

Green Technologies for 5/6G Service-Based Architecture

Grant Agreement No	101096925
Full Title	Green Technologies for 5/6G Service-Based Architectures
Start date	01 Jan 2023
End date	30 Apr 2026
Duration	40
Project URL	https://www.6green.eu
Coordinator	CONSORZIO NAZIONALE INTERUNIVERSITARIO PER LE TELECOMUNICAZIONI (CNIT)

Deliverable D5.5

6Green Validation and Performance Evaluation

Contractual due date	30/04/2026 (M40)	Actual submission date	23/06/2026
Nature	Report	Dissemination Level	Public
Lead Beneficiary	ININ		
Responsible Author	Rudolf Susnik (ININ)		
Contributions from	Rudolf Susnik, Luka Korsic, Janez Sterle (ININ); Daniele Ronzani (HPE); Marius Iordache, Catalin Brezeanu, Daniel Tichiu (ORO); Nikos Fryganiotis, Eleni Stai, Anastasios Zafeiropoulos (ICCS); Guillermo Sánchez, Luis Miguel Contreras, Javier Velázquez (TID); Jesús Benedicto (BULL); Jakob Wolf-Kämpfer (Maintastic); Alexandros Valantasis (UBIT%ECH); Chiara Lombardo, Sergio Mangialardi, Alderico Gallo (CNIT); Arman Divband, Navid Nikaein (EUR); Raffaele Bruno, Claudio Cicconetti (CNR); Orazio Toscano (TEI).		



Revision history

Version	Issue Date	Changes	Contributor(s)
v0.1	09/02/2026	Initial version (ToC)	Rudolf Susnik (ININ)
V0.2	27/05/2026	First draft completed	All authors
v0.3	04/06/2026	Content updates, ready for internal review	All authors
v0.4	16/06/2026	Internal review comments addressed	Internal reviewers and authors
v0.5	18/06/2026	Candidate version	All authors
v1.0	23/06/2026	Final revision	Chiara Lombardo, Riccardo Rapuzzi (CNIT)

Disclaimer

The content of the publication herein is the sole responsibility of the publishers and it does not necessarily represent the views expressed by the European Commission or its services.

While the information contained in the documents is believed to be accurate, the authors(s) or any other participant in the 6Green Consortium make no warranty of any kind with regard to this material including, but not limited to the implied warranties of merchantability and fitness for a particular purpose.

Neither the 6Green Consortium nor any of its members, their officers, employees or agents shall be responsible or liable in negligence or otherwise howsoever in respect of any inaccuracy or omission herein.

Without derogating from the generality of the foregoing neither the 6Green Consortium nor any of its members, their officers, employees or agents shall be liable for any direct or indirect or consequential loss or damage caused by or arising from any information advice or inaccuracy or omission herein.

Copyright message

© 6Green Consortium, 2023-2026. This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both. Reproduction is authorised provided the source is acknowledged.

Table of Contents

Table of Contents 3
List of Figures 6
List of Tables 6
Glossary of terms and abbreviations used 8
Executive Summary 11
1 Introduction 12
1.1 6Green Overview and Innovative Aspects 12
2 Validation of 6Green components 15
2.1 MDAF – Management Data Analytics Function 15
2.1.1 Component's functional requirements validation 15
2.1.2 Component's services validation 16
2.2 NWDAF - Network Data Analytics Function 16
2.2.1 Component's functional requirements validation 17
2.2.2 Component's services validation 17
2.3 NSDAF – Network Slice Data Analytics Function 18
2.3.1 Component's functional requirements validation 18
2.3.2 Component's services validation 18
2.4 PCF - Policy Control Function 19
2.4.1 Component's functional requirements validation 19
2.4.2 Component's services validation 20
2.5 NEF – Network Exposure Function 20
2.5.1 Component's functional requirements validation 20
2.5.2 Component's services validation 21
2.6 AF – Application Function 21
2.6.1 Component's functional requirements validation 22
2.6.2 Component's services validation 23
2.7 BSSF – Business Support System Function 24
2.7.1 Component's functional requirements validation 24
2.7.2 Component's services validation 26
2.8 ENIF/NSENIF – Experiential Networked Intelligence Function/Network Slice ENIF 26
2.8.1 Component's functional requirements validation 26
2.8.2 Component's services validation 27
2.9 EGMF – Exposure Governance Management Function 28

2.9.1 Component's functional requirements validation 29
2.9.2 Component's services validation 29
2.10 NSMF – Network Slice Management Function 30
2.10.1 Component's functional requirements validation 30
2.10.2 Component's services validation 31
2.11 NFMF – Network Function Management Function 31
2.11.1 Component's functional requirements validation 31
2.11.2 Component's services validation 32
2.12 EdgeMF – Edge-cloud Management Function 32
2.12.1 Component's functional requirements validation 33
2.12.2 Component's services validation 34
2.13 CCMF – Cloud-Continuum Management Function 34
2.13.1 Component's functional requirements validation 34
2.13.2 Component's services validation 35
2.14 NFVCL – Network Function Virtualization Convergence Layer 35
2.14.1 Component's functional requirements validation 35
2.14.2 Component's services validation 36
2.15 MetalCL – Metal Convergence Layer 36
2.15.1 Component's functional requirements validation 37
2.15.2 Component's services validation 37
2.16 IDAF – Infrastructure Data Analytics Function 38
2.16.1 Component's functional requirements validation 38
2.16.2 Component's services validation 38
2.17 Cloud-Continuum Wide Area 38
3 Validation of Integrated Frameworks and Solutions 40
3.1 Enhanced observability, monitoring and analytics framework 40
3.1.1 Component's functional requirements validation 40
3.1.2 Component's services validation 41
3.2 AI-driven decision and operations framework 42
3.2.1 Component's functional requirements validation 42
3.2.2 Component's services validation 43
3.3 Enhanced control policy framework 43
3.3.1 Component's functional requirements validation 44
3.3.2 Component's services validation 44
3.4 Integrated management subsystem 45

3.5 Management framework for edge-cloud resources 45
3.5.1 Component's functional requirements validation 45
4 Performance Evaluation 46
4.1 Use case 1: Critical Operation Maintenance during Energy-Constraint Disaster Scenarios 46
4.1.1 UC1 KPIs Assessment 51
4.2 Use case 2: Energy-Efficient Augmented Reality Remote Assistance System 57
4.2.1 UC2 KPIs Assessment 62
4.3 Use case 3: Zero-Carbon Clientless Virtual Enterprise Desktop as-a-Service 65
4.3.1 UC3 KPIs Assessment 69
5 Conclusions and Lessons Learned 73
6 References 74

List of Figures

Figure 1: Main innovations developed within WP2, WP3, and WP4 that have been integrated in the final prototype. 12
Figure 2: Physical setup used for UC1 demonstration and validation activities as described in [6]. 51
Figure 3: Monitoring network availability in UC1 normal operation mode scenario. 51
Figure 4: Monitoring network availability in UC1 disaster operation mode scenario. 52
Figure 5: List of Prometheus exporters. 53
Figure 6: Example of SNMP service monitor configuration for RRH energy consumption. 53
Figure 7: Time to disable (non-critical) service. 54
Figure 8: Time to disable (non-critical) UE. 54
Figure 9: Network latency variation over observed period. 54
Figure 10: Daily production of green energy over 11 days of observation. 56
Figure 11: Detailed daily comparison of environmental parameters, produced green energy and its share within total energy consumption of the ININ edge site. 56
Figure 12: UC2 Migration Procedure with CPU load and power monitoring. 58
Figure 13: UC2 UPF Offloading Procedure with Power Consumption and Bitrate. 60
Figure 14: Physical setup used for UC2 demonstration and validation activities as described in [6]. 62
Figure 15: UC2 Reference Usage Pattern. 62
Figure 16: Physical setup used for UC3 demonstration and validation activities as described in [6]. 69

List of Tables

Table 1: MDAF functional requirements validation. 15
Table 2: MDAF services validation. 16
Table 3: NWDAF functional requirements validation. 17
Table 4: NWDAF services validation. 17
Table 5: NSDAF functional requirements validation. 18
Table 6: NSDAF services validation. 18
Table 7: PCF functional requirements validation. 19
Table 8: PCF services validation. 20
Table 9: NEF functional requirements validation. 20
Table 10: NEF services validation. 21
Table 11: AF functional requirements validation. 22
Table 12: AF services validation. 23
Table 13: BSSF functional requirements validation. 24
Table 14: BSSF services validation. 26
Table 15: ENIF/NSENIF functional requirements validation. 26
Table 16: ENIF/NSENIF services validation. 27
Table 17: EGMF functional requirements validation. 29
Table 18: EGMF services validation. 29
Table 19: NSMF functional requirements validation. 30
Table 20: NSMF services validation. 31
Table 21: NFMF functional requirements validation. 31
Table 22: NFMF services validation. 32

Table 23: EdgeMF functional requirements validation. 33
Table 24: EdgeMF services validation. 34
Table 25: CCMF functional requirements validation. 34
Table 26: CCMF services validation. 35
Table 27: NFVCL functional requirements validation. 35
Table 28: NFVCL services validation. 36
Table 29: MetalCL functional requirements validation 37
Table 30: MetalCL services validation. 37
Table 31: IDAF functional requirements validation. 38
Table 32: IDAF services validation. 38
Table 33: Enhanced observability, monitoring and analytics Framework functional requirements validation. 40
Table 34: Enhanced observability, monitoring and analytics Framework services validation. 41
Table 35: AI-driven decision and operations framework functional requirements validation. 42
Table 36: AI-driven decision and operations framework services validation. 43
Table 37: Enhanced control policy framework functional requirements validation. 44
Table 38: Enhanced control policy framework services validation. 44
Table 39: Management framework for edge-cloud resources functional requirements validation. 45
Table 40: UC1 SBA actions validation. 46
Table 41: UC1 KPIs evaluation. 48
Table 42: Summary of KPI_UC1_1 (Network Availability) evaluation tests. 52
Table 43: Summary of KPI_UC1_5 (Network latency) evaluation tests. 55
Table 44: UC2 SBA actions validation. 57
Table 45: UC2 KPIs evaluation. 61
Table 46: KPI_UC2_1 evaluation results summary. 63
Table 47: KPI_UC2_2 evaluation results summary. 64
Table 48: KPI_UC2_3 evaluation results summary. 65
Table 49: KPI_UC2_4 evaluation results summary. 65
Table 50: UC3 SBA actions validation. 65
Table 51: UC3 KPIs evaluation. 67
Table 52: LibreOffice workload: CO_2e comparison (DaaS vs local laptop) across concurrency levels. Local device baseline: 14.5 W/user. 70
Table 53: FFmpeg workload: CO_2e comparison (DaaS vs local laptop) across concurrency levels. Local device baseline: 30 W/user 70

Glossary of terms and abbreviations used

Abbreviation / Term	Description
3GPP	The 3rd Generation Partnership Project
5G	5th Generation of cellular technologies
5GC	5G Core
6G	6th Generation of cellular technologies
6GC	6G Core
AF	Application Function
AI	Artificial Intelligence
API	Application Programming Interface
AR	Augmented Reality
AR/LPI	Adaptive Rate (AR) with Low Power Idle (LPI)
ARIMA	Autoregressive Integrated Moving Average
BESS	Berkeley Extensible Software Switch
BS	Base Station
BSSF	Business Support System Function
BW	Bandwidth
CCMF	Cloud-Continuum Management Function
CO2	Carbon dioxide
CPU	Central Processing Unit
CRUD	Create, Read, Update, and Delete (a concept in the API)
DaaS	Desktop as a Service
DLA	Decarbonisation Level Agreement
DPDK	Data Plane Development Kit
EdgeMF	Edge-cloud Management Function
EDN	Edge Data Network
EGMF	Exposure Governance Management Function
eMBB	enhanced Mobile Broadband
ENIF	Experiential Networked Intelligence Function
FMdata	Fault Management data
GBR	Guaranteed Bit Rate
GPRS	General Packet Radio Service
GPU	Graphical Processing Unit

Abbreviation / Term	Description
GTP-U	GPRS Tunnelling Protocol – User Plane
GUI	Graphical User Interface
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IDAF	Infrastructure Data Analytics Function
IPMI	Intelligent Platform Management Interface
JSON	JavaScript Object Notation
K8s	Kubernetes
KPI	Key Performance Indicator
kWh	kilo-Watt-hour
LCM	Life-Cycle Management
LLDP	Link Layer Discovery Protocol
MA	Moving Average
MaaS	Metal as a Service
MBR	Maximum Bit Rate
MDA	Management Data Analytics
MDAF	Management Data Analytics Function
MetalCL	Metal Convergence Layer
ML	Machine Learning
N/A	Not Available/Applicable
NAT	Network Address Translation
NEF	Network Exposure Function
NF	Network Function
NFV	Network Function Virtualization
NFVCL	NFV Convergence Layer
NFVO	NFV Orchestrator
NMDAF	Network Data Analytics Function
NSDAF	Network Slice Data Analytics Function
NSENIF	Network Slice ENIF
NSMF	Network Slice Management Function
NWDAF	Network Data Analytics Function
PCF	Policy Control Function
PDU	Packet Data Unit

Abbreviation / Term	Description
PFCP	Packet Forwarding Control Protocol
PM data	Performance Management data
PNF	Physical Network Function
QoS	Quality of Service
RAM	Random Access Memory
RAPL	Running Average Power Limit
RDP	Remote Desktop Protocol
REST	Representational State Transfer
RL	Reinforcement Learning
RTT	Round Trip Time
SBA	Service Based Architecture
SLA	Service-Level Agreement
SM	Session Management
SMF	Session Management Function
S-NSSAI	Single Network Slice Selection Assistance Information
SW	Software
TS	Technical Specification
UC	Use Case
UE	User Equipment
UI	User Interface
UPF	User-Plane Function
URL	Uniform Resource Locator
URLLC	Ultra-Reliable Low-Latency Communication
VAO	Vertical Application Orchestrator
vApp	vertical Application
VIM	Virtual Infrastructure Manager
VM	Virtual Machine
WebRTC	Web Real-Time Communication
WIM	WAN Infrastructure Manager
WP	Work Package

Executive Summary

This document, entitled "6Green Validation and Performance Evaluation" is a final document of WP5 activities, defining specific validation evaluation procedures and then providing their corresponding results.

The goal of 6Green is to conceive, design, and realize a 5/6G Service-Based Architecture (SBA) able to promote energy efficiency across the value-chain. As WP5 is specifically devoted to assessing the project outcomes in a tangible manner, the results provided in this deliverable, i.e., D5.5, rely on activities of all WP5 tasks, from testbed setup in T5.1, vertical applications integration in T5.2, and finally demonstration and validation activities in T5.3 which directly contributed to the content of D5.5.

Components forming 6Green Service Based Architecture (SBA) have been first validated on their own, i.e., as standalone artefacts. Then, particular components/functionalities of the SBA, as well as certain non-SBA components/functionalities, have been further exploited within three 6Green use cases implementation and deployment. Validation and evaluation of the use cases were mainly focused on demonstrating specific SBA functions and to performance evaluation in terms of multiple KPIs addressing environmental, network, as well as business perspective.

1 Introduction

The 6Green Project has the goal of realizing a 5/6G SBA that enables all the involved stakeholders (i.e., Infrastructure, Network Platform and Vertical Application providers) reducing their carbon footprint with respect to business-as-usual scenarios. WP5 was therefore devoted to assessing the project outcomes in a tangible manner. Activities of WP5, as also described in other deliverables, included the preparation of the testbeds for continuous development and integration of the 6Green components and vertical applications (documented in D3.3 [1], D3.4 [2], D5.1 [3], D5.3 [4]), design and integration of a demonstrator (D5.2 [5], D5.4 [6]), as well as design of an evaluation framework (D5.1 [3]).

While details on particular components of the 6Green SBA and vertical applications have been provided in other deliverables, this document focuses on validation and performance evaluation of the final hardware and software prototypes, thus providing a proof-of-concept of the 6Green ecosystem. In that sense, SBA components, as well as certain non-SBA components, deployed to the demonstrator, work as a unique platform enabling vertical applications, thus also ensuring implementation of corresponding use cases. Three use cases, as elaborated in previous deliverables, were implemented, demonstrated, validated and evaluated during the project lifetime.

1.1 6Green Overview and Innovative Aspects

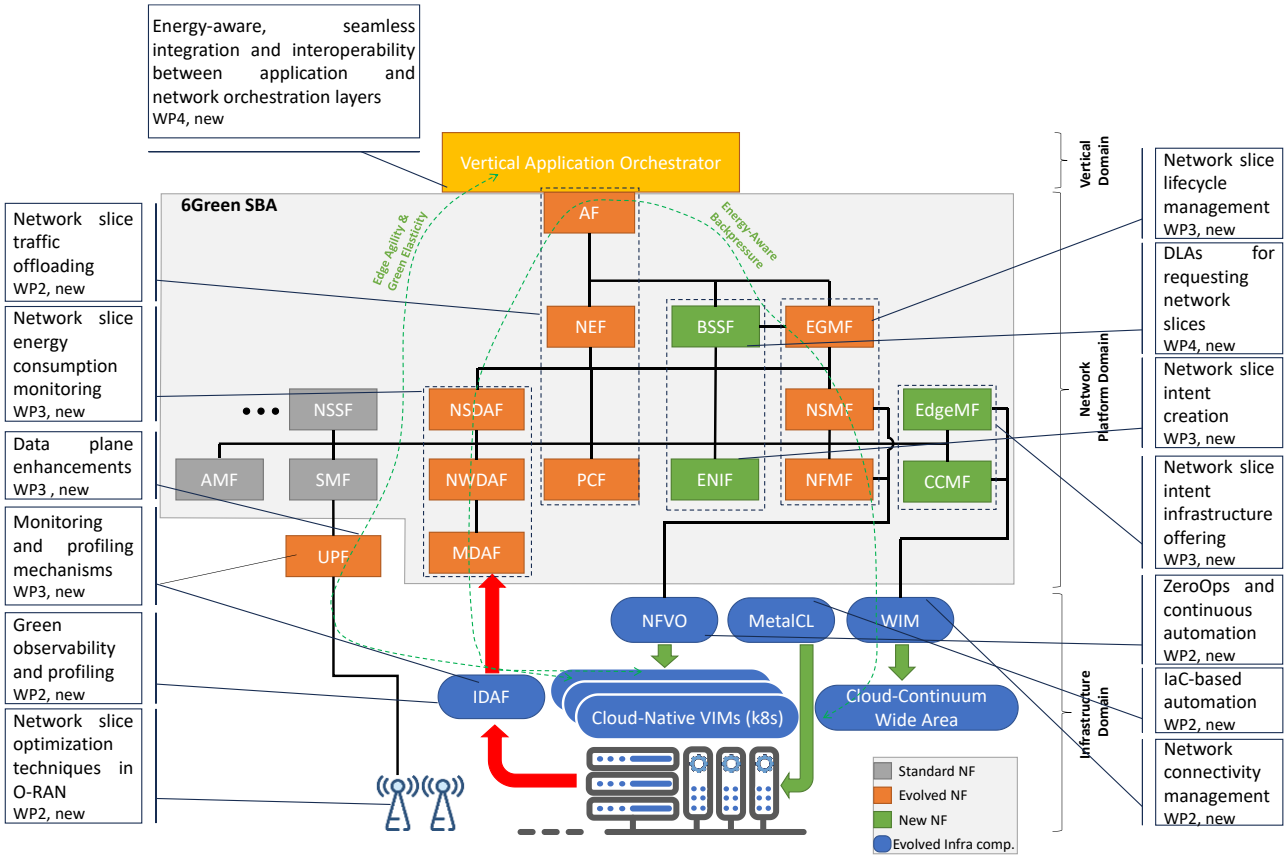


Figure 1: Main innovations developed within WP2, WP3, and WP4 that have been integrated in the final prototype.

The 6Green Project has been conceived with the ambitious goal of fostering and enabling the involvement of the stakeholders acting at the virtualized layer in the sustainability of the full 5/6G value-chain by providing them with the tools to make informed decisions on their deployments and reduce the carbon footprint ascribable to their applications and network services running on the physical infrastructure. To reflect this concept, the very structure of the project has been organized along three main research axes that correspond to the three stakeholders (namely, vertical application, network platform and infrastructure providers) to ensure that they are paid equal attention and to emphasize their synergy.

