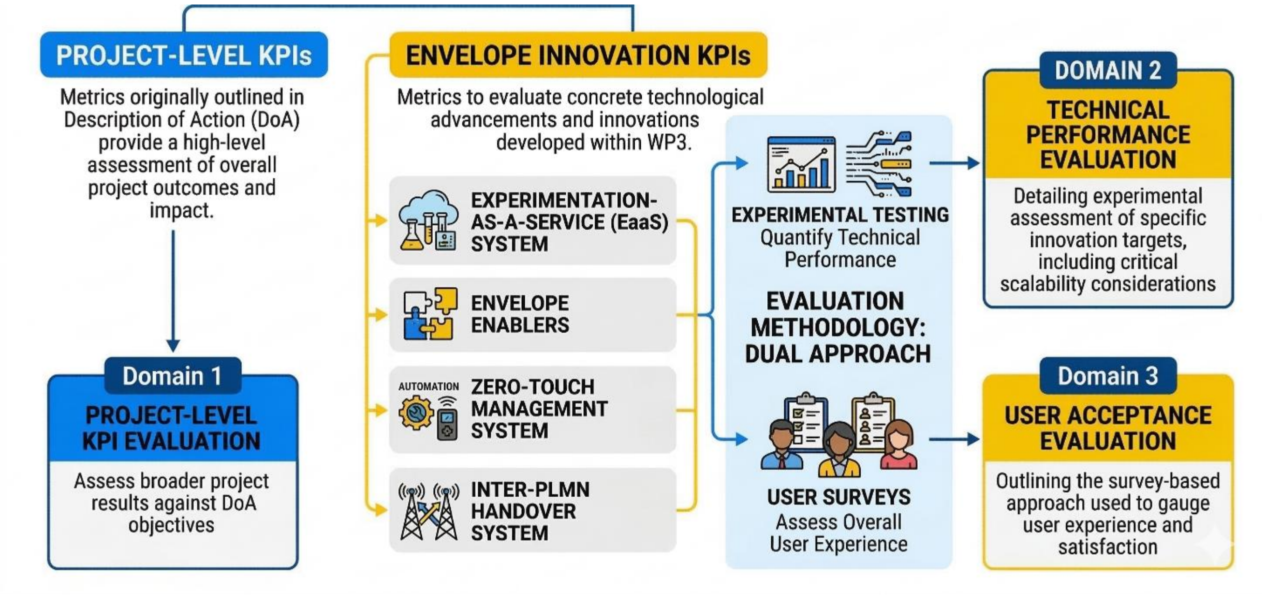

Figure 1 KPI evaluation approach

2.1 Project-level KPI evaluation

While technical KPIs measure the specific system performance of the ENVELOPE innovations (e.g., latency, availability), project-level KPIs assess the overarching execution, integration, and impact of the project (see deliverable D2.3 [1]). The evaluation methodology for these high-level KPIs follows a systematic, five-step lifecycle to ensure continuous monitoring and successful delivery of targets.

- **Step 1: Metric Operationalization and Ownership Assignment**

Before active tracking begins, each project-level KPI must be translated into measurable criteria and assigned to a specific Work Package (WP) or Task Leader.

- **Definition of "Done":** Establish the exact criteria for completion. For example, for *KPI_P_PlatformNum*, "Done" means the platform is accessible, supports dynamic reconfiguration, and is verified by all trial sites.
- **Responsibility Matrix:** Assign an owner to each KPI. The WP leader for the Open Calls would own *KPI_P_OCProjectNum*, while the Dissemination/Impact lead would own *KPI_P_SurveyResponses*.

- **Step 2: Data Collection Mechanisms**

Data collection strategies will vary based on the nature of the specific project KPI. We classify the collection into three distinct approaches:

- Deliverable & Milestone Verification (**Binary/Count Metrics**): KPIs such as *KPI_P_TSNum* (3 trial sites), *KPI_P_InnovationNum* (7 innovations), *KPI_P_PlatformNum* (1 platform), *KPI_P_UCNum* (6 Use Cases), *KPI_P_EvalMethod* (1 methodology), and

KPI_P_TestFram (1 framework) are evaluated through the formal submission and approval of associated project deliverables and successful site demonstrations.

- Operational Tracking (**Continuous Metrics**): KPIs like *KPI_P_OCProjectNum* (>= 9 projects) and *KPI_P_SurveyResponses* (100 responses) require continuous tracking.
- System-Derived Analytics (**Performance Metrics**): *KPI_P_ServiceSuccessRate* (80% success rate) acts as a bridge between technical and project KPIs. Data for this will be automatically aggregated by the employed Testing Framework (*KPI_P_TestFram*) during the validation of the CAM vertical services.

- **Step 3: Validation and Quality Assurance**

Collecting the numbers is not enough; the quality of the output must be validated against the ENVELOPE Description of Action (DoA).

- **Internal Peer Review:** For innovation and methodology-based KPIs, internal peer review by the Technical Management Team ensures the output is correctly assessed.
- **Cross-Site Consistency Checks:** For *KPI_P_TSNum*, the validation process will confirm that the trial sites in Italy, the Netherlands, and Greece actually implement the *same* ENVELOPE architecture, as specified in the target description.
- **Data Cleaning:** For survey responses, validation involves cleaning the data to ensure the 100 respondents genuinely represent the targeted CAM vertical, Economic, or Societal perspectives, removing incomplete or non-valid responses.

- **Step 4: Progress Reporting and Monitoring**

To maintain visibility over the project's health, these KPIs will be integrated into the standard project management workflow.

- **Periodical Status Updates:** KPI owners provide brief status updates during general assembly or Steering Committee meetings.
- **Milestone Reviews:** Formal evaluation of cumulative progress aligning with scheduled EC review meetings.

- **Step 5: Deviation Management and Corrective Actions**

If the evaluation reveals that a project-level KPI is at risk of not meeting its target (e.g., the Open Call only yields 6 viable projects instead of the targeted 9), a mitigation protocol is triggered.

- **Root Cause Analysis:** The KPI owner identifies why the target is lagging (e.g., insufficient marketing for the Open Call, or technical delays at a trial site).
- **Action Plan Formulation:** The KPI owner drafts a specific recovery plan (e.g., extending the Open Call deadline, hosting targeted webinars).
- **Re-evaluation:** The timeline for that specific KPI is adjusted, and evaluation frequency is increased to bi-weekly until the KPI is back on track.

2.2 Technical performance evaluation

2.2.1 Objectives and scope

The main objective of ENVELOPE is to advance and open up the reference 5G-Advanced architecture, and also to transform it into a vertical-oriented one with the necessary interfaces tailored to the CCAM vertical use cases that:

1. expose network capabilities to verticals;
2. provide vertical-information to the network; and
3. enable verticals to dynamically request and modify certain network aspects, in an open, transparent and easy to use way.

Given this architectural focus, ENVELOPE does not directly target the enhancement of lower-layer networking metrics. Instead, the project's evaluation approach assesses outcomes primarily from an API user's perspective. To support this, a specific set of KPIs tailored to ENVELOPE's objectives was defined in the deliverable D2.3 [1].

To comprehensively understand the performance of ENVELOPE's innovations and pinpoint the root causes of any potential service degradation, the project will conduct a fine-grained performance analysis. This approach is built on two main pillars:

- Multi-site capability assessment: ENVELOPE innovations will be evaluated across three distinct trial sites. These measurements will assess end-to-end, user-perceived, and application-specific performance within the context of the underlying network capabilities.
- Service impact analysis: The evaluation will pay particular attention to the impact of ENVELOPE innovations on the services deployed within the project's use cases. This involves establishing a clear cause-and-effect relationship to quantify exactly how the services are supported or enhanced.

In summary, the technical performance evaluation methodology is designed to fulfil two high-level objectives:

- Evaluate the performance of ENVELOPE innovations from the perspective of the API user.
- Assess the extent to which ENVELOPE use cases benefit from these innovations.

2.2.2 Overall approach

The technical evaluation methodology of the ENVELOPE project adapts established frameworks from the PoDIUM [2], ICT4CART [3], and 5G-IANA [4] projects. This continuity is facilitated by a shared project coordinator (ICCS) and common consortium partners across the initiatives. While these predecessor projects operate within the Cooperative, Connected and Automated Mobility (CCAM) domain and utilize comparable workplan structures, their procedural frameworks serve strictly as a baseline. Rather than directly replicating a prior model, ENVELOPE synthesizes these operational findings to develop a customized methodology, specifically tailored to its distinct requirements and large-scale pilot environment.

While ENVELOPE does not adopt the FESTA methodology [5] in its strict entirety, it strategically adapts the three core phases of the FESTA V-model to structure its technical evaluation lifecycle. This ensures a systematic progression from concept to measurable impact.

2.2.2.1 The three-phase evaluation lifecycle

ENVELOPE's technical performance evaluation methodology is divided into three phases, as depicted in Figure 2.

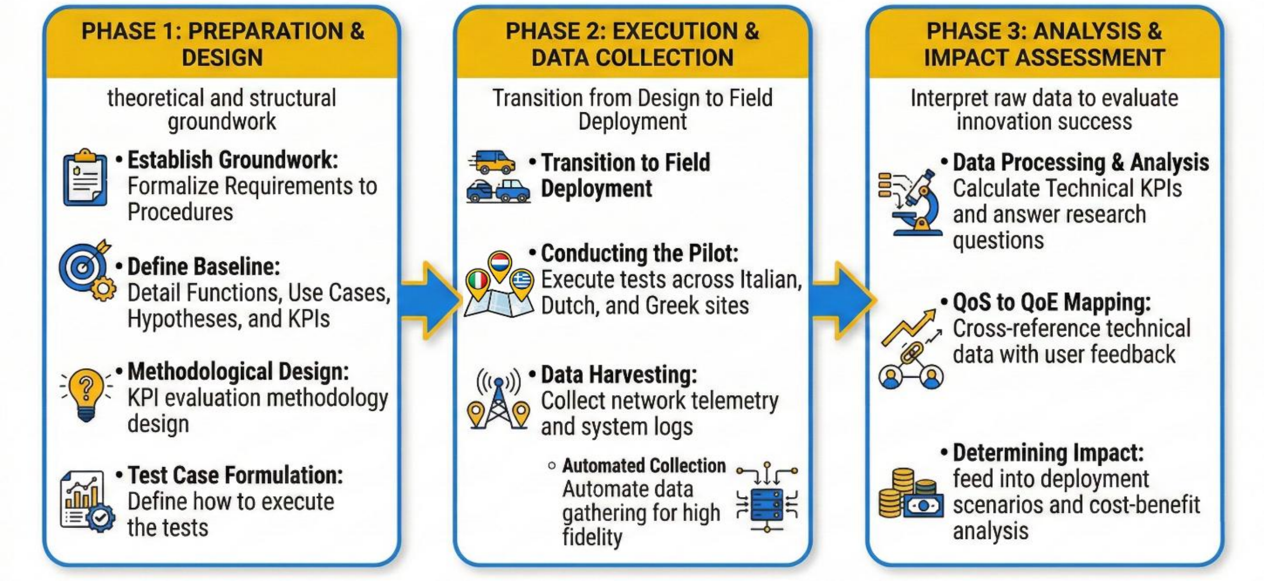

Figure 2 ENVELOPE technical performance evaluation methodology

- **Phase 1: Preparation and Design**

This phase establishes the theoretical and structural groundwork for the evaluation, moving from high-level requirements to actionable testing procedures (see Figure 3).

- Defining the Baseline: This involves formalizing the use cases, functions, research questions, and hypotheses. In ENVELOPE, this foundational work was established in WP2, with project use cases detailed in Deliverable D2.1 [6], ENVELOPE architecture and requirements in Deliverable D2.2 [7], and target KPIs formalized in Deliverable D2.3 [1].
- Methodological Design: Once the specific KPIs are defined, the KPI evaluation methodology is designed. The present deliverable D6.1 describes precisely this methodology.
- Test Case Formulation: Test cases must be defined to validate that the ENVELOPE innovations meet the intended architectural and performance requirements. The precise definition of these test cases constitutes a primary contribution of this deliverable and is comprehensively catalogued in Annex 1.

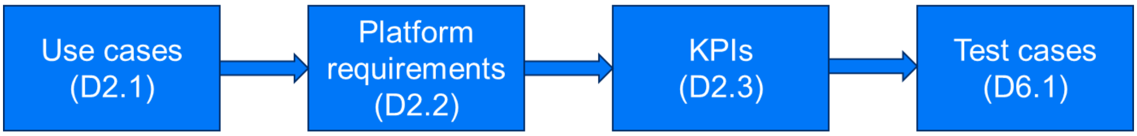

Figure 3 Steps followed to define the ENVELOPE test cases

- **Phase 2: Execution and Data Collection**

This phase transitions the methodology from design to active field deployment.

- Conducting the Pilot: Tests are executed across the designated trial sites (Italy, Netherlands, Greece). The monitoring and execution of the test cases should be automated as much as possible, especially when dealing with open APIs, as the ones used in the ENVELOPE Enablers. More information about the APIs used in the ENVELOPE Enablers can be found in the deliverables D3.2 [8] and D3.3 [9].
- Data Harvesting: This involves the active collection of network telemetry, system logs, and application-level data. The data collection must be automated wherever possible to ensure consistency, eliminate human error, and provide high-fidelity datasets for subsequent analysis. The specific data collection strategies implemented on each trial site are described in the deliverable D4.1 [10].

- **Phase 3: Analysis and Impact Assessment**

The final phase interprets the raw data collected during execution to evaluate the success of the innovations.

- Data Processing and Analysis: Raw data is cleaned, stored, and processed to calculate the final values of the defined Technical KPIs. These results are then used to answer the initial research questions.
- QoS to QoE Mapping: Technical KPIs primarily measure network Quality of Service (QoS). However, QoS directly dictates the user's perceived Quality of Experience (QoE). This technical data is subsequently cross-referenced with user feedback (gathered via surveys and interviews on a statistically significant sample, as detailed in the User Acceptance Evaluation Methodology in Section 2.3) to provide a holistic view of system performance.
- Determining Impact: Finally, the analysed data feeds into the broader impact assessment, deployment scenarios, and socioeconomic cost-benefit analyses.

2.2.2.2 Testing hierarchy and Test Suites

To ensure a structured and exhaustive evaluation, ENVELOPE's testing plan is organized into a strict hierarchy. The test plan is divided into distinct Test Suites, which act as logical containers grouping related test cases based on specific evaluation targets.

For the ENVELOPE project, there are four primary Test Suites:

1. Experimentation-as-a-Service (EaaS) System: Tests validating the performance of the EaaS platform.
2. ENVELOPE Enablers: Tests assessing the performance of specific CAM-oriented 5G/6G technological enablers.
3. Zero-Touch Management System: Tests verifying the automated, AI-driven network management and orchestration capabilities without human intervention.
4. UCs: Tests to evaluate how the ENVELOPE Innovations support or enhance the UCs. This test suite includes the tests to evaluate the inter-PLMN handover method.

2.2.3 Methodological approach for trial scalability

The scalability validation approach within the ENVELOPE evaluation methodology is systematically designed to assess how the project's innovations perform under progressively increasing levels of operational stress. To emulate realistic deployment conditions, the evaluation subjects the system to various workload types, measuring the system's ability to maintain its KPIs before reaching degradation boundaries.

To achieve this, the system under test is subjected to an incremental workload that increases iteratively based on a fixed workload step. Once the point of failure is reached, the test concludes, and final performance conclusions can be drawn. The methodological approach is depicted in Figure 4.