The resulting architecture is reported in Figure 1. The architecture itself is well-known and confirmed at the end of the Project, however this updated figure further highlights the most noteworthy innovative features and mechanisms developed in the three research axes and integrated in the final demonstrator. The building blocks composing the architecture, as described in the figure legend, have been evolved for the most part from the state of the art or from prototypes previously developed by partners and enhanced during 6Green, while the NFs depicted in green have been specifically conceived and developed within the project. The extensions have been applied to introduce functionalities and mechanisms aimed at enabling, collecting and/or communicating consumption-related information and decisions across the whole value chain. The functionalities themselves, reported in the white callouts, are all new outcomes designed and developed throughout the project activities and have been thoroughly described in D2.3, D3.3 and D4.2 in their finalized form. Here is a summary of the most relevant outcomes achieved by 6Green:

WP2: development of a set of enabling technologies that regard:

- network connectivity management and traffic offloading mechanisms;
- cloud-native orchestration mechanisms considering approaches that take advantage of service-mesh techniques, as well as automation mechanisms based on infrastructure as a code, ZeroOps and continuous automation principles
- power management mechanisms for the core, transport and access part of the continuum by considering serverless workloads
- network slice lifecycle management and optimization techniques, including energy-aware network slice management in O-RAN and multi-provider settings
- green observability and profiling mechanisms.

WP3: implementation of the energy-efficient 6Green 5/6G SBA architecture by means of:

- refinement of the 6Green SBA architecture and final implementation, which has concluded with the completion of the individual NFs and their full integration into the 6Green 5/6G SBA architecture
- integration of data plane enhancements (steering/offloading) with the segmentation lifecycle mechanisms already finalised
- completion of monitoring and profiling mechanisms, including the study of non-periodic packet sampling techniques to collect status data only during significant events
- development of multi-criteria policies designed to minimise resource wastage and implementation costs, whilst maximising revenue, based on a ML-powered slicing policy design
- development of lifecycle management mechanisms based on preliminary closed-loop automation (CLA) for the automated management of network segments.

WP4: tools to involve vertical stakeholders in the 6Green ecosystem:

- Definition of green business models to ensure the appeal and economic viability of then project's outcomes, refined using Lean Canvas and SWOT analyses

- conception of DLAs that include energy and carbon footprint considerations into the process for requesting network slices. MILP-based incentivisation framework and evaluation of the cost-effectiveness of different green ICT strategies
- 6Green VAO design for ensuring energy-aware, seamless integration and interoperability between application and network orchestration layers, including the implementation of the AF functionality
- definition of the operations to be supported by the VAO for Day-0/1/2/N, as well as the relevant workflows
- definition of vertical layer policies aligned with WP5 demonstrators
- software artefacts development at the Vertical Layer, involving: API implementation, green KPIs integration, intent management and optimisation mechanisms.

The outcomes of the three research axes have resulted in the following achievements of WP5:

- development and integration of all 6Green testbeds towards their operational state.
- integration and deployment plans for vertical applications.
- validation of SBA components and 6Green Integrated Frameworks and Solutions.
- development of the use cases and their deployment, including vertical applications, in the 6Green ecosystem.

2 Validation of 6Green components

6Green components presented in this section includes components that make up 6Green Service Based Architecture (SBA), and additional 6Green non-SBA components enabling the three 6Green paradigms – the Edge Agility, Green Elasticity and Energy-Aware Backpressure.

Non-SBA components relate to connectivity enablers in the transport network, cloud native orchestration and network automation, and to the observability and profiling in 5/6G continuum (e.g., Metal-CL, IDAF, Cloud-Continuum Wide Area).

6Green SBA is a set of network functions with a goal of improving the 5/6G ecosystem as described in more detail in previous deliverables. The final architecture of the 6Green 5/6G SBA, as elaborated in D3.3 [1], defines and implements five different frameworks, each of them focused on specific advanced features enabling the above mentioned three 6Green paradigms. While each of the frameworks consists of multiple network functions following a specific goal as further detailed in Section 3, this section focuses on every single component/function, i.e., SBA and non-SBA components. For every SBA component, functional requirements have been specified already in D3.1 [7], where services that each component should provide have been specified as well.

Prototypes of the SBA components have been, in the context of their respective frameworks, extensively tested and validated within the WP3 activities, and thus documented in D3.3 [1]. Since the conclusions of WP3/D3.3 [1] confirm meeting the requirements, this section provides details for understanding the test/validation procedure of each requirement, i.e., what is the test objective, what is acceptance criteria, and how to perform the test. The latter also applies to other, i.e., non-SBA components, which have been studied while designing, implementing and deploying 6Green testbeds.

2.1 MDAF – Management Data Analytics Function

The MDAF belongs to the Observability, monitoring, and analytics framework. Based on data retrieved from IDAF, it provides energy savings analytics, e.g., a breakdown of energy consumption against various subsystems, including network slices.

2.1.1 Component's functional requirements validation

Table 1: MDAF functional requirements validation.

Req. Label	Description	Test objective	Acceptance Criteria (based on standardized targets, if available)	Tests procedure/outputs
REQ-ES_MDA-01	MDA capability for energy saving analysis shall be able to identify the energy efficiency issues (including high energy consumption, low energy efficiency) and identify the cell/NFs or location area of where the indicated energy efficiency issues exist.	Understand which container, VM or machine consumes more energy	Yes/No	Read the MDAF logs
REQ-ES_MDA-02	MDA capability for energy saving analysis shall be able to identify the root cause of the energy efficiency issue when necessary.	Understand which container, VM or machine consumes more energy	Yes/No	Read the MDAF logs

REQ-ES_MDA-03	MDA capability for energy saving analysis shall be able to utilize the network status analysis and predictions information of the energy efficiency KPI factors (including traffic load trends) to assist achieving energy saving.	Understand which container, VM or machine consumes more energy by varying the traffic load (packet rate)	Yes/No	Read the MDAF logs
REQ_MDAF_NEW	MDAF capability for retrieving data from the IDAF includes knowledge of the URL of the Prometheus database used by IDAF.	The MDAF can query the Prometheus database	Yes/No	Read the MDAF logs

2.1.2 Component's services validation

Table 2: MDAF services validation.

Service Name	Description	Test objective	Acceptance Criteria (based on standardized targets, if available)	Tests procedure/outputs
NMDAF_PowerMetrics Enrichment	This service adds information to Scaphandre metrics	Add labels that are missing in original Scaphandre metrics.	N/A	Check queries in the Prometheus database
NMDAF_SystemLevel PowerMetricsCategorization	This service categorizes metrics regarding different tasks of the infrastructure	Categorize the different task of the infrastructure to different macro-areas (storage, memory management, net work, etc).	Yes/No	Get list of processes and do a direct mapping to category
NMDAF_Container CategoryMapping	This service maps the resource usage (system-level metrics) of the metrics to the consuming container	Identify the percentage of occupation on resources each container	N/A	Use the cAdvisor metrics in the IDAF
NMDAF_AnalyticsContainer PowerConsumptionForecast	This service forecasts the power consumption per container	Not implemented here, see NWDAF	N/A	None
NMDAF_SiteLevelCarbon EmissionsCalculator	This service calculates the total carbon emissions per site	Calculate the CO2eq emissions according to the carbon intensity of several National energy grids	N/A	The obtained CO2eq are exposed through Grafana

2.2 NWDAF - Network Data Analytics Function

The NWDAF is part of the Observability, monitoring, and analytics framework. It is responsible for collecting and analysing data from various network functions and providing insights to other network functions, such as network slice management, QoS management, and policy control.

2.2.1 Component's functional requirements validation

Table 3: NWDAF functional requirements validation.

Req. Label	Description	Test objective	Acceptance Criteria (based on standardized targets, if available)	Tests procedure/outputs
REQ_NWDAF-01	NWDAF capability for retrieving information regarding network conditions, device behaviour and service experience, in order to perform analytics on them and to send event notifications if requested.	Collect control plane data coming from the other SBA NFs through a pub/sub mechanism	Yes/no	Read the NWDAF logs
REQ_NWDAF-02	NWDAF capability for retrieving information regarding the power consumption of the single NFs. These data are collected from the MDAF.	Enable the retrieval of data computed by the MDAF	Yes/No	Read the NWDAF logs
REQ_NWDAF-03	NWDAF capability for storing data in the form of time series, in order to later perform analytics on them.	Save the time series in a database	Yes/No	Check queries in the Prometheus database

2.2.2 Component's services validation

Table 4: NWDAF services validation.

Service Name	Description	Test objective	Acceptance Criteria (based on standardized targets, if available)	Tests procedure/outputs
Nnwdaf_AnalyticsInfo	This service enables the NF service consumers to request and get specific analytics from NWDAF.	Understand the trend of the UEs in time	N/A	Read the trends from the Prometheus database
Nnwdaf_DataManagement	This service enables the NF service consumers to subscribe/unsubscribe from notifications when subscribed event(s) are detected or retrieve the subscribed data from the NWDAF.	Enable the exchange of control plane information among SBA NFs	N/A	Verify the confirmation of subscription via the NWDAF logs
Nnwdaf_AnalyticsNumberUEsSARIMA	This service provides a forecast, relying on SARIMA, on the number of UEs connected to an AMF	Knowledge of past trends of the UEs and collection of time series for forcasting	N/A	Get the data from Prometheus database.

2.3 NSDAF – Network Slice Data Analytics Function

The NSDAF is part of the Observability, monitoring, and analytics framework. It plays the role of a devoted network slice analytics function.

2.3.1 Component's functional requirements validation

Table 5: NSDAF functional requirements validation.

Req. Label	Description	Test objective	Acceptance Criteria (based on standardized targets, if available)	Tests procedure/outputs
REQ_NSDAF-01	NSDAF retrieving network slice information KPIs and user service experience, per slice analytics.	NSDAF software module functionality component functionality	Yes/No. NSDAF slices ID Endpoint confirmation.	List with all slice ids available Output "slice_ids": [slice_id_1_string, slice_id_2_string, etc],
REQ_NSDAF-02	NSDAF retrieving slice power consumption information.	NSDAF software module for power consumption within the network slice.	Yes/No. Slice Power Consumption	Historical energy consumption data. Output: "data": [(date_string, value_float) ,.....], "timestamp": String,
REQ_NSDAF-03	NSDAF provides slice analytics data information. Decision engine NF integration (VAO/ENIF)	NSDAF single slice power consumption analytics. API services exposure.	Yes/No NSDAF analytics results and prediction/	Forecasting future energy consumption Output: "data": [(date_string, value_float), ...], "timestamp": String, "error": Boolean,

2.3.2 Component's services validation

Table 6: NSDAF services validation

Service Name	Description	Test objective	Acceptance Criteria (based on standardized targets, if available)	Tests procedure/outputs
Nnsdaf_EventsSubscription_Slice	Service consumers to subscribe for slice specific information.	Slice subscription	API endpoints for retrieving data	Service subscription query validation. GET /slice_ids
Nnsdaf_AnalyticsInfo_Slice	Specific Slice analytics information	Slice analytics	API endpoints for retrieving data	Service prediction data POST/ predicted_power_microwatts

Nnsdaf_MLModel_Provisioning_Slice	ML Model matching the subscription parameters.	Model Components	ARIMA model Forecast energy consumption	ML to predict usage
Nnsdaf_DataManagement_Slice	Service slice data management	Power measurement data	ML Data Prediction	Slice forecasting

2.4 PCF - Policy Control Function

The PCF belongs to the Enhanced policy control framework. Its main responsibility is handling QoS flows within the 5/6G system.

2.4.1 Component's functional requirements validation

Table 7: PCF functional requirements validation.

Req. Label	Description	Test objective	Acceptance Criteria (based on standardized targets, if available)	Tests procedure/outputs
REQ_PCF-01	The PCF shall support the Npcf_PolicyAuthorization service to process an AF request and update the bandwidth parameters (Maximum Bit Rate – MBR / Guaranteed Bit Rate – GBR) of an active PDU Session. This is performed through an SM Policy Association Modification procedure toward the SMF, as defined in 3GPP TS 23.502 [8].	Verify that an Npcf_PolicyAuthorization_Create/Update request from the AF triggers the correct update of bandwidth limitations in the subscriber's active PDU Session.	The SMF applies the new MBR/GBR values to the PDU Session. The PDU Session reflects the updated bandwidth configuration. The PCF returns a Success response to the AF.	AF sends an Npcf_PolicyAuthorization_Update request SMF updates the bandwidth parameters through PFCP modification request. Expected Outputs: PFCP modification request and response in Traces Bandwidth limited in devices during speed test.
REQ_PCF-02	The PCF shall allow the modification of the priority associated with a QoS Flow within an active PDU Session upon receiving an AF request via the Npcf_PolicyAuthorization service, as specified in 3GPP TS 23.502 [8]. The PCF triggers an SM Policy Association Modification to the SMF, which applies the updated QoS parameters.	Verify that an AF-triggered Npcf_PolicyAuthorization_Update request successfully updates the priority of the appropriate QoS Flow	The SMF updates the ARP priority. The PDU Session reflects the updated ARP priority parameter. The PCF returns a Success response to the AF.	AF sends an Npcf_PolicyAuthorization_Update request SMF updates the ARP priority through PFCP modification request. Expected Outputs: PFCP modification request and response in Traces with ARP priority value changed.

2.4.2 Component's services validation

Table 8: PCF services validation.

Service Name	Description	Test objective	Acceptance Criteria (based on standardized targets, if available)	Tests procedure/outputs
Npcf_PolicyAuthorization	Authorizes an AF or other NF consumer request and creates/updates policies for the PDU session to which the AF is bound to.	Validate that the Npcf_PolicyAuthorization service correctly receives, processes, and applies AF policy requests	PCF successfully receives and parses Npcf_PolicyAuthorization request from AF SMF applies requested changes PCF returns a Success response to AF	API requests with different parameters, as shown in Section 2.4.1. Successful output is the response from PCF to the client and actual modification of parameters in Traces.

2.5 NEF – Network Exposure Function

The NEF is part of the Enhanced policy control framework. It enables exposure of network events, capabilities, and analytics for external/internal NFs.

2.5.1 Component's functional requirements validation

Table 9: NEF functional requirements validation.

Req. Label	Description	Test objective	Acceptance Criteria (based on standardized targets, if available)	Tests procedure/outputs
REQ_NEF-01	The NEF prototype shall be able to establish a connection with the target 5/6G Core Network APIs	Verify that the NEF prototype can connect to the 5/6G Core Network's API	The NEF component establishes an API request to the 5/6G Core Network. 5/6G Core Network responds successfully with 201 or 200 OK code state.	A client contacts the NEF making a RESTful API request. The NEF receive the request, maps the parameters and forward the request to the proper 5/6GC API endpoint. Successful of the test occurs when logs and traces from NEF indicates a positive response from 5/6GC to NEF and then to the Client.
REQ_NEF-02	The NEF shall support the EventExposure Service, allowing the AF to subscribe to, receive, and manage notifications for 5/6GC events. Supported events include session state changes, UE mobility and registration	Verify that the AF can successfully subscribe to an event type via NEF, and that NEF correctly pushes notifications based on real changes occurring in the 5/6GC.	NEF validates and registers the subscription from an AF. NEF sends a correct notification to AF. AF receives and logs the notification.	AF sends POST request to /event-exposure/subscriptions to NEF, which registers the subscription and returns a Subscription ID. After an event Core event, NEF sends event notification to the prototype callback endpoint.

REQ_NEF-03	The NEF shall support the SliceOffloading operation, allowing the AF to request the redirection or steering of traffic toward a specific network slice (S- NSSAI)	Verify that the AF can successfully request a slice offloading operation via NEF, and that the NEF correctly forwards the slice influence policy to the 5/6GC, resulting in traffic being routed through the intended slice	NEF validates and registers the request. NEF triggers slice routing enforcement in the 5/6GC. AF receives confirmation that traffic is being routed through the target slice.	AF sends a POST request to /slice-offload to NEF, specifying the target Slice ID (S- NSSAI). NEF registers the request and confirms it to the AF. When matching traffic occurs, NEF triggers slice steering rules in the 5/6GC components. Traffic is observed to flow through the target slice.

2.5.2 Component's services validation

Table 10: NEF services validation.

Service Name	Description	Test objective	Acceptance Criteria (based on standardized targets, if available)	Tests procedure/outputs
Nnef_EventExposure	The NEF Event Exposure API shall allow the AF to create, update, retrieve and delete an event subscription using the standardized endpoint /event-exposure/subscriptions.	Verify the correct behaviour of the NEF Event Exposure API when handling AF requests for subscription lifecycle management.	AF sends a POST to create a subscription (expect 201 Created). AF sends a GET to retrieve the subscription (expect 200 OK). AF sends a DELETE request to remove the subscription (expect 204 No Content). NEF updates its internal state accordingly.	The AF interacts with the NEF by issuing the standard CRUD operations on /event-exposure/subscriptions. Each API call returns the correct HTTP status code and expected payload, confirming that the NEF correctly manages subscription lifecycle operations.
Nnef_SliceOffload	The NEF SliceOffload API shall allow the AF to execute traffic offloading using the endpoint /slice-offload	Verify that the NEF TrafficOffload API behaves correctly during all operations.	AF sends a POST to create a slice offloading NEF applies the changes internally and propagates them to the 5/6GC.	The AF interacts with the NEF by issuing POST operation on /traffic-offload endpoint. The operation returns the appropriate status code and payload, confirming correct NEF API behaviour.

2.6 AF – Application Function

The AF is part of the Enhanced policy control framework. It interacts with the network platform via the NEF and on behalf of the (vertical) application (vApp) for observability and control, as well as it provides services/information related to the application.

2.6.1 Component's functional requirements validation

Table 11: AF functional requirements validation.

Req. Label	Description	Test objective	Acceptance Criteria (based on standardized targets, if available)	Tests procedure/outputs
REQ_AF-01	AF capability for retrieving information regarding available compute resources (BSSF/Location API for geographical area discovery)	Validate that the AF can accurately retrieve up-to-date information on available compute resources, specifically as a list of geographical areas suitable for deploying a vApp.	Yes/No AF to BSSF to retrieve the list of geographical areas.	To validate REQ_AF-01, the AF issues a GET request to the BSSF northbound API endpoint /api/v1/providers/locations, using also the Location API of VAO to retrieve the list of registered geographical areas (Points-of-Presence), each including geographicalAreaId, location name, coordinates, and coverage radius. Verify the response through AF/VAO system logs and confirm accuracy by comparing the returned geographical area list with the VAO UI display. Capture screenshots of the VAO dashboard and API response as evidence. Expected output: a valid JSON response listing the available geographical areas (e.g., Genova/Area 1, Milan/Area 2, Athens/Area 3 in the UC3 CNIT testbed), consistent with what is displayed in the VAO UI.
REQ_AF-02	AF capability for retrieving Energy-aware KPIs and creating policies to optimise vApp energy consumption.	Validate that the AF can retrieve relevant Energy-aware KPIs from Prometheus (via the Rating Operator) and create policies based on those KPIs to optimise energy consumption of the vApp.	Yes/No AF to consume energy related metrics stored in Prometheus and apply policies to the vApp via the Policy API.	To validate REQ_AF-02, register a metric expression via the VAO Metric Expression API referencing a valid PromQL query served by the Rating Operator. Then use the VAO Policy API to define a policy rule with a threshold condition on that metric and an associated orchestration action (e.g., migration to a green DLA node). Confirm policy creation through VAO system logs. Trigger the policy condition by inducing the relevant metric threshold and verify that the corresponding action is executed. Expected output: VAO system logs confirming metric registration, policy creation, alert detection, and action execution; screenshots from the Policy API response and VAO UI showing the active policy and triggered action.
REQ_AF-03	AF capabilities for requesting initial service slice compute and network resources and supporting renegotiation of those resources via the BSSF interface.	Validate that the AF can request initial compute and network resources for a service slice and dynamically renegotiate those resources using the BSSF interface when runtime conditions change.	Yes/No The AF must successfully submit a 3GPP TS 28.312 [9]-compliant slice intent to BSSF with compute, network, and location constraints; receive infrastructure credentials; deploy the vApp; and subsequently update the slice intent during renegotiation.	To validate REQ_AF-03, submit a slice intent from the VAO to the BSSF, embedding CPU, RAM, storage, geographicalAreaId, slice type (eMBB), sliceAmbr, and DLA constraints for vApp deployment. Confirm the request and BSSF response (including Kubernetes cluster credentials and network slice instantiation confirmation) through VAO and BSSF system logs. Verify that the vApp instance is successfully deployed on the selected Kubernetes node. For renegotiation validation, simulate a policy-triggered condition (e.g., high

				energy consumption on a red DLA node detected via Rating Operator) and verify that the VAO submits an updated slice intent (PUT /api/bssf/{id}) with revised DLA constraints (preference for green nodes). Confirm that ENIF re-optimises placement and NSMF updates the network slice. Expected output: system logs showing initial slice creation and vApp deployment, followed by a renegotiation request and updated deployment on a green DLA node; VAO UI screenshots showing slice creation and the migration event.