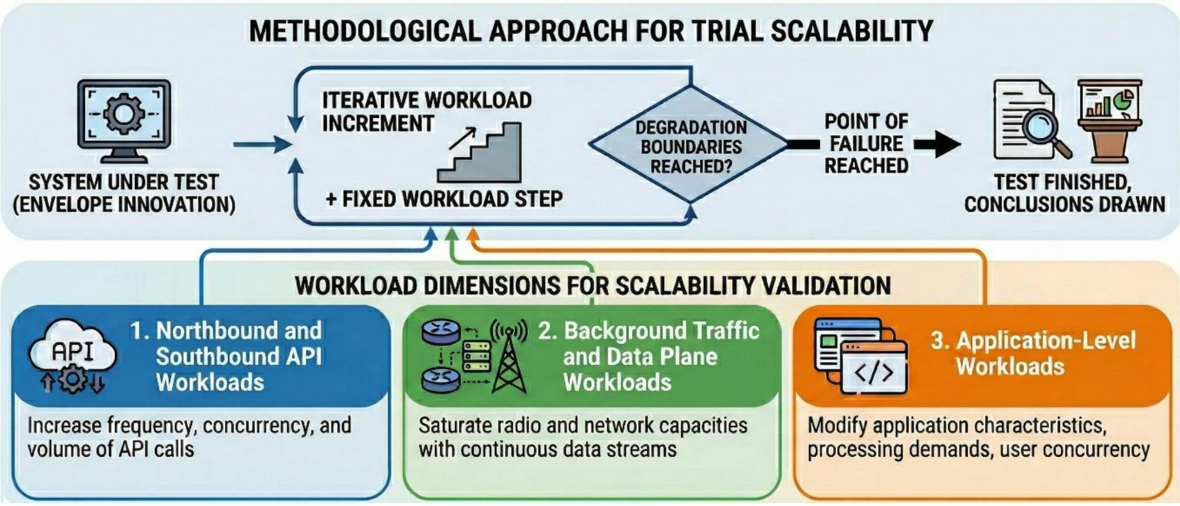

Figure 4 Methodological approach for trial scalability

2.2.3.1 Workload dimensions for scalability validation

The methodology categorizes scalability stress testing into three primary workload dimensions to ensure both the control plane and data plane are comprehensively evaluated. While Annex 1 details the specific stress-testing procedures for each ENVELOPE innovation, the following sections summarize concrete examples of the planned workloads for scalability validation.

2.2.3.1.1 *Northbound and southbound API workloads*

This category stresses the system by increasing the frequency and concurrency of API calls sent to the enablers. Increasing these API requests emulates the coexistence of multiple users.

- **Enabler Availability:** The system will perform periodic enabler API requests, starting at an initial load of 1 Hz and stepping up by increments of +10 Hz. The objective is to observe service behaviour due to a high volume of concurrent user requests.
- **pQoS Enabler:** API robustness will be tested through multiple concurrent inference requests, sequential training requests, and active subscriptions. API call frequencies will step increase by +10 Hz to stress the system. This evaluates how the enabler manages simultaneous predictive QoS demands from multiple applications and how the service degrades when the API is stressed.
- **ATSSS Enabler:** The load will be scaled by increasing the number of concurrent ATSSS-T API calls used for establishing ATSSS sessions across multiple UEs. Emulating multiple UEs in this manner aims to evaluate the system's behavior under stress.
- **Edge Enabler:** Testing will increase both the frequency and concurrency of northbound API calls. This includes generating high-frequency API calls for traffic influence (re-routing). The objective is to see how the edge routing service behaves and whether network performance degrades when the traffic influence API is overwhelmed by multiple concurrent application requests.
- **Device Location Enabler:** The approach will increase the frequency of location query and subscription requests (Northbound), starting at 1 Hz and increasing by +10 Hz. It will also increase the number of devices tracked in the system (Southbound) in increments of +10. This emulates a dense environment of tracked devices to observe 5GS performance and service responsiveness when location APIs are heavily stressed.
- **QoD Enabler:** The system will process concurrent Quality on Demand (QoD) API calls, scaling up the frequency by +10 Hz steps to stress the API layer. Similar to ATSSS, emulating multiple UEs requesting QoD simultaneously aims to check for the depletion of network resources and observe how the service acts when the API becomes unresponsive.
- **Performance Metrics Enabler:** The workload will scale by establishing and increasing the number of concurrent performance metric subscriptions across multiple UEs. The objective is to check how the telemetry system and 5GS behave when the performance metric API is saturated by simultaneous user requests.
- **Zero Touch Management (ZTM):** Testing will scale the Northbound API workloads by increasing the number of concurrent inputs (e.g., vehicles) provided to the Decision Engine, starting from 1 and stepping up by +4 concurrent inputs. This tests how the automated management service behaves when the ZTM API is stressed by the coexistence of multiple entities generating simultaneous inputs.

2.2.3.1.2 *Background traffic and data plane workloads*

This category aims to emulate realistic network congestion, radio contention, and varying throughput by introducing background data.

- **pQoS Enabler:** Background traffic from UEs will be applied to vary RAN load and emulate radio contention. This will step increase by +10 background UEs to create compounded operational stress.
- **ATSSS Enabler:** Workloads will scale by generating and progressively increasing background traffic from multiple UEs toward the 3GPP access (5G) network to simulate heavy traffic and congested conditions.
- **QoD Enabler:** The data plane will be stressed by generating continuous iPerf background traffic from multiple UEs until the radio capacity becomes saturated.
- **Inter-PLMN Handover (Greek Trial Site's UC6):** Concurrent use of the mobile network will be simulated using both physical and simulated UEs, increasing background traffic to emulate congested conditions during handovers.
- **Dutch Trial Site Use Cases (UC3, UC4 & UC5):** iPerf background traffic will be generated until radio capacity gets saturated.

2.2.3.1.3 Application-level workloads

This approach tests the system by modifying application characteristics, processing demands, and user concurrency.

- **EaaS Platform:** Scalability will be tested by increasing the frequency of application onboarding and experiment start requests (from 1 Hz to +5 Hz increments). It will also scale by increasing the application size (e.g., +250 MB steps) and the number of concurrent onboardings and running experiments (e.g., +5 increments).
- **pQoS Enabler:** Synthetic application traffic and varying server loads will be introduced to evaluate how application-level processing affects prediction timeliness.
- **ATSSS Enabler:** Testing will vary application-level workloads to artificially increase the CPU utilization and load on the UE (e.g., using sysbench) to evaluate the decision time of the Reinforcement Learning (RL) agent.
- **Edge Enabler:** The system will scale application lifecycle requests, such as onboarding, instantiation, termination, and purging, by issuing them in rapid succession or concurrently.
- **Inter-PLMN Handover (Greek Trial Site's UC6):** Workloads will scale by maintaining continuous MQTT message exchanges between UEs, stepping up the MQTT message frequency by increments of +10 Hz
- **Italian Trial Site Use Cases (UC1 & UC2):** Application workloads will scale by increasing the number of Autonomous Vehicles (UEs) actively sending sensor data (uplink) or downloading updated maps (downlink). This load will increment by +2 vehicles at a time.

2.2.4 Test case template

The Test Case is the foundational, smallest unit in the hierarchy of ENVELOPE's testing approach. To ensure standardization across all trial sites and work packages, every test case defined in Annex 1 adheres to a strict template. This test case template is based on the one used in the 5G-IANA project [11], with some modifications done inspired by the conformance testing defined in 3GPP TS 38.523-1 V17.6.0 [12], and the scalability concept described in Section 2.2.3.

Table 1 Test case template

Test Case ID	TC_xxx
ENVELOPE Innovation	EaaS, Enabler, Zero Touch Management or Inter-PLMN handover.
KPIs	Reference measured KPI(s).
Test purpose	Describe the objective of the test.
Scalability validation approach	Select the approach used in this test case to emulate large-scale operational conditions for scalability validation. Select at least one of the following complementary workload categories (ideally, address all three where applicable): • Background Traffic and Data Plane Workload • Application-Level Workloads • Northbound and Southbound API Workloads
Scalability configurations	Based on the selected workload categories above, define the parameters for incremental workload configurations used in scalability validation. Each test case must be executed under each of these configurations to evaluate system behaviour across varying scalability levels.
Pre-test conditions	List any setup or preconditions necessary to execute the test successfully. This might include initializing the system, configuring settings, or establishing specific starting conditions.
Test procedure sequence	List the steps or actions required to execute the test case. Provide clear instructions for carrying out the test, including any interactions with the system or inputs provided.
Evaluation criteria	Specify the criteria used to evaluate the correctness or success of the test. This must include comparing actual outcome against expected result (i.e. the target values defined in D2.3, for each KPI this test case measures).

2.2.5 Documentation and reporting

The evaluation documentation fulfils a dual function: (i) it provides a structured reference framework for the replication and extension of future test campaigns, and (ii) it constitutes a comprehensive and traceable record of the evaluation activities performed. Upon completion of data acquisition, the collected datasets will be subjected to a rigorous statistical analysis. The analytical process will combine descriptive and inferential statistical methods to ensure robustness and reproducibility of results. Findings will be presented through appropriate graphical representations to facilitate interpretability and transparency.

The primary objective of this analysis is to identify patterns, trends, deviations, and potential outliers with respect to the predefined Key Performance Indicators (KPIs). Within this framework, a KPI is defined as a scalar performance indicator that characterizes the behaviour of the system under test or the quality of the evaluated scenario. KPIs may be classified as continuous variables (quantitative metrics) or binary variables (pass/fail criteria). For KPIs defined as binary pass/fail criteria, the primary evaluation metric will be the percentage of successful outcomes (i.e., compliance rate).

The measured values will be systematically compared against the target thresholds defined for each KPI in order to assess compliance and performance levels. Where applicable, benchmarking will be conducted against relevant standards, reference implementations, or industry best practices. This comparative assessment will enable the identification of performance gaps, strengths, and areas requiring optimization.

2.2.5.1 Scalability assessment

Given the large-scale nature of the ENVELOPE project, a core outcome of this process is providing an evidence-based evaluation of the innovations, specifically highlighting their operational effectiveness, scalability, and limitations under the tested conditions. To properly assess scalability, testing will involve evaluating the system under progressively increasing workloads. The same statistical descriptors will be applied during these scale-up scenarios, ensuring methodological uniformity across all KPI types.

Based on the analytical results, consolidated conclusions will be drawn for each test case. These conclusions will assess the extent to which the ENVELOPE solutions meet their intended functional and non-functional requirements, including their ability to scale. Furthermore, targeted recommendations will be formulated to support effective deployment and operational integration. These recommendations will address identified shortcomings and propose technically grounded corrective or optimization measures to enhance performance in real-world scenarios.

2.2.5.2 Reporting template

To ensure methodological consistency and comparability across test cases, results and conclusions will be documented using a standardized table template (see Table 2) derived from the structure employed in 5G-MOBIX Deliverable D5.2, "Report on Technical Evaluation" [13]. Each table will summarize the evaluation results using descriptive statistical indicators and an associated graphical representation. To specifically evaluate the large-scale capabilities of the ENVELOPE project, the table is designed to track system degradation and resource consumption across defined scaling tiers.

Table 2 Template for the presentation of the technical evaluation results

<table>
<tr><th colspan="5">Test case identifier</th></tr>
<tr><td>Materials used</td><td colspan="4">Include specification and/or settings of the equipment used</td></tr>
<tr><td>KPI</td><td colspan="4"></td></tr>
<tr><td>Scalability Parameter</td><td colspan="4">Define the scaling factor (e.g., number of concurrent users).</td></tr>
<tr><td rowspan="2">Scalability Results</td><td>Load Level</td><td>Mean</td><td>Std. Deviation</td><td>95th Percentile</td></tr>
<tr><td>Baseline
…
Stress limit</td><td></td><td></td><td></td></tr>
</table>

Plot	*Include a line chart showing KPI performance vs. Load Level*
Conclusions	*Discuss the system's performance boundaries, point of degradation (bottlenecks), and maximum supported scale before breaching acceptable KPI thresholds.*

2.3 User acceptance evaluation

2.3.1 Objective and scope

The User Acceptance Evaluation assesses how stakeholders perceive the ENVELOPE platform and its associated Use Cases (UCs) in terms of usability, usefulness, trustworthiness, and adoption potential. This evaluation complements the technical KPI assessment presented in Section 2.2 by introducing a human-centric dimension to performance evaluation.

While KPIs quantify measurable technical performance (e.g., scalability, latency, reliability), user acceptance focuses on perceived value and behavioural intention to use the ENVELOPE platform in realistic operational environments. The objective is not to derive statistically representative conclusions for a broader population, but to obtain structured and methodologically grounded feedback from stakeholders directly exposed to the platform during trial activities and Open Call experimentation.

The results of this evaluation will contribute to assessing adoption readiness and will support the Proof-of-Value analysis conducted under Task 6.4.

2.3.2 Conceptual framework

The methodology is grounded in the Technology Acceptance Model (TAM)[1], which explains user adoption of information systems through two primary determinants:

- **Perceived Ease of Use (PEOU)**: the degree to which a person believes that using a system would be free of effort.
- **Perceived Usefulness (PU)**: the degree to which a person believes that using a system would enhance their performance.

According to TAM, these constructs influence behavioural intention to use, which in turn affects actual adoption. The model has been extensively applied in evaluating emerging digital platforms, enterprise systems, and advanced technological infrastructures, making it particularly suitable for assessing adoption readiness.

Given the nature of ENVELOPE, the classical TAM framework is extended with additional constructs relevant to mission-critical and mobility-oriented systems:

- Willingness to use and recommend the platform
- Social influence and organisational support

[1]Davis, F.D., User acceptance of information systems: the technology acceptance model (TAM). 1987

- Trust and security perception (particularly data handling and GDPR compliance)

A Likert-scale is being used (five-point scale ranging from “Strongly Disagree” to “Strongly Agree”). Questions are formulated in relation to the specific UC or functionality that the respondent has tested or experienced.

1. **Perceived Ease of Use (PEOU)**

This dimension evaluates how intuitive and understandable the system is. Example question formulations include:

- “The platform was easy to use and learn.”
- “My interaction with the platform was clear and understandable.”
- “I find it easy to get the platform to do what I want.”

These items assess cognitive effort, clarity of interface design, and operational transparency.

2. **Perceived Usefulness (PU)**

This dimension evaluates whether stakeholders perceive measurable benefits in task performance or operational efficiency. Example question formulations include:

- “The ENVELOPE services improve my task performance.”
- “Using the platform enables me to accomplish tasks more quickly.”
- “The platform is reliable and consistent.”

These items capture perceived productivity enhancement, reliability improvements, and added value in professional contexts.

3. **Behavioural Intention / Willingness to Use**

Behavioural intention is assessed through items such as:

- “I would use this platform regularly.”
- “I would recommend this solution to others in my field.”
- “I believe this platform has the potential for widespread adoption.”

These questions provide insight into future adoption likelihood and diffusion potential.

4. **Social Influence**

Given that ENVELOPE operates in institutional and organisational environments, social norms and organisational support are also examined. Example items include:

- “People I consider knowledgeable see value in the platform.”
- “My organisation supports the use of this platform.”

This construct evaluates whether the ecosystem environment facilitates or hinders adoption.

5. Trust and Security Perception

Considering that ENVELOPE services involve data exchange and decision support, trust is a critical adoption factor. Example items include:

- "I trust that my data is handled securely."
- "I am aware of how my data is being used."
- "I believe the platform complies with privacy standards such as GDPR."

Open-ended questions complement the multiple-choice questions by allowing respondents to identify perceived benefits, improvement suggestions, and potential legal, ethical, or technical barriers to adoption.

The User Acceptance Survey follows a structured design aligned with the Technology Acceptance Model and its extensions. The overall structure of the questionnaire is presented in Table 3.

Table 3 Overall structure of the User Acceptance Survey.

Section	Description	Constructs Covered	Type of Questions
Section 1 – General Information	Collects contextual information about the respondent, including trial site, use cases evaluated, involvement in Open Call projects, professional role, and occupation	Respondent profile and contextual variables	Multiple choice and open-ended
Section 2 – Willingness to Use and Recommend	Assesses the respondent's intention to use the ENVELOPE platform, recommend it to others, and their perception of its potential for widespread adoption	Behavioural intention to use and adoption potential	Five-point Likert scale
Section 3 – Perceived Usability and Ease of Use	Evaluates how easy the platform is to learn, understand, and interact with	Perceived ease of use	Five-point Likert scale
Section 4 – Perceived Usefulness	Assesses whether the platform improves task performance, supports professional activities, increases efficiency, and provides a reliable and user-friendly experience	Perceived usefulness	Five-point Likert scale
Section 5 – Social Influence and Organisational Support	Examines whether peers, experts, and organisations support or value the use of the platform	Social influence and organisational support	Five-point Likert scale
Section 6 – Trust and Security Perception	Evaluates user trust in data handling, awareness of data usage, and compliance with privacy regulations such as the General Data Protection Regulation	Trust and security perception	Five-point Likert scale

Section 7 – Open Feedback on ENVELOPE	Collects qualitative feedback on useful features, suggested improvements, and potential legal, ethical, or technical barriers	Qualitative insights on benefits, challenges, and adoption barriers	Open-ended
Section 8 – Open Feedback on the Questionnaire	Gathers feedback on the clarity, structure, and completeness of the survey itself	Survey quality assessment	Open-ended

2.3.3 Target groups and sampling approach

The survey targets stakeholders who have directly interacted with or evaluated ENVELOPE functionalities. These include end-users participating in trial site activities, project stakeholders with hands-on involvement, and Open Call participants deploying ENVELOPE capabilities.

The sampling approach follows a case-based design, where each trial site and associated UC is treated as an analytical unit. The intention is to ensure that each site is represented by a sufficient number of responses to enable meaningful descriptive analysis and cross-site comparison.

To ensure consistency in stakeholder identification across trial sites, a common stakeholder classification is adopted based on the roles defined in the survey instrument. The following stakeholder categories are considered:

- End-users (e.g., drivers, mobility service users, operators interacting with CAM services).
- Car manufacturers
- Road or motorway operators
- Public authorities and public companies
- Project stakeholders (consortium members involved in implementation and validation)
- Open Call participants deploying or testing ENVELOPE solutions
- General public participants involved in pilot demonstrations
- Technical experts (e.g., developers, system integrators, engineers)
- Business experts (e.g., innovation managers, exploitation and strategy stakeholders)

To support a structured and balanced distribution of the survey, a stakeholder mapping process will be conducted. Trial site leaders, in coordination with T6.4 task leader, will identify the appropriate stakeholder categories, contact points, and corresponding Use Cases. This mapping will define which type of questionnaire (e.g., User Acceptance, Impact Assessment, or Expert Assessment) is applicable to each stakeholder group. The mapping document will remain dynamic and may be updated as additional stakeholders become involved, for example following pilot events or Open Call evaluation results.

A minimum threshold of approximately 10 completed questionnaires per trial site or UC is considered appropriate to ensure statistically meaningful analysis. Where feasible, additional responses are encouraged in proportion to the number of actively involved stakeholders. In cases where participation is limited due to the scale or maturity of the trial, results will be interpreted as indicative and contextualised accordingly.

2.3.4 Data collection procedure

The User Acceptance Survey will be implemented through a structured online questionnaire. Distribution is coordinated via trial site leaders and relevant task leaders, based on a mapping file that identifies the applicable UCs, stakeholder categories, and responsible dissemination partners. Respondents are instructed to answer questions only in relation to the specific UC or functionality they have tested or experienced. Sections that are not applicable to a respondent's context may be left unanswered.

2.3.5 Data analysis method

The analysis relies primarily on descriptive statistical techniques. For each construct (e.g., Perceived Ease of Use, Perceived Usefulness, Trust), mean values and standard deviations will be calculated at trial site level. Where appropriate, aggregated comparisons across sites will be performed to identify trends and variations in acceptance levels. Given the applied and pilot-oriented nature of the project, inferential statistical testing is not the primary objective. Instead, the analysis aims to identify patterns, strengths, and potential adoption barriers that may affect scalability and market uptake. Qualitative responses will be analysed thematically to extract recurring concerns, perceived advantages, and recommendations. These findings will complement the quantitative results and support the overall interpretation of user acceptance within the ENVELOPE ecosystem.

2.3.6 Conclusion and next steps

The User Acceptance Evaluation introduces a structured and theory-based approach to assessing the perceived value and adoption readiness of the ENVELOPE platform and its associated Use Cases. By grounding the methodology in the Technology Acceptance Model and adapting it to the specific characteristics of B5G-enabled mobility services, the project ensures that technical performance indicators are complemented by a behavioural and trust-oriented assessment. This integrated perspective is essential for evaluating not only whether the platform performs as expected, but also whether it is perceived as usable, reliable, and beneficial by its intended stakeholders. Such insights are critical for supporting scalability, sustainability, and long-term exploitation of the project's results.

In the next phase, the methodology described in this section will be operationalised through structured online data collection across trial sites and Open Call activities. The collected evidence will contribute to the Proof-of-Value assessment under Task 6.4 and will inform the overall impact evaluation of ENVELOPE. The results will also provide feedback for refining adoption strategies and strengthening the alignment between technological innovation and stakeholder expectations.

3 ENVELOPE's KVI evaluation methodology

The evaluation of Key Value Indicators (KVIs) in ENVELOPE follows a structured, multi-layered assessment framework that complements KPI-based technical validation with value-oriented impact analysis. While KPIs quantify system performance, KVIs assess how those technical achievements translate into societal, environmental and economic outcomes, as defined in Deliverable D2.3 [1], and aligned with the SNS JU perspective on value-driven evaluation.

3.1 Overall approach

The ENVELOPE KVI evaluation methodology is based on four core principles, as represented in Figure 5, to ensure that value assessments are robust, traceable and contextually relevant to the technologies developed in ENVELOPE and its Use Cases:

1. **Alignment with sustainability dimensions:** ENVELOPE KVIs are classified into societal, environmental or economic categories, following the approach in the 6G-IA White Paper "What societal values will 6G address? Societal Key Values and Key Value Indicators analysed through 6G use cases" [14]. These categories reflect the sustainability perspective consistent with the United Nations Sustainable Development Goals framework [15], and ensure a balanced and comprehensive evaluation that accounts for diverse impacts, such as reduced emergency response times (societal), decreased energy consumption (environmental), and operational costs savings (economic).
2. **Traceability to technical enablers:** Every KVI is explicitly linked to specific ENVELOPE innovations. The definition of the KVIs, established in Deliverable D2.3 [1], already provides a clear association between each KVI and the corresponding ENVELOPE innovation that enables it.
3. **UC-centric evaluation:** Overall, value materializes in operational contexts rather than at platform level. Consequently, KVIs are assessed on a per-UC basis, where interaction between technology and end-user behaviour can be observed in a real-world context.
4. **Mixed-method validation strategy:** To ensure an appropriate and reliable assessment of each KVI, a dedicated validation method has been assigned to each indicator based on its nature, measurement feasibility and impact horizon. This assignment, which has been defined in Deliverable D2.3, ensures that each KVI is evaluated using the most suitable validation strategy among the three approaches defined below.
    a. **Quantitative empirical evaluation:** This is the primary method to assess KVIs that can be expressed through measurable, objective and reproducible parameters.
    b. **User surveys:** This method is used to assess the subjective and human-centric nature of some KVIs, such as those based on societal values, which cannot be captured solely through technical instrumentation.
    c. **Expert assessment:** This approach provides a structured evaluation pathway for KVIs that cannot be directly captured by technical parameters, due to either earlier Technology Readiness Levels (TRL) or limited market penetration that prevents immediate observation of systemic impact.

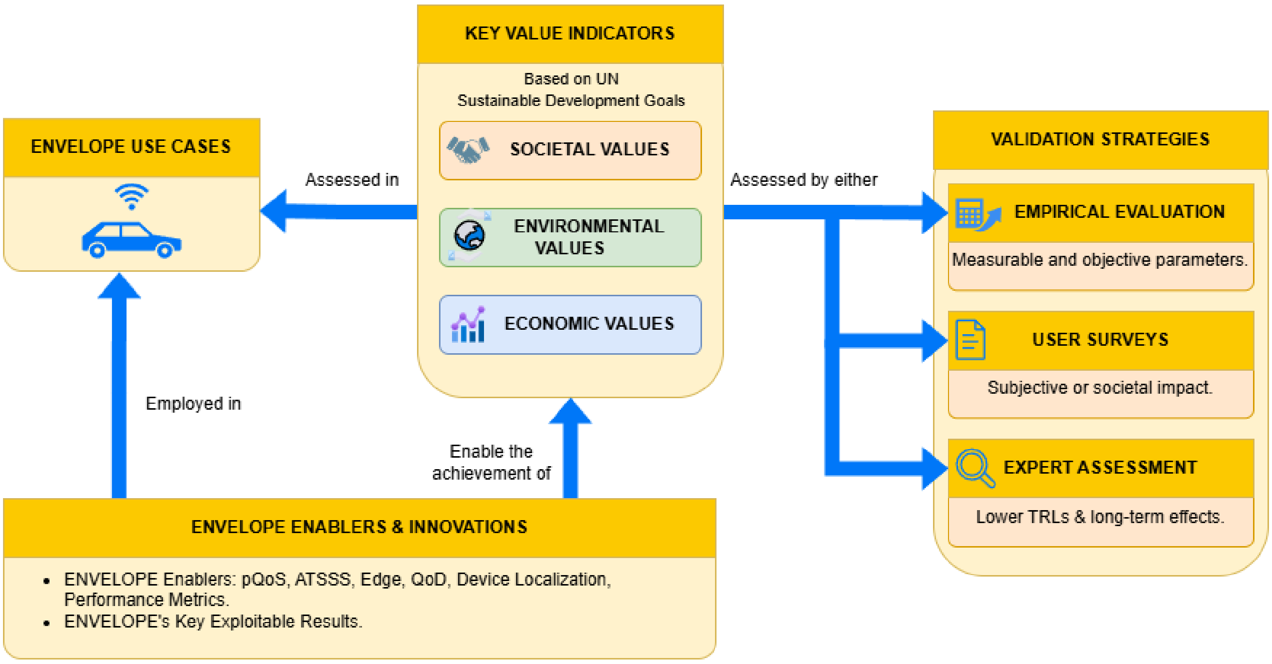

Figure 5 ENVELOPE's Key Value Indicators evaluation methodology

For reference, Table 4 summarizes the ENVELOPE KVIs, the Use Cases in which they will be assessed, and the validation strategy assigned to each indicator, as established in Deliverable D2.3.

Table 4 Summary of ENVELOPE KVIs, corresponding UCs and validation methods

KVI identifier	KVI category	UC(s)	Validation strategy
KVI_S_TrustInCCAM	Societal	Dt-UC4, Dt-UC5, Gr-UC6	User Surveys
KVI_S_DigitalAccessibility	Societal	All UCs	User Surveys
KVI_S_ EmergencyResponseTime	Societal	It-UC1, Dt-UC5	Expert Assessment (It-UC1) Empirical Evaluation (Dt-UC5)
KVI_Env_ ResourceUsageOptimization	Environmental	It-UC1, It-UC2	Expert Assessment
KVI_Env_EnergyConsumption	Environmental	Dt-UC5	Empirical Evaluation
KVI_Econ_ SustainableDevelopment	Economic	All UCs	User Surveys
KVI_Econ_ OperationalCostSavings	Economic	It-UC1, It-UC2, Dt-UC4	Expert Assessment
KVI_Econ_ TimeToMarketReduction	Economic	All UCs	User Surveys
KVI_Econ_ServiceAdoption	Economic	All UCs	User Surveys

A key element of the ENVELOPE's KVI evaluation methodology, regardless of the applied validation strategy, is the establishment of a reference baseline. The baseline represents a comparable configuration against which the ENVELOPE-enabled solution is evaluated. It may correspond to a legacy network architecture, a non-optimised deployment, or a pre-existing operational condition. While some ENVELOPE KVIs have target values defined in absolute terms, others are specified relative to baseline performance (e.g., as a percentage reduction in cost or energy consumption). In such cases, the baseline definition, which is already described at a high level in Deliverable D2.3, is documented in detail to ensure transparency and avoid ambiguity in the interpretation of impact results.

3.2 Quantitative empirical evaluation

Quantitative empirical evaluation constitutes the primary method for assessing KVIs that can be expressed through measurable, objective and reproducible parameters. This approach ensures that value-oriented indicators are grounded in experimentally verified system behaviour and that the claimed benefits or values are supported by demonstrable evidence.

Within ENVELOPE, quantitative KVI evaluation builds upon the three-phase evaluation lifecycle established for technical KPIs in Section 2.2.2.1: (1) preparation and design, (2) execution and data collection, and (3) analysis and impact assessment. This alignment ensures methodological coherence between performance evaluation and value-oriented assessment within ENVELOPE. By leveraging the established technical foundation, ENVELOPE's quantitative KVI assessment translates technical performance metrics into higher-level impact indicators representing value generation at system and use case level.

- **Phase 1: Preparation and Design**

In the preparation and design phase, the foundational requirements for the corresponding KVIs are established. These include KVI definition, evaluation criteria, relevant technical enablers and the Use Cases in which the indicators are assessed. The definition and formalisation of KVIs were performed in Deliverable D2.3, where KVIs were aligned with project Use Cases and developed enablers and innovations.

- **Phase 2: Execution and Data Collection**

In the execution and data collection phase, quantitative KVI data is acquired through experimental validation activities. Data is collected during field trials conducted at the designated trial sites and Use Cases, through controlled laboratory experimentation or operational field testing. Where possible, data acquisition is automated to ensure measurement consistency, reduce human-induced bias, and guarantee high-fidelity datasets for subsequent analysis.

- **Phase 3: Analysis and Impact Assessment**

The final phase focuses on processing and interpreting the collected data to derive value-oriented impact metrics. Raw measurement data is first processed and validated, and then used to compute final KVI values. Statistical analysis is performed to characterise data distribution, variability and reliability of the measurements, and to ensure robustness and transparency of the results.

For reporting purposes, and to ensure uniformity in result presentation, the standardized tabular template introduced in Section 2.2.5 has been adapted to present the results of KVIs assessed through quantitative empirical evaluation. The reporting table, as shown in Table 5, will include key statistical descriptors of the collected data, such as mean values, standard deviation, the 95% confidence interval and the 95th percentile. The inclusion of these statistical indicators provides a transparent representation of central tendency, variability and distribution characteristics, thereby strengthening the credibility and interpretability of the quantitative KVI assessment.