2.6.2 Component's services validation

Table 12: AF services validation.

Service Name	Description	Test objective	Acceptance Criteria (based on standardized targets, if available)	Tests procedure/outputs validation tools)
Naf_ServiceSlicePolicies	This service enables the AF/VAO to retrieve energy-related KPIs from Prometheus (via the Rating Operator) and define policies that optimise vApp energy consumption through the Policy API.	Validate that the service retrieves valid Energy-aware KPIs and allows policy creation and deployment via the VAO Policy and Metric Expression APIs.	Yes/No — The service must successfully retrieve valid energy KPIs from Prometheus and apply a policy to a vApp through the Policy API, with confirmed execution of the triggered orchestration action.	Confirm via VAO system logs, Prometheus metrics endpoint, and screenshots from the Policy API (/api/v1/policy/create) and Metric Expression API (/api/v1/expression/create). Verify that the metric expression is correctly resolved by the Rating Operator via PromQL, that the policy condition is evaluated in real time, and that the triggered action (e.g., migration to a green DLA node, scale-to-zero) is executed as expected. Expected output: VAO logs confirming metric ingestion, policy evaluation, and action execution; screenshots from the Policy API and VAO UI showing active policies and their effects on the deployed vApp.
Naf_Resources	This service enables the AF/VAO to retrieve information on available compute resources from the BSSF, specifically the list of geographical areas (Points-of-Presence) suitable for vApp placement.	Validate that the service correctly interacts with the BSSF northbound API and returns an up-to-date list of geographical areas with available compute resources.	Yes/No — The service must retrieve a valid list of geographical areas (each including geographicalAreaId, name, coordinates, and coverage) via the BSSF Location API.	Issue a GET request to /api/v1/providers/locations via the VAO and verify that the returned Location Data Model includes valid and current geographical areas matching the registered infrastructure (e.g., Genova/Area 1, Milan/Area 2, Athens/Area 3 on the CNIT testbed). Cross-check

				the data against BSSF and VAO system logs and verify consistency with the VAO dashboard display. Expected output: a valid JSON response listing active geographical areas with correct metadata; VAO UI screenshot confirming the available deployment locations.
Naf_ServiceSliceIntent	This service enables the AF/VAO to request initial service slice (compute and network) resources from the BSSF via a 3GPP TS 28.312 [9]-compliant slice intent, and to renegotiate those resources dynamically at runtime.	Validate that the service successfully sends an initial slice intent request to the BSSF and handles renegotiation when triggered by runtime conditions.	Yes/No — The service must successfully submit a slice intent (POST /api/bssf) and receive valid infrastructure credentials; and, when triggered, update the slice intent (PUT /api/bssf/{id}) with revised constraints, with BSSF/ENIF/NSMF completing the renegotiation cycle.	Validate using VAO, BSSF, and NSMF system logs. Observe the VAO UI for slice creation confirmation and verify that the vApp instance is deployed on the target Kubernetes node with the requested resources. To validate renegotiation, simulate an energy policy trigger and confirm that the updated slice intent is submitted (PUT /api/bssf/{id}), processed by the BSSF, forwarded to ENIF for placement re-optimisation, and reflected in an updated NSMF network slice configuration. Expected output: VAO and BSSF logs confirming initial slice intent acceptance and vApp deployment; renegotiation logs showing updated intent submission, ENIF placement decision, and NSMF slice update; VAO UI screenshots of the slice lifecycle and migration event.

2.7 BSSF – Business Support System Function

The BSSF is part of the AI-driven decision and operation framework. It handles the business side of the network and slice reconfiguration, facing demands from verticals.

2.7.1 Component's functional requirements validation

Table 13: BSSF functional requirements validation.

Req. Label	Description	Test objective	Acceptance Criteria (based on standardized targets, if available)	Tests procedure/outputs
REQ_BSSF_01	BSSF enables creation of a network slice and component slice in intent format.	Validate that the BSSF can request initial compute slice to ENIF and network resources to NSMF	Yes/No BSSF must provide to AF the slice placement by ENIF an successfully	To validate that the component slice is correctly created BSSF returns the placement of the components.

		for a service slice request from AF. Validate that the complete workflow of intent classification, translation and execution works.	create the network slice resource to be used by NSMF.	To validate that the slice profile is created, check if the slice profile exists as a kubernetes resource in NSMF namespace.
REQ_BSSF_02	BSSF enables update of a component slice in intent format including DLA parameters.	Validate that the BSSF can update the compute slice to ENIF with DLA label included as optimization parameter. Validate that the complete workflow of intent classification, translation and execution works.	Yes/No BSSF must retrieve the updated component slice from ENIF and return it to AF.	To validate that the component slice is correctly updated, BSSF returns the placement of the components.
REQ_BSSF_03	Mapping of available geographical areas requested by AF and provided by Edge-MF.	Validate that BSSF can correctly map the current geographical areas provided by Edge-MF into a user manageable format.	Yes/No BSSF must return to AF the areas from Edge-MF with the correct mapping.	To validate that the areas are correctly requested, BSSF returns the areas mapped.
REQ_BSSF_04	Intent format validation following the 3GPP as base specification.	Validate that the intent format is correct an follows the proposed specification.	Yes/No BSSF will return an error when the format is not compliant.	To validate that the format is correctly fulfilled, the BSSF returns error when is not followed and the possible source of the error.
REQ_BSSF_05	Correct classification of the input intent being the format common for different intents.	Validate that the BSSF classification mechanism make the correct split of the different requirements for the underlaying components.	Yes/No BSSF will correctly determine given the type of expectation objects to which underlaying NF corresponds	Doing a unique request to BSSF in intent format both types of slices are correctly managed (ENIF and NSMF)
REQ_BSSF_06	BSSF is able to add specific network and application requirements from an abstract request. Network requirements are defined in the slice using Generic Slice Template requisites. The energy aware requirements are added to support ENIF component placements.	Validate that an abstract requirement is translated correctly to ENIF and NSMF.	Yes/No BSSF adds information such as uplink and downlink bandwidth given a use case (e.g. URLLC, eMBB). BSSF supports DLA targets and could split them following the intent catalogue approach.	To validate this functionality AF inputs a slice type as a GST slice name or a specific use case and the BSSF translate it to specific network requirements.

2.7.2 Component's services validation

Table 14: BSSF services validation.

Service Name	Description	Test objective	Acceptance Criteria (based on standardized targets, if available)	Tests procedure/outputs)
Nbssf_IntentManagement	Intent-based service creation	The service enables the AF to create an application slice and a network (Service profile) slice from a common intent.	Yes/No The ENIF and the NSMF get the correct information in the corresponding slice format.	AF inputs an intent to create both type of slices and it returns a component placement, and the service profile is created.
Nbssf_IntentManagement	Intent-based service update	The service enables the AF to update an application slice and a network (Service profile) slice from an existing identifier.	Yes/No ENIF gets the correct component slice application with the modified requirements of the slice.	AF inputs an intent to request a new placement and BSSF is able to achieve it on ENIF.
Nbssf_GeographicalAreas	Mapping of geographical areas available to the AF.	The service enables the AF to request the geographical areas where the slices can be created.	Yes/No AF get the correct mapping of the geographical areas and later this mapping is used to pass the correct geographical area identifiers to ENIF and NSMF.	When AF request the geographical areas, the BSSF pass them mapped with a format a user can manage.

2.8 ENIF/NSENIF – Experiential Networked Intelligence Function/Network Slice ENIF

The ENIF and NSENIF are part of the AI-driven decision and operation framework. They are responsible to inject intelligence into the supported network management functions and network services orchestration mechanisms. They can do so by taking advantage of AI techniques, as well as the increase of the context-awareness of the applied mechanisms based on data fusion techniques.

2.8.1 Component's functional requirements validation

Table 15: ENIF/NSENIF functional requirements validation.

Req. Label	Description	Test objective	Acceptance Criteria (based on standardized targets, if available)	Tests procedure/outputs
REQ-ENIF-01	The provision of a set of mechanisms to support automation based on the translation of the end user intent/requirements to specific configuration and orchestration actions;	Validate that ENIF is capable of generating a deployment plan based on a translated intent descriptor of a vApp received from BSSF	Yes/No ENIF generates a valid deployment plan when feasible; otherwise, it returns an empty deployment plan.	To validate this functionality, ENIF provides BSSF with a valid deployment plan when feasible; otherwise, it returns an error message along with an empty deployment plan.

REQ-ENIF-02	The interaction with a set of AI mechanisms for analysis of data made available by the various monitoring probes and the provision of recommendations to orchestration mechanisms;	Validate that ENIF interacts with NWDAF/NSDAF to obtain the forecasts required as input for its proactive orchestration mechanisms	Yes/No ENIF must accept forecasts from NWDAF/NSDAF and delivers them in the correct format to its internal optimization engine for proactive orchestration	To validate this functionality, ENIF ingests forecasts from NWDAF/NSDAF, reformats them, and verifies whether the deployment plans generated by its proactive orchestration optimization mechanism are valid.
REQ-ENIF-03	The optimization and adjustment of resources and services provided by a network provider/operator, considering network slice lifecycle management operations;	Validate that ENIF's internal optimization engine can solve orchestration problems—such as virtual network embedding—either in an online manner or as multi-period optimization problems for proactive orchestration.	Yes/No When the optimization engine yields optimal solutions, ENIF provides the corresponding orchestration policies.	To validate this functionality, either BSSF requests ENIF to recommend an orchestration action by solving an optimization problem, or ENIF autonomously triggers orchestration actions by solving optimization problems.
REQ-ENIF-04	The continuous training of ML models to support decision making processes;	Validate that ENIF retrieves data from monitoring tools provided by other SBA functions to continuously train reinforcement learning (RL) agents for network slicing orchestration actions.	Yes/No ENIF collects data from SBA monitoring tools, including compute and network capacity information as well as network slice metrics such as power consumption and end-to-end delay, to feed RL agents.	To validate this functionality, ENIF is tested to ensure it continuously ingests SBA monitoring data and maintains an up-to-date, time-consistent dataset for ML/RL model training. The expected output is a continuous training pipeline that supports real-time learning and improves decision-making for network slicing orchestration.
REQ-ENIF-05	The consideration of the intent provided by end users towards the formulation of optimization mechanisms and deployment policies.	Validate that ENIF receives a vApp intent descriptor as input and translates it into an optimization problem whose solution generates a recommended deployment plan.	Yes/No ENIF accepts a slice intent descriptor from BSSF and, if it is correctly formatted, formulates an optimization problem for placing the vApp components.	To validate this functionality, BSSF requests a deployment plan from ENIF by providing a slice intent descriptor as input. If the descriptor is correctly formed, ENIF attempts to generate and return a deployment plan; otherwise, it returns an error message.

2.8.2 Component's services validation

Table 16: ENIF/NSENIF services validation.

Service Name	Description	Test objective	Acceptance Criteria (based on standardized targets, if available)	Tests procedure/outputs
Input Processing and Normalization	Ingest structured, semi-structured, and unstructured data from different data sources	This service enables **ENIF** to interface with heterogeneous data environments by successfully capturing and normalizing data from diverse sources e.g. CCMF, BSSF, NWDAF/NSDAF regardless of the data format.	Yes/No ENIF uses APIs exposed by these functions to retrieve the data and route it to its various mechanisms as appropriate for each case.	ENIF properly maps the raw inputs into its internal translation process and forwards them to the appropriate mechanism.

Data Analysis	Trigger the execution of analysis processes/scripts	This service enables ENIF to incorporate forecasting data into its optimization engine to derive proactive orchestration policies.	Yes/No ENIF accepts forecast data and deliver it in the correct format to its internal optimization engine for proactive orchestration.	ENIF ingests forecast data, reformats it, and verifies whether the deployment plans generated by its proactive orchestration optimization mechanism are valid.
Recommendation	Provide recommendations for policies enforcement or orchestration actions	This service enables orchestration network functions, such as AF, to obtain deployment plans from ENIF based on its optimization engine or RL function blocks.	Yes/No ENIF provides deployment plans and orchestration policies by leveraging solutions produced by its optimization engine or RL agents.	When ENIF recommends an orchestration policy, orchestration functions such as AF attempt to apply it; in some cases, the policy is delivered to these functions via BSSF.
Optimization Engine	Support solving of optimization problems related to network slice lifecycle management	This service enables the solution of orchestration problems—such as virtual network embedding—either in an online manner or as multi-period optimization problems for proactive orchestration.	Yes/No When the optimization engine produces optimal solutions, ENIF delivers the corresponding network slice deployments.	Either BSSF requests ENIF to recommend a network slice deployment by solving an optimization problem, or ENIF autonomously network slice deployments by solving optimization problems.
AI Functional Block Management	Support integration and interaction with AI functional blocks, considering a bidirectional information flow with ENI	This service enables interaction with AI function blocks that provide forecasts required for ENIF's decision-making, which is supported by ENIF's RL agents for network slicing.	Yes/No ENIF requests future network metrics from NSDAF for deployed network slices and updates its deployment decisions based on the RL agent's output.	When ENIF triggers a network slice deployment update using information from NSDAF AI blocks, a corresponding deployment status update is generated.
Intent Handling	Support intent handling mechanisms where intent is given as input to the ENI decision making modules	This service enables ENIF to take a slice intent descriptor as input and convert it into an optimization problem whose solution yields a recommended deployment plan.	Yes/No ENIF receives a slice intent descriptor from BSSF and, if it is properly formatted, constructs an optimization problem for the placement of vApp components.	BSSF requests a deployment plan from ENIF by providing a slice intent descriptor as input. If the descriptor is correctly formed, ENIF attempts to generate and return a deployment plan; otherwise, it returns an error message.

2.9 EGMF – Exposure Governance Management Function

The EGMF belongs to the Integrated (network) management subsystem. It is responsible for abstracting, simplifying, filtering, and aggregating the management services in order to provide a controlled way of management over Network Service or Network Slice instances, not only for some NFs like BSSF but also for vertical applications in the 6Green SBA subject to access control.

2.9.1 Component's functional requirements validation

Table 17: EGMF functional requirements validation.

Req. Label	Description	Test objective	Acceptance Criteria (based on standardized targets, if available)	Tests procedure/outputs
REQ-EGMF-01	Providing Access for trusted or untrusted VAOs to 6Green NFs	Verify that VAOs have constrained access to management NFs according to their category: trusted or untrusted.	Yes/NO	AFs receive a secret token based on their category. Untrusted AFs shall not communicate directly with management NFs; all their requests shall be routed through the BSSF. Trusted AFs, however, may directly consume the services exposed by the integrated management NFs.

2.9.2 Component's services validation

Table 18: EGMF services validation.

Service Name	Description	Test objective	Acceptance Criteria (based on standardized targets, if available)	Tests procedure/outputs
Negmf_AccessProvMns	It is an endpoint that can be reached through Kuberntes API Server.	Verify that VAO access to integrated management NFs is enforced according to its category. A trusted VAO shall be allowed to directly perform NSI lifecycle management operations, whereas an untrusted VAO shall be prevented from directly accessing the integrated management NFs. In such cases, the Kubernetes API server shall reject the request with HTTP status code 403 Forbidden.	Yes/NO	This is a unit test procedure and consists of two phases. First, an untrusted VAO configured with its unique token and secret files attempts to perform NSI lifecycle management operations. Direct access to the integrated management NFs shall be rejected, and the VAO shall only be permitted to access the BSSF. Second, a trusted VAO configured with its corresponding token and secret files attempts to perform NSI lifecycle management operations. The request shall be authorized, allowing the trusted VAO to directly create, modify, and delete NSIs without relying on the BSSF.

2.10 NSMF – Network Slice Management Function

The NSMF is part of Integrated (network) management subsystem. It inspects the lifecycle management of network slices, providing essential services to create, maintain, and terminate network slice instances.

2.10.1 Component's functional requirements validation

Table 19: NSMF functional requirements validation.

Req. Label	Description	Test objective	Acceptance Criteria (based on standardized targets, if available)	Tests procedure/outputs
REQ-PRO_NSSI-FUN-1	NSMF fulfils a request for a network slice instance that has specific characteristics, either by creating a new network slice instance or by utilizing an existing one.	Examine the lifecycle management operations for NSIs	Performing the lifecycle management of NSIs based on taking decisions by VAOs or ENIF	Verify that an NSI can be created, updated, upgraded, and deleted at runtime.
REQ-PRO_NSSI-FUN-2	NSMF allows the consumer to access further details about the involved Containerized Network Functions (CNFs) and NSs (Network Slices).	Verify the results of lifecycle management operations for NSIs by checking the status of the CNFs serving them.	Yes/No	Getting the status of NSIs after creating, updating, upgrading, or deleting them.
REQ-PRO_NSI-FUN-3	NSMF deallocates a network slice instance.	Verify the deletion of an NSI and the idempotency of the action.	Release network and computational resources by deleting NFs that exclusively serve the NSIs marked for deletion.	Trigger the NSI deletion request.
REQ-PRO_NSI–FUN-4, REQ-PRO_NSSI–FUN-9	NSMF activates an existing network slice or slice subnet instance which is inactive.	Ensure allocating network or computational resources to NSI	VAO applications/services receive the requested QoS	Trigger NSI creation; then determining the requested QoS
REQ-PRO_NSI–FUN-5, REQ-PRO_NSSI–FUN-10	NSMF deactivates an existing network slice or slice subnet instance which is active.	Ensure the network resources are not allocated to NSIs anymore	VAO applications/services cannot send and receive traffic anymore	Set the QoS of NSIs to zero throughput for both downlink and uplink.
REQ-PRO_NSI-FUN-6	NSMF modifies an existing network slice instance.	Ensure NSIs adapt to the evolving behaviours of the applications	Enabling VAOs to adapt the NSIs used by their applications and services according to energy consumption, performance, cost, and customer-specific requirements.	When Service Profiles or Slice Profiles are updated, the associated NSIs should be updated accordingly. These updates may involve upgrading the NSIs by changing the topology of the NFs serving them, or simply modifying the NSI-level QoS parameters without altering the underlying topology.
REQ-PRO_NSI-FUN-7	Enable network slice management service consumer to obtain network slice management data (e.g., PM data, FM data).	Ensure that the NSIs remain active and compliant with the QoE/QoS requirements requested by VAOs.	Enabling VAOs to know about the status of their NSIs.	The status of NSIs is reported in their associated Service Profiles/Slice Profiles.

2.10.2 Component's services validation

Table 20: NSMF services validation.

Service Name	Description	Test objective	Acceptance Criteria (based on standardized targets, if available)	Tests procedure/outputs
Nnsmf_NSProvMns_AllocateNsi	This management service enables its consumer to request allocating or modifying a Network Slice Instance (NSI)	Creating or updating an NSI depending on the type of request.	Provisioning and updating an isolated network through network slicing, allowing it to adapt to the behavior and requirements of VAO applications without affecting other applications or services.	Creating or updating a Service Profile/Slice Profile and sending a POST request to create the corresponding NSI, or a PATCH request to update or upgrade the corresponding NSI.
Nnsmf_NSProvMns_DeallocateNsi	This management service enables its consumer to request deallocating an NSI	Deleting the corresponding NSI upon receiving a DELETE request for the associated Service Profile or Slice Profile.	Deleting an NSI that is no longer required in order to save costs and reduce energy consumption.	Sending DELETE request for an NSI; therefore, the corresponding NSI should be deleted.
Nnsmf_NSProvMns_GetStatusNsi	This management service enables its consumer to request getting the status of an NSI	Reporting the status of NSIs in terms of being created, deleted, or modified.	Yes/No	Sending the GET request for a Service Profile/Slice Profile and receiving the status of the associated NSI.