Table 5 Template for presentation of KVIs assessed through quantitative empirical evaluation

KVI Identifier				
Materials used	*Include specification and/or settings of the equipment used*			
Target value	*Based on the KVI definition in Deliverable D2.3. If applicable, describe the baseline configuration and/or values used for comparison.*			
Results	**Mean**	**Std. Deviation**	**95th Percentile**	**95% confidence interval**
Conclusions	*Discuss measured results, improvement with respect to baseline conditions (if applicable), and how ENVELOPE innovations and technical enablers contribute to value generation for the evaluated KVI.*			

Through this structured methodology, quantitative empirical evaluation ensures that ENVELOPE's KVIs are supported by measurable evidence, analytically derived and statistically validated. This approach reinforces the credibility of the project's sustainability claims by demonstrating how technical performance improvements translate into tangible societal, environmental and economic value outcomes under controlled and documented conditions.

3.3 User surveys

While quantitative empirical evaluation focuses on system-level measurable metrics, user surveys provide a human-centric perspective on perceived value creation, technology acceptance and behavioural adoption potential. User surveys are used to assess KVIs that capture subjective, behavioural, and societal impact dimensions that cannot be directly measured through technical instrumentation. This validation strategy is particularly relevant for indicators related to accessibility, trust and adoption potential of ENVELOPE-enabled services, which can only be assessed through the direct feedback of stakeholders.

Therefore, the objective of this evaluation strategy is to obtain structured and reliable feedback from stakeholders directly exposed to ENVELOPE technologies during trial activities, demonstrations and Open Call projects. Given this significant overlap between the target groups for survey-based KVI evaluation and user acceptance assessment, the same respondent profiles are considered appropriate for both analyses.

Consequently, and for reasons of methodological coherence, simplicity of implementation, and to minimise response burden for participants, a single integrated questionnaire is used to assess both User Acceptance indicators and survey-based KVIs. This approach avoids duplicating surveys for the same stakeholders, increases the likelihood of higher response rates, and ensures consistency in how perception-related data is collected and interpreted across different evaluation dimensions. Consequently, the data collection procedure (including sampling approach, questionnaire deployment and data protection measures) follows the same framework described for the User Acceptance Evaluation.

Survey design for KVI evaluation follows a structured questionnaire-based approach, where questions are formulated to be concise, unbiased and directly related to the evaluated KVI and the specific ENVELOPE enabler or innovation experienced by each respondent. Following the same approach as the questionnaires for user acceptance evaluation in Section 2.3, the questions for KVI evaluation employ a Likert-scale response format (i.e., a five-point scale ranging from "strongly disagree" to "strongly agree"), to enable statistical aggregation and comparison across trial sites and use cases.

For all survey-based KVIs, a target value of 80% agreement was defined in Deliverable D2.3. This means that at least 80% of respondents are expected to select either "agree" or "strongly agree" with the statement associated with each specific KVI (e.g., improved trust, increased accessibility, accelerated innovation adoption, etc.). Therefore, in addition to the descriptive statistical analysis described in the User Acceptance methodology (e.g., distribution analysis and mean score calculation), the survey-based KVI evaluation also calculates the proportion of respondents who select "agree" or "strongly agree" for each of the corresponding KVI statements. The resulting percentage is then compared against the predefined 80% target threshold to determine whether the indicator has been successfully achieved.

Through this structured survey-based evaluation methodology, perception-oriented KVIs are supported by structured stakeholder evidence, systematically collected and statistically analysed. This approach strengthens the credibility of ENVELOPE's impact assessment by demonstrating how stakeholder perceptions regarding trust, accessibility and adoption potential reflect the broader societal and operational value generated by ENVELOPE under real experimentation and deployment conditions.

3.4 Expert assessment

Expert assessment constitutes a structured and methodologically rigorous evaluation strategy for KVIs that cannot be directly quantified through empirical measurements or reliably captured through user surveys. This situation may arise when technological components are at an early Technology Readiness Level, when deployment scale is insufficient to observe measurable societal or economic impact, or when the assessed value concerns long-term or systemic effects that extend beyond the project's experimental timeframe. In such cases, expert-based evaluation enables a reasoned and transparent estimation of expected impact, grounded in domain knowledge, analytical reasoning and comparative assessment.

Unlike quantitative and survey-based validation strategies, expert assessment primarily focuses on structured reasoning, evidence-supported judgement, and scenario-based impact estimation. The objective is not to replace empirical validation, but to complement it by providing analytical insight into value creation mechanisms that cannot be directly measured yet.

ENVELOPE adopts a systematic approach to expert-based evaluation:

- **Delphi Method-inspired assessment** [16]**:** When appropriate, a Delphi-style process may be used, in which experts provide independent assessments over multiple iterations. Between rounds, anonymised aggregated feedback is received to promote convergence and reduce individual bias. This iterative approach improves reliability and consistency of expert judgements.

- **Comparative benchmarking analysis:** In contexts where direct impact measurement is not feasible, expert evaluation may rely on structured benchmarking against comparable technological initiatives, deployments or operational environments. Experts analyse available performance, adoption or impact indicators from comparable initiatives and assess the degree of similarity with the ENVELOPE context. The considered reference initiatives include publicly available reports from European telecommunications operators, international organisations and peer-reviewed scientific literature evaluating technologies analogous to the ENVELOPE enablers in terms of societal, environmental and economic impact. For instance, existing studies indicate that zero-touch automation can increase operational efficiency by up to 20% [17], reduce effort in engineering tasks by up to 80% [18], and achieve network operational expenditure (OpEx) savings of approximately 18% [19]). Differences in scale, technological architecture or regulatory environment are explicitly identified and incorporated into the impact estimation process. This benchmarking approach enables reasoned extrapolation of potential value outcomes while maintaining methodological rigour. It also ensures that expert assessments are anchored in documented evidence rather than speculative projections.
- **Deductive impact reasoning and modelling:** For higher-level societal, environmental or economic outcomes, experts may construct causal reasoning models based on established theoretical relationships, operational principles or recognised performance-impact correlations. The process begins with clearly defined premises derived from measured KPIs, validated architectural characteristics or documented system capabilities. From these premises, experts derive structured inferences regarding the expected magnitude and direction of value impact. Each step in the reasoning chain is explicitly articulated, including intermediate variables and underlying assumptions. This approach strengthens analytical robustness by grounding impact estimation in explicit reasoning rather than implicit judgement.

Experts are required to provide structured justification accompanying their assigned scores, detailing assumptions, boundary conditions and contextual considerations. This narrative component ensures interpretability and facilitates subsequent review or sensitivity analysis. When consolidating multiple expert inputs, aggregation methods are selected to minimise distortion caused by outliers or asymmetric distributions. Measures of dispersion are reported alongside central tendency indicators to reflect the degree of consensus achieved. Where divergence remains significant, this variability is explicitly acknowledged and discussed as part of the evaluation outcome.

To preserve methodological integrity, several quality assurance measures are applied throughout the expert assessment process. Experts will be selected from within the project consortium and selection criteria include proven domain expertise, prior involvement in similar European initiatives or research activities, and the ability to provide a multidisciplinary perspective. Identified experts will be invited to participate through a coordinated engagement process, which may include dedicated workshops, structured questionnaires, and iterative consultation rounds (e.g., Delphi-inspired exercises). This approach ensures both the availability of high-quality expertise and the practical feasibility of conducting the assessment within the project framework. Conflict-of-interest declarations are collected when appropriate. Independence of judgement is maintained through controlled information sharing and anonymised response handling.

The results of the expert assessment evaluation are reported using a structured template designed to ensure transparency, traceability, and consistency in the presentation of impact estimations. Table 6 summarises the main evaluation elements, including the KVI identifier, evaluation method, expert profile, supporting evidence, and interpretation of results, providing a clear overview of the expert-based KVI assessment outcomes.

Table 6 Template for presentation of KVIs assessed through expert assessment

KVI Identifier	
Expert Evaluation Method	*Method used for evaluation: Delphi method, comparative benchmarking analysis, or deductive impact modelling.*
Target value	*Based on the KVI definition in Deliverable D2.3. If applicable, describe the baseline configuration and/or values used for comparison.*
Expert's domain competence	*Brief description of each expert's competence profile demonstrating expertise across relevant technical, societal, and/or economic domains required for KVI assessment.*
Expert Assessment Result	*Quantitative or qualitative estimate of the KVI provided by the expert panel.*
Supporting Evidence and Justification	*Supporting rationale depending on the evaluation method:* • *Delphi Method: statistical analysis of expert responses (mean, median, minimum and maximum values, and standard deviation).* • *Comparative benchmarking: References from comparable initiatives, studies or deployments. Justification of relevance is required, including discussion of similarities in scale, architecture, operational context, etc. Where possible, supporting literature, project results or publicly available performance data should be referenced.* • *Deductive Modelling: Structured logical reasoning linking technical performance premises to expected impact outcomes. The reasoning chain should explicitly identify intermediate variables, causal relationships, and modelling assumptions used to derive the estimated KVI value.*
Confidence level	*Expert confidence in the assessment result, based on evidence availability and level of expert agreement. Confidence level may be either Low, Medium, or High.*
Conclusions	*Interpretation of the expected ENVELOPE impact and key insights derived from expert evaluation.*

Through these structured mechanisms, expert assessment within ENVELOPE provides a robust and transparent methodology for evaluating KVIs in contexts where direct measurement is impractical. Rather than replacing empirical validation, it complements quantitative and survey-based approaches, ensuring that all relevant dimensions of societal, environmental and economic impact can be systematically assessed within the project's timeframe and technological maturity constraints.

4 Evaluation of Open Call projects

4.1 Overall approach

In addition to the evaluation of the technical performance, user acceptance and impact of the enablers and Use Cases developed by ENVELOPE partners, this deliverable also defines a common evaluation methodology for the assessment of Open Call (OC) projects. This methodology is applied consistently across all Open Call activities, including both Open Call #1 and Open Call #2, ensuring harmonised monitoring, validation and impact assessment of externally developed solutions integrated into the ENVELOPE ecosystem.

The objective of this common methodology is to guarantee that Open Call experiments are evaluated under comparable conditions, while maintaining alignment with the overall technical, operational and impact objectives of the project. Furthermore, the adoption of a shared methodology ensures that results obtained from Open Call projects contribute to a comprehensive assessment of the ENVELOPE platform capabilities and overall service-level value creation.

To ensure that the experiments carried out by Open Call participants are properly monitored, validated and aligned with the project objectives, ENVELOPE establishes a structured mentoring and evaluation framework. Each selected Open Call project is supported by designated mentors from the ENVELOPE consortium who provide technical guidance and oversee the progress of the experiment throughout its lifecycle.

The mentoring process includes an initial kick-off meeting between the Open Call teams and the relevant trial site partners, followed by regular technical meetings throughout the project execution. This support includes periodic coordination meetings, as well as additional ad-hoc meetings to address technical challenges to integration with the ENVELOPE platform or trial environments. This continuous interaction ensures that Open Call experiments remain aligned with ENVELOPE's trial site capabilities and evaluation objectives, while also enabling early identification and mitigation of potential technical or organisational risks.

4.2 Open Call monitoring and deliverable-based evaluation

The execution of each Open Call project follows a structured evaluation process consisting of three main phases, each associated with the submission and evaluation of a specific deliverable. This staged approach ensures continuous monitoring of the Open Call projects' progress and enables the timely verification of the experiment's technical implementation and validation activities. The evaluation phases are summarised in Table 7.

Table 7 Evaluation phases and deliverables for ENVELOPE Open Call projects

Phase	Deliverable	Description	Due Month
Kick-off	-	Initial meeting between trial sites and Open Call teams	M24 (OC1) / M28 (OC2)
Phase 1	D1 – Detailed Experiment Description and Implementation/Execution Plan	Definition of experiment design, implementation and validation plan, and project's KPIs and KVIs.	Month 1 after OC start

Phase 2	D2.1 – Intermediate Implementation Report on sub-project implementation	Progress report describing experiment execution and preliminary results.	Month 5 after OC start
Phase 3	D2.2 – Final Report and Experiment Results	Final results and validation outcomes of the experiment.	Month 6 after OC start

Each deliverable is reviewed by the assigned mentors and evaluated according to predefined criteria (see Section 4.4). The outcome of the evaluation determines whether the corresponding payment is approved.

This staged monitoring approach ensures that Open Call experiments remain aligned with the ENVELOPE objectives and that the proposed solutions are implemented and validated according to the agreed experimental plan.

4.3 Open Call experiment definition and KPI/KVI design

Each Open Call project is responsible for defining its own experimental design, including the description of the proposed solution, integration approach, validation activities and evaluation metrics. These elements are formally documented in the first deliverable submitted by each Open Call project: "D1 – Detailed Experiment Description and Implementation/Execution Plan".

This document describes the objectives of the experiment, the technical components involved, the integration with the ENVELOPE platform and the expected outcomes and benefits. In addition, Open Call participants define a set of KPIs and KVIs, which are used to evaluate the performance and impact of the proposed solutions. To support this evaluation, ENVELOPE provides access to its exposure APIs and monitoring platform, enabling Open Call participants to systematically collect and analyse the data for KPI and KVI assessment.

The proposed KPIs and KVIs, together with their associated data collection methodologies, are reviewed by the ENVELOPE mentors during the initial evaluation stage to ensure that the indicators are relevant, measurable and aligned with the overall objectives of the ENVELOPE experimentation framework.

4.4 Evaluation criteria and scoring methodology

The evaluation of Open Call deliverables is performed using a structured milestone evaluation form, which is used internally by ENVELOPE mentors to assess the quality of and progress of each project. This evaluation framework considers three main assessment dimensions:

1. **Deliverable Quality: 30%**
   This criterion evaluates the overall quality, completeness and clarity of the submitted deliverable. The assessment verifies whether the document aligns with the agreed mentoring plan, includes all required elements, demonstrates sound technical reasoning and professional presentation, and shows progress toward achieving technical milestones and goals.
2. **Technical Performance Indicators: 60%**
   This criterion evaluates the achievement, soundness and feasibility of the technical objectives defined in the project's experimental plan.

3. **Deadline Compliance: 10%**
   This criterion assesses the project's adherence to the agreed timeline, including timely submission of deliverables and completion of milestones.

Each evaluation criterion is scored on a 0-10 scale, with a minimum threshold of 5/10 for each individual criterion. The final evaluation score is calculated as a weighted aggregation of the three components. To successfully pass an evaluation stage, the project must obtain an overall score above the minimum acceptance threshold of 7/10.

Annex B presents the evaluation form provided to the Open Call mentors for the evaluation of the deliverables submitted by the Open Call participants.

4.5 Outcome of the evaluation process

The results of the deliverable evaluations determine both the continuation of the Open Call project within the programme and the release of the corresponding funding instalments. Projects that successfully meet the evaluation thresholds proceed to the next stage and continue receiving mentoring support from the ENVELOPE consortium. Projects that do not reach the thresholds may be subject to additional review by the ENVELOPE Selection Committee, which determines whether the project may continue in the programme or should be excluded.

Beyond monitoring project progress, the evaluation process also serves to ensure that the results produced by the Open Call experiments are credible, well-documented and aligned with the broader ENVELOPE experimentation objectives. The validated outcomes of these projects contribute to the overall assessment of the ENVELOPE architecture by demonstrating additional application scenarios, technological extensions and innovation opportunities enabled by the platform.

5 Conclusions

The proposed KPI and KVI evaluation methodologies will guide the evaluation of technical performance and societal, environmental, and economic impacts, ensuring a comprehensive understanding of the project's advancements on 1) exposing network capabilities to verticals; 2) providing vertical information to the network; and 3) enabling verticals to dynamically request and modify key network aspects.

The KPI evaluation methodology encompasses three primary domains: a project-level assessment of high-level outcomes; a technical performance evaluation driven by trial experimentation; and a user acceptance evaluation grounded in the Technology Acceptance Model (TAM). The technical performance evaluation focuses on four evaluation targets: the EaaS system, the ENVELOPE Enablers, the Zero-touch management system, and the Inter-PLMN handover system.