2.11 NFMF – Network Function Management Function

The NFMF is part of Integrated (network) management subsystem. It is responsible for setting up NFs provisioning service by generating their configuration files, managing their lifecycle, and ensuring that they are working properly with one another.

2.11.1 Component's functional requirements validation

Table 21: NFMF functional requirements validation.

Req. Label	Description	Test objective	Acceptance Criteria (based on standardized targets, if available)	Tests procedure/outputs
REQ-NFMF-01	Rendering NF configuration files and performing lifecycle management for NFs.	Verifying the rendered configuration files for NFs based on slice-specific configurations issued by the NSMF, and starting or stopping NFs according to the lifecycle management of NSIs.	An NF should serve the NSIs that it hosts.	Performing NSI lifecycle management and verifying that NFs are properly configured, started, and stopped to support the creation, deletion, and modification of NSIs in practice.

REQ-NFMF-02	Monitoring the NFs and reporting their status.	Monitoring the status of NFs serving NSIs, including whether they are Running, Pending, or Crashing.	Yes	Checking NFMF logs and the status of the NFs itself over the cluster.
REQ-NFMF-03	Dynamically resolving dependencies among NFs based on their functionality or their associated NSI service lists.	NF dependencies specified in configuration files should be resolved to ensure that the NFs operate properly.	It's necessary for NFs to operate properly.	Checking NF configurations to verify that dependencies are resolved and that NFs are connected to other NFs to establish NSIs in practice.

2.11.2 Component's services validation

Table 22: NFMF services validation.

Service Name	Description	Test objective	Acceptance Criteria (based on standardized targets, if available)	Tests procedure/outputs
Nnfmf_ProvMns_WMI	This endpoint belongs to a sidecar manager deployed alongside each NF and is responsible for starting, stopping, and reporting the status of the associated NF.	Ensuring the relevant NF are run, stop, and their status are correctly reported.	Yes	Communicating with the specified endpoint by sending POST or GET requests, as shown below: curl -X POST http://127.0.0.1:60002/api/v1/wmi/{start, stop}/{NF} curl -X GET http://127.0.0.1:60002/api/v1/wmi/status/{NF}
Nnfmf_ProvMns_MCI	This endpoint is provided by the same sidecar manager deployed alongside each NF. It resolves the dependencies of the associated NF by communicating with corresponding sidecar managers associated with other NFs.	Verifying that all inter-NF dependencies specified in the configuration file of each NF has been properly resolved.	Yes	Resolving the dependency, such as obtaining the IP address of another NF, by sending a GET request as shown below: curl –X Get http://nrf.ogs-nrf-shared-50331649.networkso.t9s.local.:60001 /api/v1/resolve/eth0

2.12 EdgeMF – Edge-cloud Management Function

The EdgeMF is part of Management framework for edge-cloud resources. The EdgeMF is responsible for natively providing computing resources at the network edge for industry-specific applications, i.e., vApps.

2.12.1 Component's functional requirements validation

Table 23: EdgeMF functional requirements validation.

Req. Label	Description	Test objective	Acceptance Criteria (based on standardized targets, if available)	Tests procedure/outputs
REQ-EDGEMF-01	Support the provision of computing resources available in a specific Service Area as EDN resources.	Verify that the component can publish and expose EDN computing resources associated with a specific Service Area.	Provisioned resources are successfully registered and retrievable via API.	Submit a provisioning request and receive an HTTP response containing the computing resources.
REQ-EDGEMF-02	Enable the management of EDN resources via CRUD operations.	Validate that the system supports Create, Read, Update, and Delete operations for EDN resources via API.	CRUD operations return standard HTTP status codes and maintain resource state consistency.	Execute a sequence to create, read, and delete the EDN resource, send the corresponding requests, and verify the responses to confirm that the resource no longer exists.
REQ-EDGEMF-03	Support the creation of EDN resources covering a provided Service Area in the form of Geographical Area or Topological Area.	Confirm that EDN resources can be defined using geographical service areas.	The system accepts both area types and persists them according to the defined geographical schema.	Create resources using geographical area and submit a request to retrieve them, returns a valid geographical information.
REQ-EDGEMF-04	Support the reception of computing resources requirements in a format to be defined as an input.	Ensure the component receives and validates computing resource requirement requests.	Valid requests pass schema validation, while invalid requests return a validation error.	Submit valid and invalid payloads receiving different HTTP responses.
REQ-EDGEMF-05	Be able to find available resources matching the requirements in the requested areas by interfacing with other NFs such as the CCMF.	Verify that the component can query the CCMF external network function to retrieve resources that meet the requirements.	The CCMF APIs are invoked, and a list of resources that meet the defined requirements is returned.	Submit a discovery request and receives an aggregated list of compatible resources.
REQ-EDGEMF-06	Support the clustering of the EDN resources into orchestration domains.	Validate that EDN resources can be grouped into orchestration domains.	Resources can be assigned to a domain and queried by domain with consistent membership results.	Assign resources to a domain and query the domain to obtain a response that includes the correct resource members.

2.12.2 Component's services validation

Table 24: EdgeMF services validation.

Service Name	Description	Test objective	Acceptance Criteria (based on standardized targets, if available)	Tests procedure/outputs
Nedgemf_EDNManagement	Management of EDN objects that provide services in a specific edge area.	Verify that the component can manage EDN service objects associated with a specific edge area.	EDN service objects can be created, retrieved, updated, and deleted via API while preserving their association with the specified edge area and service metadata.	Execute CRUD operations on EDN service objects with edge area attributes and receive successful HTTP responses and consistent data from the service objects.
Nedgemf_EdgeClusterManagement	Management of the compute clusters composed of one or multiple EDNs.	Validate that the component can manage compute clusters composed of one or multiple EDNs.	Clusters can be created, modified, queried, and deleted, maintaining consistent membership relationships with associated EDNs.	Create cluster with multiple EDNs and receive successful HTTP responses with EDN members.

2.13 CCMF – Cloud-Continuum Management Function

The CCMF is part of Management framework for edge-cloud resources. It acts as a central repository responsible for maintaining an up-to-date registry of the volatile cloud-continuum computing resources and their location and topology, especially at the edge, of the different geographical areas covered by the network.

2.13.1 Component's functional requirements validation

Table 25: CCMF functional requirements validation.

Req. Label	Description	Test objective	Acceptance Criteria (based on standardized targets, if available)	Tests procedure/outputs
REQ-CCMF-01	Maintain an up-to-date registry of available computing resources at each network location.	Ensure that CCMF can collect complete information about the computing resources of registered providers.	Show completion of information collection	CCMF queries Information on the pre-configured K8s cluster through the K8s API. For each node, in addition to the K8s labels, the capacity information is stored in the in-memory repository of the CCMF and assigned to labels whose key is prepended with the status/capacity/ string
REQ-CCMF-02	Fulfil information requests of compute resources availability at a provided set of network locations.	Ensure that CCMF sends responses with information about compute resources to requesters	Show exchange of messages	CCMF receives a GET request on the path /geographical areas and sends a response with the full list of the geographical areas and their node resources

2.13.2 Component's services validation

Table 26: CCMF services validation.

Service Name	Description	Test objective	Acceptance Criteria (based on standardized targets, if available)	Tests procedure/outputs
Nccmf_ComputeResourcesProvisioning	Configure and deploy virtual and/or physical compute resources	Show the list of available compute resources	Yes/no	CMF receives a GET request on the path /geographical areas and sends a response with the full list of the geographical areas and their node resources. Configuration of the compute resources is then enforced by the EdgeMF.
Nccmf_ComputeResourcesRegistration	Maintain a registry of computing resource providers	Validate a JSON configuration file providing details on the geographical areas	Yes/no	CCMF reads a JSON configuration file providing the full geographical areas details.

2.14 NFVCL – Network Function Virtualization Convergence Layer

The NFVCL is a network-oriented meta-orchestrator, specifically designed for zeroOps and continuous automation. It can create, deploy and manage the lifecycle of different network ecosystems by consistently coordinating multiple artefacts at any programmability levels (from physical devices to cloud-native microservices).

2.14.1 Component's functional requirements validation

Table 27: NFVCL functional requirements validation.

Req. Label	Description	Test objective	Acceptance Criteria (based on standardized targets, if available)	Tests procedure/outputs
REQ-NFVCL-01	Have the ability to interact with external NFVOs or to work in a standalone fashion	Assess that the NFVCL can perform LCM operations without the presence of an NFVO	Performance of LCM operations, especially Day-1 (e.g., NS instantiation: select facilities and networks for attaching xNFs)	Updated Blueprints interact with the proper provider rather than with an NFVO

REQ-NFVCL-02	Have access to a database to maintain information on the instantiated NSs	Since the NFVCL is cloud-native, it has no state, so it needs to read/write information related to the topology and the instantiated Blueprints on a Mongo DB.	Show completion of read/write operations	Read the NFVCL logs
REQ-NFVCL-03	Receive requests for Day-2 operations from the NSMF and the NFMF	Show the exchange of messages between the NSMF/NFMF and the NFVCL	Messages are exchanged	Read the NFVCL logs

2.14.2 Component's services validation

Table 28: NFVCL services validation.

Service Name	Description	Test objective	Acceptance Criteria (based on standardized targets, if available)	Tests procedure/outputs
Nnfvcl_nbi	Expose CRUD REST APIs for lifecycle management through the methods defined in the Blueprint Category meta-models	Show interactions with the SBA	Yes/no	Read the NFVCL logs
Nnfvcl_Topology	Expose VIM list and networks status, available Kubernetes clusters, PNFs and metric servers. Terraform resources on the VIMs	Show that all the resources are available in the topology	Ability to handle with any type of resources	Read the NFVCL logs upon GET Topology request
Nnfvcl_BlueprintManager	Handle the LCM operations of different NSs	Measure the time it takes for Day-0/Day-N operations on the most relevant Blueprints	Entire LCM executed in <900 s	Deployment of the main SBAs, execution times reported in D2.3 [10]
Nnfvcl_ProviderAggregator	Interact directly with external entities, like Kubernetes or OpenStack, to enable features not supported by the VIM	Show the operations being passed from the Aggregator and performed by the Providers	Interact with VIMs and K8s clusters without an NFVO	Read the NFVCL logs

2.15 MetalCL – Metal Convergence Layer

The MetalCL is a service for the dynamic management and terraforming of bare-metal infrastructure, including physical servers, network equipment, operating systems, and virtualization environments, following an Infrastructure as Code (IaC) paradigm. It enables on-demand provisioning, configuration, and orchestration of IaaS/PaaS environments and acts as an actuator for 5/6G platforms by supporting deployment of complex distributed systems such as OpenStack and Kubernetes.

2.15.1 Component's functional requirements validation

Table 29: MetalCL functional requirements validation

Req. Label	Description	Test objective	Acceptance Criteria (based on standardized targets, if available)	Tests procedure/outputs
REQ-METALCL-01	The MetalCL must be granted admin access to the HW/SW resources available in the infrastructure.	Verify that the MetalCL is allowed to perform bare-metal configurations, install SW, etc. on servers and network devices.	Yes/no	Admin privileges verified via server command line
REQ-METALCL-02	IPMI must be available on the servers	Ensure that all machines within the infrastructure are equipped with IPMI	Yes/no	Presence of IPMI verified via server command line
REQ-METALCL-03	Receive requests for Day-2 operations from the control and management NFs	Show the exchange of messages between the NFs and the MetalCL	The MetalCL can receive requests from the NFVCL	Upon request for a new power configuration, the MetalCL GUI exposes the updated metrics

2.15.2 Component's services validation

Table 30: MetalCL services validation.

Service Name	Description	Test objective	Acceptance Criteria (based on standardized targets, if available)	Tests procedure/outputs
Nmetalcl_MaaS	Discovery, commissioning, deployment, and dynamic reconfiguration of physical servers	Assess that MaaS operations are performed properly	Yes/no	Use the MetalCL API for retrieving information about the underlying hardware configuration. See D2.3 [10] for output examples
Nmetalcl_Ansible	Drive any software installations, applications and OS reconfigurations over the bare-metal servers installed by MaaS	Assess that Ansible has installed/modified the requested SW	Yes/no	Verify the output of the LLDP protocol by command line or REST interfaces
Nmetalcl_NetCL	Discovery of networking devices to enable the configuration of overlay networks.	Assess that the NetCL has installed/modified the requested SW	Yes/no	Verify the updated topology by command line or REST interfaces
Nmetalcl_CState	Change the current C-state	Assess the ability to modify the current C-state in a timely manner	The request for a new C-state is actuated	The time needed to enter a different C-State is negligible; see D2.3 [10] for results
Nmetalcl_Governor	Change the current Governor	Assess the ability to modify the current Governor in a timely manner	The request for a new Governor is actuated	Measure the time needed to adopt a different Governor is negligible; see D2.3 [10] for results

2.16 IDAF – Infrastructure Data Analytics Function

IDAF is an analytics function focused on the infrastructure layer. Thus, it collects and analyses infrastructure-related metrics from, e.g., bare-metal servers, virtual machines, containers, Kubernetes clusters, GPUs, etc. It complements NWDAF, which focuses on network-level analytics.

2.16.1 Component's functional requirements validation

Table 31: IDAF functional requirements validation.

Req. Label	Description	Test objective	Acceptance Criteria (based on standardized targets, if available)	Tests procedure/outputs
REQ-IDAF-01	The servers must support the Intel RAPL module.	Ensure that all the servers have the Intel RAPL module.	YES	Test if powertop command works.
REQ-IDAF-02	The IDAF must have access to a Prometheus db.	Ensure that a Prometheus db instance is up and running.	YES	Check if Prometheus web UI works

2.16.2 Component's services validation

Table 32: IDAF services validation.

Service Name	Description	Test objective	Acceptance Criteria (based on standardized targets, if available)	Tests procedure/outputs
Nidaf_cadvisor	The IDAF must have access to cAdvisor metrics on the Prometheus db.	Ensure that cAdvisor metrics are scraped by Prometheus.	Target is UP.	Check the targets' section on Prometheus.
Nidaf_kubestatemetrics	The IDAF must have access to Kube state metrics metrics on the Prometheus db.	Ensure that Kube state metrics metrics are scraped by Prometheus.	Target is UP.	Check the targets' section on Prometheus.
Nidaf_scaphandre	The IDAF must have access to Scaphandre metrics on the Prometheus db.	Ensure that Scaphandre metrics are scraped by Prometheus.	Target is UP.	Check the targets' section on Prometheus.

2.17 Cloud-Continuum Wide Area

The Cloud-Continuum Wide Area requires the implementation of a fully bidirectional, dynamic, and intent-based system that moves away from traditional top-down segmentation approaches. The component must fundamentally provide intent-based interoperability through the Business Support System Function (BSSF), using declarative interfaces compliant with the 3GPP TS 28.312 [9] standard, to hide technical complexity from vertical stakeholders. This requires a Dynamic Resource Registry managed by the Cloud-Continuum Management Function (CCMF) and EdgeMF, which must maintain a real-time, consistent view of volatile edge computing resources and their network topology. In addition, the system requires a Kubernetes-native

management framework (NSMF/NFMF) capable of automating the entire lifecycle of network functions using the operator pattern. Therefore, the implementation and execution of the cloud-continuum wide area depend on the operational synchronization between ENIF, BSSF, EdgeMF, CCMF and NSMF during the Cloud Continuum slicing workflow. Initially, the ENIF drives the optimization process by identifying and instantiating the optimal slice based on infrastructure availability, subsequently returning the designated deployment cluster. The result activates the BSSF coordination layer, which connects the ENIF to the AF by translating the location of the selected segment into a specific network segment resource.

In this stage, the NSMF takes over and relies on the resources provided by the BSSF to oversee the lifecycle and day-to-day operations of the portion. By working together, these components enable the wide-area cloud continuum to function as intended, combining ENIF's resource allocation, the BSSF's exposure and translation, and the NSMF's lifecycle management.

3 Validation of Integrated Frameworks and Solutions

6Green Integrated Frameworks and Solutions are five sets of SBA components, where each SBA component is part of one framework/solution. Sets provide specific advanced features, thus enabling improvement of the future 6G ecosystem through the three 6Green paradigms. In this section, each framework/solution is presented holistically by addressing their unique functional requirements and services they provide, thus not focusing on specific features of every single component which has been elaborated in section 2. As also mentioned in section 2, the 6Green SBA has been extensively tested within WP3 activities, therefore, details for understanding the test/validation procedure of each requirement are presented here, i.e., what is the test objective, what is acceptance criteria, and how to perform the test.

3.1 Enhanced observability, monitoring and analytics framework

The Observability, monitoring and analytics framework supports energy-aware relevant metrics and KPIs up to the slice level. It is composed of MDAF, NSDAF, and NWDAF network functions.

3.1.1 Component's functional requirements validation

Table 33: Enhanced observability, monitoring and analytics Framework functional requirements validation.

Req. Label	Description	Test objective	Acceptance Criteria (based on standardized targets, if available)	Tests procedure/outputs
Req_1_mdaf	MDAF availability to collect data measurements from various sources. MDAF per slice-id data queue production and categorization (labelling).	MDAF data and KPIs collection, data producer for Req_2_nsdaf	Data successfully collected from Prometheus timeseries database (CPU/RAM metrics) and processed with slice_id labels	Subscribe to MDAF interface and verify produced monitoring measured data. Validate data is stored in Redis channels and database with proper slice_id labelling.
Req_2_nsdaf	NSDAF availability to analyse various data measurements produced by MDAF. NSDAF per slice-id data analytics and predictions based on labelling.	NSDAF data analytics data produced for Req_4_analytics.	Power measurement data for containers and machines successfully received from Redis database and processed to calculate daily average energy consumption per slice_id	Subscribe to NSDAF interface and verify produced analytics data. Validate ARIMA model forecasts energy consumption 1 to 28 days into the future.
Req_3_mon	Monitoring/observability data availability in time series data bases.	Monitoring and time series data and KPIs collection available to be consumed by Req_1_mdaf.	Prometheus timeseries database contains CPU/RAM metrics accessible by MDAF	Subscribe to Prometheus database interface and verify monitoring/observability data availability
Req_4_analytics	NSDAF availability to analyse the labelled slice data.	NSDAF module capability to consume data from	NSDAF successfully processes power measurement data	Verify NSDAF data processing through Flask API endpoints. Validate data returned as list

		Req_2_nsdaf and produce analytics and feed Req_5.	and calculates daily average energy consumption per slice_id with granularity of one day (finer granularity possible if needed)	of tuples (date string, energy value in float).
Req_5_pred	NSDAF prediction of analysed labelled slice data.	NSDAF module capability to predict future energy data consumption based on Req_4_analytics.	ARIMA model (p=5, d=1, q=1) successfully forecasts energy consumption for 1 to 28 days into the future per slice_id	Validate prediction through POST predicted_power_microwatts endpoint. Verify output format includes data array, timestamp, error status, and error_details.

3.1.2 Component's services validation

Table 34: Enhanced observability, monitoring and analytics Framework services validation.