Building upon methodologies from successful prior initiatives, the KPI framework has been specifically tailored to ENVELOPE's large-scale trial nature. A dedicated methodological approach for trial scalability has been introduced to assess how innovations perform under progressively increasing operational stress. By subjecting the system to diverse, realistic workload types, the evaluation measures its capacity to maintain target KPIs under demanding deployment conditions.

A critical aspect of the KVI evaluation methodology is the strategic assignment of each KVI to its most appropriate validation strategy. For KVIs defined by measurable, objective, and reproducible parameters, quantitative empirical evaluation is selected as the primary method. Conversely, when evaluating subjective, human-centric dimensions, such as user trust or digital accessibility that cannot be captured solely through technical instrumentation, user surveys are deployed to gather direct, structured stakeholder feedback. Finally, expert assessment is reserved for evaluating long-term or systemic effects, or in scenarios where direct empirical measurement is unfeasible due to early TRLs or limited market scale. By systematically applying these criteria, the methodology ensures that every facet of ENVELOPE's societal, environmental, and economic value is assessed with the highest degree of methodological rigor and contextual relevance.

Furthermore, the document establishes the evaluation process for Open Call projects, aligning closely with standard Horizon Europe practices. Open Call beneficiaries are required to submit three deliverable reports throughout their lifecycle. These reports are subsequently evaluated against predefined criteria by assigned experts (mentors) selected from within the ENVELOPE consortium.

The delivery of this document successfully completes the first major milestone of WP6. Subsequent steps will focus on the preparation and coordinated execution of the planned evaluation activities, culminating in the analysis of results and the reporting of final conclusions.

6 References

[1] ENVELOPE, "Deliverable D2.3 - ENVELOPE availability and cooperation enablers definition and evaluation data specifications," 2025.

[2] PoDIUM, "Deliverable D5.1 - PoDIUM Evaluation Methodology," 2024.

[3] ICT4CART, "Deliverable D8.1 - Evaluation Methodology," 2020.

[4] 5G-IANA, "Deliverable D5.2 - Validation Methodology," 2023. [Online]. Available: https://zenodo.org/records/10533956. [Accessed 25 03 2026].

[5] ARCADE, "FESTA Handbook Version 8," 2021. [Online]. Available: https://www.connectedautomateddriving.eu/wp-content/uploads/2024/07/FESTA-Handbook-Version-8-FINAL-Version-20-09.pdf. [Accessed 02 03 2026].

[6] ENVELOPE, "Deliverable D2.1 - Use cases refinement and specifications," 2024. [Online]. Available: https://envelope-project.eu/wp-content/uploads/2024/04/D2.1-ENVELOPE-use-cases-description-and-specifications-Final.pdf. [Accessed 02 03 2026].

[7] ENVELOPE, "Deliverable D2.2 - Platform Architecture Requirements and Specification," 2024. [Online]. Available: https://envelope-project.eu/wp-content/uploads/2024/04/ENVELOPE-D2.2-v1.0-FINAL.pdf. [Accessed 02 03 2026].

[8] ENVELOPE, "Deliverable D3.2 - ENVELOPE open and dynamically reconfigurable B5GS description," 2025.

[9] ENVELOPE, "Deliverable D3.3 - ENVELOPE Innovation Developments Report," 2025.

[10] ENVELOPE, "Deliverable D4.1 - Deployment of ENVELOPE architecture to the sites and development of data collection tools and internet service repository," 2026.

[11] 5G-IANA, "D5.1 - Initial validation KPIs and metrics," 2022. [Online]. Available: https://zenodo.org/record/7304938#.Y6w6Z3bMJD9.

[12] 3GPP, "User Equipment (UE) conformance specification (3GPP TS 38.523-1 V17.6.0)," 2024. [Online]. Available: https://www.3gpp.org/ftp/Specs/archive/38_series/38.523-1/38523-1-h60.zip.

[13] 5G-MOBIX, "Deliverable D5.2 - Report on Technical Evaluation," 2023.

[14] G. Wikström, A. Schuler Scott, I. Mesogiti, R.-A. Stoica, G. Georgiev, S. Barmpounakis, A. Gavras, P. Demestichas, M.-H. Hamon, H.-S. Hallingby and D. Lund, "What societal values will 6G address?," 17 May 2022. [Online]. Available: https://doi.org/10.5281/zenodo.6557534. [Accessed 02 03 2026].

[15] United Nations, "The 17 Goals," 2020. [Online]. Available: https://sdgs.un.org/goals. [Accessed 02 03 2026].

[16] B. B. Brown, Delphi Process: A Methodology Used for the Elicitation of Opinions of Experts, Santa Monica, CA: RAND Corporation, 1968.

[17] Arthur D. Little, “Extracting network operations efficiency in the post-COVID-19 era,” 1 February 2021. [Online]. Available: https://www.adlittle.com/uk-en/insights/viewpoints/extracting-network-operations-efficiency-post-covid-19-era. [Accessed 24 March 2026].

[18] Arthur D. Little, “Operations 4.0 - The five tech pillars that will make or break telco operators in 2022,” June 2022. [Online]. Available: https://www.adlittle.com/en/insights/report/operations-40. [Accessed 24 March 2026].

[19] Capgemini Research Institute, “Networks with intelligence - Why and how the telecom sector should accelerate its autonomous networks journey,” February 2024. [Online]. Available: https://www.capgemini.com/wp-content/uploads/2024/02/CRI_Autonomous-Network_Final_Web-version-1.pdf. [Accessed 24 March 2026].

A. Annex 1: Test cases for technical performance evaluation

This annex outlines the test cases defined to evaluate the technical performance of ENVELOPE's innovations, in accordance with the methodology detailed in Section 2.2. As experimental testing is an iterative process, these test cases may be subject to minor adjustments and refinements during the evaluation stage. Such modifications may naturally occur to accommodate practical constraints, optimize testing procedures, or adapt to unforeseen technical variables encountered during execution, ensuring the most accurate and robust evaluation possible.

A.1 EaaS test cases

Table 8 TC_EaaS_1 test case definition

Test Case ID	TC_EaaS_1
ENVELOPE Innovation	EaaS
KPIs / KVIs	KPI_EaaS_Availability
Test purpose	Verify the percentage of time the EaaS module is fully operational
Scalability validation approach	Application-Level Workloads • Increase the frequency of application onboarding requests. • Increase the frequency of experiment instance start requests.
Scalability configurations	Initial load: • Frequency of application onboarding requests: 1 Hz. • Frequency of experiment instance start requests: 1 Hz. Step increase: • Frequency of application onboarding requests: +5 Hz. • Frequency of experiment instance start requests: +5 Hz.
Pre-test conditions	• EaaS module up and running. • Application package created. • Experiment package created and onboarded. • Frequency of application onboarding requests defined. • Frequency of experiment instance start requests defined. • Test window defined.

Test procedure sequence	• Start performing the application onboarding requests at the defined frequency. • Start performing experiment instance start requests at the defined frequency. • Keep performing the requests for the defined test time window. • Verify that the EaaS module is healthy.
Evaluation criteria	The percentage of time the EaaS module is fully operational should be larger than the 99.9% within the service time window.

Table 9 TC_EaaS_2 test case definition

Test Case ID	**TC_EaaS_2**
ENVELOPE Innovation	EaaS
KPIs / KVIs	KPI_EaaS_AppOnboardingTime
Test purpose	Verify the time requirements for onboarding an application to the Application Repository.
Scalability validation approach	Application-Level Workloads • Increase the size of the application to be onboarded. • Increase the number of concurrent application onboardings.
Scalability configurations	Initial load: • Size of application: 1 MB • Number of concurrent application onboardings: 1 Step increase: • Size of application: +250 MB • Number of concurrent application onboarding: +5
Pre-test conditions	• EaaS module up and running. • Application package created. • Size of the application defined. • Number of concurrent application onboardings defined.

Test procedure sequence	• For all concurrent application onboardings
	  ○ Perform the POST requests for creating the application package info resource.
	  ○ Wait for the previous step to be completed for all onboardings.
	  ○ Perform the application onboarding requests for all onboardings.
	• Wait for the completion of the onboardings.

	• Retrieve the metrics from Prometheus.
Evaluation criteria	The Application Onboarding Time is in line with respect to the values introduced in D2.3 for this KPI.

Table 10 TC_EaaS_3 test case definition

Test Case ID	**TC_EaaS_3**
ENVELOPE Innovation	EaaS
KPIs / KVIs	KPI_EaaS_ExperimentStartupTime KPI_EaaS_ExperimentStopTime
Test purpose	Verify the time requirements for starting and stopping an experiment by the Experiment Lifecycle Manager.
Scalability validation approach	Application-Level Workloads • Increase the number of the applications to be started in the experiment • Increase the number of concurrent experiment instances to be started
Scalability configurations	Initial load: • Number of applications in an experiment: 1 • Number of concurrent experiments: 1 Step increase: • Number of applications in an experiment: +5 • Number of concurrent experiments: +5
Pre-test conditions	• EaaS module up and running. • Experiment package created and onboarded. • Number of applications in the experiment defined. • Number of concurrent experiment instances defined.

Test procedure sequence	• For all concurrent experiment instances
	  ○ Perform the POST requests for starting the experiment.
	• Wait for the previous step to be completed for all the experiment instances.
	• For all concurrent experiment instances
	  ○ Perform the POST requests for stopping the experiment.
	• Wait for the previous step to be completed for all the experiment instances.
	• Retrieve the metrics from Prometheus.

Evaluation criteria	The Experiment startup and Experiment stop times are in line with respect to the values introduced in D2.3 for these KPIs.

A.2 Enabler Availability test case

This is a common test case to be used by all ENVELOPE Enablers.

Table 11 TC_Enabler_Availability test case definition

Test Case ID	**TC_ Enabler_Availability**
ENVELOPE Innovation	All enablers
KPIs / KVIs	KPI_Enab_Availability
Test purpose	Verify the percentage of time the enabler is operational and able to respond to requests successfully within the defined service time window.
Scalability validation approach	Northbound API Workloads • Perform enabler API requests periodically during the defined service time window
Scalability configurations	Initial load: • Frequency of concurrent enabler API requests: 1 Hz • Time window: 1 hour Step increase: • Frequency of concurrent enabler API requests: +10 Hz • Time window: 1 hour
Pre-test conditions	• Enabler is up and running • All underlying enabler components (transformation functions, 5G core, RAN, and UEs) are available
Test procedure sequence	• Start performing concurrent enabler API requests at the defined frequency. • Keep performing the requests for the defined test time window. • Monitor the amount of time the enabler is up during the test time window.
Evaluation criteria	Tests will be considered successful if: • Enabler availability is ≥ 99,9% within service window

A.3 pQoS Enabler test cases

Table 12 TC_pQoS_1 test case definition

Test Case ID	TC_pQoS_1
ENVELOPE Innovation	pQoS Enabler
KPIs / KVIs	KPI_Enab_pQoS_EffectLatency KPI_Enab_pQoS_SuccessRate KPI_Enab_pQoS_InferenceAccuracy KPI_Enab_pQoS_InferenceIQNMessages (Greek Site only) KPI_Enab_pQoS_InferenceDuration KPI_Enab_pQoS_InferenceConfidence
Test purpose	Validate that the pQoS enabler produces accurate, timely and reliable predictions under various configurations, including: static/mobility-related, variability in RAN load and application-level scenarios. Ensure the system maintains performance and stability under increasing operational load.
Scalability validation approach	(a) Background Traffic and Data Plane Workload: UE background traffic and varying RAN load are applied to emulate radio contention and throughput variations. This stresses inference pipelines and IQN delivery under realistic network conditions; (b) Application-Level Workloads: Synthetic application traffic and varying server load evaluate the effect of application-level processing on prediction timeliness; (c) Northbound and Southbound API Workloads: Multiple concurrent inference requests, active subscriptions and continuous monitoring data ingestion assess API robustness under load.
Scalability configurations	(a) RAN Load Increase: Initial load: 1 reference UE Step increase: +10 background UEs. (b) pQoS API call frequency: Initial load: Single pQoS API call. Step increase: +10 Hz frequency of pQoS API calls. (c) Application Load: Initial configuration: no load Step increase: Application/server load (Greek Site-specific, exact increment to be defined). (d) Mixed Mobility + RAN Load Increase: Initial load: Single UE with variable mobility pattern. Step increase: +10 background UEs to emulate compounded operational stress (Greek Site-specific).
Pre-test conditions	• ENVELOPE pQoS enabler, NWDAF, and monitoring infrastructure are deployed and operational. • Prometheus-based (or similar) monitoring platform is collecting RAN, core, UE, and application metrics.

	• AI/ML models are pre-trained or baseline models are available. • Authentication, and subscription mechanisms are configured. • Reference UE and background UEs are registered and connected to the network. Background UEs are able to communicate with different patterns (e.g., different uplink bitrate and 5QI profiles) • Vehicular OBUs are functional, on-boarded and connected to the monitoring platform (e.g., GPS connectivity) • Application server and relevant services (e.g., MQTT) are deployed (if applicable in the scenario) • CAM application is configured to receive pQoS notifications (if applicable in the scenario). Logging and performance monitoring tools are enabled.
Test procedure sequence	1) Initialize system and verify API availability. 2) Configure scenario parameters and apply the selected scalability configuration. 3) Collect baseline QoS metrics (throughput, latency, RSSI, SINR). 4) Submit inference requests (request-response and subscription modes). 5) Collect predicted QoS values, confidence scores, and IQN notifications. 6) Record timestamps for effect latency and inference duration. 7) Repeat steps 4–6 under increasing load per scalability configuration. 8) Collect logs from Application Server, NWDAF (backend), metrics, and monitoring data for analysis.
Evaluation criteria	Successful if all KPIs are within their defined targets: availability ≥ 99.9%, effect latency < 300 ms, success rate ≥ 99%, inference accuracy MAPE ≤ 5% and RMSE ≤ 10%, IQN messages ≥ 100/day, inference duration < 200 ms, inference confidence ≥ 75%. Additionally, no critical failures occur, monitoring and notification data remain intact, and IQN delivery respects latency bounds.

Table 13 TC_pQoS_2 test case definition

Test Case ID	**TC_pQoS_2**
ENVELOPE Innovation	pQoS Enabler
KPIs / KVIs	KPI_Enab_pQoS_TrainingDuration KPI_Enab_pQoS_TrainingConvergence

Test purpose	Verify that the pQoS enabler can train or retrain predictive models efficiently and that models converge to acceptable accuracy. Assess the performance of the pQoS training algorithm.
Scalability validation approach	Northbound and Southbound API Workloads: • Multiple training requests in a sequence / Service-level requests to online model training and/or retraining a (simplified) existing model to be enhanced, continuous monitoring data ingestion. Usage of NWDAF (Southbound) online training interfaces.
Scalability configurations	(a) PQoS training requests: Initial load: single request call Step increase: requests in succession to stress test CPU/GPU load (exact increment to be defined); (b) Training Model Variants: QoS Sustainability, DN performance, Hybrid (simple and enhanced versions) – Greek Site specific; (c) Training Model Params: UL/DL Throughput, Latency – Greek Site specific
Pre-test conditions	• Monitoring database with historical data available. Training algorithm available at NWDAF component in the core. • PQoS-T APIs exposed and operational • Baseline models or training configurations defined • Logging and performance monitoring tools are enabled.
Test procedure sequence	1) Trigger model training for a selected scenario. 2) Monitor training lifecycle events and metrics. 3) Record training duration and convergence values (MAE/RMSE). 4) Analyse enabler performance using the logs recoded at NWDAF 5) Repeat for additional scenarios.
Evaluation criteria	Successful if training duration is within GPU/CPU thresholds (Training Duration: < 600 s for GPU, < 3600 s for CPU) and convergence metrics meet defined targets (MAE ≤ average value, RMSE ≤ 10% of target range).

Table 14 TC_pQoS_3 test case definition

Test Case ID	**TC_pQoS_3**
ENVELOPE Innovation	pQoS Enabler
KPIs / KVIs	KPI_Enab_pQoS_CAMSafeModeTA

Test purpose	Assess the impact of the pQoS inference response on the CAM application behaviour. Validate that the CAM application activates Safe Mode correctly based on predictive QoS notifications during various scenarios.
Scalability validation approach	(a) Background Traffic and Data Plane Workload: UE background traffic and varying RAN configurations are applied for radio contention and throughput variations, to emulate realistic conditions for pQoS notification consumption by CAM app, (b) Northbound and Southbound API Workloads: Multiple service requests in a sequence from one of multiple applications towards pQoS inference APIs.
Scalability configurations	(a) PQoS API call frequency: Initial load: Single pQoS API call. Step increase: +10 Hz frequency of pQoS API calls. (b) Mixed Mobility + RAN Load Increase: Initial load: Single UE with variable mobility pattern. Step increase: +10 background UEs to emulate compounded operational stress (Greek Site-specific).
Pre-test conditions	• CAM application integrated with PQoS enabler • Monitoring and inference services active • Test UEs and application server configured • CAM application sending out API calls • All conditions of TC_pQoS_1 testcase are met.
Test procedure sequence	1) Initialize CAM application and PQoS subscriptions. 2) Execute scenario e.g., with UE mobility and moderate network load. 3) Test device (vehicle) positioned in the radio cell with only CAM application realistic data traffic (for multi-cell scenario) 4) Deliver PQoS predictions to CAM and monitor Safe Mode activation. 5) Record temporal alignment with actual network degradations. 6) Analyse in the CAM application if Safe Mode was correctly triggered or not triggered.
Evaluation criteria	Successful if Temporal F1 score ≥ 85% and Safe Mode activation corresponds to predicted network conditions.

A.4 ATSSS Enabler test cases

Table 15 TC_ATSSS_1 test case definition

Test Case ID	TC_ATSSS_1
ENVELOPE Innovation	ATSSS Enabler
KPIs / KVIs	KPI_Enab_ATSSS_SuccessRate KPI_Enab_ATSSS_SetupTime
Test purpose	Measure the success rate of ATSSS-related API calls, and evaluate the time required for an ATSSS session to be established and available to the application.
Scalability validation approach	Northbound and Southbound API Workload • Increase the number of concurrent ATSSS-T API calls for ATSSS session establishment.
Scalability configurations	Number of concurrent ATSSS-T calls • Initial load: Single ATSSS-T API call to establish an ATSSS session. • Step increase: Increase number of concurrent ATSSS-T API calls to establish ATSSS sessions across multiple UEs (exact number to be defined).
Pre-test conditions	• Operational ENVELOPE 5G network with AF deployed at the edge. • Operational Wi-Fi network enabling UE connectivity to the 5G Core via the N3IWF function. • UEs/OBUs subscribed to the ATSSS Policy Function and identified as ATSSS-capable. • AFs subscribed to the ATSSS Policy Function and requesting ATSSS-capable UEs/OBUs.
Test procedure sequence	• Step 1: The UE/OBU registers with the 5G network and establishes a PDU session via the 3GPP access interface. • Step 2: The UE/OBU registers with the ATSSS Policy Function, indicating its ATSSS capability to the ENVELOPE 5GS. • Step 3: The Application Function subscribes to the ATSSS Policy Function to receive notifications about ATSSS-capable UEs/OBUs. • Step 4: The AF triggers the ATSSS-T API to establish an ATSSS session with a selected UE/OBU and records the timestamp of the request initiation. • Step 5: The ATSSS Policy function receives the ATSSS-T request from the AF and configures the ATSSS session between the UE/OBU and the ENVELOPE 5G core network. • Step 6: The UE registers with the 5GS and establishes an additional PDU session with the 5G Core Network via the non-3GPP (Wi-Fi) access interface.

	• Step 7: The UE notifies the ATSSS Policy Function of the successful or failed establishment of the secondary network path. • Step 8: The ATSSS Policy Function informs the AF of the session establishment outcome (success or failure). • Step 9: The Application Function records the session establishment outcome (success or failure) and, in case of success, measures the end-to-end session setup time.
Evaluation criteria	When ATSSS-T is applied requesting an ATSSS session, the following target thresholds are expected to be met (single-UE baseline scenario): • ATSSS service setup time between the UE and the edge server to be <6 seconds • ATSSS-T success rate > 95%; Percentage of ENVELOPE ATSSS-T API calls that complete successfully without network-side, server-side, or client-side errors

Table 16 TC_ATSSS_2 test case definition

Test Case ID	**TC_ATSSS_2**
ENVELOPE Innovation	ATSSS
KPIs / KVIs	KPI_Enab_ATSSS_PLRreduction
Test purpose	Evaluate the impact of ATSSS on packet loss rate reduction by comparing multi-path ATSSS sessions with single-path 5G sessions under low and heavy traffic conditions.
Scalability validation approach	Background Traffic and Data Plane Workload • Increase the background traffic load on the 3GPP access (5G) network.
Scalability configurations	Background Traffic and Data Plane Workload • Initial load: no background traffic generated towards the 3GPP RAN. • Step increase: increase background traffic by multiple UEs towards the 3GPP RAN (exact configuration to be defined).
Pre-test conditions	• Operational ENVELOPE 5G network with AF deployed at the edge. • Operational Wi-Fi network enabling UE connectivity to the 5G Core via the N3IWF function. • iPerf server deployed at the edge and configured as an AF. • iPerf client installed and configured on the UEs. • UEs are able to connect via a single path connection to the 5G network.

	• UEs are able to establish an ATSSS session via the ATSSS-T API explained in the test case (TC_ ATSSS_1).
Test procedure sequence	The test case is composed of three subcases. The first subcase establishes a baseline single-path 5G session and evaluates the packet loss rate as iPerf traffic from the selected UE is progressively increased. The second subcase performs the same evaluation using ATSSS with MP-QUIC enabled (load balancing scheduler). The above subcases are then repeated while introducing additional UEs to increase the load on the 3GPP RAN and assess performance under further congested conditions. • Step 1: The UE/OBU registers with the 5G network and establishes a PDU session via the 3GPP access interface. • Step 2: The UE initiates UDP traffic towards the edge-hosted iPerf server over the 5G network. • Step 3: The packet loss rate is measured at the UE and/or AF using the iPerf tool while progressively increasing the traffic load. • Step 4: The AF triggers the ATSSS-T to establish an ATSSS session with the selected UE/OBU. • Step 5: steps 2 and 3 are repeated under ATSSS operation, and the packet loss rate after traffic splitting across multiple links is calculated using the iPerf tool. • Steps 6: Repeat the procedure under increasing RAN load conditions by adding additional UEs to stress the 3GPP access network.
Evaluation criteria	When ATSSS-T is applied to request an ATSSS session, the following performance target is expected to be met: • Packet loss reduction ≥10% compared to single-path operation. Compliance with this threshold will be evaluated under increasing traffic load and growing numbers of UEs.

Table 17 TC_ATSSS_3 test case definition

Test Case ID	**TC_ATSSS_3**
ENVELOPE Innovation	ATSSS
KPIs / KVIs	KPI_Enab_ATSSS_LatencyReduction
Test purpose	Evaluate the impact of ATSSS on end-to-end latency reduction by comparing multi-path ATSSS sessions with single-path 5G sessions under low and heavy traffic conditions.

Scalability validation approach	Background Traffic and Data Plane Workload • Increase the background traffic load on the 3GPP access (5G) network.
Scalability configurations	Background Traffic and Data Plane Workload • Initial load: no background traffic generated towards the 3GPP RAN. • Step increase: increase background traffic by multiple UEs towards the 3GPP RAN (exact configuration to be defined).
Pre-test conditions	• Operational ENVELOPE 5G network with AF deployed at the edge. • Operational Wi-Fi network enabling UE connectivity to the 5G Core via the N3IWF function. • iPerf server deployed at the edge and configured as an AF. • iPerf client installed and configured on the UEs. • UEs are able to connect via a single path connection to the 5G network. • UEs are able to establish an ATSSS session via the ATSSS-T API explained in the test case (TC_ ATSSS_1).
Test procedure sequence	The test case is composed of three subcases. The first subcase establishes a baseline single-path 5G session and evaluates end-to-end latency using ICMP ping measurements towards the AF. The second subcase increases background traffic on the 3GPP RAN by introducing additional UEs generating iPerf traffic, in order to assess its impact on UE-to-AF latency. The third subcase establishes an ATSSS session with the Min-RTT MP-QUIC scheduler enabled and evaluates latency performance under congested network conditions. • Step 1: The UE/OBU registers with the 5G network and establishes a PDU session via the 3GPP access interface. • Step 2: The UE initiates ICMP ping measurements towards the edge-hosted AF over the 5G network. • Step 3: The end-to-end latency is measured under baseline conditions, with no background traffic generated by other UEs. • Step 4: Background UEs initiate TCP and/or UDP traffic towards the edge-hosted iPerf server over the 5G network to generate additional load. • Step 5: Measure end-to-end latency while the network is subjected to high background traffic from multiple UEs.

	• Step 6: The AF triggers the ATSSS-T API to establish an ATSSS session with the selected UE/OBU, enabling the MP-QUIC Min-RTT scheduler • Step 7: The end-to-end latency (ICMP Ping) is measured for UEs operating under the ATSSS-enabled Min-RTT scheduling solution.
Evaluation criteria	When ATSSS-T is applied to request an ATSSS session, the following performance target is expected to be met: • End-to-end latency reduction ≥ 20% compared to single-path operation. Compliance with this threshold will be evaluated under increasing traffic load and growing numbers of UEs.

Table 18 TC_ATSSS_4 test case definition

Test Case ID	**TC_ATSSS_4**
ENVELOPE Innovation	ATSSS
KPIs / KVIs	KPI_Enab_ATSSS_ThroughputGain
Test purpose	Evaluate the impact of ATSSS on uplink and downlink throughput gain by comparing multi-path ATSSS sessions with single-path 5G sessions under low and high traffic load conditions.
Scalability validation approach	Background Traffic and Data Plane Workload • Increase the background traffic load on the 3GPP access (5G) network.
Scalability configurations	Background Traffic and Data Plane Workload • Initial load: no background traffic generated towards the 3GPP RAN. • Step increase: increase background traffic by multiple UEs towards the 3GPP RAN (exact configuration to be defined).
Pre-test conditions	• Operational ENVELOPE 5G network with AF deployed at the edge. • Operational Wi-Fi network enabling UE connectivity to the 5G Core via the N3IWF function. • iPerf server deployed at the edge and configured as an AF. • iPerf client installed and configured on the UEs. • UEs are able to connect via a single path connection to the 5G network. • UEs are able to establish an ATSSS session via the ATSSS-T API explained in the test case (TC_ATSSS_1).
Test procedure sequence	The test case is composed of four subcases. The first subcase establishes a baseline single-path 5G session and

	evaluates uplink and downlink throughput under no-background-traffic conditions. The second subcase enables ATSSS multi-path connectivity via the ATSSS-T API and evaluates throughput gain under the same no-load conditions. The third subcase measures throughput under single-path operation while introducing background traffic from additional UEs to create network congestion. The fourth subcase re-enables ATSSS multi-path connectivity under congested conditions and evaluates the resulting throughput gain. • Step 1: The UE/OBU registers with the 5G network and establishes a PDU session via the 3GPP access interface. • Step 2: The UE initiates TCP or UDP traffic towards the edge-hosted AF over the single-path 5G connection. • Step 3: The total uplink and downlink throughput of the UE is measured under baseline conditions with no additional background load. • Step 4: The AF triggers the ATSSS-T API to establish an ATSSS session with the selected UE/OBU, enabling the MP-QUIC max-bandwidth scheduler. • Step 5: The total uplink and downlink throughput gain of the UE is measured under baseline conditions with no additional background load. • Step 6: The AF deletes the ATSSS session and returns back to single path 5G connection. • Step 7: Background UEs initiate TCP/UDP traffic using iPerf towards the edge-hosted server to increase data-plane load. • Step 8: Measure the impact on uplink and downlink throughput resulting from the additional traffic generated by background UEs (single path). • Step 9: The AF triggers the ATSSS-T API to re-establish an ATSSS session with the selected UE/OBU, enabling the MP-QUIC max-bandwidth scheduler. • Step 10: Evaluate uplink and downlink throughput performance under ATSSS operation with Max-Bandwidth MP-QUIC scheduling enabled.
Evaluation criteria	When ATSSS-T is applied to request an ATSSS session, the following performance target is expected to be met: • Uplink and downlink throughput increase ≥ 50% compared to single-path operation, subject to the

	applied congestion control algorithm and scheduling policy. Compliance with this threshold will be evaluated under increasing traffic load and growing numbers of UEs.

Table 19 TC_ATSSS_5 test case definition

Test Case ID	**TC_ATSSS_5**
ENVELOPE Innovation	ATSSS
KPIs / KVIs	KPI_Enab_ATSSS_RLDecisionTime
Test purpose	Evaluate the impact of application-level workloads on the decision time of the RL agent located at the UE. The goal is to identify under which conditions the target can be met.
Scalability validation approach	• Variation of application-level workloads to increase the CPU utilization/load on the UE.
Scalability configurations	Considered application-level workloads: • Full ATSSS deployment. • Begin with application that utilizes very little compute resources (no load). • Successively increase artificial workload on the UE.
Pre-test conditions	• All preconditions to establish an ATSSS connection between the UE and the CAM application server must be fulfilled (all API endpoints are online, the UE has active WiFi and 5G connections). • The UE must be in an idle state. • A (pre-trained) model for the RL agent is loaded and the container providing the endpoints to access the agent's functionality is running. • The AUE must be configured to log the response times of the agent during deployment.
Test procedure sequence	• Step 1: Establish an ATSSS connection with low application workload (e.g., iperf -b 1k) and the "load-balancing" rule to utilize the RL agent. • Step 2: The application must be active for at least 100 seconds. The AUE automatically queries the RL agent and logs the decision times. • Step 3: Add artificial workload on the UE, e.g., using sysbench cpu run --threads=X --time=Y. • Step 4: Wait for at least 100 seconds (decision time is logged). Optionally, go back to step 3 and increase the workload. • Step 5: Obtain the log file generated by the ANE.
Evaluation criteria	The time taken by the RL agent to output a decision is expected to be at most 500 ms.