Service Name	Description	Test objective	Acceptance Criteria (based on standardized targets, if available)	Tests procedure/outputs
Service_req_1 analytics	NSDAF energy consumption analytics for the labelled slice data information.	Validate the energy analytics functionality per slice data availability.	NSDAF successfully processes power measurement data and calculates daily average energy consumption per slice_id with configurable granularity (default: one day, finer granularity possible if needed)	NSDAF data processing for analytics; energy consumption analysis. Verify through POST /current_power_microwatts endpoint. Validate output format includes data array (list of tuples with date string and energy value in float), timestamp, error status, and error_details.
Service_req_2_prediction	NSDAF energy consumption prediction for the labelled slice data information.	Validate the prediction functionality per slice data availability.	ARIMA model (p=5, d=1, q=1) successfully forecasts energy consumption for 1 to 28 days into the future per slice_id. Model effectively captures temporal dependencies while maintaining interpretable structure.	NSDAF data processing for prediction; energy consumption prediction for one week (configurable up to 28 days). Verify through POST /predicted_power_microwatts endpoint. Validate ARIMA model components: AR (p) models' relationship with past observations, I (d) applies differencing for stationarity, MA (q) model's relationship with past forecast errors.
Service_req_3_exposure	NSDAF energy consumption exposure for the analytics and predicted	Validate the NF data energy consumption exposure functionality per	Flask web service successfully exposes REST API endpoints accessible by ENIF/VAO and other	NSDAF data exposure for analytics and prediction through Flask API. Validate GET /slice_ids endpoint returns list of available

	labelled slice data information.	slice data availability.	systems. All slice_ids retrievable and data properly formatted.	slice_ids. Verify POST endpoints for current and predicted power consumption return properly formatted responses. Test API accessibility from ENIF/VAO integration points.

3.2 AI-driven decision and operations framework

The AI-driven decision and operations framework is related to AI-driven decisions and operations. In particular, its network functions (BSSF and ENIF/NSENIF) are cooperating in the slice creation process.

3.2.1 Component's functional requirements validation

Table 35: AI-driven decision and operations framework functional requirements validation.

Req. Label	Description	Test objective	Acceptance Criteria (based on standardized targets, if available)	Tests procedure/outputs
Req_1_bssf	BSSF availability to transform a VAO-provided vApp descriptor into an enriched network slice request by aggregating both SLA and DLA requirements into a 3GPP-compliant Slice Intent format.	BSSF Slice Intent creation and provisioning for Req_3_create_slice and Req_4_update_slice	The vApp descriptor received from the VAO is correctly transformed into a valid 3GPP-compliant format to create a Slice Intent.	The Slice Intent is provided as input to the ENIF decision-making components, and BSSF waits for the ENIF response containing the deployment plan.
Req_2_enif	ENIF availability to generate orchestration actions in the form of a deployment plan by leveraging its decision-making mechanism, either through optimization problem solving or RL-agent-based decision-making.	Provide the deployment plan for Req_3_create_slice and Req_4_update_slice.	ENIF generates a feasible deployment plan that respects infrastructure resource constraints, ensures the vApp's QoS requirements, and optimizes the objective function defined by the Slice Intent and the provider.	The deployment plan generated by ENIF's decision-making components is returned to BSSF.
Req_3_create_slice	NF Set-3 shall provide support for network and application slice creation with the optimal placement.	Verify that NF Set-3 correctly places the application components following ENIF placement optimization.	The constraints specified in the slice intent must be satisfied by the solution, including location requirements and compute resource demands, while ensuring that no available infrastructure capacity is violated in the optimized placement.	Make a slice request to BSSF (method GET) with some target requirements and get the placement of the components fulfilling those requirements.
Req_4_update_slice	NF Set- 3 shall provide support for network and application slice update with the optimal assignation of application components.	Verify that NF Set-3 correctly updates the application components for an existing placement.	The slice is updated following the new constraints.	Make a slice update request to BSSF (PUT method) with extra target requirements such as DLA and get the updated slice placement.

3.2.2 Component's services validation

Table 36: AI-driven decision and operations framework services validation.

Service Name	Description	Test objective	Acceptance Criteria (based on standardized targets, if available)	Tests procedure/outputs
Service_req_1_intent_handling	BSSF receives a vApp descriptor, translates it into a 3GPP-compliant Slice Intent, and provides it as input to ENIF.	Validate that BSSF can convert the VAO descriptor into a valid Slice Intent format and deliver it to ENIF through the appropriate API call.	BSSF translates the vApp descriptor through its translation mechanism into a Slice Intent format suitable for ENIF's decision-making components.	Through a POST request, BSSF sends the Slice Intent to ENIF, and ENIF returns a deployment plan to BSSF as the original Slice Intent enriched with deployment policies generated by ENIF.
Service_req_2_infra_provision	ENIF retrieves the available infrastructure resources by leveraging information provided by CCMF/EdgeMF.	Validate that ENIF retrieves infrastructure availability information and provides it to its decision-making components in the correct format.	ENIF receives information on available memory, storage, and CPU resources across infrastructure worker nodes, enabling vApp placement, update orchestration actions, and network slice deployment.	By utilizing this information, ENIF's decision-making for orchestration actions ensures that physical infrastructure resource constraints are not violated.
Service_req_3_rl_decision_making	ENIF trains an RL agent using training data from NWDAF/NSDAF to support decision-making for optimal network slicing.	Validate that ENIF generates a network slice deployment plan based on the decision-making output of each trained RL agent.	ENIF receives a slice request via a POST API call and returns a slice deployment plan if the request is feasible.	This deployment plan is returned to BSSF and subsequently forwarded to the SBA orchestration components for execution of the decision plan.
Service_req_4_optimization	ENIF includes an optimization mechanism that enables either reactive or proactive decision-making—leveraging forecasts from NWDAF/NSDAF—by solving an optimization problem.	Validate that ENIF generates a vApp deployment plan based on the output of its optimization mechanism.	ENIF receives a Slice Intent via a POST API call and returns a vApp deployment plan if the request is feasible.	This deployment plan is returned to BSSF and subsequently forwarded to the SBA orchestration components for execution of the decision plan.

3.3 Enhanced control policy framework

The Enhanced control policy framework includes PCF, NEF, and AF network functions. The framework deals with control plane policies as well as network capability exposure – both for observability and control purposes.

3.3.1 Component's functional requirements validation

Table 37: Enhanced control policy framework functional requirements validation.

Req. Label	Description	Test objective	Acceptance Criteria (based on standardized targets, if available)	Tests procedure/outputs
REQ_NS2-01	NF Set- 2 shall provide integrated support for policy management and Slice Offloading, triggered by 6Green SBA NFs (ENIF, VAO)	Verify that NF Set- 2 supports SliceOffloading and can modify some QoS parameters, based on requests coming from ENIF/VAO.	NF Set- 2 accepts ENIF/VAO requests for slice update/offloading. NF Set-2 forwards profiles modification requests to UDR correctly.	VAO/ENIF initiates SliceOffloading procedure. Observe NF Set- 2 behaviour and network reaction.
REQ_NS2-02	NF Set- 2 shall provide integrated support for network capability exposure in terms of events triggered by subscriber mobility or state change.	Verify that NF Set- 2 supports Network exposure.	NF Set-2 accepts AF requests for event subscriptions. NF Set-2 notifies the client when an event occurs.	AF initiates Event Exposure subscription via NF (NEF) for a specific event to be notified. NF Set-2 notifies real network event (UE movement, session change, traffic flow).

3.3.2 Component's services validation

Table 38: Enhanced control policy framework services validation.

Service Name	Description	Test objective	Acceptance Criteria (based on standardized targets, if available)	Tests procedure/outputs
Service_req_slice_offloading	Slice Offloading and slice policy modification capabilities through a NEF, enabling ENIF and VAO to request slice updates and QoS parameter modifications according to 6Green SBA procedures.	Validate that NF Set- 2 provides integrated support for Slice Offloading and policy management, correctly handling requests triggered by ENIF/VAO and propagating slice profile and policy updates towards the relevant core network functions.	NEF Flask web service successfully exposes REST API endpoints for Slice Offloading and policy management, accessible by ENIF and VAO. NF Set- 2 correctly accepts and processes ENIF/VAO SliceOffloading and slice update requests. Policy and slice profile modification requests are correctly forwarded by NF Set- 2 to UDR. Requests and responses are properly formatted and compliant with the expected interfaces.	ENIF/VAO initiates a Slice Offloading procedure by invoking the REST API exposed. Verify that service is reachable and correctly handles incoming SliceOffloading requests. Observe NF Set- 2 behaviour in response to the request, including validation and internal processing. Validate that NF Set- 2 forwards slice policy and QoS parameter modification requests to UDR Verify successful end-to-end interaction between ENIF/VAO, NF Set- 2 and downstream network functions.
Service_req_event_exposure	NF Set- 2 network capability exposure service for event	Validate that NF Set- 2 supports network capability	NF Set- 2 successfully accepts AF event	AF initiates an event exposure subscription via the NEF.

	notifications triggered by subscriber mobility or network state changes.	exposure by allowing AFs to subscribe to network events and by correctly notifying subscribed clients when relevant events occur.	subscription requests via NEF. Event subscriptions are correctly registered and managed by NF Set- 2. NF Set- 2 detects real network events (e.g. UE mobility, session changes, traffic flow). NF Set- 2 notifies the subscribed AF when the corresponding event occurs. Requests and notifications are properly formatted and delivered end- to- end.	Verify that NF Set- 2 accepts and registers the subscription request. Trigger a real network event (UE movement, session change or traffic flow variation). Confirm that NF Set- 2 detects the event and sends a notification to the AF. Verify correct end- to- end interaction.

3.4 Integrated management subsystem

Within the NF set 4, the NSMF and NFMF are responsible for providing the basic functionalities required for the lifecycle management of network slices, including creating, updating, upgrading, and deleting them. No specific requirements have been identified for this set/subsystem of functions next to requirements/test and corresponding evaluation provided in D3.3 [1].

3.5 Management framework for edge-cloud resources

The Management framework for edge-cloud resources introduces two management network functions, i.e., EdgeMF and CCMF, which aim at empowering the SBA with the native capability of providing computing resources to vertical applications. In that regard, EdgeMF and CCMF collaborate with other NFs, such as BSSF and ENIF, to identify the optimal infrastructure for implementing a vApp in a specific geographic area.

3.5.1 Component's functional requirements validation

Table 39: Management framework for edge-cloud resources functional requirements validation.

Req. Label	Description	Test objective	Acceptance Criteria (based on standardized targets, if available)	Tests procedure/outputs
REQ_NS5-01	NF Set 5 support the creation of EDN resources covering a provided Service Area in the form of Geographical Area or Topological Area, as well as the grouping of EDNs into orchestration domains	Confirm that EDN resources can be defined using geographical service areas and grouped into orchestration domains.	Yes/no	Execute a sequence to create, read, delete and group the EDN resource, send the corresponding requests, and verify the responses.
REQ_NS5-02	NF Set- 5 shall provide support for maintaining a registry of computing resource providers and available physical/and or virtual computing resources.	Ensure that CCMF can collect complete information about the computing resources of registered providers.	Yes/no	CCMF reads a configuration file with list of geographical areas and K8s clusters in them. Then, for each node in the K8s cluster capacity information is queried through the K8s API. For each node, in addition to the K8s labels,

4 Performance Evaluation

Performance evaluation of the 6Green ecosystem is based on use cases, whose general description and set up, goals, as well as validation and evaluation methodology definitions can be found in the deliverable D5.4 [6]. Within each use case, specific 6Green SBA actions have been validated, and the use case performance have been evaluated against each use cases' specific KPIs.

4.1 Use case 1: Critical Operation Maintenance during Energy-Constraint Disaster Scenarios

Table 40: UC1 SBA actions validation.

SBA Action Name	Description of operations at 6Green SBA level	Cause (trigger event)	Tests procedures/outputs
Scale to zero	Turning off unused components (e.g., pods, containers) and automatically switching them back on when required, e.g., traffic (re)directed to the component.	No load for a specific component, and/or no load in a specific area/time for a specific slice	**Test procedure**: Scaling to zero is demonstrated with 5/6G core network container and vApp container when operational mode switches from normal to disaster (both containers are scaled to zero at cloud site), and while switching from disaster scenario to normal one (containers located at edge site are scaled to zero). **Output**: components are properly deactivated and reactivated at the cloud and edge sites upon switching from normal to disaster mode and vice versa
Migration	support interoperable, fast and seamless geographical relocation of images, sessions and related workloads of slices/NFs and/or vApps.	Availability of (more) green energy available at specific site; SLA change; infrastructure/resource consolidation or interruptions	**Test procedure**: Migration is demonstrated as a complementary scenario to "scale to zero" - during the operational mode switch, 5/6G CN and vApp operations are migrated to another infrastructure site without or with minimal service interruption. **Output**: instances are migrated within the same cluster without any service interruptions.
Infrastructure power management	Managed by updating infrastructure components' configurations according to the requirements and current situation; including HW-level operations in particular: consolidation of resources in the continuum, e.g., AR/LPI operations on servers, bootstrap, etc.	Traffic volume change, SLA modification, re-optimization (e.g., lower latency UPF for supporting migration to a farthest facility)	**Test procedure**: Infrastructure power management is demonstrated for RAN part of the 5/6G network (i.e., RAN is deployed at the edge site of cloud-continuum) - agentic AI system enforces configuration modifications (MIMO level, output radio power, radio channel bandwidth) based on data available from observability system and SLA requirements. **Output**: critical operations are preserved in the presence of power management operations applied to the infrastructure.
Observability	Needed as an input into functions that exploit observability/monitoring data for their purposes (e.g., energy-aware scaling, migration, etc.), as well as for various (post-) analysis.	ININ site provides continuous power consumption monitoring data (collected by qMON platform and its underlying tools, such as Scaphandre and others, deployed at the edge site) which are available	**Test procedure**: Observability function is demonstrated by showing multiple graphs, logs and real time insights into the system, as well as source of evidence for evaluation of UC1 KPIs.

		directly from qMON infrastructure of via 6Green observability solution	**Output:** the collected and elaborated metrics are made available though Prometheus exporters and exposed by means of tools such as Graphana.

UC1 Scale-to-zero

The scale-to-zero demonstrates (i) turning off unused components, e.g., pods and/or containers that have no load/traffic to process, and (ii) automatically switching these components back on when load/traffic is (re)directed to the particular component.

Within the UC#1, scaling to zero is demonstrated within the context of 5/6G core network and UC#1 vApp whose components are distributed across cloud and edge sites. Trigger point for the scale to zero functionality activation is operational mode switch, i.e., switching from normal operational mode to disaster operational mode of UC#1. As elaborated in the use case description [11], switching to disaster mode ensures continuity of the operations during unlikely circumstances.

Scale-to-Zero Procedure (Manual Demonstration)

- Disaster scenario is activated.
- 5/6G core network container and vApp container running at cloud site are scaled to zero (the equivalent containers are activated at the edge site).
- Normal operational scenario is activated.
- 5/6G core network container and vApp container running at the edge site are scaled to zero (the equivalent containers are activated at the cloud site).

UC1 Migration

The migration functionality demonstrates ability of 6Green to support interoperable, fast and seamless geographical relocation of images, sessions and related workloads of slices/NFs and/or vApps.

Within the UC#1, the migration is demonstrated in conjunction with the Scale-to-Zero function. As described above, scaling to zero of 5/6G core network and vApp containers causes 5/6G core network and vApp containers migration/activation at another site, i.e., executing scaling to zero at cloud site causes the migration of components to the edge site and vice versa.

Migration Procedure (Manual Demonstration)

- 5/6G core network container and vApp container running at cloud site are scaled to zero.
- 5/6G core network container and vApp container are migrated and activated at the edge site.
- 5/6G core network container and vApp container running at the edge site are scaled to zero.
- 5/6G core network container and vApp container are migrated and activated at the cloud site

Instances are migrated without service interruptions.

UC1 Infrastructure power management

The infrastructure power management allows for updating configuration of infrastructure components, thus consolidating resources and contributing to energy consumption and carbon footprint reduction. While updating configurations, user experience/SLA should not be degraded.

Within the UC#1, infrastructure power management functionality is demonstrated for the RAN part of the 5/6G network which is deployed at the edge site. Updating the configuration is enabled by agentic AI system which relies on real time values of multiple parameters collected by the 6Green Observability (sub)system (see below).

Configuration update Procedure

- Agentic AI collects data available from observability system and SLA requirements.
- It evaluates impact on user experience/SLA (in case of UC1 this specifically relates to operational continuity of critical services) potentially caused by modifying MIMO level, output radio power, and radio channel bandwidth.
- One or multiple of the above RAN related parameters may be modified.
- According to the (temporal) requirements, configuration update may cause energy consumption reduction or increase.
- Updated configuration is enforced, critical operations are preserved.

UC1 Observability

The observability function is one of key support functions providing data required for other functionalities decision making, as well as for various (post-) analysis. Since UC#1 is predominantly focused on the edge site, part of the use case preparation work was implementation of tools enabling continuous power consumption monitoring data and integrate this (sub)system into overall 6Green Observability framework. At the edge site, qMON platform and its underlying tools, such as Scaphandre is deployed as part of 6Green Observability.

Observability Procedure

- Observability function is demonstrated by showing multiple graphs, logs and real time insights into the system, as well as source of evidence for evaluation of UC1 KPIs.
- Collected metrics are made available through Prometheus exporters.

Table 41: UC1 KPIs evaluation.

KPI label	KPI	Test Objective	Acceptance Criteria (based on standardized targets, if available)	Tests procedure/outputs
KPI_UC1_1	Network Availability	Critical operations rely on availability of the network at every time, i.e., also during the energy-constraint disaster scenario. The KPI will help design and optimise the network to meet the availability requirements.	Preserving the same availability rate (e.g., 99,99 %) during both the energy non-constraint and the energy constraint scenario.	**Test procedure:** Network Availability KPI is validated through packet loss measurements. ININ's qMON tools are utilized for the test, i.e., qMON agents are deployed in the cloud (CNIT testbed), at the edge (ININ testbed), and at the far-edge (UE). Packet loss is measured for packets traveling between (i) UE and cloud (normal operational mode scenario), and between (ii) UE and edge (critical operational mode scenario). **Output**:in both cases there were no packet loss detected – the availability is 100 %. See Section 4.1.1 for details.

KPI_UC1_2	Energy-constraint conditions detection time	To enable service prioritization during the occurrence of disaster scenario, the system needs to detect energy constraint conditions has occurred, i.e., available energy resources does not guarantee unlimited operation time which could be, for example, due to interrupted energy supply through electrical grid.	30 seconds	**Test procedure:** Energy-constraint conditions are detected by monitoring power consumption via custom Prometheus exporters. Fully integration into Kubernetes using Service Monitor resources assures time synchronization of all components involved in the "detection process". **Output**: actual detection time depends also on time resolution supported by the power consumption metering source (SNMP, Scaphandre, etc.), i.e., besides responsiveness of the metering tool, appropriate time interval between reporting two consecutive metering reports should be selected, In particular case, interval of 3 seconds have been identified sufficient to detect energy-constrained conditions in less than 10 seconds. See Section 4.1.1for examples (screen shots) of service monitors.
KPI_UC1_3	Time to disable non-critical services	To enable service prioritization during the occurrence of disaster scenario, it would be required in certain situation to disable non-critical services to keep critical services operating. As mentioned before, available energy resources may not guarantee unlimited operation time, while by disabling certain services, it might be possible to significantly prolong the operation time for critical services.	30 s (for all non-critical services are disabled)	**Test procedure:** Disabling service, i.e., scaling it down, is triggered by energy-constraint detection mechanism (elaborated within KPI_UC1_2 – see above). The system is able to disable the service almost instantly after the disable/scale down procedure is triggered (see below for relevant log snippet). **Output**: time for disabling services stays below the desired 30 s target.
KPI_UC1_4	Time to disable non-critical UEs connected	To enable service prioritization during the occurrence of disaster scenario, in certain situation it could be also required to disable non-critical UEs still connected to the network. This would lower the energy consumption of the system and thus add extra time for critical services to keep operating.	30 s (for all non-critical services are disabled)	**Test procedure:** Similarly, as in case of KPI_UC1_3, disabling the non-critical UEs connected to the 5/6G network is triggered by the energy-constraint detection mechanism (elaborated within KPI_UC1_2). The system disables non-critical UE almost instantly after the disable/scale down procedure is triggered (see below for relevant log snippet). **Output**: time for disabling services stays below the desired 30 s target.
KPI_ UC1_5	Network Latency	In a disaster recovery situation, the latency for a 5/6G mobile network may be impacted in several ways. Firstly, there may be a high volume of traffic on the network due to increased demand for communication and information sharing. This can lead to congestion and network slowdowns, resulting in higher latency. Secondly, damage to network infrastructure or power outages may cause disruptions to	20 ms (considering all non-critical services are disabled) This value strongly depends on the type of critical services under analysis.	**Test procedure:** Network latency measurement is based on measuring RTT (round-trip-time), utilizing ININ's qMON tools with ICMP protocol as enabler for RTT measurements. qMON agents have been deployed to UE for the purpose of KPI_UC1_5 evaluation. **Output**: As the goal for the KPI is evaluation of latency in disaster operational mode scenario, the agent pings 5/6G core network component deployed at the edge site. Latency (i.e., average RTT of ICMP ping packet) measured varies between 10 ms and 15 ms, thus acceptance criteria are met. See Section 4.1.1 for graphs depicting measurement results.

		network connectivity, leading to longer latency times. Thirdly, the prioritization of emergency communications may take precedence over other types of traffic, leading to longer latencies for non-emergency traffic. Finally, the availability and location of network resources such as base stations, antennas, and backhaul links may be affected, leading to increased latency as traffic is rerouted to alternative resources. Overall, it is important to plan for disaster recovery situations and implement strategies to minimize the impact on network latency.		
KPI_ UC1_6	Green energy utilization rate	To monitor the proportion of locally produced zero-emission energy that is consumed by the end-to-end service during normal operation with the aim to compare the scenarios when the use of this energy sources is maximized versus the standard scenario when these resources are only utilized to maximize performance.	The target of this KPI is to observe a significant increment on the proportion of locally produced zero-emission energy used by the service. The amount of this increase will be dependent on the site´s energy production capacity and the weather conditions.	**Test procedure:** required measurements: energy production of local zero-emission source, energy consumption of (local) facilities contributing to the service provision. 6Green monitoring and observability framework is utilized for energy consumption measurements, while energy production is monitored by the measurement/monitoring tool provided by the energy source equipment. **Output**: a photovoltaic system has been utilized as a local green energy source. During the observation period of 11 days (Apr 1 – Apr 11, 2026), 78 % of energy consumed by local infrastructure (ININ testbed) has been provided by local green source. See below for details.