A.5 Edge Enabler test cases

Table 20 TC_Edge_1 test case definition

Test Case ID	TC_Edge_1
ENVELOPE Innovation	Edge enabler
KPIs / KVIs	KPI_Enab_Edge_ServiceDeploymentTime
Test purpose	To assess the ability of the compute continuum to correctly and efficiently execute application lifecycle operations under representative operational load, including application deployment to an edge server. In parallel, validate the sustained correct operation of the coordinating enabler, as evidenced by continuous successful request handling and service responsiveness throughout the test.
Scalability validation approach	Scalability Validation is considered in terms of: • Application-Level Workloads: increase application lifecycle requests (such as onboarding, instantiation, termination and purge). • Northbound API Workloads: increase the frequency and concurrency of northbound API calls.
Scalability configurations	• Initial load: Single API call and no background data-plane traffic. • Step increase: Increase number of API calls in rapid succession (or concurrent) and increase of application lifecycle requests (such as onboarding, instantiation, termination and purge).
Pre-test conditions	Italian trial site: • Edge Cloud enabler up and running • Edge server available • Application to be deployed already onboarded • Define the size of the application • Define the number of concurrent deployments Dutch trial site: • One edge zone available for deploying applications • V2X server sending out API calls Greek trial site: • aeriOS-based continuum deployed and operational, including two administrative domains and the interconnecting routed network. • Basic connectivity verified between domains and edge sites. • Registered users and/or service accounts available in the continuum.

	• Authentication and authorization mechanisms active, allowing token issuance to access: •CAMARA Edge Application Management northbound APIs. •aeriOS continuum southbound APIs. • Network configuration complete and validated: •Routing between domains functional. •Firewall and access control rules properly configured. •Control-plane and data-plane traffic verified. • Application descriptors for onboarding prepared, validated, and aligned with target edge environments (including resource definitions and compatibility checks). • Tooling and automation scripts available for: •Invoking CAMARA northbound API calls (onboarding, instantiation, termination, status queries). •Querying aeriOS continuum APIs for resource and instance status. •Generating background inter-domain traffic (e.g., iperf-based). •Logging activities across enabler and continuum components. • Logging and time synchronization in place: NTP or equivalent ensures aligned timestamps for KPI measurement.
Test procedure sequence	1. Initialization: Ensure the continuum and enabler are in the defined initial state. Verify that all required services are running, network connectivity between the two domains is operational and authentication tokens for API access can be successfully obtained. 2. Workload configuration selection: Select the workload configuration to be executed and API call frequency/concurrency. 3. Execute application lifecycle operations: Using scripted or automated clients, perform application lifecycle actions via the CAMARA northbound API, including onboarding (if applicable), instantiation, termination and purge of application instances. Requests are issued across both domains and, where applicable, concurrently, in line with the selected workload configuration. 4. Generate control-plane query activity: In parallel with lifecycle operations, issue northbound API requests for application and resource status information, generating concurrent query traffic throughout the test execution. 5. Observe and record system behavior: Allow the system to process the combined workloads for the defined test duration, while logs are collected from the enabler and relevant continuum components for subsequent KPI calculation. 6. Terminate workloads: Stop API workload scripts in an orderly manner once the test duration or sample count is reached.

Evaluation criteria	Edge service deployment time: For each successful application instantiation, the deployment time is calculated as the elapsed time between the deployment request submission and the moment the application instance is reported as ready or running on the target edge, based on logged timestamps. Aggregated deployment time statistics are compared against the target value of 180 seconds.

Table 21 TC_Edge_2 test case definition

Test Case ID	**TC_Edge_2**
ENVELOPE Innovation	Edge enabler
KPIs / KVIs	KPI_Enab_Edge_HandoverTime KPI_Enab_Edge_HandoverSR
Test purpose	To assess the performance of migrating (handover) an application instance from one edge server to a new one.
Scalability validation approach	Scalability Validation is considered in terms of: • Application-Level Workloads: increase application lifecycle requests (such as onboarding, instantiation, termination and purge). • Northbound API Workloads: increase the frequency and concurrency of northbound API calls.
Scalability configurations	• Initial load: Single API call and no background data-plane traffic. • Step increase: Increase number of API calls in rapid succession (or concurrent) and increase of application lifecycle requests (such as onboarding, instantiation, termination and purge).
Pre-test conditions	Italian trial site: • MEC Handover feature available • Two edge servers available • Applications deployed on a UE • Define the size of the application • Define the number of concurrent migrations Dutch trial site: • Two edge zones available for deploying applications • V2X server sending out API calls Greek trial site: • aeriOS-based continuum deployed and operational, including two administrative domains and the interconnecting routed network. • Basic connectivity verified between domains and edge sites. • Registered users and/or service accounts available in the continuum.

	• Authentication and authorization mechanisms active, allowing token issuance to access: •CAMARA Edge Application Management northbound APIs. •aeriOS continuum southbound APIs. • Network configuration complete and validated: •Routing between domains functional. •Firewall and access control rules properly configured. •Control-plane and data-plane traffic verified. • Application descriptors for onboarding prepared, validated, and aligned with target edge environments (including resource definitions and compatibility checks). • Tooling and automation scripts available for: •Invoking CAMARA northbound API calls (onboarding, instantiation, termination, status queries). •Querying aeriOS continuum APIs for resource and instance status. •Generating background inter-domain traffic (e.g., iperf-based). •Logging activities across enabler and continuum components. • Logging and time synchronization in place: NTP or equivalent ensures aligned timestamps for KPI measurement.
Test procedure sequence	1. Initialization: Ensure the continuum and enabler are in the defined initial state. Verify that all required services are running, network connectivity between the two domains is operational and authentication tokens for API access can be successfully obtained. 2. Workload configuration selection: Select the workload configuration to be executed (no or high load) and configure the corresponding parameters for data-plane traffic generation and API call frequency/concurrency. 3. Execute application lifecycle operations: Using scripted or automated clients, perform application lifecycle actions via the CAMARA northbound API, including onboarding (if applicable), instantiation, termination and purge of application instances. Requests are issued across both domains and, where applicable, concurrently, in line with the selected workload configuration. 4. Generate control-plane query activity: In parallel with lifecycle operations, issue northbound API requests for application and resource status information, generating concurrent query traffic throughout the test execution. 5. Observe and record system behaviour: Allow the system to process the combined workloads for the defined test duration, while logs are collected from the enabler and relevant continuum components for subsequent KPI calculation. 6. Terminate workloads: Stop API workload scripts in an orderly manner once the test duration or sample count is reached.

Evaluation criteria	Edge handover time: For each successful stateless application migration, the handover time is calculated as the elapsed time between the migration request submission and the moment the new application instance is reported as ready or running on the target edge. Aggregated time statistics are compared against the target value of 60 seconds. Edge handover success rate: The application migration success rate is calculated as the ratio of successful migration operations to the total number of attempted migrations during the test run. A migration is considered successful when a new application instance becomes ready on the target edge within the defined timeout. The measured success rate is compared against the target value of at least 90%.

Table 22 TC_Edge_3 test case definition

Test Case ID	**TC_Edge_3**
ENVELOPE Innovation	Edge Enabler
KPIs / KVIs	KPI_Enab_Edge_TrafficInfluenceUEPolicyUpdateTime
Test purpose	Assess the performance of the re-configuration of UE policy via URSP (UE Route Selection Policy) to achieve traffic influence (re-routing) of UE's data traffic to newly selected edge server
Scalability validation approach	Northbound API Workloads • High-frequency API calls
Scalability configurations	Northbound API Workload • Initial load: Single traffic influence API call. • Step increase: Increase frequency of traffic influence API calls (exact increment to be defined).
Pre-test conditions	• Two application instances deployed in two different edge zones • V2X server sending out API calls • UE supporting URSP (UE Route Selection Policy)
Test procedure sequence	1. Starting the logging in the V2X server (northbound API) and UE. 2. Connect UE to one application instance of one of the edge zones 3. Trigger API calls based on the scalability configurations 4. After each API call, measure the time it takes to update a UE policy and verify that the UE's traffic is

	redirected to the new application instance at the new edge zone. 5. Repeat test in succession by sending a policy to switch UE's data traffic between edge zones back and forth. 6. Stop logging in the V2X server and at UE. 7. Analyse enabler performance using the logs recoded at V2X server and UE. The analysis step uses other scripts processing the saved logs.
Evaluation criteria	Tests will be considered successful if: • UE policy update time <= 30 seconds

A.6 Device Location Enabler test case

Table 23 TC_DeviceLocation_1 test case definition

Test Case ID	**TC_DeviceLocation_1**
ENVELOPE Innovation	Device Location enablers
KPIs / KVIs	KPI_Enab_DL_EffectLatency, KPI_Enab_DL_DeviceLocationGeofenceSR, KPI_Enab_DL_DevicesInAreaQuerySR, KPI_Enab_DL_CellIdLocationRetrievalSR
Test purpose	Verify the time requirement and the success rate of Device Location enablers.
Scalability validation approach	Northbound API Workloads • Increase the frequency of Device Location query and/or subscription requests. Southbound API Workloads (if applicable) • Increase the number of devices in the system.
Scalability configurations	Frequency of Device Location query and/or subscription requests • Initial load: 1 Hz. • Step increase: +10 Hz. Number of devices in the system (if applicable) • Initial load: 1 • Step increase: +10
Pre-test conditions	• Device Location enabler is up and running. • All underlying enabler components (transformation functions, 5G core, RAN, and UEs) are available.
Test procedure sequence	• Start performing the Device Location APIs query and/or subscription requests at the defined frequency

	• Keep performing the requests for the defined test time window • Wait for the responses • Compute the time elapsed between the requests and the responses • Check if the response from the Device Location corresponds to the actual situation
Evaluation criteria	Tests will be considered successful if time response < 1 s and success rate > 95%

A.7 QoD Enabler test cases

Table 24 TC_QoD_1 test case definition

Test Case ID	**TC_QoD_1**
ENVELOPE Innovation	QoD enabler
KPIs / KVIs	KPI_Enab_QoD_QoSSessionRetainability
Test purpose	Determine the percentage of time the requested QoS profile for a user session (e.g. a PDU session) is guaranteed in a pre-defined time window.
Scalability validation approach	Background Traffic and Data Plane Workload • Generate iPerf background traffic until radio capacity gets saturated.
Scalability configurations	Background Traffic and Data Plane Workload • Initial load: no presence of iPerf background traffic. • Step increase: increment presence of iPerf background traffic from multiple UEs (exact increment to be defined).
Pre-test conditions	• Multiple UEs are connected to the network: one main UE for which the KPIs will be assessed and other UEs that will be used to send background traffic. • Service with API consumer module is available
Test procedure sequence	3 tests sequences will be executed covering combinations of {QoD disabled, QoD enabled} and {Background traffic disabled, Background traffic enabled} configurations: 1. Tests with no QoD and no background traffic 2. Tests with no QoD and incremental background traffic until saturating the radio capacity. 3. Tests with QoD enabled and incremental background traffic until saturating the radio capacity. The test sequence comprises of these steps: • Starting the logging at the service backend • Starting the logging and iPerf traffic at the main UE

	• Applying QoD via the service's API consumer module (for configuration 3) • Starting the iPerf traffic at the other UEs in order to saturate the radio capacity (for configurations 2 and 3) • Stopping the loggings at the main UE and at the service backend • Analysing session retainability KPI using data stream statistics (throughput, latency) during the defined time window for the test case performed
Evaluation criteria	Tests will be considered successful if: • Session retainability ≥ 95%

Table 25 TC_QoD_2 test case definition

Test Case ID	**TC_QoD_2**
ENVELOPE Innovation	QoD enabler
KPIs / KVIs	KPI_Enab_QoD_QoSProfileEstablishmentTime KPI_Enab_QoD_SuccessRate
Test purpose	Determine the percentage of QoD API requests which are successfully accepted and processed by the QoD enabler and also calculate the time required by the QoD enabler to complete the provisioning and configuration of the required QoS profile for each QoD API request.
Scalability validation approach	Northbound API Workload • Concurrent QoD API calls
Scalability configurations	Northbound API Workload • Initial load: without other concurrent QoD API calls. • Step increase: + 10 Hz concurrent QoD API calls frequency.
Pre-test conditions	• One UE connected to the network • Service with API consumer module available to: (i) send QoD API requests for the UE and to (ii) generate high load concurrent QoD API calls.
Test procedure sequence	2 tests sequences will be executed to test the different scalability parameters: • Tests with sequential QoD API calls for the connected UE without the presence of other concurrent QoD API calls (Configuration 1) • Tests with sequential QoD API calls for the connected UE with the presence of other concurrent QoD API calls for either other UEs or non-existing UEs (triggering an error), as the main goal is to stress the API layer with a high number of calls (Configuration 2) The test sequence comprises of these steps:

	• Starting the logging in the service backend • Trigger QoD APIs based on the scalability parameters defined above (configuration 1 or 2) • For each QoD API call for the connected UE, calculate the QoD establishment time and whether the QoD API call was successful • Stop logging in the service backend • Analyse enabler performance using the API responses of the enabler. The analysis step uses other scripts processing the saved logs.
Evaluation criteria	Tests will be considered successful if: • Success rate is ≥ 99,9%. • Establishment time is < 1000 ms

A.8 Performance Metric Enabler test cases

Table 26 TC_PM_1 test case definition

Test Case ID	**TC_PM_1**
ENVELOPE Innovation	Performance Metrics Enabler
KPIs / KVIs	KPI_Enab_PerMetrics_NotificationLatency KPI_Enab_PerMetrics_NotificationSR
Test purpose	Determine the enabler availability, notification latency, and notification success rate via maintaining a single subscription or multiple concurrent subscriptions.
Scalability validation approach	Northbound and Southbound API Workload • Establishing multiple concurrent subscriptions.
Scalability configurations	Northbound and Southbound API Workload • Initial load: single API subscription. • Step increase: increase number of concurrently established subscriptions for all UEs (exact increment to be defined).
Pre-test conditions	• Main UE performing MQTT stream and UEs generating iperf background traffic are connected to the 5G cell. • V2X server with API consumer module available. • Define the test case duration parameter: $T_{test.}$
Test procedure sequence	1. Start MQTT stream on the main UE and iperf on the UEs generating background traffic 2. Start test case-specific logging in the V2X backend 3. Create one Performance Metrics subscription for the main UE

	4. Gather data for T_{test} duration (baseline with single subscription) 5. Create concurrent subscriptions for all UEs (scripted) 6. Gather data for T_{test} duration (concurrent subscriptions) 7. Cancel all subscriptions and stop the applications (MQTT, iperf) 8. Stop backend logging, evaluate data
Evaluation criteria	Tests will be considered successful if: • KPI_Enab_PerMetrics_NotificationLatency < 1000ms • KPI_Enab_PerMetrics_NotificationSR ≥ 95%

A.9 Zero-Touch Management test cases

Table 27 TC_ZTM_1 test case definition

Test Case ID	**TC_ZTM_1**
ENVELOPE Innovation	Zero Touch Management
KPIs / KVIs	KPI_ZT_NativeAIframeworkServiceTime KPI_ZT_NativeAIframeworkServiceAccuracy KPI_ZT_Decision&ProvisioningTime
Test purpose	Verify the time requirement for intent translation, decision engine reconfiguration and outcome provisioning.
Scalability validation approach	Northbound API Workloads • Increase the number of concurrent inputs (number of vehicles) provided to the Decision Engine. The Intent Engine will reconfigure accident detection thresholds for the monitored vehicles.
Scalability configurations	• Initial load: 1 concurrent input. • Step increase: +4 concurrent inputs.
Pre-test conditions	Intent and Decision Engines up and running
Test procedure sequence	• User provides input in the chatbox of the Intent Engine • Intent Translation • Reconfiguration of Decision Engine • User notification of the outcome
Evaluation criteria	The test is considered successful • For KPI_ZT_NativeAIframeworkServiceTime if the zero-touch action service (including the front-end elements) is less than 5 minutes. • For KPI_ZT_NativeAIframeworkServiceAccuracy if the accuracy in the translation from high level natural language to specific values is over the 95%.

	• For KPI_ZT_Decision&ProvisioningTime if the time to perform a zero-touch decision (not including the front-end elements) is less than 1 minute.