4.1.1 UC1 KPIs Assessment

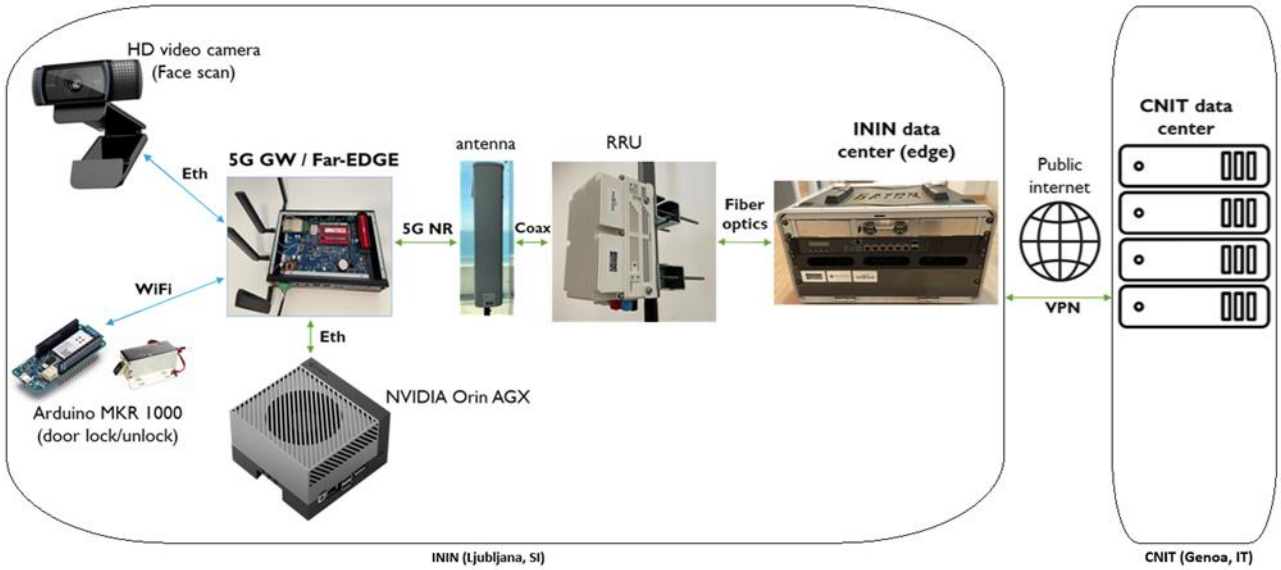


Figure 2: Physical setup used for UC1 demonstration and validation activities as described in [6].

KPI_UC1_1 - Network Availability

Network availability has been observed for both UC1 scenarios, i.e., normal operation scenario and disaster operation scenario:

- Normal operation scenario:
  - 100 % availability observed (Figure 3);
  - 5/6G core network functionalities have been provided by CNIT core/cloud, thus latency results observed are impacted by VPN established between the gNB (edge/ININ site) and CN (CNIT site).



Figure 3: Monitoring network availability in UC1 normal operation mode scenario.

- Disaster operation scenario:
  - 100 % availability observed (Figure 4);
  - 5/6G core network functionalities provided locally on ININ edge.



Figure 4: Monitoring network availability in UC1 disaster operation mode scenario.

Network availability measurements were performed using the qMON Network Monitoring suite (http://www.qmon.eu). Each scenario was monitored over a continuous 4-hour period. Availability was assessed via the ping measurement module, which issued ICMP ping probes at one-minute intervals to continuously verify end-to-end connectivity between the UE and internet. Throughout the measurement period, background traffic was generated by a live video stream (camera stream to vApp) at approximately 8 Mbps, ensuring availability was evaluated under representative load rather than on an idle link. See Table 42 for details.

Table 42: Summary of KPI_UC1_1 (Network Availability) evaluation tests.

Metric	Normal operation (VPN to CNIT core)	Disaster operation (local ININ edge)
Availability	100%	100%
Duration	4 h	4 h
Background traffic	Video stream (6Green demo) ~8 Mbps	Video stream (6Green demo) ~8 Mbps
Successful probes	1110	1160
RTT - Min	10.6 ms	10.7 ms
RTT - Mean	60.8 ms	24.9 ms
RTT - Max	393 ms	55.4 ms
RTT - Last	20.1 ms	15.1 ms
Target IP	1.1.1.1	1.1.1.1

KPI_UC1_2 - Energy-constraint conditions detection time

Figure 5 depicts screenshot of Prometheus exporters list. Exporters are crucial for detecting energy-constraint conditions. For the demonstration purposes, we focus on RRH exporter, i.e., `snmp-exporter-prometheus-snom-exporter-swallow`. In Figure 6 we further see configuration of SNMP service monitor for RRH energy consumption, showing 3 seconds interval set, which suffices to detect energy-constrained conditions in less than 10 seconds.

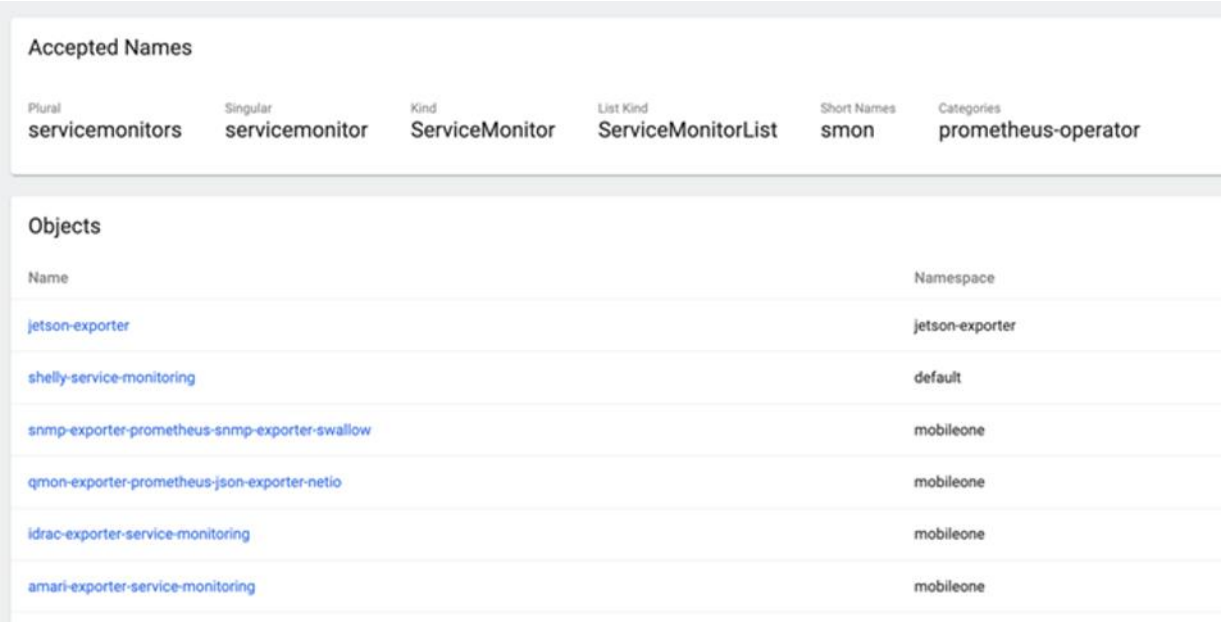


Figure 5: List of Prometheus exporters.



Figure 6: Example of SNMP service monitor configuration for RRH energy consumption.

KPI_UC1_3 -Time to disable non-critical services

Figure 7 show that critical service(s) are disabled – Pod is deleted – almost immediately after the disablement has been triggered. The latter follows energy constraint detection as elaborated in KPI_UC1_2.



Figure 7: Time to disable (non-critical) service.

KPI_UC1_4 – Time to disable non-critical UEs connected

Figure 8 shows that the system disables non-critical UE almost instantly after the disable/scale down procedure is triggered.



Figure 8: Time to disable (non-critical) UE.

KPI_UC1_5 – Network Latency

Figure 9 demonstrates that the latency is not constant, the figure thus shows how it varies during the ca. 21 hours time period observed, but for the whole observation it never goes above the 20 ms target.



Figure 9: Network latency variation over observed period.

The latency was measured using the qMON Network Monitoring suite over a continuous observation period of approximately 21 hours, with the RTT measurement module issuing ICMP ping probes at one-minute intervals towards the target IP 192.168.203.1 (5/6G Network Core in ININ's testbed). Throughout the measurement, background traffic was generated by a live video stream (6Green demo) at approximately 8 Mbps, so that latency was evaluated under representative load rather than on an idle link. See Table 43 for details.

Table 43: Summary of KPI_UC1_5 (Network latency) evaluation tests.

Metric	Value
Duration	~21 h
Background traffic	Video stream (6Green demo) ~8 Mbps
RTT - Min	10.4 ms
RTT - Mean	12.0 ms
RTT - Max	14.4 ms
RTT - Last	11.6 ms
Latency target	20 ms
Target IP	192.168.203.1 (5/6G Network Core)

KPI_UC1_6 - Green energy utilization rate

Carbon emissions could be reduced either by reducing energy consumption and/or by consuming energy with lower carbon emissions. Figure 10 and Figure 11 presents how energy from green/solar sources can be variable due to the weather conditions, i.e., cloudiness as main differentiator in this case. However, even with variable green source, total carbon emissions of the site are reduced.

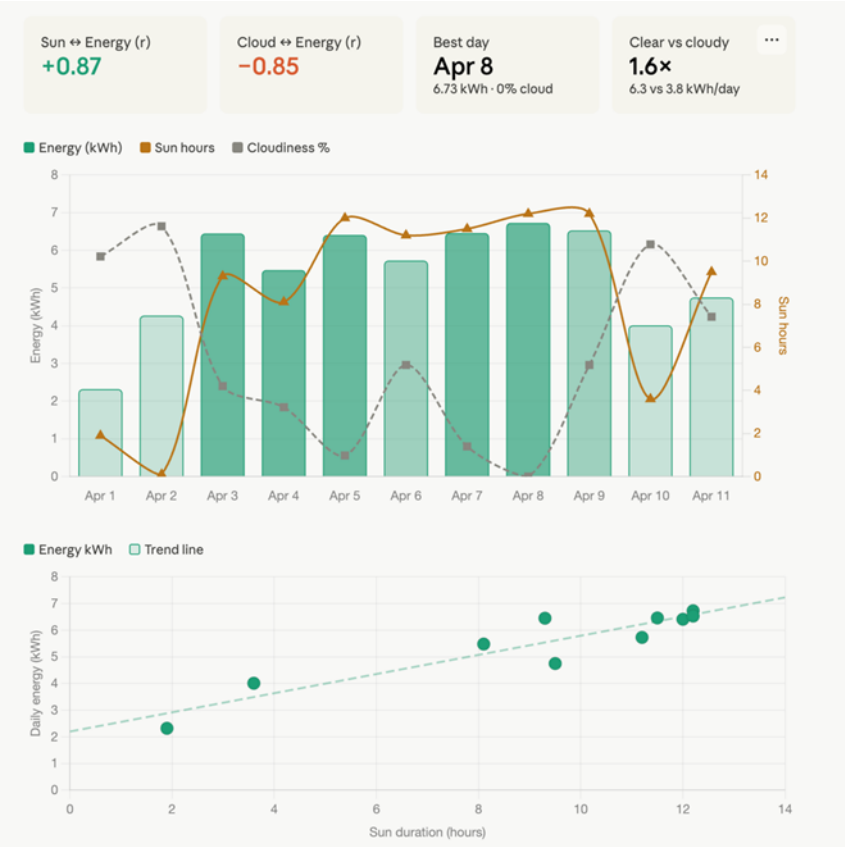


Figure 10: Daily production of green energy over 11 days of observation.

Full Daily Comparison

DATE	SUN (H)	CLOUD %	SOLAR KWH	SERVER KWH	RRH KWH	M1 TOTAL KWH	GAP KWH	GREEN %	COVERAGE
Apr 1	2	58%	2.30	4.76	2.25	7.01	-4.71	32.8%	32.8%
Apr 2	1	67%	4.30	4.78	2.31	7.09	-2.79	60.6%	60.6%
Apr 3	11	1%	6.40	4.60	2.29	6.89	-0.49	92.9%	92.9%
Apr 4	8	18%	5.40	4.42	2.30	6.72	-1.32	80.4%	80.4%
Apr 5	12	5%	6.40	4.53	2.30	6.83	-0.43	93.7%	93.7%
Apr 6	1.5	29%	5.70	4.49	2.30	6.79	-1.09	83.9%	83.9%
Apr 7	11	8%	6.40	4.43	2.29	6.72	-0.32	95.2%	95.2%
Apr 8	12	0%	6.73	4.56	2.29	6.85	-0.12	98.2%	98.2%
Apr 9	12	28%	6.50	4.64	2.33	6.97	-0.47	93.3%	93.3%
Apr 10	3.5	61%	4.00	4.49	2.30	6.79	-2.79	58.9%	58.9%
Apr 11	9	43%	4.70	4.52	2.30	6.82	-2.12	68.9%	68.9%
TOTAL / AVG	-	-	58.83	50.2	25.3	75.47	-16.64	78%	

Figure 11: Detailed daily comparison of environmental parameters, produced green energy and its share within total energy consumption of the ININ edge site.

4.2 Use case 2: Energy-Efficient Augmented Reality Remote Assistance System

Table 44: UC2 SBA actions validation.

SBA Action Name	Description of Ops at 6Green SBA level	Cause (trigger event)	Tests procedure/outputs
Scale to zero	Turning off unused components (e.g., pods, containers) and automatically switching them back on when required, e.g., traffic (re)directed to the component.	No load for a specific component, and/or no load in a specific area/time for a specific slice	**Test procedure:** demonstration of the Scale-to-Zero concept by shutting down the media server during periods of no active calls in the CNIT testbed, due to missing automation and reverse SSH proxy constraints. **Output:** Idle media server components were successfully deactivated, illustrating the intended reduction of unused compute resources and the expected energy and cost savings potential in a fully automated 6Green deployment.
Migration	Support interoperable, fast and seamless geographical relocation of images, sessions and related workloads of slices/NFs and/or vApps*.	Availability of (more) green energy available at specific site; SLA change; infrastructure/resource consolidation or interruptions	**Test procedure:** Demonstration of 6Green SBA migration by relocating media server workloads from non-green to renewable-energy nodes once sessions end and utilization is low, with automatic backend reconnection and service continuity validation. **Output:** Successful seamless migration enabling infrastructure consolidation and relocation to greener/lower-cost nodes, contributing to reduced CO_2 emissions and operational expenditure.
UPF offloading	Steering traffic to a HW/DPDK-based UPF instead of a VNF when deemed more efficient	Triggered by changes in traffic volume, SLA requirements, or network re-optimization needs (e.g., latency optimization for migration scenarios).	**Test procedure:** Demonstration of 6Green UPF offloading by dynamically switching traffic between a DPDK-based UPF (BESS) and a kernel-based UPF (Free5GC) using a PFCP Agent/GTP-U Router, including session transfer, NAT rule update, and controlled datapath activation/deactivation. **Output:** Successful seamless UPF switching with maintained service continuity, enabling reduced idle energy consumption and improved energy efficiency through traffic steering toward lower-power network functions.

UC2 Scale-to-zero

The Scale-to-Zero capability demonstrates the concept of shutting down unused media server components during periods of inactivity to reduce resource consumption. In an ideal 6Green deployment, components would be automatically deactivated when no traffic is present and reactivated on demand.

The showcase validates the underlying principle of reducing idle resource usage and highlights its expected contribution to improved energy efficiency and reduced operational costs in a fully automated deployment.

Scale-to-Zero Procedure (Manual Demonstration)

- No active calls are running on the media server.
- Media server is shut down after confirming zero load.
- Reactivation would normally occur automatically upon incoming traffic but is not supported in the current setup.
- Automatic reconnection of media server to Maintastic backend works as expected.

This demonstration illustrates the intended KPI-relevant effect of scale-to-zero: reducing idle compute usage, lowering energy consumption, and contributing to both carbon and cost reduction in fully supported deployments.

UC2 Migration

The UC2 AR Remote Assistance application demonstrates the Migration capability of the 6Green Service-Based Architecture (SBA) by dynamically relocating the media server workload to infrastructure with a more favourable energy profile.



Figure 12: UC2 Migration Procedure with CPU load and power monitoring.

Migration Procedure

- (1) No active call is running on the media server.
- (2) A two-participant video call is established on the red energy profile media server.
- (3) Two additional participants join the session, resulting in a four-participant call.
- (4) The platform detects the availability of a renewable-energy-powered media server.
- (5) The active call is terminated, and the media server CPU utilization drops below 1%, triggering the migration process.
- (6) A new two-participant video call is established on the migrated green energy profile media server.
- (7) Two additional participants join the session, resulting in a four-participant call.

The migration process is validated through monitoring of CPU utilization and power consumption on both the source and target infrastructure nodes, demonstrating the successful relocation of the workload to a renewable-energy-powered environment.

Beyond demonstrating the migration capability of the 6Green SBA, this scenario also contributes to the improvements measured in KPI_UC2_1 – Carbon Reduction and KPI_UC2_2 – Operational Expenditure Reduction.

The migration mechanism enables two further optimization strategies which will be explained in the following.

Infrastructure consolidation

By relocating the media server workload to another node, the original physical cluster could potentially be completely shut down when no longer required. This reduces idle resource consumption and lowers overall energy usage, directly contributing to reduced CO_2 emissions.

Energy- and cost-aware placement

The media server can be migrated to a node with lower resource requirements and a more favourable energy profile. For example, workloads can be executed on lower-performance infrastructure when demand is low, reducing the required compute resources and associated infrastructure costs while maintaining the required service quality.

As a result, migration acts as an enabling mechanism for both sustainability and cost optimization by reducing the number of active resources and increasing the utilization of renewable-energy-powered infrastructure.

UC2 UPF Offloading

The UC2 AR Remote Assistance application demonstrates the **UPF Offloading** capability of the 6Green Service-Based Architecture (SBA) by dynamically steering network traffic between different User Plane Function (UPF) implementations based on efficiency and energy-awareness criteria.

The showcase is based on two alternative data path implementations: a DPDK-based UPF (Berkeley Extensible Software Switch – BESS), which provides high performance but consumes power even in idle state, and a kernel-based UPF (Free5GC), which has near-zero power consumption when no traffic is present. The PFCP Agent / GTP-U Router acts as the switching component responsible for coordinating the transition between both data paths.

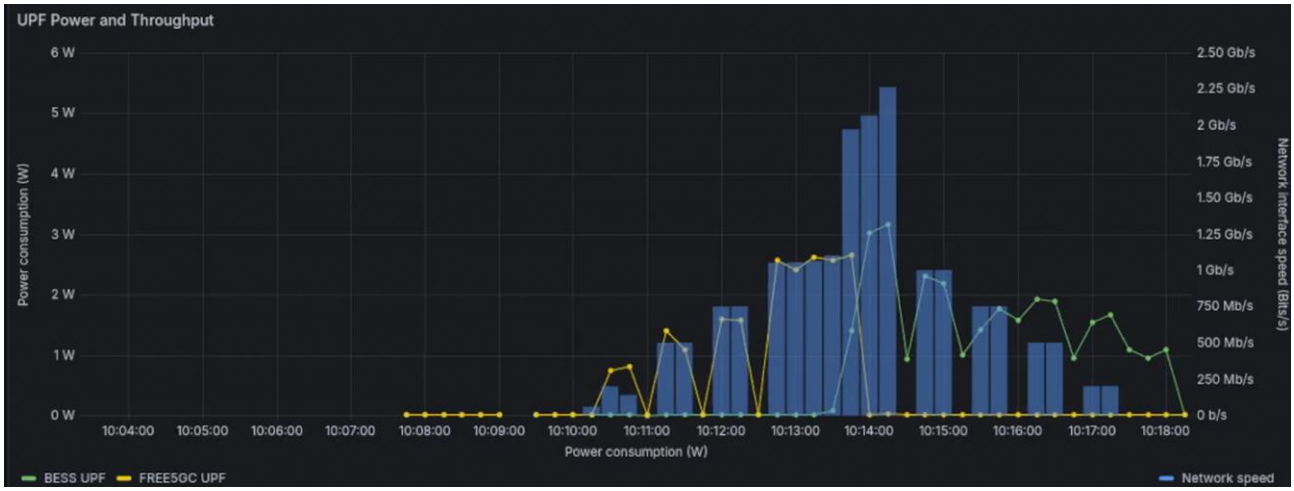


Figure 13: UC2 UPF Offloading Procedure with Power Consumption and Bitrate.

When a switch is triggered, the system powers on the target data path if required, transfers active session state, updates NAT routing rules to redirect traffic, and finally deactivates the previously used data path. This ensures seamless continuity of communication while optimizing energy consumption.

UPF Switching Procedure

- BESS (DPDK-based UPF) and Free5GC data paths are available (Free5GC always active, BESS consumes power even without traffic).
- The switching component (PFCP Agent / GTP-U Router) initiates a UPF switch via web UI.
- The new data path is powered on if required (Free5GC is always on).
- Active session state is copied to the new data path.
- NAT rules on both routers are updated to redirect traffic to the selected UPF.
- The old data path is powered off.

Contribution to 6Green Actions

This showcase demonstrates UPF offloading as an energy- and performance-aware re-optimization mechanism, enabling:

- Traffic steering based on efficiency and load conditions.
- Reduction of idle energy consumption in high-performance data paths.
- SLA-aware selection of UPFs depending on latency or performance requirements.
- Dynamic re-optimization of network paths during operation.

The result is a flexible network function placement strategy that balances performance and energy efficiency while maintaining service continuity for AR Remote Assistance communication.

The following section describes the UC2 specific KPI test and evaluation procedures, including the measured and observed results.

Table 45: UC2 KPIs evaluation.

KPI label	KPI	Test Objective	Acceptance Criteria (based on standardized targets, if available)	Tests procedure/outputs
KPI_ UC2_1	Carbon reduction, Zero-Carbon Service	Improve Maintastic network architecture to get closer to their carbon neutrality, e.g., by lowering resource use according to its usage, distributing server loads to reduce minimal resource use, handling energy-efficient vApp deployment locations and shutting down unused server under low loads.	Decreased by 30%	**Test procedure:** Comparison of 24/7 Azure baseline vs 6Green scale-to-zero deployment using vCPU-hour–based energy model (0.25 kWh/vCPUh, EF = 0.3 $kgCO_2/kWh$) with usage-driven active time estimation **Output:** ~51.7% reduction in vCPU-hours, energy consumption, and CO_2 emissions per workspace per day.
KPI_ UC2_2	Operational Expenditure for media server cluster	Improve Maintastic network architecture and make use of 6Green SBA to decrease operation expenditure to run media server cluster.	Decreased by 50%	**Test procedure:** Comparison of baseline 24/7 Azure deployment vs 6Green scale-to-zero architecture using the same workload-driven operating hours model as the carbon KPI. OPEX is derived from VM runtime (€/VM-hour) based on Standard D4ls v5 instances and active usage profiles. **Output:** ~51.7% reduction in operational expenditure due to reduced VM runtime, scale-to-zero behaviour, and workload-driven resource activation.
KPI_ UC2_3	Adaptive bandwidth depending on UE 5/6G signal quality for UC2_Adaptive bandwidth	Client application of Maintastic vApp dynamically adapts to the changing and demanding bandwidth changes by adjusting the video encoding parameters accordingly.	Boolean – Verify the accomplishment by accessing and measuring bandwidth and encoding parameters under real changing or simulated network conditions.	**Test procedure:** Evaluation of UC2 application performance under adaptive video bandwidth settings using low (640p, ~600 Kbit/s), medium (720p, ~1.5 Mbit/s), and high (1080p, ~3 Mbit/s) profiles across multiple receivers. **Output:** The application maintained stable and fluent video transmission across all bandwidth profiles.
KPI_ UC2_4	Network Latency	Remote Visual Assistance application demands specific network latency to provide the supported service with "proper" QoE	10 - 100 ms	**Test procedure:** Latency evaluation of UC2 AR Remote Assistance using two simulated UEs connected via a real 5/6G network on the CNIT infrastructure. 100 consecutive ping measurements were collected to assess network performance. **Output:** Minimum latency 4.5 ms, average 7.1 ms, maximum 24.3 ms, significantly below the 10 - 100 ms target range, confirming highly responsive AR communication performance.

4.2.1 UC2 KPIs Assessment

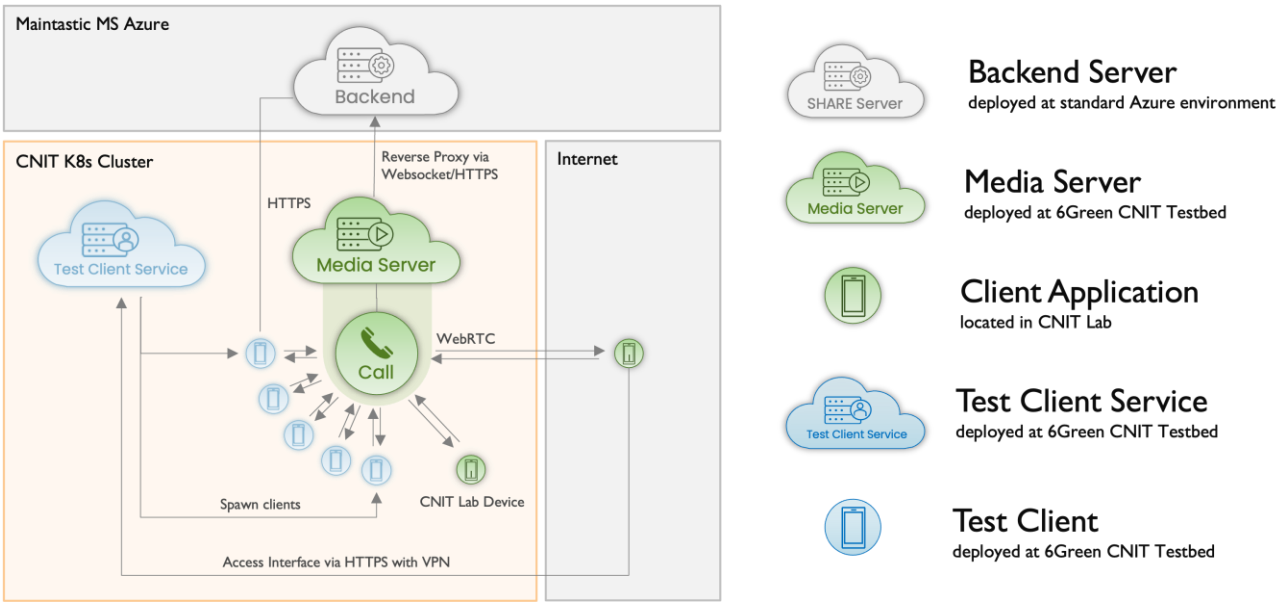


Figure 14: Physical setup used for UC2 demonstration and validation activities as described in [6].

KPI_UC2_1 – Carbon reduction, Zero-Carbon Service

This KPI chapter defines a comparative energy and carbon model for AR-based remote assistance services. The model compares the operational scenarios of a baseline Azure 24/7 deployment and an optimized 6Green adaptive infrastructure.

Reference Usage Model

Figure 15 illustrates a representative usage profile aggregated across 10 workspaces, showing the number of active calls and active participants over time.

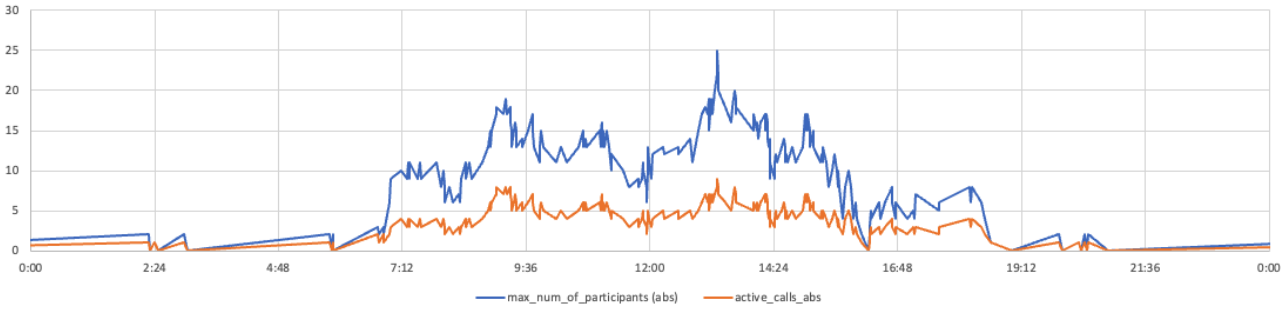


Figure 15: UC2 Reference Usage Pattern.

The data indicates a clear usage pattern. During peak hours (approximately 07:00–18:00), the system reaches a maximum of 8 concurrent calls and approximately 25 active participants across all workspaces. This corresponds to an average of one active call per workspace, with approximately 2–3 participants per call.

Outside of business hours, system utilization decreases significantly. Between approximately 18:00 - 02:00 and 02:00 - 06:00, only sporadic call activity is observed. No call activity is recorded between approximately 20:00 - 02:00, indicating a complete idle phase during this period.

Baseline Azure 24/7 Deployment (Business as Usual)

The baseline scenario assumes a static 24/7 deployment of the AR Remote Assistance media server infrastructure on Microsoft Azure. Each workspace operates one dedicated media server instance (4 vCPUs, Standard D4ls v5), which remains continuously allocated regardless of actual call activity.

This results in a constant resource consumption of 96 vCPU-hours per workspace per day. Using a conversion factor of 0.25 kWh per vCPU-hour and an emission factor of 0.3 kg CO_2/kWh, the baseline results in:

- Energy consumption: 24 kWh/day
- CO_2 emissions: 7.2 kg CO_2/day

6Green Scale-to-zero Deployment

The 6Green scenario introduces a dynamic, demand-driven deployment model using scale-to-zero functionality. Media server instances are only active during call sessions and are deallocated during idle periods.

Based on observed usage patterns:

- Active period: 07:00–18:00 (11 hours/day of full operation)
- Night period: 20:00–02:00 with rare usage, approximated as 1 call every 10 days

This results in an expected active time of 11.6 hours per day.

Consequently, the system consumes:

- 46.4 vCPU-hours per workspace per day
- 11.6 kWh per day
- 3.48 kg CO_2 per day

Evaluation Result

Comparing both scenarios under equivalent functional workload conditions shows a consistent reduction across all metrics (Table 46):

- vCPU-hour reduction: **51.7%**
- Energy reduction: **51.7%**
- CO_2 reduction: **51.7%**

Table 46: KPI_UC2_1 evaluation results summary.

KPI label	KPI	Target value	Is-value
KPI_ UC2_1	Carbon reduction, Zero-Carbon Service	30%	51.7%

The 6Green deployment achieves approximately 51.7% reduction in carbon emissions per workspace per day compared to the baseline Azure 24/7 deployment. This reduction is primarily driven by scale-to-zero operation during inactive periods and reduced compute utilization aligned with actual user demand, demonstrating the effectiveness of energy-aware infrastructure management in AR-based remote assistance systems.

Further possible carbon reductions are described in the sections of UC2 Migration and UC2 UPF Offloading but were not additionally considered for this KPI.

KPI_UC2_2 – Operational Expenditure for media server cluster

The evaluation of KPI_UC2_2 is based on the same workload and operating profile as defined in the carbon reduction analysis. Operational expenditure is directly derived from the media server runtime, where costs are incurred only during active VM operation periods.

Baseline Azure Scenario (Business as Usual)

The baseline scenario assumes a static 24/7 deployment of the AR Remote Assistance media server infrastructure on Microsoft Azure. Each workspace operates one continuously running Standard D4ls v5 virtual machine, independent of actual call activity.

This results in constant operational costs based on full-time VM availability (24 hours per day), regardless of workload demand or utilization.

6Green Scenario

The 6Green scenario applies the same operating profile as the carbon reduction model, using scale-to-zero mechanisms.

VMs are only active during call sessions (07:00–18:00 and rare night calls), while being fully deallocated during idle periods. As operational costs are directly linked to VM runtime, this leads to proportional cost savings compared emission savings to the baseline always-on deployment.

Operational expenditure is calculated using a time-based cost model:

- VM cost is proportional to active runtime (€/VM-hour)
- Total cost is derived from cumulative VM uptime per workspace
- The same active time assumptions used in the CO_2 model are applied to ensure consistency between energy and cost evaluation

Applying the reduced runtime profile of the 6Green architecture leads to a proportional reduction in operational expenditure. Overall, the 6Green deployment achieves an estimated ~51.7% reduction in operational expenditure, consistent with the reduction in compute runtime (Table 47).

Table 47: KPI_UC2_2 evaluation results summary.

KPI label	KPI	Target value	Is-value
KPI_ UC2_2	Operational Expenditure for media server cluster	Decreased by 50%	51.7%

Operational expenditure reduction directly follows the workload-driven runtime model defined in the carbon KPI. By aligning VM provisioning with actual usage patterns, the 6Green architecture eliminates idle resource costs and significantly improves cost efficiency without affecting service availability during active periods.

KPI_UC2_3 – Adaptive bandwidth

The evaluation of KPI_UC2_3 assesses the ability of the UC2 application to operate under different video quality and bandwidth settings. Tests were performed using low (640p, ~600 Kbit/s), medium (720p, ~1.5 Mbit/s), and high (1080p, ~3 Mbit/s) video profiles with multiple receivers.

CPU utilization, estimated energy consumption, and video stability were monitored during the tests. No significant impact on CPU load was observed across the different quality settings, while all profiles provided stable and fluent video transmission (Table 48).

Table 48: KPI_UC2_3 evaluation results summary.

KPI label	KPI	Target value	Is-value
KPI_ UC2_3	Adaptive bandwidth	Verify the accomplishment	True

The UC2 application successfully adapted to all tested bandwidth profiles while maintaining stable video quality and user experience.

KPI_UC2_4 – Network Latency

The evaluation of KPI_UC2_4 assesses the network latency of the UC2 AR Remote Assistance application during operation over a 5/6G connection. Low latency is essential to ensure responsive audio/video communication and effective remote guidance.

The KPI is evaluated using two simulated UEs connected via a real 5/6G network and communicating through the same CNIT infrastructure cluster. Latency is measured using ping requests, with 100 consecutive measurements collected to determine a minimum latency of 4.5 ms, an average of 7.1 ms and maximum 24.3 ms (Table 49).

Table 49: KPI_UC2_4 evaluation results summary.

KPI label	KPI	Target value	Is-value
KPI_ UC2_4	Network latency	10-100 ms	7.1 ms

With an average latency of 7.1 ms, remaining well below the target threshold of 100 ms and confirming that the UC2 application provides highly responsive communication suitable for AR-based remote assistance scenarios.

4.3 Use case 3: Zero-Carbon Clientless Virtual Enterprise Desktop as-a-Service

Table 50: UC3 SBA actions validation.

SBA Action Name	Description of operation at 6Green SBA level	Cause (trigger event)	UC
Scale to zero	turning off unused pods and automatically switching them back on when needed (e.g., traffic directed to that pod is arriving)	No load in a specific area/time for a specific slice	**Test procedure:** in the UC3 DaaS scenario, the scale-to-zero capability is triggered based on infrastructure and network-level metrics collected during virtual desktop usage over the 5/6G network slice. When no activity is detected on the Guacamole GUI or on connected virtual desktop sessions,

			unused DaaS components are automatically scaled down to zero to reduce resource consumption and energy usage. Components are dynamically reactivated when a new user session or activity is initiated. **Output:** the system is able to automatically scale down unused DaaS components and scale them back up as resuming traffic triggers the scale to one.
Migration	support interoperable, fast and seamless geographical relocation of images, sessions and related workloads of slices/NFs and/or vApps*. (dynamic flow steering: move apps from edge to core depending volumes, ee, …)	Availability of green energy on a site (Infrastructure) Resource consolidation SLA change	**Test procedure:** in the UC3 DaaS scenario, migration dynamically relocates virtual desktop workloads across the cloud–edge infrastructure based on application-level metrics, SLA requirements, and energy conditions. The 6Green platform selects the most appropriate execution nodes to ensure energy-efficient operation while maintaining service continuity and user QoE. **Output:** migration occurs seamlessly and does not cause any interruptions.

UC3 Scale-to-zero

UC3 Scale-to-Zero The scale-to-zero functionality demonstrates the capability of the 6Green SBA to dynamically deactivate unused DaaS application components and automatically reactivate them when user demand resumes.

Within UC3, end users (UEs) access the DaaS service through the 5/6G network via the Guacamole web interface, which provides access to their associated virtual desktop sessions running on the Kubernetes-based cloud–edge infrastructure. During operation, the 6Green monitoring framework continuously evaluates infrastructure-level and network-level metrics associated with the service. When no activity is detected, unused DaaS components are automatically scaled down to zero in order to reduce resource utilization and energy consumption. Upon the arrival of new user traffic, the required components are automatically re-instantiated and the service is restored.

Scale-to-Zero Procedure (Manual Demonstration)

- Users establish virtual desktop sessions through the 5/6G network using the Guacamole web interface.
- User activity is terminated and no traffic is generated towards the DaaS service.
- The 6Green SBA detects inactivity and automatically scales down unused DaaS components.
- New user traffic is generated, and the SBA automatically restores the required components, re-establishing access to the virtual desktop service.

UC3 Migration

The migration functionality demonstrates the capability of the UC3 DaaS application to relocate virtual desktop workloads across the cloud–edge infrastructure according to application-level policies, SLA requirements, and service-specific metrics. These policies are defined by the user and communicated to the Vertical Application Orchestrator (VAO), while application-level metrics can be provided by components such as the Rating Operator. The objective is to ensure that workloads are executed on the most suitable infrastructure resources while maintaining service continuity and the required Quality of Experience (QoE).

Within the UC3 DaaS scenario, the 6Green platform continuously monitors infrastructure and energy-related conditions and provides energy-awareness information to the orchestration framework. By combining application-level requirements with infrastructure-level energy metrics, virtual desktop workloads can be relocated to more energy-efficient execution environments while preserving application state and user experience.

Migration Procedure (Manual Demonstration)

- The DaaS application is initially deployed on the default execution infrastructure, and virtual desktop sessions are established for active users.
- Users access their virtual desktops through the DaaS platform and perform their normal workloads.
- Application-level policies and SLA requirements are configured in the VAO. For example, a policy may specify that if the power consumption of the hosting infrastructure exceeds a predefined threshold, the workload should be relocated to a more energy-efficient ("green") execution environment.
- During operation, application-level metrics are collected regarding the DaaS application (e.g., through the Rating Operator), while the 6Green platform continuously monitors infrastructure and energy-related conditions.
- When the policy conditions are met (e.g., the measured energy consumption exceeds the configured threshold), the VAO triggers a migration request.
- Based on the infrastructure and energy-awareness information provided by the 6Green platform, a more suitable execution environment is selected.
- The virtual desktop workload is relocated to the selected cloud or edge site.
- User access to the DaaS service continues through the migrated workload while preserving application state and maintaining the required QoE.

Table 51: UC3 KPIs evaluation.

KPI label	KPI	Description and Test Objective	Acceptance Criteria (based on standardized targets, if available)	Tests procedure/outputs
KPI_ UC3_1	Carbon reduction, Zero-Carbon Service	Less processing power on the client side leads to lower carbon emissions from the laptop/desktop usage.	Decreased by 85%	**Test procedure:** Power consumption measured for local vs DaaS execution (LibreOffice, FFmpeg) using powerstat. CO_2e derived over an 8-hour operation using a standard emission factor (0.30 kg CO_2e/kWh). QoE used to define SLA-compliant operating region. **Output:** results show consistent CO_2e savings at all SLA-compliant points, ranging from approximately 24% to 39% depending on workload and concurrency level. See Section 4.3.1 and **Error! Reference source not found.** for full methodology and results.
KPI_ UC3_2	Maintenance costs	Virtual Desktop Infrastructure can reduce Capital Expenses /Operational Expanses more than 85% compared to standard hardware-based solutions	Decrease the maintenance cost by 40%	**Test procedure:** operational costs were evaluated per user/month comparing traditional laptop-based deployment with the UC3 DaaS model. The analysis includes device amortization, IT support, infrastructure, and energy costs.

				Output: results show cost reductions ranging from approximately 44% to 53% depending on workload, exceeding the KPI target of ≥40%. See . See Section 4.3.1 and Deliverable D5.4 [6] for full methodology and results.
KPI_ UC3_3	Performance - adaptive Network Bandwidth/Latency	Virtual Desktop application demands specific network latency to provide the supported service with "proper" QoE	15-100 ms	**Test procedure:** end-to-end latency evaluation follows the methodology described in Deliverable D5.4. **Output:** the mean response time over the 5/6G network was measured at 17.1 ms, remaining within the acceptable operational range of 15–100 ms required to maintain satisfactory Quality of Experience (QoE). Application-level observations during virtual desktop interaction, including typing responsiveness, mouse movement, and screen updates, indicated smooth and responsive operation with no perceptible delay under the evaluated conditions. Therefore, KPI_UC3_3 is considered successfully validated.
KPI_ UC3_4	Mobility support	Virtual Desktop applications can support migration of the user state during handovers of his/her end device	Boolean -- Verify the accomplishment by accessing and measuring a moving end-device to the central VDI deployment in CNIT	**Test procedure:** boolean validation: an active DaaS virtual desktop session (Ubuntu RDP via Guacamole browser interface) was maintained while the end-user device moved across network access points connected to the CNIT 5/6G infrastructure. **Output:** session continuity and user state preservation confirmed without interruption. Validated: True.
KPI_ UC3_5	Zero-client	Virtual Desktop application will only require a smartphone with 5/6G connectivity and a Web Browser	Boolean -- Laptop/mobiles will access the DaaS service through HTML 5 supporting browsers	**Test procedure:** boolean validation: the DaaS service accessed exclusively through standard HTML5-compatible web browsers (Guacamole) on lightweight devices including smartphones and Chromebook-class laptops via 5/6G, with no client software, local installation, or specialised hardware required. **Output:** all computation executed centrally on the CNIT Kubernetes infrastructure. Validated: True.

4.3.1 UC3 KPIs Assessment

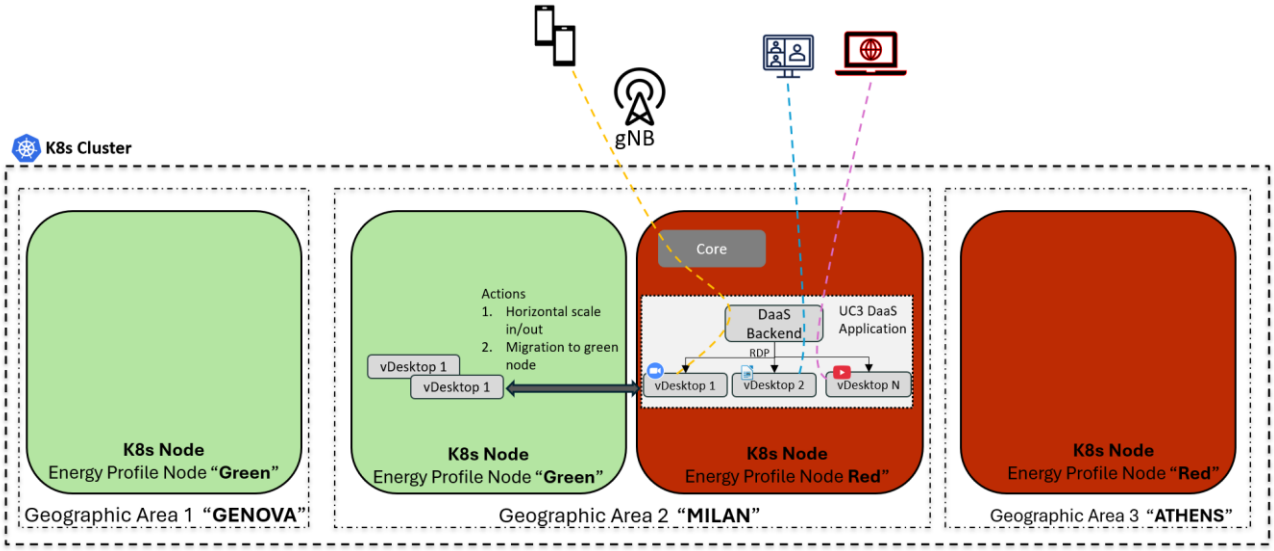


Figure 16: Physical setup used for UC3 demonstration and validation activities as described in [6].

KPI_UC3_1 – Carbon reduction, Zero-Carbon Service

Power consumption measurements, workload execution, QoE evaluation, and CO_2e estimation follow the methodology described in Deliverable D5.4 [6]. For light office workload (e.g., LibreOffice usage), the local device baseline is 14.5 W/user, while for compute-intensive workload (e.g., FFmpeg usage), it is 30 W/user. The DaaS thin-client (Chromebook-class) accounts for 8 W/user.

Server-side scaling S(N) was derived from measurements up to 20 concurrent users. For LibreOffice, server-side power increase exhibited near-linear behaviour, and a linear approximation was used to extrapolate results beyond N = 20. For FFmpeg, a saturating exponential model was adopted to capture resource contention at higher concurrency levels. Full measurement methodology and scaling models are described in Deliverable D5.4 [6].

CO_2e estimation was performed by deriving energy consumption from measured power values over a representative 8-hour operating day and converting it to CO_2-equivalent emissions using a standard grid emission factor of 0.30 kg CO_2e/kWh.

Quality of Experience (QoE) was measured as typing throughput ratio relative to single-user performance (LibreOffice) and frames-per-second ratio (FFmpeg). Results are reported at the maximum concurrency level satisfying each SLA constraint. Full QoE methodology is provided in Deliverable D5.4 [6].

For the LibreOffice workload (Table 52), at the SLA-80% operating point (QoE ≥ 0.80), N = 28 users with savings of +33.4%. At SLA-70% (QoE ≥ 0.70), N = 44 users with savings of +34.8%. The maximum savings asymptotically approaches ~36.2% at N = 100, although this lies below acceptable QoE levels. For the FFmpeg workload (Table 53), at SLA-80% (QoE ≥ 0.80), N = 13 users with savings of +24.1%. At SLA-70% (QoE ≥ 0.70), N = 20 users with savings of +32.5%, and at SLA-60% (QoE ≥ 0.60), N = 27 users with savings of +39.1%.

At low concurrency levels (N < 5), DaaS total system energy consumption exceeds local execution due to fixed server overhead. However, carbon savings become positive and increase consistently beyond N ≈ 5, confirming the efficiency of centralised DaaS execution at realistic enterprise scales.

Overall, total system CO_2e savings are positive and substantial at all SLA-compliant operating points (e.g., concurrency levels that maintain acceptable QoE while achieving efficient infrastructure utilization and carbon reduction), ranging from approximately 24% to 39% depending on workload type and concurrency level. These results demonstrate effective carbon reduction under realistic operating conditions.

Table 52: LibreOffice workload: CO_2e comparison (DaaS vs local laptop) across concurrency levels. Local device baseline: 14.5 W/user.

Users [#]	S(N) [W]	DaaS [W]	Local [W]	QoE [%]	DaaS CO_2e [kg]	Local CO_2e [kg]	Savings [%]
1	16.58	24.58	14.5	100.0	0.059	0.035	−69.0
5	20.98	60.98	72.5	95.0	0.146	0.174	+15.9
10	26.48	106.48	145.0	93.1	0.256	0.348	+26.6
15	31.98	151.98	217.5	89.4	0.365	0.522	+30.1
20	37.48	197.48	290.0	85.8	0.474	0.696	+31.9
28*	45.08	269.08	406.0	80.2	0.646	0.974	+33.4
30	48.48	288.48	435.0	78.3	0.692	1.044	+33.7
40	59.48	379.48	580.0	72.0	0.911	1.392	+34.6
44*	63.08	415.08	638.0	70.4	0.996	1.531	+34.8
50	70.48	470.48	725.0	66.2	1.129	1.74	+35.1
70	92.48	652.48	1015.0	56.1	1.566	2.436	+35.7
100	125.48	925.48	1450.0	44.1	2.221	3.48	+36.2

Note: SLA operating points are indicated with (*). For LibreOffice: N=28 (SLA-80%, QoE=0.802, savings +33.4%) and N=44 (SLA-70%, QoE=0.704, savings +34.8%). DaaS power is calculated as S(N) + 8×N (thin-client consumption: 8 W/user), while local execution is 14.5×N. CO_2e estimation assumes an 8-hour operating day and an emission factor of 0.30 kg CO_2e/kWh. S(N) is extrapolated beyond N=20 using a linear approximation.

Table 53: FFmpeg workload: CO_2e comparison (DaaS vs local laptop) across concurrency levels. Local device baseline: 30 W/user.

Users [#]	S(N) [W]	DaaS [W]	Local [W]	QoE [%]	DaaS CO_2e [kg]	Local CO_2e [kg]	Savings [%]
1	32.0	40.0	30.0	100.0	0.096	0.072	−33.3
2	50.0	66.0	60.0	98.1	0.158	0.144	−10.0
5	108.0	148.0	150.0	92.7	0.355	0.36	+1.3
10	160.0	240.0	300.0	84.3	0.576	0.72	+20.0
13*	192.0	296.0	390.0	80.0	0.711	0.936	+24.1
15	210.0	330.0	450.0	76.7	0.792	1.08	+26.7
20*	245.0	405.0	600.0	70.0	0.972	1.44	+32.5
27*	276.9	492.9	810.0	61.0	1.183	1.944	+39.1
30	287.4	527.4	900.0	57.6	1.266	2.16	+41.4
40	310.9	630.9	1200.0	47.7	1.514	2.88	+47.4
50	323.6	723.6	1500.0	39.4	1.737	3.6	+51.8
70	334.3	894.3	2100.0	27.0	2.146	5.04	+57.4
100	338.0	1138.0	3000.0	15.2	2.731	7.2	+62.1

Note: SLA operating points are indicated with (*). For FFmpeg: N=13 (SLA-80%, QoE=0.800, savings +24.1%), N=20 (SLA-70%, QoE=0.700, savings +32.5%) and N=27 (SLA-60%, QoE=0.610, savings +39.1%). DaaS power is calculated as S(N) + 8×N (thin-client consumption: 8 W/user), while local execution is 30×N. CO_2e estimation assumes an 8-hour operating day and an emission factor of 0.30 kg CO_2e/kWh. S(N) is extrapolated beyond N=20 using saturating exponential model.

KPI_UC3_2 – Maintenance costs

Operational cost evaluation follows the methodology and cost model described in Deliverable D5.4.

Local model

The local deployment cost is defined as:

$C_{local} = A_L + S_L + E_L$

where:

- A_L (device amortization): €1,500 enterprise laptop over 60 months = €25.00/user/month.
- S_L (IT support): ≈1 ticket/month × 30 min resolution at €30/h = €15.00/user/month.
- E_L (energy): 14.5 W avg (LibreOffice) and 30 W avg (FFmpeg) × 176 h/month at €0.20/kWh = €0.51 and €1.06/user/month.

Total cost:

- C_{local} = €40.51/user/month (LibreOffice)
- C_{local} = €41.06/user/month (FFmpeg)

DaaS model

The DaaS deployment cost is defined as:

$C_{daas} = A_{CB} + S_D + A_{SRV} + M_{SRV} + H_{SRV} + E_D$

where:

- A_{CB} (thin-client amortization): €250 Chromebook-class device over 60 months = €4.20/user/month.
- S_D (centralized support): 55% helpdesk reduction vs local = 0.45 × €15.00 (S_L) = €6.75/user/month.
- A_{SRV} (server amortization): €5,500 server / 60 months = €91.67/server/month
  This cost is shared among users according to the operating points derived from the combined QoE and energy-efficiency analysis presented in KPI_UC3_1. For the office (LibreOffice) workload, the selected operating point corresponds to approximately 28 users per server (QoE ≈ 0.8), while for the compute-intensive (FFmpeg) workload the selected operating point corresponds to approximately 20 users per server (QoE ≈ 0.7). Based on these operating points, the amortized server cost per user becomes:
  €3.27/user/month (LibreOffice)
  €4.58/user/month (FFmpeg)
- M_{SRV} (server maintenance): €500/year contract = €41.66/server/month
  The maintenance cost is distributed across users following the same server-sharing model derived from the QoE-aware operating points, 28 users for LibreOffice and 20 for FFmpeg. This results in:
  €1.49/user/month (LibreOffice)
  €2.08/user/month (FFmpeg)

- H_{SRV} (system administration): 3 h/server/month × €30/h = €90/server/month
  In the centralized DaaS model, administration effort is shifted from per-device maintenance to shared infrastructure management. The administration cost is therefore distributed across users according to the same server-sharing model derived from the QoE-aware operating points (28 users for LibreOffice and 20 for FFmpeg). This results in:
  €3.21/user/month (LibreOffice)
  €4.50/user/month (FFmpeg)
- E_D (DaaS energy):
  The DaaS energy cost combines server-side infrastructure consumption and thin-client energy usage under the evaluated workloads. Based on the measurements obtained in KPI_UC3_1, the total DaaS power consumption corresponds to approximately 270.28 W for the LibreOffice workload and 405 W for the FFmpeg workload. These values are distributed across users according to the selected QoE-aware operating points derived in KPI_UC3_1, corresponding to approximately 28 users for the LibreOffice workload and 20 users for the FFmpeg workload. This results in an average energy consumption of approximately 9.65 W per user for LibreOffice and 20.25 W per user for FFmpeg. Assuming 176 operating hours (opHours) per month and an electricity tariff of €0.20/kWh, the monthly energy cost is calculated as E_D = (Power × opHours / 1000) × 0.20, resulting in:
  €0.34/user/month (LibreOffice)
  €0.71/user/month (FFmpeg)
- **Total cost:**
  C_{daas} = €19.23/user/month (LibreOffice)
  C_{daas} = €22.79/user/month (FFmpeg)

Results and validation

The DaaS model achieves:

- 52.5% cost reduction for LibreOffice
- 44.5% cost reduction for FFmpeg

Both results exceed the KPI_UC3_2 target of ≥40%.

Therefore, the KPI is validated.

5 Conclusions and Lessons Learned

This document presents the validation and performance evaluation results of the technologies designed, implemented, and deployed within the 6Green project, including SBA components, non-SBA components, integrated frameworks and solutions, and the three project use cases. The components, together with the frameworks and solutions integrating these components, were validated against their corresponding functional requirements through verification of the services and functionalities they provide. In parallel, the use cases underwent performance evaluation, including demonstration of key 6Green functionalities (edge agility, green elasticity, and energy-aware backpressure) and assessment of the KPIs defined for each use case.

For each individual 6Green SBA component, as well as non-SBA component, the document first introduces the related functional requirements, followed by the corresponding test objectives and acceptance criteria. This is complemented by a detailed description of the validation procedures, including execution steps and instructions for assessing the obtained verdicts. The same methodology was then applied to the validation of the services exposed by the components.

An equivalent validation and evaluation approach was also applied to the integrated frameworks and solutions. In the case of use case performance evaluation, the methodology focused on KPI assessment, while the demonstration results highlighted the execution and effectiveness of SBA-related actions and functionalities within realistic business-oriented scenarios.

The obtained validation and performance evaluation results provide tangible proof on how the developed components, integrated frameworks and solutions, and the considered use cases successfully support the whole 6Green technological viability, meeting the requirements and KPIs defined within the project. The reported activities on experimental evaluation highlighted the feasibility and effectiveness of the key 6Green paradigms, namely edge agility, green elasticity, and energy-aware backpressure, which were successfully demonstrated and exploited, at least partially, across the implemented use cases.

In this process, it is worth mentioning that the Consortium has also assessed in a quantifiable manner that the cooperation among stakeholders can lead to significant energy and economic savings: for instance, infrastructure consolidation resulting in reduced CO_2 emissions directly translates into tangible economic savings, confirming the mutual benefits of the win-win approach promoted by 6Green.

Furthermore, activities carried out during the lifetime of 6Green have also allowed identifying some current technological limitations that still prevent the full achievement of the most ambitious sustainability goals addressed by the project. For instance, dematerialization solutions are still suffering from a lack of maturity at both hardware and software level: the presence of hardware accelerators and GPUs that do not natively allow any power management capabilities, and software is usually not developed with energy efficiency as a top priority; nevertheless, significant carbon reductions have been achieved (up to 39% maintaining an acceptable level of QoE). Moreover, while live migrations are not supported between independent Kubernetes clusters, 6Green has been able to bypass this issue by designing, introducing, and prototyping the scale-to-zero paradigm, which has proved the ability to support vital applications such as critical infrastructure systems and maintain the ability to save energy while still providing some level of redundancy.

Overall, the results presented in this deliverable confirm the maturity, applicability, and operational potential of the 6Green technologies and provide a solid basis for further development, integration, and large-scale deployment of sustainable and intelligent beyond-5/6G network solutions. 6Green has made some initial but significant strides in transforming the flexibility/programmability of technologies in complex multi-tenant and -domain architectures not towards additional energy consumption, but instead towards opportunities for sustainable businesses.

6 References

[1] 6Green, “D3.3: The 6Green 5/6G Service Based Architecture,” 2026.

[2] 6Green, “D3.4: Final Release of Network Function Prototypes of the 6Green 5/6G Service Based Architecture,” 2026.

[3] 6Green, “D5.1: First Release of the 6Green Pilot,” 2024.

[4] 6Green, “D5.3: Final Release of the Vertical Applications,” 2026.

[5] 6Green, “D5.2: Final Release of the 6Green Pilot,” 2025.

[6] 6Green, “D5.4: Final Integration of the 6Green Demonstrator,” 2026.

[7] 6Green, “D3.1: Work-in-Progress: Enabling Green Interoperability in the 5/6G SBA,” 2023.

[8] 3GPP, “TS 23.502 - Procedures for the 5G System (5GS), Release 19,” 2025.

[9] 3GPP, “TS 28.312 - Management and orchestration; Intent driven management services for mobile networks (Release 19),” 2025.

[10] 6Green, “D2.3: The 6Green Enabling Technologies,” 2025.

[11] 6Green, “D2.1: Use and Business Cases, Design and Technology Requirements, and Architecture Specification,” 2023.