Table 28 TC_ZTM_2 test case definition

Test Case ID	TC_ZTM_2
ENVELOPE Innovation	Zero Touch Management
KPIs / KVIs	KPI_ZT_AutomatedManagementProcess
Test purpose	Verify that the process for network management, or application management, or security management is fully automated
Scalability validation approach	N/A
Scalability configurations	N/A
Pre-test conditions	Management process under test is deployed
Test procedure sequence	• Start the management process • Monitor the steps of the management process • Verify that the process is being executed without the need of human intervention
Evaluation criteria	The test is considered successful if there is one automated process not requiring human intervention for network, application or security management domain.

A.10 Italian Trial Site's UC-specific test cases

The Italian Trial Site's UC-specific test cases evaluate how the UCs' performance is improved when the QoD enabler is activated.

UC1-specific test case:

Table 29 TC_IT_UC1_UL test case definition

Test Case ID	TC_IT_UC1_UL
ENVELOPE Innovation	QoD Enabler
KPIs / KVIs	KPI_UC1_UL_Increase
Test purpose	Verify the increase in the uplink rate when the AV needs to send the sensor data due to accident
Scalability validation approach	Application-level Workloads • increase the number of UEs sending the sensor data
Scalability configurations	• Initial load:1 vehicle. • Step increment: +2 vehicles.

Pre-test conditions	• QoD enabler is up and running. • All underlying enabler components (transformation functions, 5G core, RAN, and UEs) are available.
Test procedure sequence	• Verify the UL rate before the QOD request. • Trigger the QoD request. • Verify the UL rate after the QoD request triggering.
Evaluation criteria	The test is considered successful if increase corresponds to what requested at the QoD

UC2-specific test cases:

Table 30 TC_IT_UC2_UL test case definition

Test Case ID	**TC_IT_UC2_UL**
ENVELOPE Innovation	QoD Enabler
KPIs / KVIs	KPI_UC2_UL_Increase
Test purpose	Verify the increase in the uplink rate when the AV needs to send the sensor data due to map update
Scalability validation approach	Application-level Workloads • increase the number of UEs sending the sensor data
Scalability configurations	• Initial load:1 vehicle. • Step increment: +2 vehicles.
Pre-test conditions	• QoD enabler is up and running • All underlying enabler components (transformation functions, 5G core, RAN, and UEs) are available
Test procedure sequence	• Verify the UL rate before the QOD request • Trigger the QoD request • Verify the UL rate after the QoD request triggering
Evaluation criteria	The test is considered successful if increase corresponds to what requested at the QoD

Table 31 TC_IT_UC2_DL test case definition

Test Case ID	**TC_IT_UC2_DL**
ENVELOPE Innovation	QoD Enabler
KPIs / KVIs	KPI_UC2_DL_Increase
Test purpose	Verify the increase in the downlink rate when the AV needs to download the updated map
Scalability validation approach	Application-level Workloads • increase the number of UEs downloading the map
Scalability configurations	• Initial load:1 vehicle. • Step increment: +2 vehicles.

Pre-test conditions	• QoD enabler is up and running • All underlying enabler components (transformation functions, 5G core, RAN, and UEs) are available
Test procedure sequence	• Verify the DL rate before the QOD request • Trigger the QoD request • Verify the DL rate after the QoD request triggering
Evaluation criteria	The test is considered successful if increase corresponds to what requested at the QoD

A.11 Dutch Trial Site's UC-specific test cases

The Dutch Trial Site's UC-specific test cases evaluate how the QoD, Device Location and pQoS enablers support the UCs through a clear cause-and-effect relationship.

UC3-specific test case:

Table 32 TC_DT_UC3 test case definition

Test Case ID	**TC_DT_UC3**
ENVELOPE Innovation	QoD and Device Location Enablers
KPIs / KVIs	• KPI_DT_UC3_E2E_Latency • KPI_DT_UC3_DeviceLocationGeofenceSR
Test purpose	Verify that the recorded digital twin data is uploaded with a maximum E2E latency to the server (when QoD is enabled) and only at the desired cell location (based on Device Location notifications).
Scalability validation approach	Background Traffic and Data Plane Workload • Generate iPerf background traffic until radio capacity gets saturated.
Scalability configurations	Background Traffic and Data Plane Workload • Initial load: no presence of iPerf background traffic. • Step increase: increment presence of iPerf background traffic (exact increment to be defined).
Pre-test conditions	• Vehicle is equipped with a traffic sign recognition (TSR) system. • Digital twin instance is running on edge/cloud infrastructure. • V2X server is running and can reach the Device location and QoD enablers.

Test procedure sequence	2 tests sequences will be executed covering combinations of {QoD disabled, QoD enabled} with background traffic saturating the radio capacity: • Tests with no QoD with background traffic • Tests with QoD enabled with background traffic The test sequence comprises of these steps: 1. Continuously transmit lightweight telemetry and interpreted sign metadata to the digital twin. 2. Subscribe to Device Location Geofencing region for notification. 3. When the vehicle enters the geofenced location and receives a notification 4. (Configuration 2) the bandwidth required for uploading the file is requested by activating the QoD enabler. 5. The data transmission for uploading the file to the server is initiated. 6. Transmit recorded TSR camera streams from the vehicle to the server according to scalability configurations defined.
Evaluation criteria	Tests will be considered successful if: • KPI_DT_UC3_E2E_Latency < 100 ms. • KPI_DT_UC3_DeviceLocationGeofenceSR > 95%

UC4-specific test case:

Table 33 TC_DT_UC4 test case definition

Test Case ID	**TC_DT_UC4**
ENVELOPE Innovation	QoD Enabler
KPIs / KVIs	• KPI_DT_UC4_E2E_Latency • KPI_DT_UC4_Jitter
Test purpose	Verify that QoD enables guaranteed E2E latency to support repeatable mixed reality testing even under high network load condition.
Scalability validation approach	Background Traffic and Data Plane Workload • Generate iPerf background traffic until radio capacity gets saturated.
Scalability configurations	Background Traffic and Data Plane Workload • Initial load: no presence of iPerf background traffic. • Step increase: increment presence of iPerf background traffic (exact increment to be defined).

Pre-test conditions	• Real vehicle is operating on the test track while a digital twin runs in parallel. • Clock synchronization is established across systems to ensure accurate comparison of events. • V2X service is running and can reach the QoD enabler.
Test procedure sequence	2 tests sequences will be executed covering combinations of {QoD disabled, QoD enabled} with background traffic saturating the radio capacity: • Tests with no QoD with background traffic • Tests with QoD enabled with background traffic The test sequence comprises of these steps: 1. (Configuration 2) Activate QoD for guaranteed latency and jitter. 2. Transmit continuous vehicle state updates from the vehicle to the digital twin server. 3. Generate virtual objects (e.g., pedestrians) within the digital twin environment. 4. Transmit these virtual objects to the vehicle perception and control stack. 5. Record timestamps at each stage of virtual object generation, transmission, and vehicle reaction. 6. Calculate the latency difference between the simulated reaction and the real vehicle reaction.
Evaluation criteria	Tests will be considered successful if: • KPI_DT_UC4_E2E_Latency < 100 ms • KPI_DT_UC4_Jitter < 50 ms

UC5-specific test case:

Table 34 TC_DT_UC5 test case definition

Test Case ID	**TC_DT_UC5**
ENVELOPE Innovation	QoD and pQoS Enablers

KPIs / KVIs	• Teleoperation KPIs
	  ○ KPI_DT_UC5_Teleop_E2E_Latency
	  ○ KPI_DT_UC5_Teleop_MinUplinkRate
	  ○ KPI_DT_UC5_Teleop_Service_Reliability
	  ○ KPI_DT_UC5_Teleop_Disengagements
	• Perception set KPIs:
	  ○ KPI_DT_UC5_RSU_E2E_Latency:
	• Digital Twin KPIs:

	○ KPI_DT_UC5_DTupdate_E2E_Latency
Test purpose	Demonstrate that pQoS can support the tele-operating system in requesting QoD only when it is possible, i.e., there is enough bandwidth to be reserved in the desired future location. In addition, verify that QoD enables vehicle tele-operation even under high network load condition when QoD can be enabled.
Scalability validation approach	Background Traffic and Data Plane Workload • Generate iPerf background traffic until radio capacity gets saturated.
Scalability configurations	Background Traffic and Data Plane Workload • Initial load: no presence of iPerf background traffic. • Step increase: increment presence of iPerf background traffic (exact increment to be defined).
Pre-test conditions	• Vehicle is placed into a controlled failure scenario where primary perception sensors or processing are unavailable. • Nearby parked vehicle or infrastructure is configured to provide external camera and LiDAR feeds. • V2X service is running and can reach the QoD and pQoS enablers.
Test procedure sequence	2 tests sequences will be executed: • Tests using pQoS to predict that QoD can be enabled for the tele-operated vehicle in the future radio cell. In this scenario, the future radio cell has background traffic with low priority (best effort) • Tests using pQoS to predict that QoD cannot be enabled for the tele-operated vehicle in the future radio cell. In this scenario, the future radio cell has background traffic with high priority already reserved for other UEs The test sequence comprises of these steps: 1. Introduce a controlled failure, disabling onboard perception capability. 2. Request pQoS to predict of QoD can be enabled for the tele-operated vehicle in the future cell. a. (Configuration 1) Activate QoD to ensure required low latency and bandwidth and proceed with following steps below b. (Configuration 2) Stop test, since QoD cannot be enabled 3. Process LiDAR data on site to locally generate bounding boxes via an edge node. 4. Stream oriented bounding boxes from the site via MQTT broker to the remote station. 5. Convert edge data into object information and fuse it into the digital twin representation .

	6. Transmit live video stream to the remote operator via direct communication path . 7. Operator uses combined video and digital twin inputs to remotely drive the vehicle.

Evaluation criteria	Tests will be considered successful if:
	
	• Teleoperation KPIs
	  ○ KPI_DT_UC5_Teleop_E2E_Latency < 150 ms
	  ○ KPI_DT_UC5_Teleop_MinUplinkRate >= 12 Mbps.
	  ○ KPI_DT_UC5_Teleop_Service_Reliability >= 99.999%
	  ○ KPI_DT_UC5_Teleop_Disengagements < 1/hour safety driver needs to take over the vehicle control
	• Perception set / DT KPIs:
	  ○ KPI_DT_UC5_RSU_E2E_Latency < 150 ms
	• Digital Twin KPIs:
	  ○ KPI_DT_UC5_DTupdate_E2E_Latency < 100ms

A.12 Greek Trial Site's UC-specific test case

The Greek Trial Site's UC6-specific test case is focused on evaluating the performance of the inter-PLMN handover method, which is a key element of the UC.

Table 35 TC_GR_Inter-PLMN test case definition

Test Case ID	TC_GR_Inter-PLMN
ENVELOPE Innovation	Inter-PLMN Handover
KPIs / KVIs	KPI_InterPLMN_ServiceDisruption KPI_InterPLMN_AppRTT
Test purpose	To evaluate the service disruption due to Inter-PLMN Handover during LBO Roaming by measuring: (i) the elapsed time between the UE's session release from the Home network (HPLMN) and establishment of the new session in the visited PLMN (VPLMN). (ii) the application-level round-trip time (RTT) of MQTT message exchanges between two UEs during and after the network transition from the HPLMN to the visited VPLMN.

	(iii) C-ITS CAM and DEMN messages produced and RTT CAM messages. The ENVELOPE enablers directly support the successful testing of the inter-PLMN handover, which is the core part of the Greek Use Case (Data Sharing for Real-Time Situation Awareness), establishing a clear cause-and-effect relationship. More specifically, as a vehicle transitions from the Home PLMN to the Visited PLMN, the Device Location enabler continuously tracks the vehicle's position to detect PLMN boundaries, triggering the migration of the edge application to the target VPLMN. Concurrently, the pQoS enabler forecasts network performance degradation, allowing to trigger a handover in case of low signal quality. Upon successful handover to the VPLMN, the QoD enabler can be immediately triggered under LBO roaming to dynamically secure guaranteed priority treatment for critical hazard notification flows, thereby keeping MQTT and C-ITS message Round-Trip Times (RTT) within strict limits and ensuring continuous local situational awareness.
Scalability validation approach	Scalability Validation is considered in terms of: • Background Traffic and Data Plane Workload: Concurrent use of the mobile network, using physical and simulated UEs • Application-Level Workloads: Continuous MQTT message exchange between UEs across the continuum.
Scalability configurations	• Initial load: (1) No background traffic/load (2) Single MQTT session (3) Low RTT CAM MQTT message frequency [every 2 secs] • Step increase: (1) +10 Hz MQTT message frequency (2) Increase background traffic to emulate congested conditions (exact step to be defined)
Pre-test conditions	• HPLMN Network (OTE) operational at the Trial site. • VPLMN Network (NCSRD) operational at the Trial site. • Application Servers operational in AeriOS continuum at the Home (OTE) and Visited (NCSRD) domains • UEs furnished with HPLMN SIM Cards
Test procedure sequence	1. UE establishes MQTT communication 2. UE starts at HPLMN coverage 3. UE establishes continuous CAM application message exchange with HPLMN edge domain 4. UE moves to VPLMN coverage 5. UE establishes continuous CAM application message exchange with VPLMN edge domain 6. Observe and log the Inter-PLMN handover event 7. Measure service disruption time

	8. Measure MQTT application RTT during and after the handover 9. Repeat the procedure with smartphone UE, OBU and simulated UE.
Evaluation criteria	Post process logs to measure: • n_t1: the time between the PDU Closing Time from HPLMN (connection lost with OTE) and the PDU Establishment Time to VPLMN (connection restored with NCSRD). • MQTT RTT of a message generated when UE in HPLMN (OTE) and received when UE in VPLMN (NCSRD) KPI_InterPLMN_ServiceDisruption is validated if n_t1 is always lower than the CAM MQTT thresholds KPI_InterPLMN_AppRTT is validated when MQTT RTT is < 1s

B. Annex 2: Open Call Milestone Evaluation Form

Evaluation Criteria

1. Deliverable Quality (weight 30%)

Description: Evaluate the submitted deliverable based on its quality, completeness, and alignment with project objectives.

Criteria:

- Alignment with project objectives and the agreed Mentoring Plan.
- Completeness and inclusion of all required elements.
- Clarity, organisation, and professional presentation.
- Technical accuracy and adherence to defined methodologies or standards.
- Demonstration of progress toward achieving technical milestones and goals.

Score (0–10/10; Threshold: 5/10): ……..

Comments on the quality of deliverable in case of below-average performance threshold:

2. Technical Performance Indicators (60%)

Description: Assess the achievement and/or soundness and feasibility of the project's technical KPIs and milestones, as outlined in the agreed Mentoring Plan.

Score (0–10/10; Threshold: 5/10): ……..

Comments on Technical Performance, in case of below-average performance threshold:

3. Deadline Compliance (10%)

Description: Evaluate the project's adherence to deadlines for deliverables and milestones.

Score (0–10/10; Threshold: 5/10): ……..

Comments on Deadline Compliance, in case of below-average performance threshold:

Summary and Final Recommendation

Final Score (Score: 0–10/10; Threshold: 7/10):……..

Threshold Assessment (Minimum Required: 7/10):
☐ Above Threshold
☐Below Threshold

Mentor’s Proposal:
☐ Continue with Funding
☐Under Threshold (Provide Justifications Below)

Justification for Recommendation (if applicable):

Selection Committee Validation

(To be filled by the Selection Committee)

☐ Validated
☐ Exclusion from Program

Committee Comments: