Deliverable D4.1

Initial report on test-plan creation and methodology, and development of test orchestration framework

Editor:	Andrea F. Cattoni (KEYS)
Deliverable nature:	Report (R)
Dissemination level: (Confidentiality)	Public (PU)
Contractual delivery date:	30th June 2019
Actual delivery date:	12th July 2019
Suggested readers:	ICT-19 projects, industry verticals seeking to conduct Testing on 5G systems
Version:	1.0
Total number of pages:	62 (without annexes)
Keywords:	5G, KPI, Use Cases, Test Plan, Test Framework,

Abstract

WP4 is focused on ensuring that the 5G-VINNI facility delivered by the project is verified with a uniform testing procedure, allowing the Mobile Network Operators that are part of the consortium to have confidence in the system. The focus of WP4 is on testing methodology, test execution and E2E network validation to verify that KPIs across different verticals are met. The WP is intended to provide all the needed testing and validation tools and support across the project, as well as providing the support needed by vertical projects hosted by the 5G-VINNI facility. This document is structured as a report on the different activities of the Tasks T4.1 and T4.2 carried out during the first Period, while the Annexes represent the initial structure of a live document that will become, at the end of the project, with D4.4 the official Test Specification handbook of the 5G-VINNI project.

[End of abstract]

Disclaimer

This document contains material, which is the copyright of certain 5G-VINNI consortium parties, and may not be reproduced or copied without permission.

In case of Public (PU): All 5G-VINNI consortium parties have agreed to full publication of this document.

In case of Restricted to Programme (PP): All 5G-VINNI consortium parties have agreed to make this document available on request to other framework programme participants.

In case of Restricted to Group (RE): All 5G-VINNI consortium parties have agreed to full publication of this document. However this document is written for being used by <organisation / other project / company etc.> as <a contribution to standardisation / material for consideration in product development etc.>.

In case of Consortium confidential (CO): The information contained in this document is the proprietary confidential information of the 5G-VINNI consortium and may not be disclosed except in accordance with the consortium agreement.

The commercial use of any information contained in this document may require a license from the proprietor of that information.

Neither the 5G-VINNI consortium as a whole, nor a certain part of the 5G-VINNI consortium, warrant that the information contained in this document is capable of use, nor that use of the information is free from risk, accepting no liability for loss or damage suffered by any person using this information.

The EC flag in this document is owned by the European Commission and the 5G PPP logo is owned by the 5G PPP initiative. The use of the flag and the 5G PPP logo reflects that 5G-VINNI receives funding from the European Commission, integrated in its 5G PPP initiative. Apart from this, the European Commission or the 5G PPP initiative have no responsibility for the content.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 815279.

Impressum

Full project title	5G Verticals Innovation Infrastructure
Project acronym	5G-VINNI
Number and title of work-package	WP4 Execution and Validation of Use Cases and KPIs on 5GVINNI End-to-End Facility
Number and title of task(s)	T4.1 Definition of Test Methods and Test Cases for validation of network service-level KPIs and use case-level KPIs T4.2 Development and maintenance of a test orchestration framework
Document title	Initial report on test-plan creation and methodology, and development of test orchestration framework
Editor: Name, company	Andrea F. Cattoni (KEYS)
Work-package leader: Name, company	Andrea F. Cattoni, (KEYS)

Copyright notice

Executive summary

The envisaged landscape of pervasive 5G services calls for solid and extensive trials, validation tests and measurements that innovating vertical customers strongly need to carry out for extracting reliable conclusions related to 5G performance.

The trial environments, as the one built by the 5G-VINNI project, should mirror the conditions and configuration that the vertical applications will face on their launch in production networks, to verify whether the vertical application can be considered 5G-ready. Even more importantly, by early testing their innovative use cases over a standards-based full-chain 5G facility, and following a systematic approach, a wide range of vertical industries may timely make well-informed business decisions on launching their services with guaranteed performance levels and therefore with higher chances of business success.

Knowing the expected performance is unfortunately not enough for providing a concrete validation of the network. To support such an effort, the 5G ecosystem needs to accommodate for frameworks that easily test, measure and demonstrate – for the wide range of new use cases and foreseeable conditions of deployments and scenarios – the Key Performance Indicators (KPIs) of the new softwarized 5G network carrier-grade environments.

New approaches and types of testing shall allow the fastest upgrade and deployment paces enabled by the software. Bridging together telco and Cloud Computing changes how testing is performed, and even the type of tests. There is still a gap in End-to-End (E2E) and Network Function Virtualization (NFV) characterizations, as well as in performance evaluation of the new 5G heterogeneous infrastructure. Test and Measurements (T&M) procedures, tools, and methodologies (i.e. testing, monitoring, analytics, and diagnostics) need to be agreed upon, to ensure a proper functioning of the deployed networks.

An entire WP in the 5G-VINNI project, WP4, is dedicated to such scope, making sure that whatever is built and delivered by the project is verified with a uniform testing procedure, allowing the Mobile Network Operators (MNOs) that are part of the consortium to have confidence in the system via Continuous Integration and Continuous Deployment (CI/CD) of the network components.

The focus of WP4 is on testing methodology, test execution and E2E network validation to verify that KPIs across different verticals are met. The WP is intended to provide all the needed testing and validation tools and support across the project, as well as providing the support needed by vertical projects hosted by the 5G-VINNI facility.

The WP is also the provider of technical content and support for the work carried out by the project within the 5G PPP Test, Measurement, and KPIs Validation (TMV), Working Group (WG). The alignment with such a group is deemed an essential part of the technical work performed within the WP, to be able to share and unify methodologies across different ICT-17 projects, leveraging on the experience of Phase II ones.

This document is structured as a report on the different activities of the Tasks T4.1 and T4.2 carried out during the first project Period, while the Annexes represent the initial structure of a live document that will become, at the end of the project, with D4.4 the official Test Specification handbook of the 5G-VINNI project.

List of authors

Company	Author	Contribution
Telenor ASA (TNOR)	Ole Grøndalen	Section 2.2
Keysight Technologies (KEYS)	Michael Dieudonné	Section 4.3
Keysight Technologies (KEYS)	Lars Mikkelsen	Section 4.4
Telefonica Investigacion y Desarollo SA (TID)	Sonia Fernández Tejería	Section 5
EANTC	Mohammad Othman	Section 2.1
EANTC	Haitao Shi	Section 2.3, 2.4, 3.1, 3.2, 3.3
Keysight Technologies (KEYS)	Andrea F. Cattoni	Section 2.2, 2.3, 2.4, 4
Simular Research Laboratory (SRL)	Ahmed Elmokashfi	Section 4.3, 4.5

Table of Contents

Executive summary 4
List of authors 5
Table of Contents 6
List of figures 8
List of tables 9
Abbreviations 10
Definitions 13
1 Introduction 14
2 Testing Methodology and Formalization 16
2.1 Considered Use Cases 16
2.1.1 Drone control for Public Protection and Disaster Relief 16
2.1.2 Drone Usage to add Coverage 17
2.1.3 Dynamic enrichment of video 17
2.1.4 Enhanced Mobile Broadband (eMBB) 17
2.1.5 Augmented Reality 18
2.1.6 Massive Machine Type Communications (mMTC) 18
2.1.7 Remote Ultrasound Examination 18
2.1.8 Telemedicine Using HoloLense 18
2.1.9 Ultra-Reliable Low-Latency Communications 18
2.1.10 Coverage Study and Slicing for Narrowband Internet of Things 19
2.1.11 First Person View (eMBB/URLLC) 19
2.1.12 360° Video broadcast (eMBB) 19
2.2 Testing and Monitoring: Differences and Priorities 19
2.2.1 Testing, Monitoring, and Telemetry Formal Definitions 20
2.3 Establishing a Testing Process for 5G Networks 20
2.3.1 Full Stack Testing Approach 20
2.3.2 Overview of Existing Standard Test Cases and Methodologies 21
2.3.3 Considered Test Areas 24
2.3.4 Test Case Formalization Process 26
2.3.5 Test Cases Overview 27
2.4 Testing KPIs Overview 30
2.4.1 Core KPIs Defined by 5G-PPP Test, Measurement, and KPIs Validation Working Group30
2.4.2 KPIs in Focus During the Rel’0 Testing Phase 31
3 5G-VINNI Site Facilities Release 0 Testing Plan 32
3.1 Testing Scope 32
3.2 Expected Testing Periods 32
3.3 Testing Focus: Expected Test Procedures and Test Cases 33
4 Test as a Service Development 34
4.1 Motivation 34
4.2 Testing Architecture 35

4.2.1 Alignment with WP1 35
4.2.2 Components 35
4.3 Traffic Generation Components Overview 36
4.3.1 IxLoad VE 36
4.3.2 Hawkeye 38
4.3.3 Yardstick 39
4.3.4 MONROE Network Monitoring Node 40
4.4 Automation Solution 43
4.4.1 Cloud-Native, Microservice-based Architecture 44
4.4.2 Microservices Overview 45
4.4.3 OpenTAP: Test Execution Service 49
4.4.4 Plugins Overview 51
4.5 Prototyping and Staging Deployment Status of TaaS 54
4.5.1 TaaS Traffic Generation Components 54
4.5.2 TaaS Automation Solution 56
5 Telemetry Components 57
6 Conclusions 61
References 62

List of figures

Figure 1 - The 5G system layers 20
Figure 2 – Expected Testing Periods for Release 0 32
Figure 3 - TaaS system overview from a tooling perspective 34
Figure 4 – Testing architecture introduced in D1.1 [14] 35
Figure 5 – User interface for multi-play user environment with realistic subscriber modelling 37
Figure 6 - Library containing different traffic and protocols 38
Figure 7 - Hawkeye real time dashboard 39
Figure 8 - example of System level test with Yardstick 40
Figure 9 - Grafana result visualization of Yardstick test results 40
Figure 10 - MONROE measurement node - Hardware 41
Figure 11 – Software processes on the node 42
Figure 12 – A high level overview of networking configurations 42
Figure 13 – Experiments scheduling workflow 43
Figure 14 – Cloud-native, Microservice-base system architecture 45
Figure 15 – The Executor manages resources in different execution environments 48
Figure 16 – overview of logging and telemetry components foreseen for the Automation system 49
Figure 17 – NB-IoT power measurement setup 55
Figure 18 - Telemetry architecture 57
Figure 19 – MON collector with model-based metrics 57
Figure 20 – Monitoring Service Architecture with model-based metrics 58
Figure 21 – Current deployment of telemetry service 59
Figure 22 – VNFD for OSM VCA metrics 59
Figure 23 – VNFD for Model-based metrics 60
Figure 24 – MON service with gNMI client 60

List of tables

Table 1 – Description of the Document Annexes 15
Table 2 – List of Use Cases Considered for Test Case Definitions 16
Table 3 –Considered Test Areas in Release 0 25
Table 4 - Test case template 26
Table 5 - Overview of Unit Test Cases for NFVI/MANO Interoperability Test 28
Table 6 - Overview of Sub-Module Test Cases for NFV Performance Test 29
Table 7 - Overview of Sub-Module Test Cases for Radio Access Network 29
Table 8 - Overview of Sub-Module Test Cases for Mobile Core Network 29
Table 9 - Overview of E2E Test Cases 30
Table 10 - Identified Technical KPIs 30
Table 11 – Focused KPIs in Release 0 31
Table 12 – Definition of requirements for the Automation Solution 44
Table 13 – Overview of Telemetry categories and components 49

Abbreviations

A list of abbreviations is strongly recommended

3GPP	3rd Generation Partnership Program
4G	Fourth Generation (mobile/cellular networks)
5G	Fifth Generation (mobile/cellular networks)
5G PPP	5G Infrastructure Public Private Partnership
AMF	Access and Mobility Management Function
API	Application Programming Interface
AR	Augmented Reality
CI/CD	Continuous Integration, Continuous Deployment
CLI	Command Line Interface
COTS	Commercial-Off-The-Shelf
CPE	Customer Premises Equipment
CPU	Central Processing Unit
DC	Data Centre
DFKI	Deutsches Forschungszentrum für Künstliche Intelligenz
DUT	Device Under Test
E2E	End-to-End
eMBB	enhanced Mobile Broadband
ESB	External Stakeholder Board
ETSI	European Telecommunications Standards Institute
FPV	First Person View
gNB	5G Node B
GUI	Graphical User Interface
HD	High Definition
HOT	HEAT Orchestration Template
IoT	Internet of Things
IT	Information Technology
KPI	Key Performance Indicator
MANO	Management and Orchestration
MEC	Multi-access Edge Computing
mMTC	Massive Machine Type Communications
MNO	Mobile Network Operator

MON	Monitoring
NB-IoT	Narrowband IoT
NetOps	Network Operations
NFV	Network Function Virtualization
NFVI	NFV Infrastructure
NFVO	Network Function Virtualization Orchestrator
NGMN	Next Generation Mobile Network
NGMN TTI	NGMN Test Trial Initiative
NS	Network Service
NSA	Non-stand-alone
OPEX	Operational Expenditure
PCF	Policy Control Function
PM	Performance Management
PNF	Physical Network Function
PPDR	Public Protection and Disaster Relief
QoE	Quality of Experience
QoS	Quality of Service
R&D	Research and Development
RAN	Radio Access Network
REST	REpresentational State Transfer
SBC	Single Board Computer
SDN	Software Defined Networks
SLA	Service Level Agreement
SMF	Session Management Function
SoA	State of Art
SUAS	Small Unmanned Aircraft Systems
SUAV	Small Unmanned Aerial Vehicle
SUT	System Under Test
T&M	Test and Measurement
TaaS	Test as a Service
TMG	Test, Measurement, and KPIs Validation
UAV	Unmanned Aerial Vehicle
UDM	Unified Data Management
UE	User Equipment

UPF	User Plane Function
URLLC	Ultra-Reliable Low Latency
VCA	VNF Configuration Adapter
vEPC	virtual Evolved Packet Core
VIM	Virtual Infrastructure Manager
VNF	Virtual Network Function
VNFD	Virtual Network Function Descriptor
VNFM	Virtual Network Function Manager
VoIP	Voice over IP
VR	Virtual Resource
WG	Working Group
ZSM	ETSI Zero-touch Network and Service Management

Definitions

This document contains specific terms to identify elements and functions that are considered to be mandatory, strongly recommended or optional. These terms have been adopted for use similar to that in IETF RFC2119, and have the following definitions.

MUST This word, or the terms "REQUIRED" or "SHALL", mean that the definition is an absolute requirement of the specification.

MUST NOT This phrase, or the phrase "SHALL NOT", mean that the definition is an absolute prohibition of the specification.

SHOULD This word, or the adjective "RECOMMENDED", mean that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.

SHOULD NOT This phrase, or the phrase "NOT RECOMMENDED" mean that there may exist valid reasons in particular circumstances when the particular behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.

MAY This word, or the adjective "OPTIONAL", mean that an item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because the vendor feels that it enhances the product while another vendor may omit the same item. An implementation which does not include a particular option MUST be prepared to interoperate with another implementation which does include the option, though perhaps with reduced functionality. In the same vein an implementation which does include a particular option MUST be prepared to interoperate with another implementation which does not include the option (except, of course, for the feature the option provides.)

1 Introduction

The envisaged landscape of pervasive 5G services calls for solid and extensive trials, validation tests and measurements that innovating vertical customers strongly need to carry out for extracting reliable conclusions related to 5G performance.

The trial environments, as the one built by the 5G-VINNI project, should mirror the conditions and configuration that the vertical applications will face on their launch in production networks, to verify whether the vertical application can be considered 5G-ready. Even more importantly, by early testing their innovative use cases over a standards-based full-chain 5G facility, and following a systematic approach, a wide range of vertical industries may timely make well-informed business decisions on launching their services with guaranteed performance levels and therefore with higher chances of business success.

Several use cases are considered by the 5G-VINNI consortium as representative of the potential applications that a future 5G network should support. The presence of the External Stakeholder Board (ESB) within the project made possible to have a preliminary overview of which type of performances are requested to the 5G network by the different vertical customers.

Knowing the expected performance is unfortunately not enough for providing a concrete validation of the network. To support such an effort, the 5G ecosystem needs to accommodate for frameworks that easily test, measure and demonstrate – for the wide range of new use cases and foreseeable conditions of deployments and scenarios – the Key Performance Indicators (KPIs) of the new softwarized 5G network carrier-grade environments.

New approaches and types of testing shall allow the fastest upgrade and deployment paces enabled by the software. Bridging together telco and Cloud Computing changes how testing is performed, and even the type of tests. There is still a gap in End-to-End (E2E) and Network Function Virtualization (NFV) characterizations, as well as in performance evaluation of the new 5G heterogeneous infrastructure. Test and Measurements (T&M) procedures, tools, and methodologies (i.e. testing, monitoring, analytics, and diagnostics) need to be agreed upon, to ensure a proper functioning of the deployed networks.

An entire WP in the 5G-VINNI project, WP4, is dedicated to such scope, making sure that whatever is built and delivered by the project is verified with a uniform testing procedure, allowing the Mobile Network Operators (MNOs) that are part of the consortium to have confidence in the system via Continuous Integration and Continuous Deployment (CI/CD) of the network components.

The focus of WP4 is on testing methodology, test execution and E2E network validation to verify that KPIs across different verticals are met. The WP is intended to provide all the needed testing and validation tools and support across the project, as well as providing the support needed by vertical projects hosted by the 5G-VINNI facility. The objectives of the WP are:

1. Develop a 5G-specific test methodology that encompasses test cases from component integration to E2E performance tests.
2. Define a set of test cases that can be recognized as industry reference for 5G network validation and testing.
3. Develop, deploy, and maintain a testing framework and infrastructure for 5G-VINNI that can be assumed as reference by the industry.
4. Validate the 5G KPIs via execution of cross-facilities test campaigns.
5. Support the validation of third parties' use cases by customizing the fundamental knowledge acquired during the project.

The WP is also the provider of technical content and support for the work carried out by the project within the 5G PPP Test, Measurement, and KPIs Validation (TMV), Working Group (WG). The alignment with such a group is deemed an essential part of the technical work performed within the

WP, to be able to share and unify methodologies across different ICT-17 projects, leveraging on the experience of Phase II ones.

During the first Period of the 5G-VINNI project, that this Deliverable is reporting, the activities covered the points 1 to 3, while the practical field testing and engagement with is expected for the second year. In this Period the WP has been strongly active in contributing to the 5G PPP TMV-WG, as later explained in Section 2.4.1.

This document is structured as a report on the different activities of the Tasks T4.1 and T4.2, while the Annexes represent the initial structure of a live document that will become, at the end of the project, with D4.4 the official Test Specification handbook of the 5G-VINNI project.

Section 2 provides an overview of the use cases considered by the 5G-VINNI project and describes the initial work for the definition of the KPIs to be used in the testing. Such an effort comprises of a deep State of Art (SoA) overview on the existing standard white paper documentation, as well as the alignment work performed within the scope of the 5G PPP TMV-WG. Having this in mind, the Section provides an initial definition of what 5G-VINNI should consider a Testing Process for 5G.

In Section 3 is described the expected testing Plan for the site facilities in Release 0 of the 5G-VINNI infrastructure. The first two Sections are related to the work performed in T4.1.

Section 4 is instead an overview of the concept of Test as a Service (TaaS) as interpreted by the 5G-VINNI project. The Section describes the needs and purpose of such a system, identifies a practical architecture and components, and report on the current development status of the system. This Section is related to the work performed in T4.2.

An introductory overview of Monitoring components developed in 5G-VINNI is presented in Section 5. The idea is to have a preliminary assessment of how Monitoring can be implemented in the site facilities using OSM as orchestration component.

Finally, Section 6 provides a conclusion over the work carried out in the first Period of the project, and it is followed by the Annexes, in particular:

- Annex A: 5G-VINNI Test Specification – 5G-VINNI Test Methodologies and Test Cases
- Annex B: 5G-VINNI Key Performance Indicator Definitions
- Annex C: 5G-VINNI Pre-qualification Test Cases
- Annex D: 5G-VINNI Test Cases Handbook

A more detailed overview of the content is described in Table 1.

Table 1 – Description of the Document Annexes

Annex	Title	Content
Annex A	5G-VINNI Test Specification – 5G-VINNI Test Methodologies and Test Cases	Test methodology and definition of the testing procedures.
Annex B	5G-VINNI Key Performance Indicator Definitions	Formalization of the KPIs aligned with the 5G PPP Template format
Annex C	5G-VINNI Pre-qualification Test Cases	List of Pre-qualification tests to be performed by site facilities before handing over the infrastructure for performance testing
Annex D	5G-VINNI Test Cases Handbook	Definition of the additional Test Cases defined by the 5G-VINNI project additionally to the ones available on the SoA

2 Testing Methodology and Formalization

This section describes the development process of the test cases, starting from the considered use cases, to the testing procedure and testing methodology. A test case template is provided as a reference and guidance of all test cases format for 5GVINNI project. The overview of test area, test case, and focusing KPIs are also listed in this section, details of the definition of test plan and KPI definition can be found in D4.1 Annex B and Annex D.

2.1 Considered Use Cases

To define the test cases according to the actual usage scenarios in the vertical industry, after alignment with D3.1 [17], below use cases in Table 2 are considered while designing the test plans:

Table 2 – List of Use Cases Considered for Test Case Definitions

Use case	Vertical	Service type
Drone control for PPDR	eHealth	
Drone Usage to add Coverage		
Dynamic enrichment of video		
Enhanced Mobile Broadband (eMBB)		eMBB
Augmented Reality		
Massive Machine Type Communications (MMTC)		mMTC
Remote Ultrasound Examination	eHealth	
Telemedicine Using HoloLense	eHealth	
Ultra-Reliable Low-Latency Communications		URLLC
Coverage Study and Slicing (NB-IoT)		mMTC
FPV (eMBB/URLLC)		eMBB/URLLC
360° Video broadcast (eMBB)		eMBB

2.1.1 Drone control for Public Protection and Disaster Relief

Use of Unmanned Aerial Vehicles (UAVs), commonly referred to as drones, to assist the fire brigade or the police in downtown Oslo for Public Protection and Disaster Relief (PPDR). A drone and a pilot will be located on a fire truck or in a police car. After arriving at the scene, the drone will be airborne and provide information to the response team using the drone's camera. A scenario with a terrorist on the move will also be considered.

In case of fires, the drones can also be used after the fire has been extinguished to generate fire maps and 3D models e.g. to help determine the cause of the fire. The drone will be flown autonomously or manually up 120 meters altitude.

2.1.2 Drone Usage to add Coverage

Use of a flexible Small Unmanned Aircraft Systems (SUAS) deployments with multiple Small Unmanned Aerial Vehicle (SUAV) units, which are interconnected to form an aerial communication network, enabling the provision of value-added services, such as building aerial sensor networks to aid disaster management, over a delimited geographical area. Therefore, the main goal is to support the automated configuration and deployment of complex and heterogeneous network services over a communication platform composed by SUAVs. These network services will be composed by a set of interconnected Virtualized Network Functions (VNFs), each providing the software implementation of a physical network function. The VNFs are going to be implemented into a Single Board Computer (SBC), such as a Raspberry Pi, carried by drones as light payloads. Regarding the design of SUAV devices, they should implement a wireless interface enabling the direct communication with every other SUAV that is within its radio coverage. Additionally, some SUAVs will deploy a wireless access point, supporting the access of mobile ground stations (users) to the aerial infrastructure. Besides this, the VNFs deployed over a particular SUAV will be interconnected through a virtual private network. The services that the SUAVs are going to implement are the ones related with the establishment of a Voice over IP (VoIP) call and the distribution of images. Moreover, the VNFs would be also capable of integrate the telemetry management.

2.1.3 Dynamic enrichment of video

In this use case, a “default” video delivery service will be “enriched” over time, depending on the users’ preferences and availability of resources. This enrichment will consist on adding features such as e.g. real-time, augmented reality-like information (depending on hardware availability), video track for hearing impaired, or real-time interaction with other users. From the user perspective, it will serve to illustrate the benefits of a seamless service re-composition, adding modules to the video delivery service and re-placing functionality in order to fulfil application requirements. From the operator / technical perspective, the main objective of this use case consists in showing the feasibility and the benefits of an elastic management and orchestration of the network. By “elastic” we refer to the ability to adapt the operation of the various modules to the resources available (e.g., radio resources, CPU resources). This will entail an improvement of the network efficiency and its capability to smoothly adapt the resource allocation and utilization.

The proposed use case features two slices: i) an enhanced Mobile Broardband (eMBB) slice to serve highly-demanding videos (in terms of requirements), such as, e.g., 8K videos or 360° videos, and ii) an Ultra-Reliable Low Latency Communications (URLLC) slice to provide interaction with the environment or other users, e.g., real-time collaboration with other avatars in a virtual scenario, gaming applications. The testbed includes a set of Physical Network Functions (PNFs) that implement the radio lower layers and a cloud infrastructure (composed by a larger but farther to the radio central cloud and a closer but less capable edge cloud). In particular, besides the virtual reality application, the testbed will be composed by a set of functionalities: Network Slice Blueprinting and Onboarding, VNF Relocation due to latency, Horizontal and Vertical VNF Scaling, Admission Control.

2.1.4 Enhanced Mobile Broadband (eMBB)

eMBB is a set of services characterized by their need for high capacity links. Some example eMBB services are virtual / augmented reality and High Definition (HD) video streaming. It is expected that the mobile capacity should increase 100 times over the capacity offered by current state of the art 4G systems to meet these requirements in urban user densities. Users should be able to achieve high speeds even at challenging scenarios such as automotive content streaming, while being at the cell edge and in crowded areas.

2.1.5 Augmented Reality

This use case describes the delivery of annotations on top of a video stream captured by an end user of the service. The end user captures video which is delivered to an Augmented Reality (AR) application backend, which is responsible for object recognition, tracking and annotation of the video with related information. The video annotation is performed taking into account the position of the identified objects in the video and delivered as an enhanced video stream to the user. This use case is expected to be supported by the Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI) as a member of the ESB. The exact application context will be later defined, choosing from the wide DFKI portfolio. A candidate application targets the touristic domain, providing informational annotations on top of identified monuments. As with the use case 1, initially both ends of the use case would be located in the University of Patras facility. Depending on met KPIs the two ends could potentially be located on different facility sites. The use case targets the use of a Multi-access Edge Computing (MEC) location within the Patras facility, as a means to lower latency.

2.1.6 Massive Machine Type Communications (mMTC)

Massive Machine Type Communications (mMTC) is the communication paradigm for the next generation of embedded and smart devices, such as the ones used in transportation, utilities, healthcare, consumer goods and manufacturing. Some key aspects of these devices are the their fully automated nature and the minimal interaction with humans. They are expected to send small amounts of information to servers, cloud and some devices controlled by humans. Some of these devices are expected to be installed in remote, hard to access and / or with bad coverage locations. Thus, a key requirement is the ability to use of very robust modulation and coding scheme and in general the ability to communicate under bad signal. They are expected to last a decade in a single battery and the overall device cost should be low. Finally, their number is expected to explode in the coming years, thus base stations should be able to handle the additional overhead in signalling.

2.1.7 Remote Ultrasound Examination

An ultrasound probe will be mounted on a small robot. An expert will control this with a “joystick” type of device that get haptic feedback from the robot and perform an ultrasound examination on a patient. Existing audio/video/ultrasound streaming systems will be demonstrated over 5G networks.

2.1.8 Telemedicine Using HoloLense

Ambulances from Oslo Municipality are equipped with Nomadic HoloLens. Paramedics send over electronic patient care record and initiate video conference and/or 3D streaming with a remote expert located in the Oslo University Hospital (Rikshospitalet) in order to take the best decision for patient treatment and/or patient orientation.

The platform relies on two 5G hotspots, located in downtown Oslo and at the hospital, provided by Telenor.

2.1.9 Ultra-Reliable Low-Latency Communications

URLLC is a set of services with strict latency and reliability requirements. Some example applications in this category include tactile internet, remote control of factory machinery, professional audio and self-driving cars. Reliability increase is achieved by dedicating more resources to signalling, higher redundancy encoding and retransmissions. The above increase latency though, especially if the packet size is big. If bandwidth requirements are low, by using a smaller packet size, it is possible to achieve low latency and high reliability simultaneously. Another benefit of shorter packets is the reduction of transmission times.

2.1.10 Coverage Study and Slicing for Narrowband Internet of Things

The suggested use case, related to Internet of Things (IoT) devices, consists first of the coverage study of Narrowband IoT (NB-IoT) devices on the University of Patras facility site. A number of NB-IoT devices will be deployed and the network coverage will be examined taking into consideration various factors including, but not being limited to, distance from the antenna, penetration power etc. Subsequently, the use case will focus on the IoT slicing concept, where two distinct IoT applications will consume data provided by corresponding virtualized IoT Gateway instances, belonging to separate IoT Slices.

2.1.11 First Person View (eMBB/URLLC)

The use case suggests a remote-controlled vehicle which can provide real time video streaming from an on-board camera allowing the user to control and move the vehicle in First Person View (FPV) mode. The vehicle would be moving in rural area (University of Patras facility). Initially both the user side and the vehicle will be on the same site and 5G cell. For following iterations of the use case, the streaming end-point, e the remote-control location and the vehicle can be on different facility sites and connect to different 5G cells, assuming that the KPIs are met when the facilities sites are interconnected.

Design wise, the use case requires the implementation of the user side (control and reception of the video streaming) and the implementation of the vehicle side, which requires the use of light but powerful mini-pc and autopilot hardware for the vehicle control.

This use case includes elements from both eMBB and URLLC.

2.1.12 360° Video broadcast (eMBB)

This use case describes the video streaming from a person equipped with a 360ocamera to a viewer. As with the use case 1, initially both ends of the use case would be located in the University of Patras facility. Depending on met KPIs the two ends could potentially be located on different facility sites.

2.2 Testing and Monitoring: Differences and Priorities

Depending on how we look at the network, and in which phase of the network lifecycle we operate, there are even substantial differences in the KPIs that can be observed and measured. This difference is mostly due to the difference between Testing and Monitoring.

Testing (or Active Testing) provides greater observability due to the active control over the type and intensity of traffic that is pushed through the network and subsets of the network elements. This provides more degrees of freedom in selecting what can be tested and measured (e.g., scalability or security resilience). Monitoring is instead a generally passive process that looks at what happens in the network. For this reason, the KPIs that can be measured in testing and monitoring are substantially different. An example of these differences can be seen in two similar documents coming from the Next Generation Mobile Network (NGMN) initiative [1] and the 3rd Generation Partnership Program (3GPP) [2]. The former is focused on testing aspects, while the latter provides an overview of KPIs to be measured during normal network operations.

Testing is normally applied during the network rollout, during the onboarding process of new VNFs or new software updates, and for the validation of newly deployed network services. Monitoring is instead constantly active during network operations, and it is a key enabler of network management actions and processes. In 5G-VINNI, it has been given priority to testing. That enables, in a first place, to start validating the 5G PPP Contractual KPIs, and secondly to verify the stability and usability of the 5G-VINNI infrastructure. Furthermore, testing is also the number one priority of the 5G PPP TMV-WG, which the project wishes to be fully aligned.

2.2.1 Testing, Monitoring, and Telemetry Formal Definitions

Since the distinction between Testing, Monitoring, but also Telemetry is relevant not only for the 5G-VINNI project but also as a general differentiation in testing the 5G network, it is appropriate to provide a formal definition of these three terms.

- **(Network, Slice, Infrastructure, NFVI, VNF) Testing**: taking measures to check the quality, performance, reliability, or conformance of (Network, Slice, Infrastructure, NFVI, VNF) especially before putting it into production.
- **(Network, Slice, NFVI, Interface) Monitoring:** observing and checking the progress or quality of the infrastructure supporting and the traffic flowing through the (Network, Slice, NFVI, Interface) over a period of time. Maintaining regular surveillance over the infrastructure supporting and the traffic flowing through the (Network, Slice, NFVI, Interface).
- **(Network, Slice, Infrastructure, NFVI, VNF) Telemetry**: recording measurements or statistical data about the elements or components of the (Network, Slice, Infrastructure, NFVI, VNF) and collecting them into a remote or centralized location or database.

2.3 Establishing a Testing Process for 5G Networks

2.3.1 Full Stack Testing Approach

The 5G network is composed of several complex and heterogeneous components, blending Information Technology (IT), cloud, and telecommunication technologies. These technologies provide building blocks that can be stacked on top of each other to create the full 5G network, from the foundations up, like the pyramid in Figure 1.

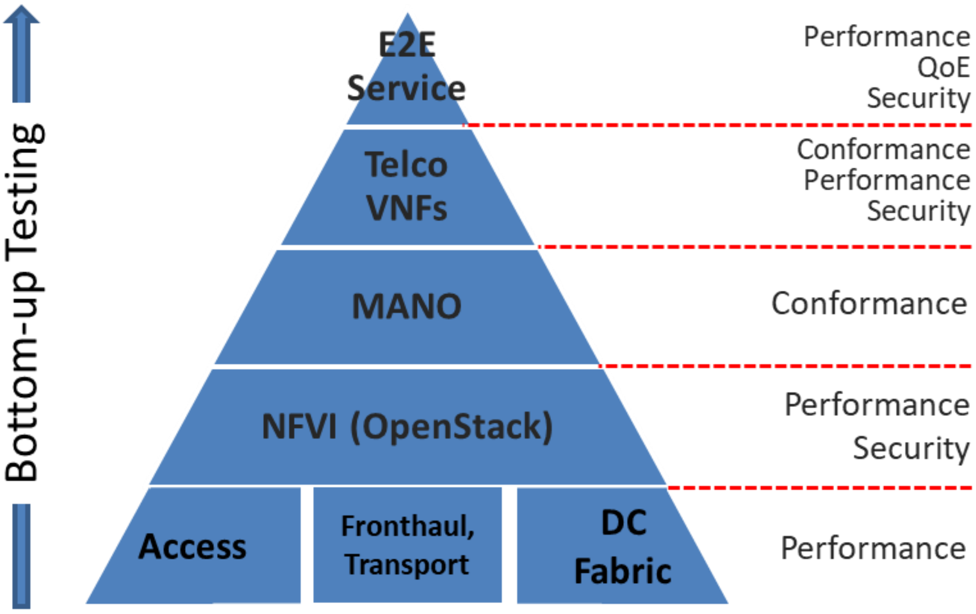


Figure 1 - The 5G system layers

At the bottom of the pyramid there are the basic transport technologies, such as front- and backhaul, and the Data Centre (DC) network fabric. On those, the NFV infrastructure (NFVI) is built, with cloud technologies such as OpenStack. The Management and Orchestration is instead a kernel component for enabling the NFV principles. The telecommunication and service components seen as VNFs can be then included in the picture, and with those, at the tip of the pyramid, there are the E2E network services.

Each level carries along with different types of tests that **should** (or more appropriately **must**) be performed while deploying and integrating the network, onboarding the VNFs, and providing the services. In 5G-VINNI, a bottom-up approach is considered, jointly with a CI/CD process, to validate and continuously guarantee the stability of the 5G network. Most of the test types and test cases would naturally come together with the technology from the different standard bodies. 5G-VINNI will try to inherit, whenever possible, such testing elements, and it will try to bridge the gaps with additional testing methodologies and components wherever deemed missing.

2.3.2 Overview of Existing Standard Test Cases and Methodologies

Several existing documents are covering different aspects of testing for the new components and environments foreseen for 5G networks. These documents have been analysed in order to establish a baseline for the 5G-VINNI investigation on testing procedures, KPIs, and test cases.

These documents, coming mostly from the European Telecommunications Standards Institute (ETSI) and 3GPP, can be categorized based on which technology and type of test are covering:

- NFV Performance Testing: describing how to analyse and profile the performances of the NFV Infrastructure
- NFV Conformance and Interoperability Testing: describing how to verify the interoperability between orchestration components and check how their interfaces conform to specific standards
- 5G Radio Access Network (RAN) Specific Testing: providing KPIs, metrics to stress or monitor in the RAN components of the 5G network
- E2E and Quality of Experience (QoE) Testing: describing how to test and verify both the network performance and the service performance from a high-level E2E perspective.

During the analysis, it emerged that the documents have different levels of details and content types. Specifically, when it comes to the actable definition of test cases and KPIs, some of them provide what can be called a “testing direction” rather than a specific Test Specification. They are nevertheless useful to establish a foundation on which the 5G-VINNI Test Specifications can be built. Other documents are more detailed are substantially ready to be handed in the hands of testers for actual test executions. In that case, in the 5G-VINNI specification, such areas will be considered already covered, and the documents directly referred to as part of the 5G-VINNI Test Methodology.

2.3.2.1 NFV Performance Testing

This first category covers the documents that describe available test methodologies, test cases, and KPIs in the area of the NFV Performance Testing. The documents are all coming from the ETSI NFV-TST working group, and they define mostly NFVI performance testing methodologies, KPIs, and a few exemplary test cases.

2.3.2.1.1 ETSI NFV-TST 001

The ETSI NFV-TST 001 [3] is the main document for NFVI performance testing found in the SoA, and it covers the generic methodology for deriving actable test cases. The document subdivides the testing area first into the NFVI areas (Compute, Storage, Network), and then into testing types (Performance/Speed, Capacity/Scale, Reliability/Availability). Several KPIs are defined in the document, though they are not formalized in a precise way. In a similar fashion, roughly twenty test cases are available, but they are missing the level of detail that would make them good candidates for being part of a Test Specification suite.

Despite they inaccuracies, the document must be considered foundational, first and foremost for setting precise directions on how to perform NFVI testing. Secondly, because it is the methodological source used to create the test cases available in the OPNFV Yardstick project (see Section 4.3.3).

2.3.2.1.2 ETSI NFV-TST 004

The ETSI NFV-TST 004 [4] is a methodology document that deepens certain specific aspects related to the NFV system, namely the Path testing. A Path is, in the ETSI terminology, the E2E interconnection provided by a chain of VNFs. The document provides a small set of KPIs covering both the Path Instantiation procedure and the Path Performance.

2.3.2.1.3 ETSI NFV-TST 009

The ETSI NFV-TST 009 [5] is a methodology document that provides an accurate definition of several KPIs (or "metrics") that are actable and practical. These definitions include test and measurement procedures needed to perform tests on several types of System Under Test (SUTs). The main covered KPIs areas include Throughput, Latency, Delay Variation, Loss, while amongst the test procedures, the document covers specifically the search algorithms for providing extensive testing.

2.3.2.2 NFV Conformance and Interoperability Testing

This category covers the documents that describe available test methodologies, test cases, and KPIs in the area of the NFV Conformance and Interoperability Testing. Even in this case the documents are all coming from the ETSI NFV-TST working group, and they are more detailed than the ones previously introduced on Performance.

2.3.2.2.1 ETSI NFV-TST 007

The ETSI NFV-TST 007 [6] is a full Test Specification document that covers the aspects of NFVI Interoperability. While the focus is obviously not on KPIs given the nature of the testing type, it contains more than 50 detailed test cases that will be considered "as is" and recommended in the 5G-VINNI Test Specification document. The test cases cover the Management of Software Images, VNF Packages, VNF Lifecycles, Fault, Performance, and Network Services.

2.3.2.2.2 ETSI NFV-TST010

The ETSI NFV-TST010 [7] is a full Test Specification document that covers the conformance of specific implementations of the NFV interfaces. The considered reference interfaces are:

- SOL002 for the Ve-Vnfm-en interface;
- SOL003 for the Or-Vnfm interface;
- SOL005 for the Os-Ma-Nfvo interface.

The document is currently in a draft state and not officially published yet. The interesting aspect is the open source availability of the test cases in a Robot implementation. Given also the presence of executable test scripts, these test cases will be considered for introduction within the 5G-VINNI Test Specification.

2.3.2.3 5G Functions Testing: 3GPP TS 28.552

This category covers the only document available that identifies KPIs in the 5G functions area: the 3GPP TS 28.552 [2]. The document identifies a series of KPIs covering the performance of the different network components, covering the RAN and part of the 5G core, namely:

- 5G Node B (gNB)
- Access and Mobility Management Function (AMF)
- Session Management Function (SMF)
- User Plane Function (UPF)
- Policy Control Function (PCF)
- Unified Data Management (UDM).

The document is intended to provide KPIs for Network Management and Orchestration, detailing which live measurements need to be performed during normal operations in production networks. This document will become useful, in its entirety only at a later stage of the project, when network monitoring and telemetry will be introduced. It is nevertheless a good starting point for identifying some KPIs that can be used during testing and how to process measurements coming from the network components.

2.3.2.4 End-to-End and Quality of Experience Testing

This first category covers the documents that describe available test methodologies, test cases, and KPIs in the area of E2E and QoE Testing. These references come from multiple sources, and they represent the most recent advances in the evaluation of the network performance from the user perspective.

2.3.2.4.1 3GPP TS 28.554

The 3GPP TS 28.554 [8] is another document focused on Network Management, but it offers interesting views on which type of KPIs and use cases would be needed to observe and validate the concept of Slice. Most of the presented KPIs look at the slice from an aggregated perspective, de facto analyzing the traffic that massively transverse the 5G core network. The document associates to each KPI a specific use case, creating examples on how monitoring that specific performance indicator can be used in the management of the 5G network. It must be noted that both the TS 28.552 and the TS 28.554 are the first 3GPP documents analyzing and detailing KPIs related to the use of virtual resources, entering the domain traditionally covered by the ETSI NFV group.

2.3.2.4.2 NGMN Test Trial Initiative Testing Framework

This document [1] has been created by the NGMN Test Trial Initiative (TTI) to harmonize how the 5G trials would have been carried out. The main intention was to create a set of common KPIs and testing methods covering several aspects of the 5G system making sure that different trials could have comparable results. The document includes KPIs and testing methods covering the RAN, the E2E QoE, and certain User Equipment (UE) aspects such as energy consumption. The KPIs identified in the document are specific and clear, but they are not formalized in depth. They are, though, a very good initial step for creating a testing framework for 5G.

2.3.2.4.3 EU H2020 TRIANGLE Project D2.6

The EU H2020 TRIANGLE Project D2.6 [9] was originally intended as a way to do QoE testing for mobile devices and mobile application. The methodology can be reused to provide a simple, human-readable grading of the network performance, as exemplified in the contribution that the project provided to the NGMN TTI Testing Framework document [1]. The methodology is founded on a clear per-application/use case set of KPIs and a set of interpolation and transformation functions, making it possible to create a “synthetic” Mean-Opinion Score. The document is a full Test Specification focused on UEs and Mobile Apps, even though a majority of the test cases could be revised to cover network testing aspects.

2.3.2.5 Summary

In this section, several documents covering existing test methodologies, KPIs, and test cases have been analyzed. Most of the documents provide useful insights into how to create Test Specifications for the 5G-VINNI project. The intent is to create a set of recommendations and Test Cases useful to perform extensive testing to the full 5G infrastructure. 5G-VINNI project partners (especially EANTC and Keysight Technologies) have actively contributed to the production of most of the analyzed documents, putting 5G-VINNI in a unique position for influencing the testing ecosystem for 5G.

2.3.3 Considered Test Areas

Based on the considered use cases discussed in section 2.1 and current existing standard test cases and methodologies in section 2.3.2, below are the considered test areas for 5G-VINNI project:

- Performance

  Performance testing, also called stress testing or load testing, aims to verify that the SUT can tolerate required constant load of service requests, and that the SUT will perform according to the performance requirements, and will have adequate response time for valid requests.

  Network performance test uses large volumes of valid traffic to find the limits of how much traffic a system is able to handle in normal and impaired (non-optimal) network conditions.

  Interested KPIs will be network throughput, latency, packet loss, jitter.

- Scalability and Capacity

  Network scalability and capacity tests ensure the ability of a unit, system or an E2E network to handle the increased number of users, data volume, process transactions or traffic throughput without any degradation in the QoE or the reliability of the service.

- Interoperability & Conformance

  Interoperability testing ensures the E2E functionality of the whole system not the individual parts or units. The results of an interoperability test usually are applicable for a particular implementation or system which was tested and not necessarily to be valid for other non-identical implementations. So, usually the interoperability is hard to be automated.

  Conformance testing gives a high-level of confidence that key components of a device or system are working as they were specified and designed to do. Usually the conformance tests are based on the industry standards, so the repetitive procedures could be automated. In general the system is greater than sum of its parts meaning that the conformance testing does not prove interoperability.

  Interoperability between different network components (e.g. between RAN and Core network) and interoperability between different facility sites will be tested. The conformance of the NFV MANO components is also considered to be tested.

- Manageability (not for the first deliverable)

  Test of service provisioning and subscriber management are considered in Release 1 tests. The network fault management and performance monitoring tests will be tested partially in Release 0 and will be completed in Release 1.

- Security (not for the first deliverable)

  Security feature tests on AAA, user management, and policy management are considered in the Release 1 test.

- Resiliency and High Availability (not for the first deliverable)

  Network resiliency and high availability tests for the infrastructure and services are not included in Release 0.

Table 3 lists the considered test areas in Release 0:

Table 3 –Considered Test Areas in Release 0

No.	Test Area	Sub- Test Area
1	NSA E2E Network Performance Test	eMBB Network Slice
2		uRLLC Network Slice
3		mMTC Network Slice
4	NSA E2E Network Scalability Test	eMBB Network Slice
5		uRLLC Network Slice
6		mMTC Network Slice
7	NFV Performance Test	NFVI Virtual Resource Performance Test
8		NFVI Network Performance Test
9		Network Service Autoscaling Validation
10		Network Service Speed of Activation Test
11	NFVI/MANO Interoperability Test	Software Image Management
12		VNF Package Management
13		VNF Lifecycle Management
14		Fault Management
15		Performance Management
16		NS Lifecycle Management
17	NSA vEPC Performance Test	vEPC Performance Measurement
18		vEPC User Capacity Measurement
19	Transport Network Performance Test	Transport Network Performance Test

2.3.4 Test Case Formalization Process

As a general reference for test case formalization in the 5GVINNI project, the basic structure and requirements in a test case are defined in this section. The test report formalization is not in the scope of this document and will be discussed separately.

We introduced modularization in the test case template shown below in Table 4. By summarizing the comment elements in each test area, we can avoid the unnecessary duplication in the test case description and therefore reducing the maintenance effort in the later stage of the project. For example:

1) the test topology and common parameters are presented at the beginning of each test area, all test cases sharing similar test environment in that test area can refer to the same section for this information, avoiding repeating same content in every test case;
2) The KPI requirements and measurements are summarized in each test area; all test cases measuring the same KPIs must follow the same requirements and measurements. When there's an update of interested KPIs in the project level or facility site level, the KPIs or means of measurements can be updated without going through all related test cases.
3) Common preambles and postambles are also listed at the beginning of each test area, as a reference for all test cases.

All test cases developed for 5G-VINNI must follow below the structure, not applicable sections can be skipped (for example, KPIs are N/A for interoperability tests):

1. **Preambles:** including pre-conditions, test setups, and requirements on pre-qualification tests.
2. **Test Parameters:** describe common test parameters for test cases in this test area.
3. **Measurements:** describe the method of computing the KPIs from the actual measurements. Optional for non-performance tests.
4. **Key Performance Indicators:** list all the KPIs measured in this test area and the required value for each KPI. Optional for non-performance tests.
5. **Postambles:** summarize the common postambles for all tests in this test area and describe the procedure and requirement of the actions that should be taken after each test case is executed.
6. **Test Cases:** list all test cases in this test area with below format in Table 4.

Table 4 - Test case template

Test Case Name	Name of the test case. Not necessary if the test case name is shown in the title of this table.
Purpose	Describes what is tested in the test case.
Description	The high-level description that clarifies what is tested in the test case
Initial Conditions	Describes the initial conditions applied for this section, as well as the reference to the Test Setup diagram.
Parameters	Use the specific test case parameters described below.

<table>
<tr><td></td><td><b>Table x: Parameters for xxx measurement</b><table><tr><th>Parameters derived from KPIs</th><th>Values</th></tr><tr><td></td><td></td></tr></table></td></tr>
<tr><td><b>Procedures & Expected Results</b></td><td>The following table describes the Procedures & Expected Results in detail.</td></tr>
<tr><td></td><td><b>Table y: Procedures & Expected Results for xxx measurement</b><table><tr><th>Steps</th><th>Procedures</th><th>Expected Results</th></tr><tr><td>1.</td><td></td><td></td></tr><tr><td>2.</td><td></td><td></td></tr><tr><td>3.</td><td></td><td></td></tr><tr><td>4.</td><td></td><td></td></tr></table></td></tr>
<tr><td><b>Measurements</b></td><td>Define the unique method of measurements if not included in the <b>Measurements</b> section at the beginning of this test area.</td></tr>
<tr><td><b>Postambles</b></td><td>Define the unique postambles if not included in the <b>Postambles</b> section at the beginning of this test area.</td></tr>
</table>

2.3.5 Test Cases Overview

The test in each facility site is divided to two test phases: 1) site acceptance test (pre-qualification tests, executed by each facility site in WP2); 2) integration test (unit test, sub-module tests, E2E tests, executed by WP4).

In this section, we will provide an overview of all test phases and test types; the actual test cases will be listed in D4.1 Annex C and Annex D.

2.3.5.1 Pre-qualification Test Cases

The site acceptance test is executed by each facility site, and test results will be used as input from WP2 to handover the facility site to WP4 to execute the integration tests.

Each facility site can select the different focus of testing areas and a different set of pre-qualification test cases to be executed. As described in the test case formalization section, pre-qualification tests are one of the preambles to perform the integration test. To be ready for a certain area of integration test, the required pre-qualification tests must be tested and passed. WP4 will execute the integration tests based on the results from the pre-qualification test.

The pre-qualification test cases can be found in D4.1 Annex C.

2.3.5.2 Unit Test Cases

Overview of Unit Test Cases is listed in Table 5. The details of the test cases can be found in D4.1 Annex D.

Table 5 - Overview of Unit Test Cases for NFVI/MANO Interoperability Test

Test Category	Test Case Name	Test Case Purpose
Software Image Management	add software image	Verify that the NFVO can add a software image to the image repository managed by the VIM
	query software image by NFVO	Verify that the NFVO can retrieve the information of a software image from the image repository managed by the VIM
	query software image by VNFM	Verify that the VNFM can retrieve the information of a software image from the image repository managed by the VIM
	update software image	Verify that the NFVO can update the metadata of a software image in the image repository managed by the VIM
	delete software image	Verify that the NFVO can delete a software image from the image repository managed by the VIM
Fault Management	virtualized resource fault alarm notification	Verify that a fault alarm notification propagates to the NFVO when a virtualized resource that is required for the NS connectivity fails
	virtualized resource fault alarm clearance notification	Verify that a fault clearance notification propagates to the NFVO when a failed virtualized resource that is required for the NS connectivity is recovered
	VNF fault alarm notification	Verify that a VNF fault alarm notification propagates via the VNFM to the NFVO when a VNF fault is triggered by a failed virtualized resource
	VNF fault alarm clearance notification	Verify that a VNF fault alarm clearance notification propagates via the VNFM to the NFVO when a VNF fault is cleared by resolving a failed virtualized resource
Performance Management	VR PM job creation and notification monitoring	Verify that the performance metrics of a virtualized resource that is required for a NS instance connectivity can be monitored using PM jobs and notifications
	VR PM job creation and threshold monitoring	Verify that the performance metrics of a virtualized resource that is required for a NS instance connectivity can be monitored using PM jobs and thresholds
	VR PM job deletion	Verify that the monitoring of performance metrics of a virtualized resource that is required for a NS instance connectivity can be stopped by deleting PM jobs
	VR PM threshold deletion	Verify that a threshold created for a virtualized resource that is required for a NS instance connectivity can be deleted

2.3.5.3 Sub-Module Test Cases

Overview of Sub-Module Test Cases is listed in Table 6, Table 7, Table 8. The details of the test cases can be found in D4.1 Annex D.

Table 6 - Overview of Sub-Module Test Cases for NFV Performance Test

Test Case Name	Test Case Purpose
NFVI Network Performance Measurement	Measure the L3 network performance between an originating virtual machine and a target entity on the NFVI. Reference: ETSI NFV-TST001
NFVI Virtual Resource Performance Measurement	Measure the Virtual Resource (Storage/processor/memory) performance in a virtual machine on the NFVI. Reference: ETSI NFV-TST001
Network Services Auto scaling validation	To verify the successful completion of NS auto scaling in response to auto scale stimuli. Reference: ETSI NFV-TST001
Network Services Speed of Activation Test	To measure the time needed to activate a Network Service that comprises multiple VNFs in a service chain Reference: ETSI NFV-TST001

Table 7 - Overview of Sub-Module Test Cases for Radio Access Network

Test Case Name	Test Case Purpose
RAN Network Performance Measurement	Measure the end to end performance of RAN based on 3GPP Rel15 NSA system architecture.

Table 8 - Overview of Sub-Module Test Cases for Mobile Core Network

Test Case Name	Test Case Purpose
vEPC NSA Network Performance Measurement	Measure the end to end performance of vEPC mobile core which supports 5G radio network based on 3GPP Rel15 NSA system architecture.
vEPC NSA Network Capacity Measurement	Measure the system capacity vEPC that will support 5G network deployments based on NSA architecture.

2.3.5.4 E2E Test Cases

Overview of E2E Test Cases are listed in Table 9. The details of the test cases can be found in D4.1 Annex D.

Table 9 - Overview of E2E Test Cases

Test Case Name	Test Case Purpose
E2E eMBB Performance Measurement	Measure the End to End performance of UE IP packets transmitted from UE to the N6 interface in the eMBB slice of the 5G network.
E2E uRLLC Performance Measurement	Measure the End to End performance of UE IP packets transmitted from UE to the N6 interface in the uRLLC slice of the 5G network.
E2E mMTC Performance Measurement	Measure the End to End performance of Machine Type Communication in the mMTC slice of the 5G network.

2.4 Testing KPIs Overview

One of the first steps for creating Test Cases is identifying which KPIs need to be stressed by the Test. As it is evidenced by the SoA in Section 2.3.2, all the different documents use slightly different definitions of the KPIs. That can lead to confusion and impossibility to compare and align the test results. The same issue is also present across the different ICT-17 projects, and therefore, there is an ongoing effort in the 5G PPP TMV-WG to uniquely define the KPIs to be used across the projects.

Starting from the existing definitions, the KPI s are fine-tuned in their definition, specifying, for example, the measurement points or the processing formula for extracting the KPI from the measurements.

The effort is though not only a TMV-WG one since the 5G-VINNI project is actively working on definitions that will be eventually proposed for alignment inside the TMV-WG.

A formalization template, agreed within the TMV-WG, is presented in D4.1 Annex B.

2.4.1 Core KPIs Defined by 5G-PPP Test, Measurement, and KPIs Validation Working Group

The priority was, for the TMV-WG, to identify those technical KPIs that were supporting the 5G-PPP contractual KPIs validation [9]. The initial identified KPIs support mostly the following two contractual KPIs:

- **P1:** Providing 1000 times higher wireless area capacity and more varied service capabilities compared to 2010.
- **P4:** Creating a secure, reliable and dependable Internet with a “zero perceived” downtime for services provision.

In practice, those two contractual KPIs can be translated into a series of **Technical KPIs**, as displayed in Table 10.

Table 10 - Identified Technical KPIs

Type	KPI name	KPI measurement points	5G-PPP KPI Validated
SLA	Minimum Expected Upstream Throughput	UE is transmitting IP packets to the N6 interface.	P1
SLA	Minimum Expected Downstream Throughput	UE receiving IP packets from the N6 interface	P1

Type	KPI name	KPI measurement points	5G-PPP KPI Validated
SLA	Maximum Expected Latency	RTT of UE IP packets transmitted to the N6 interface.	P1, P4
SLA	Network Reliability	Transport packets are lost between the UE and the N6 interface	P4
SLA, Technology Validation	Quality of Experience	Measured at the UE side at application or application API level	P1, P4
Technology Validation	UL Peak Throughput	Single UE transmitting IP packets to the N6 interface.	P1
Technology Validation	DL Peak Throughput	Single UE receiving IP packets from the N6 interface	P1

The “Type” column remarks for which purpose the KPI is useful. The Service Level Agreement (SLA) KPIs are the ones used to validate if the service design can support the SLA agreed with the Vertical, and they can be used as well during the network monitoring phases to trigger alarm and network management actions. The Technology Validation KPIs are instead focused on providing the proof that 5G is delivering the promised performances, mostly peak ones.

2.4.2 KPIs in Focus During the Rel’0 Testing Phase

In 5G-VINNI Release 0 testing phase, not all KPIs will be tested, taking into consideration of the readiness of the 5G -related standards, and the progress of facility site implementation. Lists the focused KPIs out of the identified technical KPIs in Release 0:

Table 11 – Focused KPIs in Release 0

Type	KPI name	KPI measurement points	5G-PPP KPI Validated
SLA	Minimum Expected Upstream Throughput	UE transmitting IP packets to the N6 interface	P1
SLA	Minimum Expected Downstream Throughput	UE receiving IP packets from the N6 interface	P1
SLA	Maximum Expected Latency	RTT of UE IP packets transmitted to the N6 interface	P1, P4
SLA	Network Reliability	Transport packets are lost between the UE and the N6 interface	P4
SLA	Maximum Packet Delay Variation	UE transmitting IP packets to the N6 interface	

3 5G-VINNI Site Facilities Release 0 Testing Plan

3.1 Testing Scope

In Release 0, we will test the RAN, Transport, and Mobile Core Network performance and scalability, the NFVI/MANO interoperability and performance, and E2E network performance and scalability, based on the interests and readiness of each facility site and the test tools. On RAN and Mobile Core part, only NSA 5G New Radio (NR) with vEPC will be tested, 5G Core (5GC) and 5G NR in Stand-Alone (SA) mode are not in the scope of Release 0.

3.2 Expected Testing Periods

The expected testing periods for each facility site are as follows:

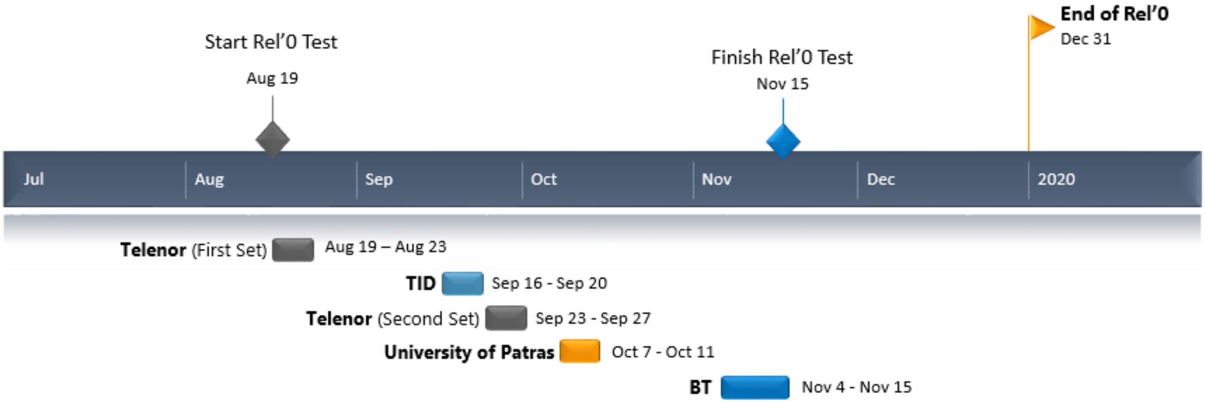


Figure 2 – Expected Testing Periods for Release 0

- Telenor: Aug 19 - Aug 23; Sep 23 - Sep 27

  According to the feedback from the Telenor facility site, part of the tests will be ready to be executed on Aug19, while some of the tests will be only ready for testing till Sep 16. So the test session is divided into two sets, one week each. For the second test set, to avoid conflict with TID testing period, the Telenor second test is planned to start from Sep 23.

- TID: Sep 16 - Sep 20

  As TID feedback that only NFV tests can be tested in Release 0, and it will be ready by Sep 16, the test period is planned for one week starting from Sep 16.

- University of Patras: Oct 7 - Oct 18

  Same with TID, University of Patras facility site will test only NFV for Release 0, and it will be ready by Oct 7. So, the test period is one week starting from Oct 7.

- BT: Nov 4 - Nov 15

  Due to a delay in the site infrastructure readiness in BT, the testing period is postponed to November. Two weeks testing period is planned.

- For all facility sites:

  Preparation of the test environment and VPN remote access are finished before the start of the testing.

3.3 Testing Focus: Expected Test Procedures and Test Cases

Based on the feedback from each facility site on the test environment planning, implementation, and interested test areas, below are the testing focus for each site, all mentioned test cases are defined in D4.1 Annex D:

- BT:
  - Unit Tests for NFVI/MANO: 7.4.1.1 - 7.4.3.4
  - Sub-module Tests for NFV Performance Test: 8.6.1 - 8.6.4
  - Sub-module Tests for RAN: 9.6.1
  - Sub-module Tests for vEPC NSA: 10.6.1 - 10.6.2
  - E2E Tests: 11.6.1 - 11.6.6
- Telenor:
  - Unit Tests for NFVI/MANO: 7.4.1.1 - 7.4.3.4
  - Sub-module Tests for NFV Performance Test: 8.6.1 - 8.6.4
  - Sub-module Tests for RAN: 9.6.1
  - Sub-module Tests for vEPC NSA: 10.6.1 - 10.6.2
  - E2E Tests: 11.6.1 - 11.6.6
- TID:
  - Unit Tests for NFVI/MANO: 7.4.1.1 - 7.4.3.4
  - Sub-module Tests for NFV Performance Test: 8.6.1 - 8.6.4
- University of Patras:
  - Unit Tests for NFVI/MANO: 7.4.1.1 - 7.4.3.4
  - Sub-module Tests for NFV Performance Test: 8.6.1 - 8.6.4

4 Test as a Service Development

In this Section it will be clarified the main development work carried out in Task 4.2. The main goal of the task is to deliver to the project an integrated set of tools that enable and facilitate the testing cycles of the 5G-VINNI infrastructure.

4.1 Motivation

Testing tools have always been an essential part of network deployment engineers and, to a very limited extent, to network management engineers. The advent in 5G of softwarization brings along an extreme number and variety of parameters that can affect the overall network performance. Constant software updates, optimization of NFVI parameters are only two examples of the huge number of possible use cases. Testing becomes then an even more important tool in the hand of network management and operations engineers for guaranteeing the SLAs contracted by the MNOs with their Vertical customers.

TaaS plays then an important role in reducing the effort that the MNOs engineers need to put in testing the 5G infrastructure and components. By Simplifying the testing operations and providing an interface to connect to the CI/CD pipelines of the MNOs' Network Operations (NetOps), TaaS promises a stable performance delivery while maintaining under control (or even reducing) the OPEX. TaaS is expected to become an essential component of the Zero-Touch philosophy that is currently tackled in standards like ETSI Zero-touch Network and Service Management (ZSM) [11] or open source communities such as OPNFV [12] and ONAP [13].

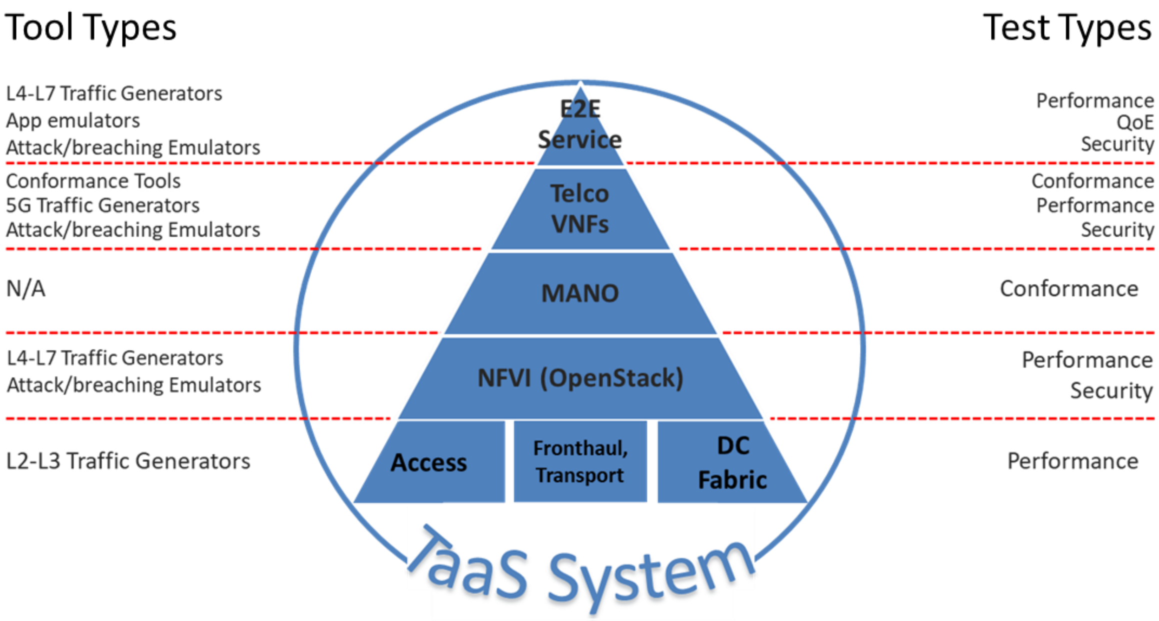


Figure 3 - TaaS system overview from a tooling perspective

But what does TaaS practically do? TaaS is an automation and interfacing layer that allows connecting all the T&M tools needed for validating and verifying the 5G system, from the individual components up to the E2E service. The automation makes possible to abstract the complexity with a series of either standard or custom Test Cases.

In 5G-VINNI, the TaaS system is designed in line with the 5G-VINNI architecture principles outlined in D1.1 [14] and further explained in the following sections. It will make use of state-of-art products coming from partners' portfolios, prototypes, and open source components to offer, not only to the project partners but also to the incoming vertical customers such as the ICT-19 projects a tool for verifying network and application performance.

4.2 Testing Architecture

In this Section is presented the Testing Architecture that defines the TaaS system developed in the framework of the 5G-VINNI project. The architecture was drawn based on the current requirements of having a cloud-based system that could offer the use of testing components to the MNOs, the vertical customers, and CI/CD pipelines.

4.2.1 Alignment with WP1

The general architecture has been introduced and discussed in D1.1 [14] and it can be seen in Figure 4. The main aspects of the architecture are related to the different components present in the system and how they interact together at a generic level.

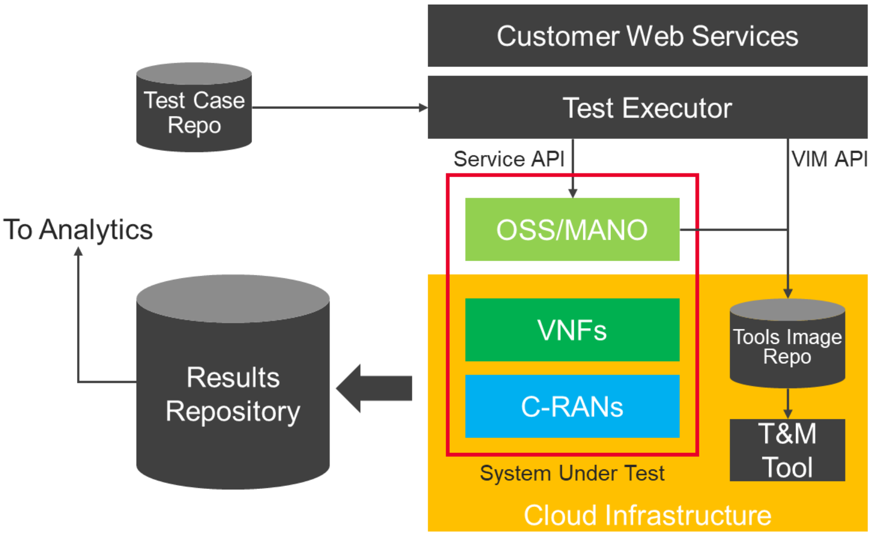


Figure 4 – Testing architecture introduced in D1.1 [14]

To be noted is the presence of a Test Cases Repository storing the execution scripts needed to perform the different types of tests in the 5G-VINNI infrastructure.

4.2.2 Components

In the architecture are several present components, of which some can be partially mapped with the elements present in the NFV architecture. The following components are included:

- Customer Web Service: the service is a web application that allows the human user to define, create, and execute test campaigns. The application also exposes an API that can be contacted by other applications (e.g., CI/CD pipelines) for consuming testing services. From here on it will be denoted as Automation Solution.
- Test Case Repository: it is a logical component used for storing and managing the available test scripts needed to execute the tests and their configuration.
- Test Executor Service: the service is the actual component coordinating and performing the test on the selected SUT. It oversees managing and coordinating all the different tools required to perform the tests.
- VIM: it is a needed third-party actor in the system. It allows the Test Executor to deploy the needed infrastructure and tools to perform the tests. In the NFV terminology, it corresponds to the VIM, and in 5G-VINNI, the most used orchestration system is Heat/OpenStack.

- Traffic Generation Tools: these are the actual T&M tools used to probe and stress the 5G infrastructure. They are needed for having granular control over the type and quantity of traffic present in the network, or for performing specific operations difficult to do with real application traffic (e.g. Latency and QoE measurements).
- Results Repository: this is a needed element for storing the results generated by the tools, but also essential status and configuration elements of the network, e.g. VNF logs. It is a logical element, since its implementation might be rather complex given the heterogeneity of the data and their quantity. It could be implemented e.g. with Big Data solutions. It provides the permanent data storage for further analytics applications.

4.3 Traffic Generation Components Overview

4.3.1 IxLoad VE

IxLoad is a test tool enabling the generation of realistic traffic to load and assess the performance of a network. IxLoad comes in 2 versions, the “physical” version of IxLoad, using a hardware platform to generate the requested traffic and IxLoad VE which is a Virtual Edition allowing to test the cloudified network. Both versions are very similar in functionality and use. In the frame of 5G-VINNI, IxLoad VE will be the tool integrated in the testbed.

IxLoad VE enables functional and performance testing to validate user QoE in virtual networks. IxLoad VE emulates web, video, voice, storage, VPN, wireless, infrastructure, and encapsulation/security protocols to create realistic scenarios. It loads the network with a set of streams as it would be used in normal production environment. A modular system design allows IxLoad VE to elastically scale with the 5G-VINNI infrastructure, while real-time QoE metrics let the testing user drill down to quickly identify network degradations and isolate breaking points. Such features will be relevant in the frame of 5G service deployment on the 5G VINNI platform. Since IxLoad is as easy to use as it is effective, we anticipate 5G VINNI vertical users to be able to benefit from the complete end-to-end service validation without requiring the Vertical user to have a deep protocol expertise.

IxLoad VE comes with a wide variety of fully stateful protocols to emulate a complete multi-play user environment. It delivers a sophisticated subscriber modelling capability that enables testers or experimenters to replicate the dynamic nature of subscriber behaviour (data traffic, connections/de-connections etc…). Supported protocols are web traffic (http, TLS…), File transfer (FTP…), email traffic, storage systems (SMB, NFS, cloud…), IPTV and OTT systems, voice traffic, TCP, UDP etc.

The tool scalability is able to generate city scale traffic. In the frame of 5G-VINNI it will enable large scale testing and emulation of large settings across the different operator domains. The system will come preconfigured with a set of predefined scenarios which can be adapted to the need of the Vertical experimenter. Along with comprehensive network support, advanced test timelines, and automatic goal-seek fine-tuning to meet test objective, enables users to comprehensively assess the performance of their service delivery networks and determine application-level performance with this solution.
IxLoad VE will be integrated and automated in the frame of 5G-VINNI to run automatically from OpenTAP.

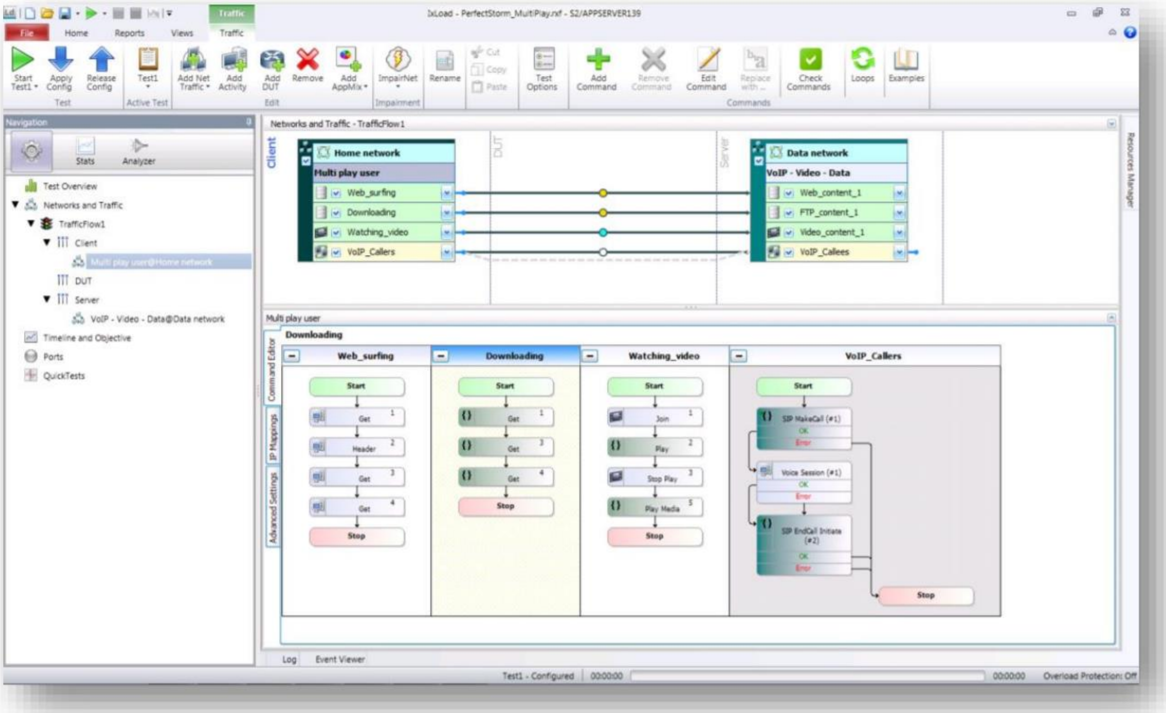
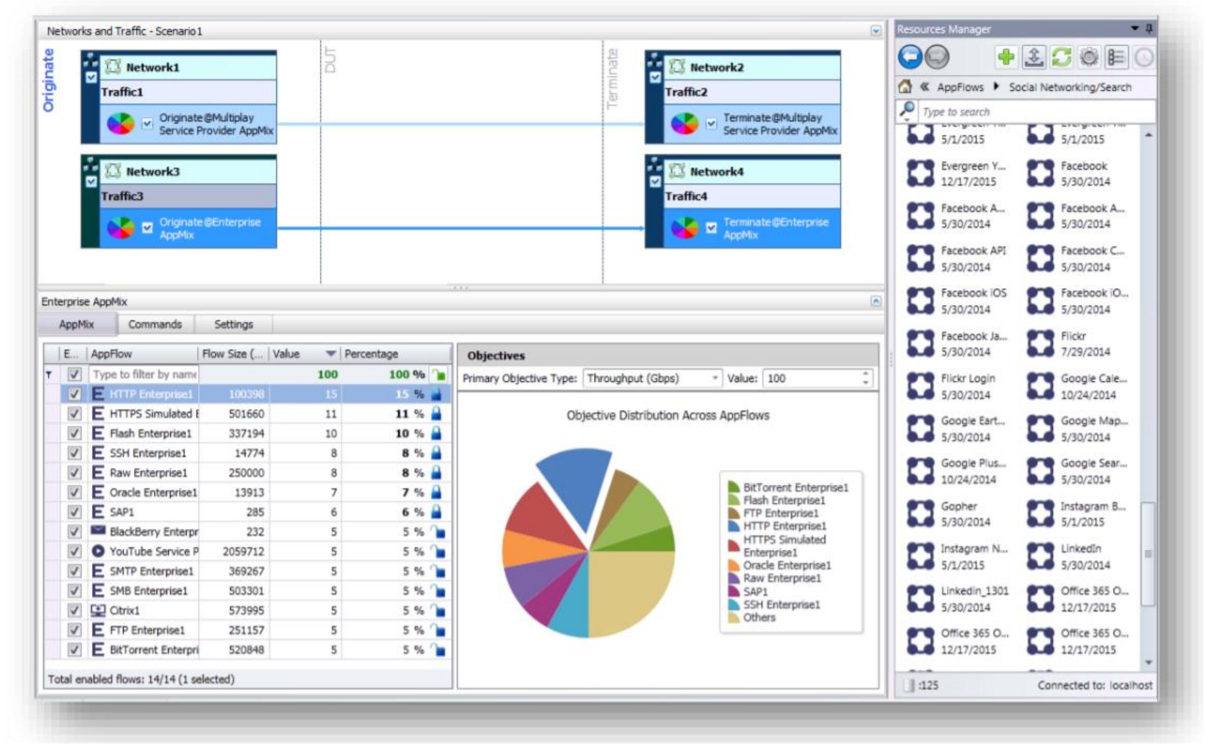


Figure 5 – User interface for multi-play user environment with realistic subscriber modelling

To model real-world device and network load, IxLoad uses an approach called subscriber modelling that enables users to define serval parameters reflecting potential traffic scenarios. The tool allows to emulate multiple application services per subscriber or user, including service sequencing, model traffic usage over time for each application with their respective timeline, simulate the temporal nature of subscribers connecting and disconnecting from the network in a dynamic manner and control usage profile—particular websites visited, e-mail servers used, transfer sizes, channel-changing behaviour, streaming content, protocol options, etc.



E...	AppFlow	Flow Size (...	Value	Percentage
☑	Type to filter by name		100	100 %
☑	HTTP Enterprise1	100398	15	15 %
☑	HTTPS Simulated E	501660	11	11 %
☑	Flash Enterprise1	337194	10	10 %
☑	SSH Enterprise1	14774	8	8 %
☑	Raw Enterprise1	250000	8	8 %
☑	Oracle Enterprise1	13913	7	7 %
☑	SAP1	285	6	6 %
☑	BlackBerry Enterpr	232	5	5 %
☑	YouTube Service P	2059712	5	5 %
☑	SMTP Enterprise1	369267	5	5 %
☑	SMB Enterprise1	503301	5	5 %
☑	Citrix1	573995	5	5 %
☑	FTP Enterprise1	251157	5	5 %
☑	BitTorrent Enterpri	520848	5	5 %

Total enabled flows: 14/14 (1 selected)

Figure 6 - Library containing different traffic and protocols

4.3.2 Hawkeye

The purpose of Hawkeye is to asses an analyse the performance of traffic between different endpoints. Hawkeye offers a single Web based interface which scales to monitor hundreds of endpoints anywhere in the network. The endpoint is a device or node that is connected to the LAN or WAN and accepts communications back and forth across the network. In a traditional sense, an endpoint can be a modem, hub, bridge, or switch but here it is a test device all controlled out of the Hawkeye software.

Endpoints are deployed at different sites e.g. Telenor, BT, secondary 5G VINNI facility, within different location inside the 5GVINNI cloud. They generate synthetic traffic to inject into and receive from the live network, reproducing real users traffic patterns. The synthetic traffic is processed for objective quality metrics and used for insights.

Endpoints are available traditionally in both hardware and software formats. In the frame of 5G VINI, only software endpoints will be deployed.

Hawkeye allows 2 main type of tests: the node-to-node tests and the real service tests.

In node-to-node tests, one endpoint generates synthetic traffic and sends it to a second endpoint on a live network. An example would be a branch office to the main office. The receiving endpoint (node) is able to relay information about the quality of the connection to the management interface - where results are displayed in real-time live dashboards. Multiple node-to-node tests can be setup, considered a mesh, where an endpoint can serve both as the origination point of synthetic traffic as well as a recipient of traffic from other endpoints. 5G VINNI will mainly rely on this kind of test.

In real service tests, an endpoint generates synthetic traffic and sends it to servers or network equipment – it will process the response it receives from the network into objective KPIs to the management interface, where again it is displayed real-time. In the 5G VINNI context, this would mean that an endpoint would be installed in the Vertical user infrastructure.

An endpoint relies on a test library and can generate different kind of tests such as:

- Verify the quality of experience (QoE) for the end users. Emulate Voice and video traffic, covering services like standard Voice over IP (VoIP), unified communications (e.g. Skype for business), or other video conferencing;
- Make sure the users can access business critical applications through different firewalls systems;
- Qualify and maintain network SLAs with diagnostic tools for IP Transport testing - Assess layer 3 network performance indicators (loss, jitter, delay)
- Validate network services like DNS and Traceroute by quickly diagnosing with ping and resolving any issues
- Qualify the real capacity of your circuits. Test TCP and UDP traffic throughput over wired or wireless connections. Synthetic traffic can be generated up to line rate

Finally, all the traffic between endpoints can be analysed and visualized in real time via the dashboard as depicted in the figure below.

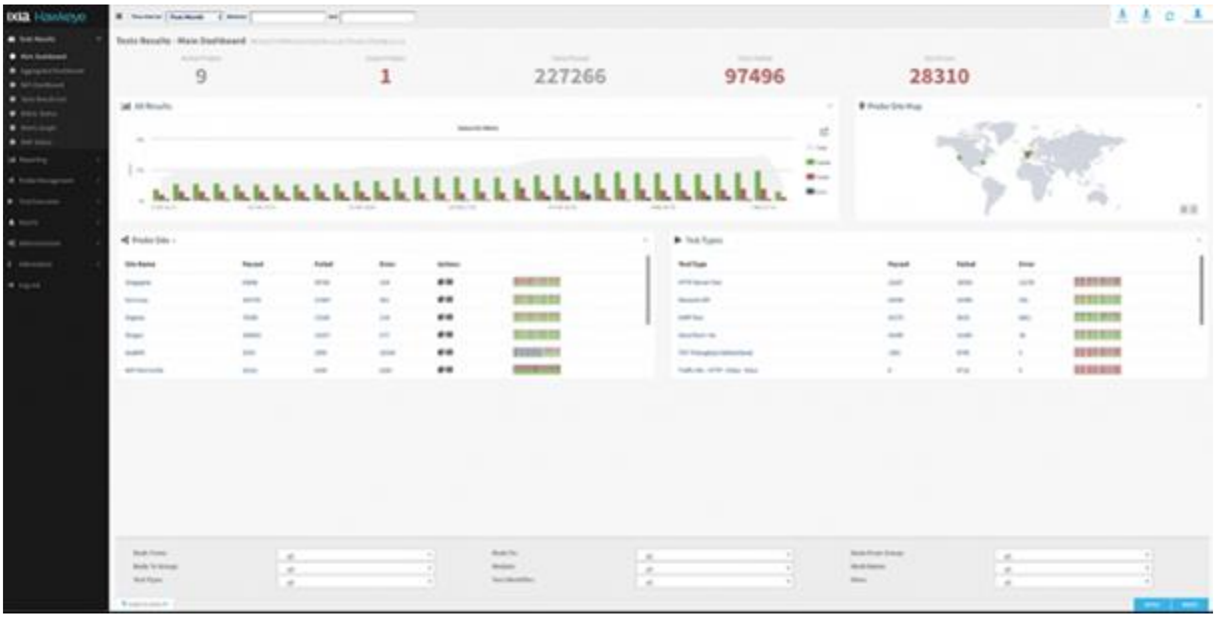


Figure 7 - Hawkeye real time dashboard

4.3.3 Yardstick

Yardstick is an Open Source project developed by the OPNFV community. The goal of the Yardstick Project is to verify the infrastructure compliance when running VNF applications.

NFV Use Cases described in ETSI GS NFV 001 show a large variety of applications, each defining specific requirements and complex configuration on the underlying infrastructure and test tools. The Yardstick concept decomposes typical VNF work-load performance metrics into several characteristics/performance vectors, which each of them can be represented by distinct test-cases.

The project's scope is to develop a test framework, test cases and test stimuli.

Yardstick allows to:

- Decompose VNF work-load performance metrics into a number of characteristics / performance vectors, identifying and categorizing the metrics related to characterization of the infrastructure, develop test case examples to realize the metrics;
- Enable verification of more complex test cases by developing functionality to run: parallel testing, inject fault, test multiple topologies, test scenarios.

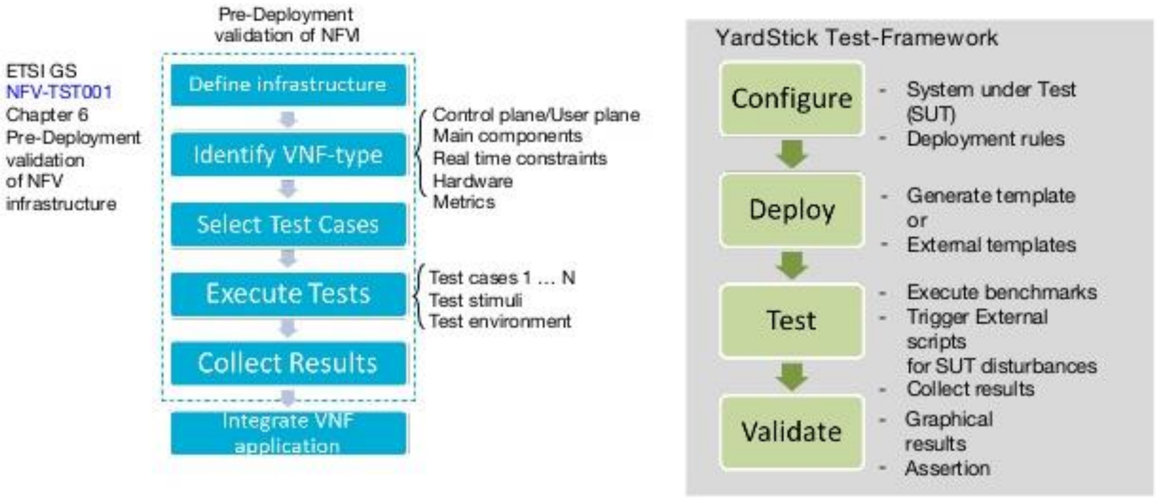


Figure 8 - example of System level test with Yardstick

The methodology used by the Project, to verify the infrastructure form the perspective of a VNF, is fully aligned with ETSI NFV-TST 001. Visualization of the test results is performed using Grafana as shown below:

Figure 9 - Grafana result visualization of Yardstick test results

More information can be found on [15].

4.3.4 MONROE Network Monitoring Node

The MONROE Mobile Broadband (MBB) measurement platform was originally developed within the EU Horizon 2020 project MONROE[1], which was in turn a Europe-wide extension of the NorNet Edge (NNE) project. NNE is a national project that is funded by the Norwegian government and the Norwegian Research Council to continuously monitor the reliability and performance of commercial Norwegian MBB operators. SRL has been successfully coordinating and running NNE since 2012. Following the conclusion of MONROE, SRL and then SimulaMet have taken over the development of the measurement platform by integrating it back into NNE.

The measurement platform comprises three parts: the measurement node, experiments scheduler and backend. In the following, we describe these components.

[1] https://www.monroe-project.eu/

4.3.4.1 The measurement node

Figure 10 shows the current measurement node, which is a single board computer that is based on PC Engines APU2D4 system board[2].

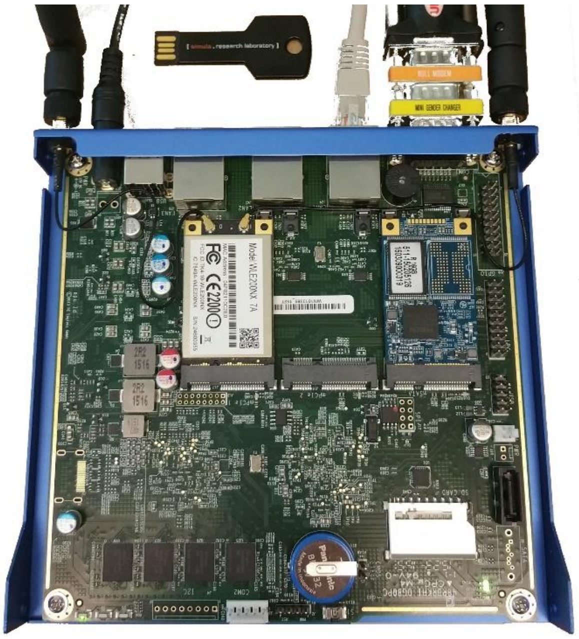


Figure 10 - MONROE measurement node - Hardware

The board comes with the following configurations:

- 1 GHz 64-bit quad core AMD Geode APU.
- GiB RAM.
- 16 GiB SSD.
- Three miniPCIe slots. Two of them can be used for connecting cellular modems.
- 2 Gigabit Ethernet channels and 2 USB 3.0 ports.

The current node connects to LTE networks using Sierra Wireless LTE Cat 6 modems.

Every MONROE node runs several background processes for managing node lifetime, network connectivity and running experiments. Figure 11 gives an overview of various processes that run on the MONROE node. Overall, we have two main components: the Core software and experiments. The Core software is responsible for managing the node’s life time and network devices. Furthermore, it allows various experiments and measurements access to available network interfaces including streaming metadata. It comprises three parts: node management software, node maintenance software and experimentation enablers. **Node management software** includes: *Device Listener*, which detects, configures and connects network devices, *Routing Daemon*, which acquires IP addresses via DHCP and set up routing tables, and *Network monitor*, which monitors state of connectivity and attempts to timely repair failing connections. Furthermore, the node operates behind a firewall.

[2] https://pcengines.ch/apu2d4.htm

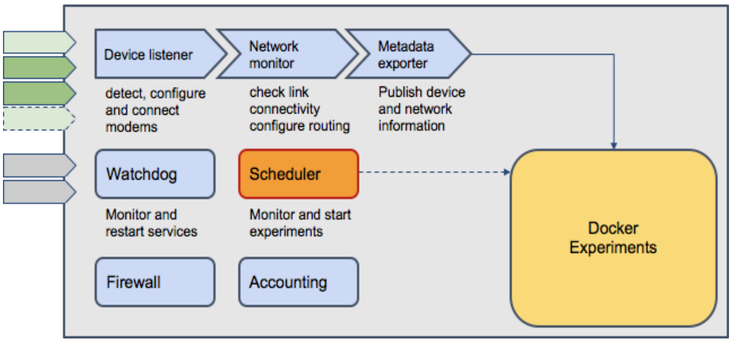

Device listener
detect, configure and connect modems
Network monitor
check link connectivity configure routing
Metadata exporter
Publish device and network information
Watchdog
Monitor and restart services
Scheduler
Monitor and start experiments
Firewall
Accounting
Docker Experiments

Figure 11 – Software processes on the node

The **node maintenance software** involves a system-wide watchdog that monitors the state of all node management and experimentation enablers components and initiates mitigating actions upon failure.

The experimentation enablers include: *the scheduling client*, which periodically communicates with the experiment scheduler in order to deploy new measurements and manage running ones, and services for existing experiments, which involve components for allowing experiments that run inside the MONROE namespace access to network devices that are available in the Host Network namespace. It also involves a metadata broadcasting service that relays network connectivity metadata e.g. RSRP, RAT and current serving cell to experiment containers.

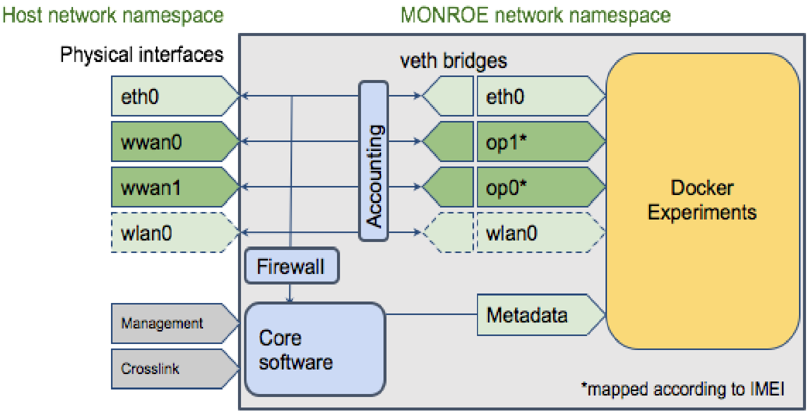


Figure 12 – A high level overview of networking configurations

This service runs continuously in the background and relays metadata via ZeroMQ in JSON format. Finally, it includes a synchronization service to copy measurements results to the backend. This service is built on top of rsync and runs continuously to transfer newly generated data files.

The experiments are packaged as docker containers that run in a namespace that is separate from the Host network namespace where the Core software runs. *Networking-wise,* we map physical interfaces in the Host network namespace to virtual interfaces in the MONROE namespace (see Fig.3). More specifically, fgor each physical interface that the network-listener detects as available, we create a virtualized ethernet, veth, interface pair, and move one end to the MONROE namespace. We then add routing rules in the network namespace to allow routing by interface. In order to allow the network devices in the host namespace to communicate with the ones in the MONROE network namespace, we define an internal Network Address Translation (NAT) function. We use iptables NAT masquerading rules in the host namespace to configure the NAT function. Finally, we add the corresponding routing rules to map each veth interface to the correct physical interface.

4.3.4.2 Experiments scheduler

The experiment scheduling system consists of two parts: the scheduler and the scheduling client.

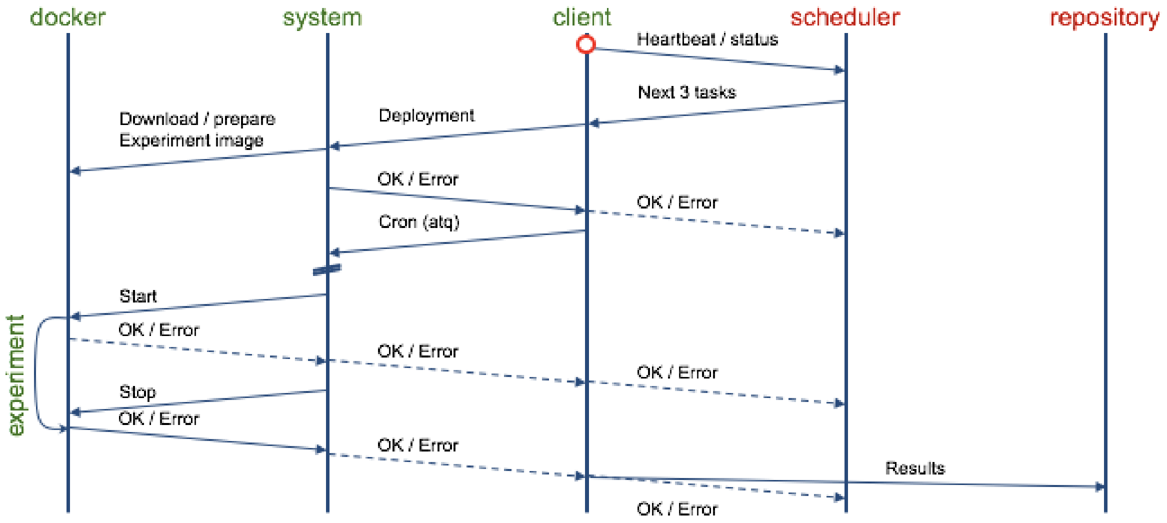


Figure 13 – Experiments scheduling workflow

The *scheduler* is hosted in the measurements platform backend and is responsible for distributing measurements to nodes and monitor their deployment and lifecycle. For running a new experiment, the user needs to start by preparing a docker container that packs the experiment workflow. Once the container is prepared, it can be uploaded to the scheduler via a web-interface or a command-line client. At this point the user can specify which nodes to run the experiment on, which network interfaces and respective timestamps.

The *scheduling client*, which runs on each node, periodically checks with the scheduler for new experiments, pulls them and schedule them. It also keeps the scheduler informed of the outcome of various scheduling steps. Figure 4 shows the scheduling workflow and signaling. The scheduler is written using Python and SQLite and is open source.

4.3.4.3 Backend

The measurements platform backend comprises measurement servers, management servers and data repository. We operate and control all servers that are needed for performing E2E measurements. For example, a UDP echo server is operated for responding to UDP echo requests that are initiated by experiments on the nodes and an HTTP server is maintained for supporting HTTP download tests. The management servers are responsible for updating the node base software and monitoring nodes health. The data repository comprises a server that is the other end of the rynsc process, a tool for parsing measurements and inserting them into a user-specified database and a database. We are currently using MySQL as a backend database, but the parser can interface with any desired database.

MONROE is an E2E platform for measuring the performance of mobile broadband networks from the user perspective. The platform also allows for deploying tailored experiments and evaluating novel protocols.

4.4 Automation Solution

In this section it will be described the Architecture of the Automation Solution. Founded on the architectural principles described in Section 4.2.1, the development architecture and several implementation details will be described.

The purpose of the Automation solution is twofold:

- Provide and interface for the user to create and manipulate test scripts and test campaigns (collection of the scripts to be executed).

- Allow the user or a third system (e.g. a CI/CD pipeline) to execute test cases and test campaigns.

The execution of the test is also a quite complex set of actions to be performed by the Automation Solution. There are several steps that the system needs to follow:

1. Instantiate the test infrastructure in the target environment (cloud or a lab machine).
2. Configure the test system and deliver the needed scripts to be executed, containing the test methodology to be used.
3. Monitor the execution of the tests and provide feedback to the user about the currently executing test cases and test campaigns.
4. Release the testing infrastructure resources and provide a clean slate environment for the next tests.
5. Provide the user access to a way of visualizing or analysing the results.

While the steps 1 to 4 and the result collection are immediate needs for the 5G-VINNI project, the ease of visualization or the advent of smart analytics will be postponed to later stages.

4.4.1 Cloud-Native, Microservice-based Architecture

Before starting the development of the system, several requirements have been collected to make the proper design decisions on how to implement the components. The requirements are listed in Table 12.

Table 12 – Definition of requirements for the Automation Solution

Requirement	Need/Explanation
Linux-capable	Stable, open source, and reduces the need of third-parties licensed products
Cloud-Native	Can be deployed in a single machine or scale at cloud level
Founded on cutting-edge technologies	Relies on stable, mature, open source technologies that are easily upgradable
Webified	Is designed to be multi-platform and run remotely
Highly modular	Is resilient due to micro-service architecture
Scalable	Can be expanded by adding micro-services, and each service is designed to be modular

The choice was clear since idea inception that the architecture had to be based on cloud-native principles and a system of interconnected microservices. The microservices can be subdivided in:

- Basic service providers (service discovery, authentication, logging)
- Stateful systems (typically databases or logical volumes)
- Business logic services (the actual working components implementing the business functionalities)

An overview on how all these categories are interconnected and interplaying together in the Automation Solution can be seen in Figure 14.

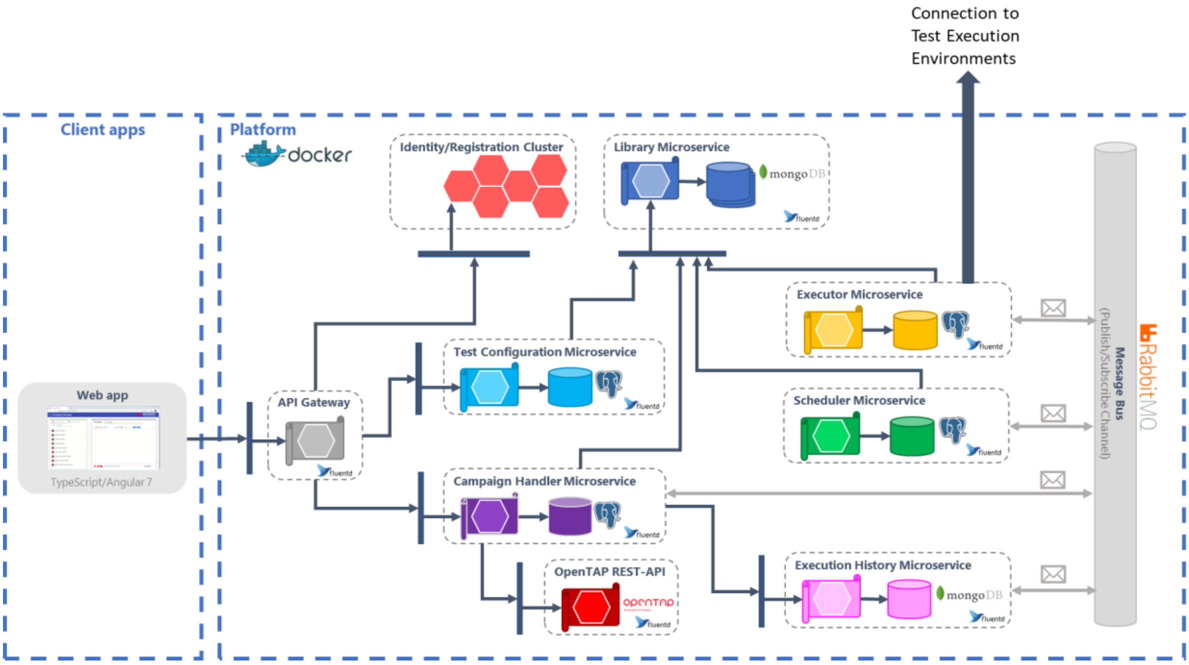


Figure 14 – Cloud-native, Microservice-base system architecture

The different microservices are joined together via synchronous or asynchronous communication. The former is typically implemented via a REpresentational State Transfer (REST) Application Programming Interface (API), while the latter assumes the form of a so-called "message bus" or "message queues".

All the microservices are composed by two logical elements: a stateless (business) logic, and a stateful component used to keep track of temporary information without leaving it to the mercy of the presence in memory. The stateful element is typically a database, and for 5G-VINNI the current choice is PostgreSQL.

The wide use of databases instead of storage volumes is due to the increased reliability and scale capabilities intrinsically offered by such systems. As a matter of fact, modern databases offer possibility of replication, distributed sharding, and scale as part of the server components, without the need for the user to re-implement it.

4.4.2 Microservices Overview

In this section, a deep dive into the different components of the architecture is presented. The purpose of the section is to explain the individual building blocks, being those basic services, stateful systems, or business logic elements.

4.4.2.1 Client Web App

The user will be able to access the system capabilities via a client web app. The app is a very thin Graphical User Interface (GUI) layer, completely decoupled from the core functionality of the system. This allows both sides (GUI and system) to be individually modifiable and upgradable without affecting each other, as long as the API between the two is maintained constant. Another advantage of such a de-coupling, is the possibility of creating multiple, customer specific GUIs for the same system, with only a minor code impact.

The GUI currently developed for the 5G-VINNI project is developed in Angular 7 and Typescript, modern technologies for web applications. While in development and small environments it is possible to run it on NodeJS, for production-grade deployments it would be recommended to move towards Nginx.

4.4.2.2 API Gateway

Microservice-based systems are extremely effective for both development speed but also reliability and maintenance. They are nevertheless vulnerable to possible cascade fails if attacked or if the web client is not properly developed. For this reason, it is always wise not to expose all the services directly to the customer but provide a firewalling mechanism that can filter and protect the core services from the external world. API Gateways and Reverse Proxies are utilized in microservice-based architectures with this purpose in mind. Another advantage offered by an API Gateway is offering a single-entry point to the web client, making its development easier.

While an API Gateway is foreseen in the 5G-VINNI Automation Solution architecture, its implementation is moved to later stage, when the core business logic is implemented and mature. An initial investigation on the available solutions in the open source communities has been performed, and it has temporarily been concluded that a possible choice for this role could be Kong.

4.4.2.3 Identity/Registration Cluster

A set of microservices are needed to perform user and identities management. From a final user's perspective that is translated into being able to log into the system, maintain a profile, being assigned to a role with specific access rights to features and functionalities. Nevertheless, in microservice-based architectures these services have a hidden but extremely important function, that is securing the system through an authorization-based transaction mechanism.

Being each microservice an independent network element, makes the system vulnerable to different exploitation types, including exposing protected information to unauthorized parties. For this reason, each transaction that occur in the system needs to be authorized. Each microservice needs to be able to verify the identity of the other microservice it is talking to, if the user that such microservice is acting on behalf of is authorized to see or use a certain resource, and so on.

Within the 5G-VINNI context, open source services are currently investigated: Ory Hydra as OpenID Connect and OAuth2 server, Ory Keto for Access Control, and Keycloak for identity and access management.

4.4.2.4 Library

The Library microservice is designed to be a multi-purpose information storage system. The main purpose is allowing the system to store in a structured and systematic fashion all the different objects that that either useful to the user or the system. The library owes its name from the architectural design behind the service. The Library provides a catalogue that can be browsed to identify which types of information the Library contains (the so-called "shelves"). Each shelf is a document-oriented database containing a different type of information. This allows high flexibility in the type of information to be stored, but also in how quickly new types can be added.

As an example, the Library is the service that will implement the Test Case Repository foreseen in the architecture depicted in Figure 4.

The current choice for the document-oriented database is MongoDB that allows a two-level element management: via individual databases, and with different collections within each database.

4.4.2.5 Test Configuration

The test configuration is a very specific business logic service that allows the configuration of the Test Systems and the System Under Test. Complex testing systems require a high number of parameters and interconnections to function properly. This service is specifically designed to facilitate such configuration, allowing the user to save configurations ready to use on a click of a button via the test execution GUI.

4.4.2.6 Asynchronous Message Bus

A microservice-based system can be implemented via REST communication only. This is useful when "online" synchronous communication is needed, and there is always a one-to-one communication involved. When systems need to be allowed to scale each microservice independently, and there is a one-to-many or a many-to-one communication, it might be wiser to use an asynchronous, queue-based communication method.

Within the context of 5G-VINNI, RabbitMQ is currently investigated as a solution. Open source technology, it is widely used for control channel communication due to its advanced features and stability. A possible use of Kafka and Zookeeper might be considered due to their higher message handling capacity for result collection.

4.4.2.7 Campaign Handler

The Campaign Handler is the core backend element of the business logic that allows the interaction with the user. As a matter of fact, users should be able to collect together different sets of Test Cases. Such a collection is hereby called a Test Campaign. This service allows the user to compose, save, and manage Test Campaigns and individually run them. While the service is designed to be as generic as possible, in the 5G-VINNI implementation it relies on the use of OpenTAP (explained in Section 4.4.3) as executor of individual Test Cases and it therefore requires the support of an OpenTAP validation service for every parameter change that is performed by the user.

4.4.2.8 OpenTAP

OpenTAP (Section 4.4.3) is assumed to be the underlying executor of individual test case. Each test case included in a Test Campaign should be able to be parametrized according to the specific needs of the user. Avoiding conflicts and errors when parameters are touched is of paramount importance.

The Campaign Handler loads the Test Cases in OpenTAP as soon as it is added to the campaign. Whenever a parameter is modified in the Campaign Handler, such modification is propagated to the Test Case loaded in OpenTAP, where the test methodology and logic is implemented, and the modification is validated. The validation is then pushed back to the Campaign Handler that can immediately (and visually) communicate it to the user.

4.4.2.9 Scheduler

The Scheduler microservice is at the core of the Automation business logic of the system. Whenever a Test Campaign needs to be executed (immediately or at a specified date and time), the Campaign Handler pushes a Scheduling Request to the Scheduler via the message bus. The Scheduler processes all the requests (coming eventually from multiple users as well) and send execution requests to the Executor service.

4.4.2.10 Executor

The role of the Executor microservice is to make sure that an execution system is available in an execution environment chosen by the user. To make a clear example, the user might select to run a Test Case on premise in the local OpenStack environment, or having the execution in a public cloud such as AWS, as displayed in Figure 15.

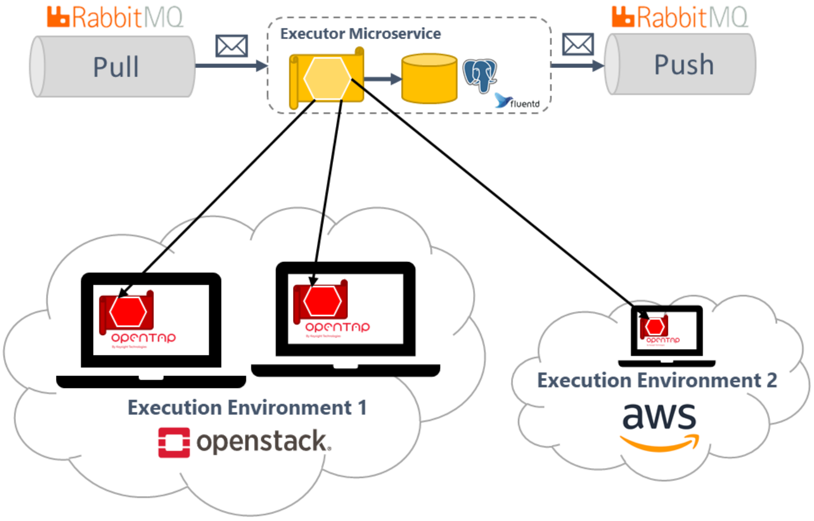


Figure 15 – The Executor manages resources in different execution environments

The Executor keeps track of the available resources, and whenever not present, instantiate the Test Executor in the target environment. For 5G-VINNI, it has been decided that the rest of the Test Infrastructure (e.g. testing tools and SUTs) is either permanently instantiated or instantiated by each individual Test Case script.

4.4.2.11 Execution History

The execution of a Test Campaign and the included Test Cases is a complex process that involves multiple actors performing portions of the process. Furthermore, the reporting and updating of the execution status can even come from a remote environment where the Test Case is executed (see Executor service in the previous Section for explanation). To facilitate the tracking of the execution status and the further analysis of the results, an intelligent collection agent capable of methodically collect all the different execution steps is needed. The Execution History service performs this job, allowing the user to quickly list the executed Campaigns and Test Cases via a unique and simple query.

4.4.2.12 Telemetry Components

Another essential structural component of a cloud-native architecture is the set of telemetry components that log, trace, and collect health metrics of the systems. A microservice-based architecture has several structural elements, together with the business logic ones, that require observation and logging in order to be able to verify the health status of the system and enable administrator to verify possible issues and bottlenecks. The telemetry of microservice-based architectures needs to be very heterogeneous and complex as it can be seen in Figure 16.

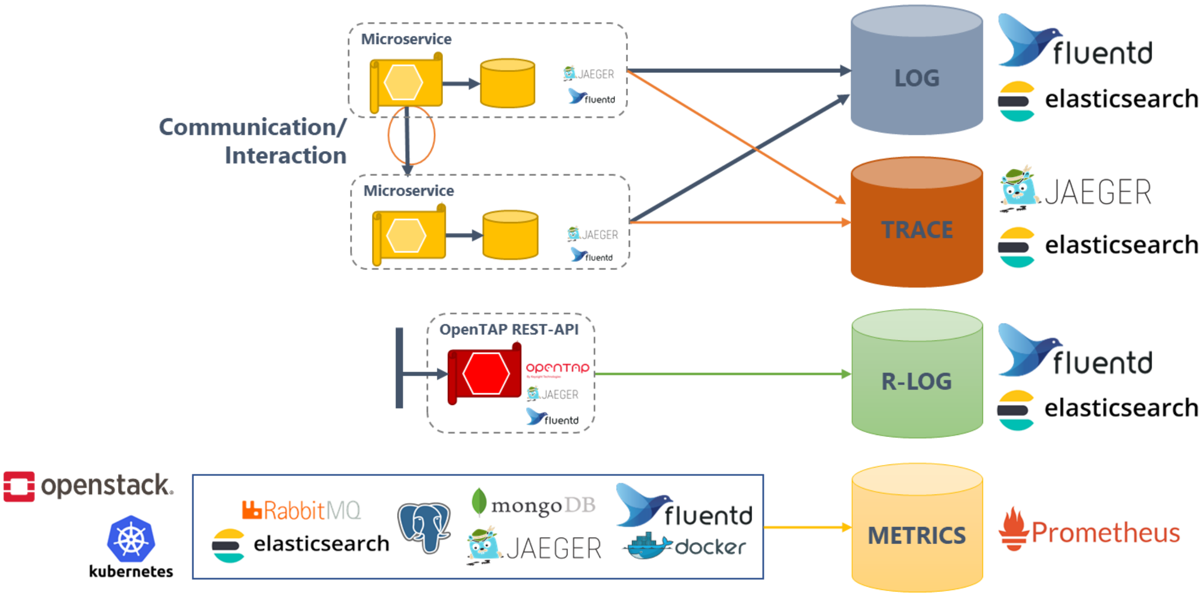


Figure 16 – overview of logging and telemetry components foreseen for the Automation system

There are different categories within the telemetry domain that needs to be taken into account when designing the telemetry architecture. A more detailed overview is highlighted in Table 13.

Table 13 – Overview of Telemetry categories and components

Category	Description	Solution	Architecture	Storage
Logs	System logs of the different microservices	Fluentd	Fully distributed	Elasticsearch
Traces	Timing and compositions of the multi-service transactions	Jaeger	Distributed agent and collector server	Elasticsearch
Result Logs	Logs related to the execution of the Test Cases	Fluentd	Fully distributed	Elasticsearch
Metrics	Measurements related to component or platform performance	Prometheus	Fully distributed	Prometheus

The table contains not only an overview of the logical subdivision of the collected information (the “Category” column), but also the currently foreseen component that will be integrated into the Automation solution in future releases.

4.4.3 OpenTAP: Test Execution Service

OpenTAP is an open source software solution, developed originally by Keysight, for fast and easy development and execution of automated test and/or calibration algorithms. These algorithms control measurement instruments and possibly vendor-specific Devices Under Test (DUTs). By leveraging the features of C#/.NET and providing an extendable architecture, OpenTAP minimizes the code needed to be written by the programmer.

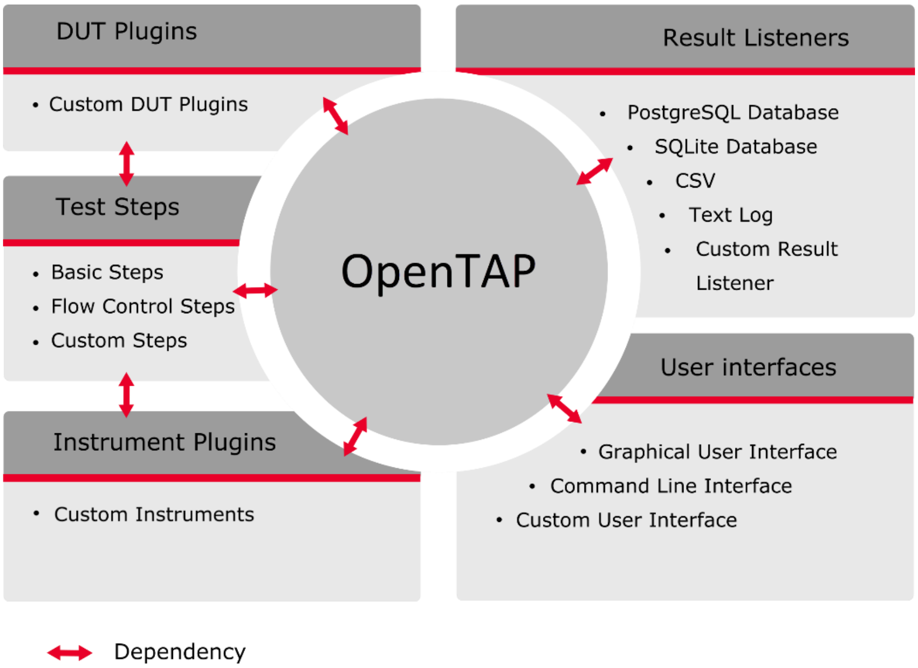
OpenTAP offers a range of functionality and infrastructure for configuring, controlling and executing test algorithms. OpenTAP provides an API for implementing plugins in the form of test steps, instruments, DUTs and more.

OpenTAP consists of multiple executables, including:

- OpenTAP (as a dll)
- Command Line Interface (CLI)
- Package Manager

Steps frequently have dependencies on DUT and instrument plugins.

The illustration below shows how OpenTAP is central to the architecture, and how plugins (all the surrounding items) integrate with it.



The OpenTAP assembly is the core and is required for any OpenTAP plugin. The most important classes in the OpenTAP are: TestPlan, TestStep, Resource, DUT, Instrument, PluginManager and ComponentSettings. OpenTAP also provides an API, which is used by the CLI, and other programs like an editor GUI.

If a graphical user interface is needed you can download the Keysight Test Automation Developer's System (Community or Enterprise Edition). It provides you with both a Software Development Kit (SDK) as well as an Editor GUI

An essential feature of OpenTAP is its flexible architecture that lets users create plugins. OpenTAP plugins can be any combination of TestStep, Instrument, and DUT implementations. Other OpenTAP components such as Result Listeners and Component Settings are also plugins. By default, OpenTAP comes with plugins covering basic operations, such as flow control and result listeners.

Plugins are managed by the PluginManager, which by default searches for assemblies in the same directory as the running executable (the GUI, CLI or API). Additional directories to be searched can be specified for the GUI, the CLI and the API. When using the API, use the PluginManager.DirectoriesToSearch method to retrieve the list of directories, and add any new directories to the list.

A test plan is a sequence of test steps with some additional data attached. Test plans are created via the Editor GUI. Creating test plans is described in the Graphical User Interface Help (GuiHelp.chm), accessible within the Editor GUI. Test plan files have the .TapPlan suffix, and are stored as xml files.

4.4.3.1 Test Plan Control Flow

To use OpenTAP to its full potential, developers must understand the control flow of a running test plan. Several aspects of OpenTAP can influence the control flow. Important aspects include:

- Test plan hierarchy
- TestStep.PrePlanRun, TestStep.Run, TestStep.PostPlanRun methods
- Result Listeners
- Instruments and DUTs
- Test steps modifying control flow

4.4.3.2 External Parameters

Editable OpenTAP step settings can be marked as External. The value of such settings can be set through the Editor GUI, through an external program (such as OpenTAP CLI), or with an external file. This gives the user the ability to set key parameters at run time, and (potentially) from outside the Editor GUI. You can also use the API to set external parameter values with your own program.

4.4.4 Plugins Overview

In this section is presented an overview of the OpenTAP Plugins designed in the context of 5G-VINNI to interconnect the used testing tools with the Automation Component.

4.4.4.1 HEAT

HEAT is an orchestration service that can be used to control OpenStack in terms of building and setting up VMs, networks, stacks, etc. HEAT provides a REST API which is used in this plugin.

The current version of the plugin supports several features, which are implemented as OpenTAP test steps. The plugin also implements 2 OpenTAP instruments which is used when executing the steps.

Instrument name	Instrument Settings	Description
Orchestration Service	-Identity Service Instrument: a reference to the identity service of the currently used OpenStack -Orchestration Service Endpoint: URL and port of the currently used orchestration service endpoint (HEAT)	Creates a reference for an orchestration service (HEAT), based on endpoint and identity service. A connection to this orchestration service is created when test steps of the plugin are used.
Identity Service	-Domain Name: the domain name of the user -Identity service endpoint: URL and port of the currently used identity service -Password: password of the user -ProjectId: id of the project -User: user ID	Creates a reference for an identity service of OpenStack. A connection to this identity service is created when test steps of the plugin are used.

The implemented test steps are described in the following table.

Test step name	Description
Create Stack	Creates a stack based on an input HEAT Orchestration Template (HOT) file and returns a reference to the created stack.
Define String	Defines a string, which is useful for defining input once to be used several times later.
Delete Stack	Deletes a stack based on an input reference to a previously created stack.
Get Output Value	Gets the value of a specific output from a stack previously created.
List Outputs	Lists the available outputs from a created stack.
List Stack Events	Lists the events of a stack.
List Stacks	Lists the stacks created in OpenStack.
SSH Client	Creates an SSH client that connects to an endpoint and executes a command.
Wait for Stack Event	Waits for a specific event in a stack.

The current version of the plugin is developed for OpenTAP version 9.0 (minimum) both for windows and Linux.

The typical flow of using the HEAT plugin in OpenTAP is to create a stack containing VMs and networks, getting specific output needed for interaction, performing actions via SSH client, and deleting the stack.

4.4.4.2 Yardstick

Yardstick is a tool for testing the performance of a Network function virtualization infrastructure (NFVI). This is done using a set of predefined tests. Yardstick consists of Yardstick Jumphost and Yardstick image. The Jumphost in turn controls HEAT for setting up networks and test instances based on the Yardstick image. Yardstick Jumphost provides a REST API which is used in this plugin.

The current version of the plugin supports several features and both version 1 and 2 of the API, implemented as OpenTAP test steps. The plugin also implements an OpenTAP instrument which is used when executing the steps.

Instrument name	Instrument Settings	Description
Yardstick	-Yardstick REST API Base URL: a reference to the Yardstick Jumphost in terms of URL and port.	Creates a reference for a Yardstick Jumphost. A connection to this Yardstick Jumphost is created when test steps of the plugin are used.

The implemented test steps are described in the following table.

Test step name	Description
V1 API - GetTaskLogStep	Get the log of a task executed by Yardstick.

Test step name	Description
V1 API - GetTaskResultStep	Get the results of a test executed by Yardstick.
V1 API - GetTaskStatusStep	Get the status of a task and evaluate the status code.
V1 API - ListTestCasesStep	List the test cases available at the Yardstick Jumphost.
V1 API - RunTestCaseStep	Run a test case based on the test case ID.
V1 API - WaitForTaskToFinish	Wait for a specific task to finish execution and evaluate the status code.
V2 API - Project - CreateTaskStep	Create a new task based on name and project ID and return the task ID.
V2 API - Project - DeleteProjectStep	Delete a project based on project ID.
V2 API - Project - GetProjectInfoStep	Get information of a project based on project ID.
V2 API - Project - GetProjectsStep	Get the current available projects in Yardstick.
V2 API - Task - AddEnvironmentStep	Add an environment to an existing task.
V2 API - Task - AddTestCase	Add a test case to be executed to an existing task based.
V2 API - Task - ListTasksStep	List the tasks currently available in Yardstick.

The current version of the plugin is developed for OpenTAP version 9.0 (minimum) both for windows and Linux.

The typical flow of using the Yardstick plugin in OpenTAP is for API version 1 to list test cases and run a test case, and for API version 2 to create a task, add environment, add test case and execute the task.

4.4.4.3 IxLoad

IxLoad VE is a comprehensive traffic generating tool (as show in Section 4.3.1) which is composed of a gateway, virtual chassis, and virtual load modules. the IxLoad gateway offers a REST API for configuring and controlling tests, executed by the chassis and load modules. The plugin utilized the REST API for controlling the tool.

The current version of the plugin supports several basic features allowing the user to manage sessions, basic configurations, executing tests and evaluating results. The plugin also implements an OpenTAP instrument which is used when executing the steps.

Instrument name	Instrument Settings	Description
IxLoad Gateway	-IpAddress: an IP of the IxLoad gateway. -Port: the IxLoad gateway port for REST API. -Type: the type of gateway, i.e. Linux or Windows.	Creates a reference for an IxLoad gateway. A connection to this gateway is created when test steps of the plugin are used.

The implemented test steps are described in the following table.

Test Step Name	Description
Evaluate test - Define stats	Define the stats that the test will be evaluated upon.
Evaluate test - Pass/Fail Criteria	Define the expressions (<,>,=) to evaluate the selected stats of the test.
Info - Get Chassis List	Get the list of chassis registered at the gateway.
Info - Get Community List	Get the community list of the current configuration at the gateway, in terms of ports, clients, servers, traffic flows, etc.
Info - Get Port List	Get list of ports registered at a chassis.
Info - Get Sessions	Get list of sessions at the gateway.
Run test - Run Test	Run the test based on the current configuration.
Setup - Add Chassis IP	Add a chassis based on its IP.
Setup - Assign Port to Chassis	Assign a port to a chassis to be used in the test configuration.
Setup - Clear Chassis List	Clear the list of chassis registered at the gateway.
Setup - Close Session	Close a specified session, e.g. the previously created session.
Setup - Create Session	Create and activate a session with the gateway.
Setup - Delete Session	Delete all sessions at the gateway.
Setup - Load Configuration	Load a configuration file that has been uploaded to the gateway.
Setup - Upload RXF File	Upload a configuration file to the gateway.

The current version of the plugin is developed for OpenTAP version 9.0 (minimum) both for Windows and Linux.

The typical flow of using the IxLoad plugin in OpenTAP is to create a session, upload RXF file, load RXF file, clear chassis list, add chassis, assign ports to chassis, define stats, run test, define and evaluate pass/fail criteria and finally delete session.

4.5 Prototyping and Staging Deployment Status of TaaS

4.5.1 TaaS Traffic Generation Components

4.5.1.1 MONROE Development

A key perquisite before making MONROE nodes ready for measuring E2E performance in 5G-VINNI facilities is the availability of suitable UEs or Customer Premises Equipments (CPEs). At the moment, there are no Commercial-Off-The-Shelf (COTS) CPEs or UEs that are available in Europe. In the meanwhile, we plan to investigate the suitability of Samsung S10 as potential UEs. We will evaluate two modes of tethering: operating the phone as a WiFi access point and connecting to the phone directly via USB. We also plan to check whether the phone chipset allows for directly accessing connection metadata. The current MONROE node can support up to 200Mbps over IEEE 802.11ac, which means that it is adequate for conducting speed tests during Rel’0 and Rel’1.

4.5.1.1.1 Measuring IoT connectivity

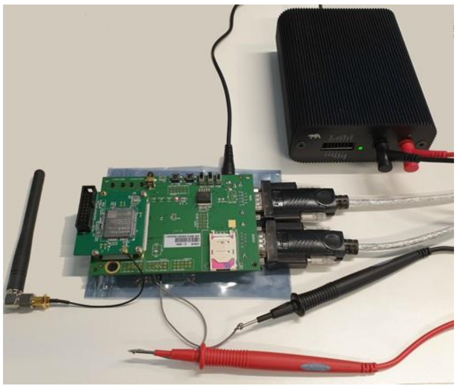


Figure 17 – NB-IoT power measurement setup

We have tested a number of COTS NB-IoT and LTE-M modules over commercial LTE networks in Norway. These include: u-blox EVK-N211, u-blox EVK-R410M, Quectel-BC95 and Quectel BC-96. More specifically, we have implemented software necessary for initiating connectivity to the network, tools for collecting connection metadata including SINR, serving cell, ECL among others as well as software for monitoring and storing control plane messages. Furthermore, we have adapted these modules to allow for instrumenting their power consumption. To this end, we use the Otii Arc power meter. Otii Arc is a power optimization tool used to measure energy consumption for embedded systems as a power supply unit for the tested IoT device or as a current and voltage measurement unit, which provides up to 5 V with a high-resolution current measurement with sample rate up to 4 ksps for the range of 1 µA to 5 A.

The form factor of the modules above, unfortunately, cannot be easily integrated with MONROE nodes. We, however, have access to USB dongles UEs that are built around u-blox SARA R-410-02B module. These UEs are compatible with MONROE nodes, but we have not managed to measure their power consumption. Accordingly, we currently have two alternatives for testing NB-IoT and LTE-M:

MONROE node-based testing that does not provide a mean for accurately measuring power consumption.

Laptop based testing using various evaluation kits that allows for tracking power consumption. We are working on supporting new modules including ones from Nordic Semiconductors and Sierra Wireless.

4.5.1.2 Summary

Tool	Status	OpenTAP plugin	Expected Release
IxLoad VE	Commercially available. Ready to be deployed when the facilities are ready for Rel’0	Yes	Rel’0
Hawkeye	Commercially available. Public server currently deployed on KEYS on-prem cloud. Ready to be deployed in Norway as central cloud	No	Rel’0
Yardstick	Openly available from community. Ready to be deployed	Yes	Rel’0

Tool	Status	OpenTAP plugin	Expected Release
MONROE	Waiting for availability of commercial devices	No	Rel'1

4.5.2 TaaS Automation Solution

Service	Status	Expected Release
Client Web App	Prototype ready, currently re-factored	Rel'0
API Gateway	State of Art investigation, decision made	Rel'2
Identity/Registration Cluster	System available, missing integration with business logic	Rel'1
Library	Basic version with limited features	Rel'0
Test Configuration	Architecture design stage	Rel'2
Message Bus	System available, currently integrated	Rel'0
Campaign Handler	Prototype version available. Must be re-factored in the future	Rel'0
OpenTAP	System fully available. Integration in progress.	Rel'0
Scheduler	Design available. Expected initial limited feature version	Rel'0
Executor	Design available. Expected initial limited feature version	Rel'0
Execution History	Architecture design stage. Expected introduction with limited capability	Rel'1
Telemetry	Architecture design. Limited capabilities expected at introduction	Rel'1

5 Telemetry Components

As it was defined in Section 2.2, telemetry is a process that allows to collect measurements about the elements and collect it into a remote or centralized location or database. Telemetry helps stream data out of the network faster, enabling the streaming of large amounts of data in real time. Therefore, telemetry can be used for helping network monitoring and troubleshooting.

Telemetry uses a subscription model to identify information sources and destinations, in which the subscriber sends a request for information upon the network element, and then a subscribed set of YANG objects are streamed to that subscriber.

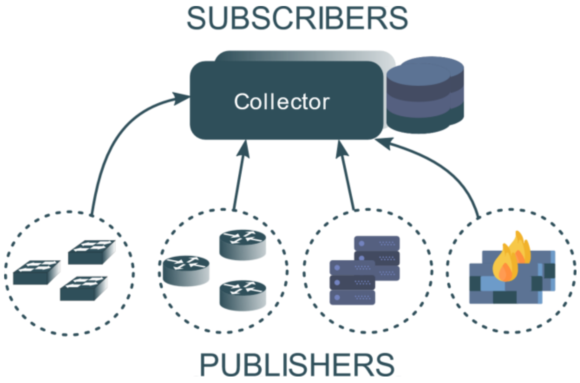


Figure 18 - Telemetry architecture

As Figure 18 depicts, network elements (whether physical or virtual) stream telemetry information from the publishers to the collector. This information is integrated in the collector, which can summarize it, correlates it, and aggregates it.

The first operational activities carried out in 5TONIC Spain facility site are described in D3.2 Annex D [16]. In this document, the components that will be used for the monitoring interface are fully detailed. As a reminder, the *mon-collector* is the one in charge of the performance monitoring activities. This module collects NFVI metrics from OpenStack Ceilometer and VNF metrics from OSM's VCA (VNF Configuration Adapter) module, through Juju metrics. Additionally, the monitoring system would be extended with the implementation of a data model-driven telemetry system, that follows the publisher-subscriber methodology defined before. Figure 19 depicts the goal to be implemented in the Spain facility site.

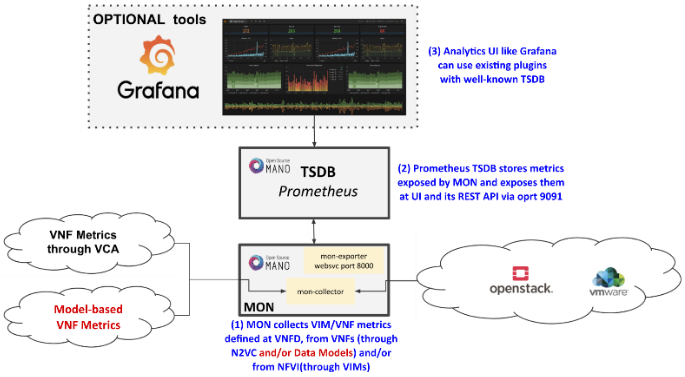

OPTIONAL tools
Grafana
(3) Analytics UI like Grafana can use existing plugins with well-known TSDB
Open Source MANO
TSDB
Prometheus
(2) Prometheus TSDB stores metrics exposed by MON and exposes them at UI and its REST API via oprt 9091
VNF Metrics through VCA
Model-based VNF Metrics
Open Source MANO
mon-exporter websvc port 8000
mon-collector
MON
openstack.
vmware
(1) MON collects VIM/VNF metrics defined at VNFD, from VNFs (through N2VC and/or Data Models) and/or from NFVI(through VIMs)

Figure 19 – MON collector with model-based metrics

As it was briefly described in D3.2 Annex D [16], the model-driven telemetry system follows a layer architecture in which each layer is independent from each other, providing this a flexible system depending on the specific environment it wants to be deployed. As a first deployment, the Spain facility will implement a client that connects to the VNF/PNF through gRPC protocol and relies on the OpenConfig YANG data models.

Taking into consideration the description of the different set of elements of the monitoring system architecture detailed in D1.1 [14], the new telemetry service can be considered as a monitoring agent[3], as it is going to be part of the OSM monitoring system, as depicted in Figure 20.

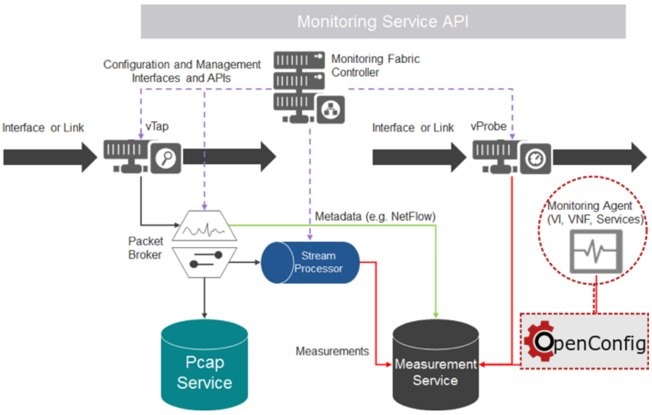


Figure 20 – Monitoring Service Architecture with model-based metrics

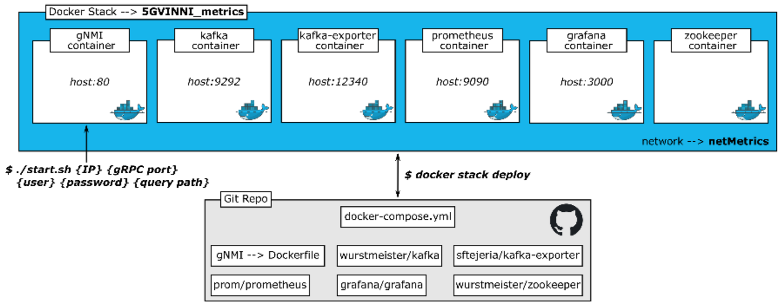
OpenConfig adopts YANG data modelling language for managing and displaying management information of each device for its use in network management configuration. Likewise, it defines a hierarchical data structure in which each node has a unique identifier that allows a clear interaction between the different nodes in the hierarchy. This also makes possible the definition of reusable grouping of nodes.

Regarding the transport protocol, gRPC is an HTTP/2 transport framework based on Protocol Buffer develop by Google. Despite it needs an additional API to support telemetry service, its implementation provides lightweight API for fast data delivery at high rates, including support of streaming mode with extremely low overhead. It allows not only YANG-based dataset and method definition, but also binary data format compression and efficient wire encoding. Previously mentioned capabilities turn gRPC into a suitable option for connecting devices and sensors for telemetry streaming purposes, even better than HTTP, RESTCONF or NETFCONF standard interfaces based on either XML or JSON.

The telemetry session can be established following two methods: dial-in or dial-out. In dial-out mode, the paths and destinations are configured in the network element and put together in one or more subscriptions. Therefore, the network element tries to continually establish a session with each destination. In contrast, in the dial-in mode, the collector connects to the device and subscribes to one or more paths. The gNMI (gRPC Network Management Interface) used in this telemetry system supports the dial-in mechanism.

[3] *"The monitoring agent is an element running as a part of the NFVI, or the VNFs, or Orchestration services providing data such as logs, events streams, infrastructure measurements (e.g. CPUs usage)"*

The current deployment of this service can be seen in the Figure 21.



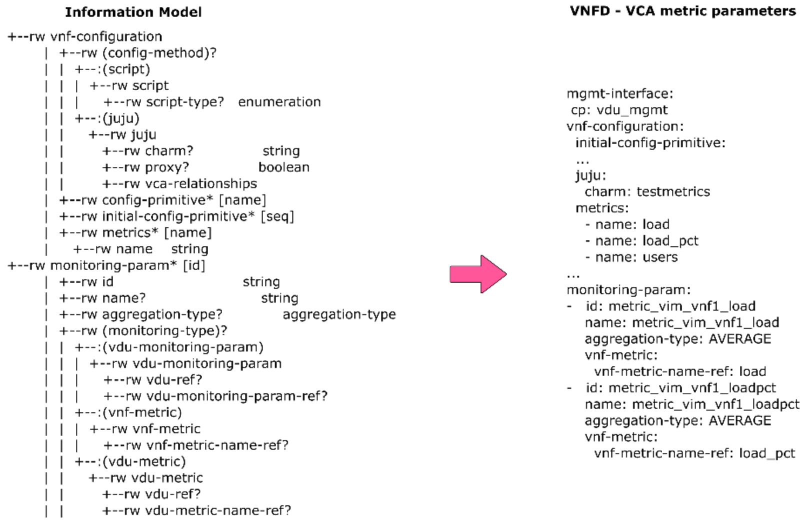
```
$ ./start.sh {IP} {gRPC port}
  {user} {password} {query path}
```

```
$ docker stack deploy
```

Figure 21 – Current deployment of telemetry service

As it can be seen, the current deployment of telemetry service is done by a Docker stack, which allows to define and coordinate the functionality of an entire application. This stack replicates the current OSM R5 services (Kafka, Prometheus and Grafana) and the gNMI client that performs the connection between the orchestrator and the VNF/PNF devices through gRPC protocol. To initiate the service, there is an external bash script that calls the gNMI Docker container for collecting the desired data in a specific VNF/PNF.

The deployment of this Docker stack is a 1st approach for the final goal depicted in Figure 19. To implement the final scenario, the Information Model and the VNFD[4] should be modified in a way the MON (MONitoring) service includes the gNMI client. In the Figure 22 is described both the information model and the VNF descriptor for collecting data through OSM VCA.



Information Model

```
+--rw vnf-configuration
      |  +--rw (config-method)?
      |  |  +--:(script)
      |  |  |  +--rw script
      |  |  |     +--rw script-type?   enumeration
      |  |  +--:(juju)
      |  |     +--rw juju
      |  |        +--rw charm?            string
      |  |        +--rw proxy?            boolean
      |  |        +--rw vca-relationships
      |  +--rw config-primitive* [name]
      |  +--rw initial-config-primitive* [seq]
      |  +--rw metrics* [name]
      |     +--rw name    string
+--rw monitoring-param* [id]
      |  +--rw id                          string
      |  +--rw name?                          string
      |  +--rw aggregation-type?              aggregation-type
      |  +--rw (monitoring-type)?
      |  |  +--:(vdu-monitoring-param)
      |  |  |  +--rw vdu-monitoring-param
      |  |  |     +--rw vdu-ref?
      |  |  |     +--rw vdu-monitoring-param-ref?
      |  |  +--:(vnf-metric)
      |  |  |  +--rw vnf-metric
      |  |  |     +--rw vnf-metric-name-ref?
      |  |  +--:(vdu-metric)
      |  |     +--rw vdu-metric
      |  |        +--rw vdu-ref?
      |  |        +--rw vdu-metric-name-ref?
```

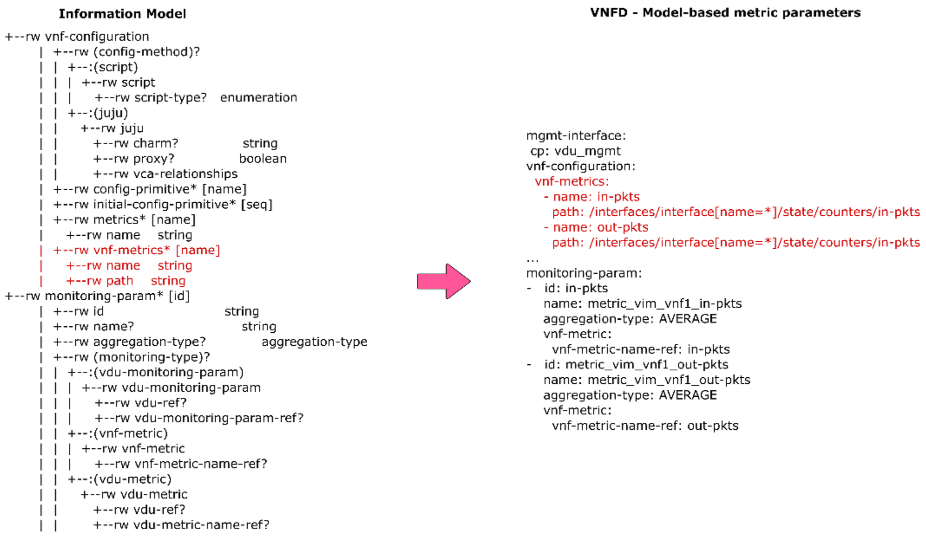
VNFD - VCA metric parameters

```
mgmt-interface:
 cp: vdu_mgmt
vnf-configuration:
  initial-config-primitive:
  ...
  juju:
    charm: testmetrics
  metrics:
    - name: load
    - name: load_pct
    - name: users
...
monitoring-param:
-   id: metric_vim_vnf1_load
    name: metric_vim_vnf1_load
    aggregation-type: AVERAGE
    vnf-metric:
      vnf-metric-name-ref: load
-   id: metric_vim_vnf1_loadpct
    name: metric_vim_vnf1_loadpct
    aggregation-type: AVERAGE
    vnf-metric:
      vnf-metric-name-ref: load_pct
```

Figure 22 – VNFD for OSM VCA metrics

In a similar way, it is proposed the information model depicted in Figure 23 for the definition of model-based metrics.

[4] Firstly, the VNFD will be modified, but the intention is to be extended also for the NST.



Information Model

```
+--rw vnf-configuration
     |  +--rw (config-method)?
     |  |  +--:(script)
     |  |  |  +--rw script
     |  |  |     +--rw script-type?   enumeration
     |  |  +--:(juju)
     |  |     +--rw juju
     |  |        +--rw charm?               string
     |  |        +--rw proxy?               boolean
     |  |        +--rw vca-relationships
     |  +--rw config-primitive* [name]
     |  +--rw initial-config-primitive* [seq]
     |  +--rw metrics* [name]
     |     +--rw name    string
     |  +--rw vnf-metrics* [name]
     |     +--rw name    string
     |     +--rw path    string
+--rw monitoring-param* [id]
     |  +--rw id                            string
     |  +--rw name?                            string
     |  +--rw aggregation-type?              aggregation-type
     |  +--rw (monitoring-type)?
     |  |  +--:(vdu-monitoring-param)
     |  |  |  +--rw vdu-monitoring-param
     |  |  |     +--rw vdu-ref?
     |  |  |     +--rw vdu-monitoring-param-ref?
     |  |  +--:(vnf-metric)
     |  |  |  +--rw vnf-metric
     |  |  |     +--rw vnf-metric-name-ref?
     |  |  +--:(vdu-metric)
     |  |     +--rw vdu-metric
     |  |        +--rw vdu-ref?
     |  |        +--rw vdu-metric-name-ref?
```

VNFD - Model-based metric parameters

```
mgmt-interface:
 cp: vdu_mgmt
vnf-configuration:
  vnf-metrics:
    - name: in-pkts
      path: /interfaces/interface[name=*]/state/counters/in-pkts
    - name: out-pkts
      path: /interfaces/interface[name=*]/state/counters/in-pkts
...
monitoring-param:
-   id: in-pkts
    name: metric_vim_vnf1_in-pkts
    aggregation-type: AVERAGE
    vnf-metric:
      vnf-metric-name-ref: in-pkts
-   id: metric_vim_vnf1_out-pkts
    name: metric_vim_vnf1_out-pkts
    aggregation-type: AVERAGE
    vnf-metric:
      vnf-metric-name-ref: out-pkts
```

Figure 23 – VNFD for Model-based metrics

As it can be seen in Figure 23, a new parameter can be defined in the VNFD: *vnf-metrics*. This parameter would be composed of a name (key identifier) and also the path to be subscribed by the gNMI client.

Furthermore, as the MON service is the one in charge of performing the monitoring activities, it should be modified for implementing the gNMI client in its code. The Figure 24 highlight this concept.

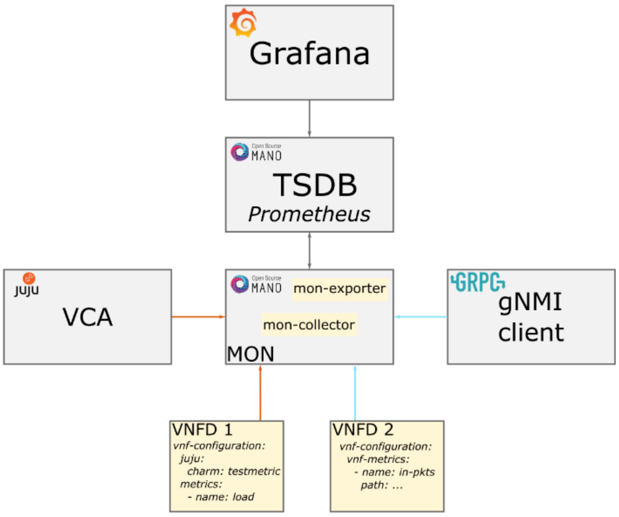


Figure 24 – MON service with gNMI client

As it is depicted, if the VNFD includes the *metric* parameter, the *mon-collector* will collect the data from the OSM VCA through Juju metrics. In contrast, if the VNFD includes the *vnf-metric* parameter, the *mon-collector* will collect it from the gNMI client by subscribing to the specific data path included in the descriptor.

6 Conclusions

This deliverable has reported on the activities carried out by Task T4.1 and Task T4.2 during the first Period of the project.

This document has briefly introduced some of the use cases that the Vertical customers will potentially bring, in order to provide a preliminary idea on what type of tests can be expected and which KPIs might be relevant. An overview of the current SoA on testing standards has also been described, to provide a solid base for the envisaged testing process that WP4 will specifically design for the 5G-VINNI project.

The foundations of such a Testing Methodology relies on the identification of a common template for designing the 5G-VINNI specific Test Cases, as establishing a formalization process that will lead, towards the end of the project, to a Test Specification Handbook, whose seminal core can be seen in the Annexes to this Deliverable.

From a tooling perspective, an overview of the initial tools used testing Rel'0 of the facility has been presented. While most of the tools can be considered a collection of SoA COTS products, coming from enterprises or open source communities, the real innovation brought by the 5G-VINNI project lies in the creation of a TaaS solution for the project.

Monitoring has also been introduced in this Deliverable, providing a first overview on how facility site using OSM can implement a network monitoring toolset that allows not only the human operators but also automated components to have an overview of the network performance.

References

[1] NGMN, Definition of the Testing Framework for the NGMN 5G Pre-Commercial Network Trials (Version 2), https://www.ngmn.org/fileadmin/ngmn/content/downloads/Technical/2019/190111_NGMN_PreCommTrials_Framework_definition_v2_small.pdf

[2] 3GPP, TS 28.552: Management and orchestration; 5G performance measurements, https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3413

[3] ETSI, NFV-TST 001: Network Functions Virtualisation (NFV); Pre-deployment Testing; Report on Validation of NFV Environments and Services, Available online: http://www.etsi.org/deliver/etsi_gs/NFV-TST/001_099/001/01.01.01_60/gs_NFV-TST001v010101p.pdf

[4] ETSI, NFV-TST 004: Network Functions Virtualisation (NFV); Testing; Guidelines for Test Plan on Path Implementation through NFVI, Available online: http://www.etsi.org/deliver/etsi_gr/NFV-TST/001_099/004/01.01.01_60/gr_NFV-TST004v010101p.pdf

[5] ETSI, NFV-TST 009: Network Functions Virtualisation (NFV) Release 3; Testing; Specification of Networking Benchmarks and Measurement Methods for NFVI, Available online: http://www.etsi.org/deliver/etsi_gs/NFV-TST/001_099/009/03.01.01_60/gs_NFV-TST009v030101p.pdf

[6] ETSI, NFV-TST 007: Network Functions Virtualisation (NFV) Release 2; Testing; Guidelines on Interoperability Testing for MANO, Available online: http://www.etsi.org/deliver/etsi_gr/NFV-TST/001_099/007/02.05.01_60/gr_NFV-TST007v020501p.pdf

[7] ETSI, NFV-TST 010: Network Function Virtualisation (NFV) Release 2; Testing; API Conformance Testing Specification, Available online: https://portal.etsi.org/webapp/workProgram/Report_Schedule.asp?WKI_ID=54031

[8] 3GPP, TS 28.554: Telecommunication management; Management and orchestration; 5G end to end Key Performance Indicators (KPI), Available online: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3415

[9] EU H2020 TRIANGLE Project, Deliverable D2.6: Final Test Scenario and Test Specifications, Available online: https://www.triangle-project.eu/wp-content/uploads/2018/11/TRIANGLE_D2-6.pdf

[10] 5G PPP, Contractual Arrangement Setting up a Public Private Partnership in the Area of Advanced 5G Network Infrastructure for the Future Internet between the European Union and the 5G Infrastructure Association, Available online: https://5g-ppp.eu/contract/

[11] ETSI Zero touch network and Service Management (ZSM) Industry Specification Group (ISG), https://www.etsi.org/technologies/zero-touch-network-service-management

[12] Open Platform for NFV (OPNFV), Linux Foundation Project, https://www.opnfv.org/

[13] Open Network Automation Platform (ONAP), Linux Foundation Project, https://www.onap.org/

[14] 5G-VINNI Project, Deliverable D1.1: Design of infrastructure architecture and subsystems v1, Available online: https://doi.org/10.5281/zenodo.2668754

[15] OPNFV Community, Project Yardstick, Available online: https://wiki.opnfv.org/display/YardStick/YardStick

[16] 5G-VINNI Project, Deliverable D3.2: Publication of first version of service catalogues to vertical consumers

[17] 5G-VINNI Project, Deliverable D3.1: Specification of services delivered by each of the 5G-VINNI facilities

[end of document]

Deliverable D4.1 Annex A

5G-VINNI Test Specification

5G-VINNI Test Methodologies and Test Cases

Editor:	Andrea F. Cattoni (Keysight)
Deliverable nature:	Report (R)
Dissemination level: (Confidentiality)	Public (PU)
Contractual delivery date:	31st February 2021 (living document)
Actual delivery date:	
Suggested readers:	ICT-19 projects, industry verticals seeking to conduct Testing on 5G systems
Version:	0.2
Total number of pages:	17
Keywords:	5G, KPI, Use Cases, Test Plan, Test Framework,

Abstract

[End of abstract]

Disclaimer

This document contains material, which is the copyright of certain 5G-VINNI consortium parties, and may not be reproduced or copied without permission.

In case of Public (PU): All 5G-VINNI consortium parties have agreed to full publication of this document.

In case of Restricted to Programme (PP): All 5G-VINNI consortium parties have agreed to make this document available on request to other framework programme participants.

In case of Restricted to Group (RE): All 5G-VINNI consortium parties have agreed to full publication of this document. However this document is written for being used by <organisation / other project / company etc.> as <a contribution to standardisation / material for consideration in product development etc.>.

In case of Consortium confidential (CO): The information contained in this document is the proprietary confidential information of the 5G-VINNI consortium and may not be disclosed except in accordance with the consortium agreement.

The commercial use of any information contained in this document may require a license from the proprietor of that information.

Neither the 5G-VINNI consortium as a whole, nor a certain part of the 5G-VINNI consortium, warrant that the information contained in this document is capable of use, nor that use of the information is free from risk, accepting no liability for loss or damage suffered by any person using this information.

The EC flag in this document is owned by the European Commission and the 5G PPP logo is owned by the 5G PPP initiative. The use of the flag and the 5G PPP logo reflects that 5G-VINNI receives funding from the European Commission, integrated in its 5G PPP initiative. Apart from this, the European Commission or the 5G PPP initiative have no responsibility for the content.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 815279.

Impressum

Full project title	5G Verticals Innovation Infrastructure
Project acronym	5G-VINNI
Number and title of work-package	WP4 Execution and Validation of Use Cases and KPIs on 5G-VINNI End-to-End Facility
Number and title of task(s)	T4.1 Definition of Test Methods and Test Cases for validation of network service-level KPIs and use case-level KPIs
Document title	5G-VINNI Test Specification
Editor: Name, company	Andrea Cattoni, Keysight
Work-package leader: Name, company	Andrea Cattoni, Keysight

Copyright notice

List of authors

Company	Author	Contribution
Keysight Technologies (KEYS)	Andrea F. Cattoni	Section 1
EANTC	Mohammad Othman	Section 2

Table of Contents

List of authors ... 3
Table of Contents ... 4
List of tables ... 5
Abbreviations ... 6
Definitions ... 8
1 Introduction ... 9
1.1 Objective of this document ... 9
Scope ... 9
Test Equipment Requirements ... 9
1.1.1 Test Equipment for interoperability tests ... 9
1.1.2 Test Equipment for performance and scalability tests ... 9
1.1.3 Test Equipment for layer 7 performance and security test areas ... 12
2 Testing Methodology Guidelines ... 14
Basic concepts for testing ... 14
2.1.1 Overview ... 14
2.1.2 Pre-qualification Testing ... 14
2.1.3 Unit Test ... 14
2.1.4 Sub-module Test ... 14
2.1.5 E2E Test ... 15
2.1.6 Test Environment ... 15
2.1.7 Test Descriptions ... 15
Test Specifications ... 15
Testing Process ... 15
References ... 17

List of tables

Table 1: Test equipment for performance tests 11
Table 2: Test equipment for layer 7 and security tests 12

Abbreviations

5G	Fifth Generation (mobile/cellular networks)
5G PPP	5G Infrastructure Public Private Partnership
IoT	Internet of Things
KPI	Key Performance Indicator
OPEX	Operational Expenditure
QoE	Quality of Experience
OPEX	Operational Expenditure
QoS	Quality of Service
R&D	Research and Development
SDN	Software Defined Networks
KPI	Key Performance Indicator
MANO	Management and Orchestration
MEC	Multi-access Edge Computing
mMTC	Massive Machine Type Communications
URLLC	Ultra-Reliable Low Latency
E2E	End-To-End
RAN	Radio Access Network
DUT	Device Under Test
E2E	End-to-End
eMBB	enhanced Mobile Broadband
gNB	5G Node B
NFV	Network Function Virtualization
NFVI	NFV Infrastructure
NFVO	Network Function Virtualization Orchestrator
NGMN	Next Generation Mobile Network

NS	Network Service
SLA	Service Level Agreement
UE	User Equipment
VIM	Virtual Infrastructure Manager
VNF	Virtual Network Function
VNFD	Virtual Network Function Descriptor
VNFM	Virtual Network Function Manager

Definitions

This document contains specific terms to identify elements and functions that are considered to be mandatory, strongly recommended or optional. These terms have been adopted for use similar to that in IETF RFC2119, and have the following definitions.

MUST This word, or the terms "REQUIRED" or "SHALL", mean that the definition is an absolute requirement of the specification.

MUST NOT This phrase, or the phrase "SHALL NOT", mean that the definition is an absolute prohibition of the specification.

SHOULD This word, or the adjective "RECOMMENDED", mean that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.

SHOULD NOT This phrase, or the phrase "NOT RECOMMENDED" mean that there may exist valid reasons in particular circumstances when the particular behaviour is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behaviour described with this label.

MAY This word, or the adjective "OPTIONAL", mean that an item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because the vendor feels that it enhances the product while another vendor may omit the same item. An implementation which does not include a particular option MUST be prepared to interoperate with another implementation which does include the option, though perhaps with reduced functionality. In the same vein an implementation which does include a particular option MUST be prepared to interoperate with another implementation which does not include the option (except, of course, for the feature the option provides.)

1 Introduction

1.1 Objective of this document

The present document provides the test methodologies and test descriptions for 5G-VINNI project testing, including the test methodology and test specification overview. KPIs definition, and detailed test description definitions are described in D4.1 Annex B [3], Annex C [4] and Annex D [5]. By customizing the test methodology and interested KPIs according to the network performance requirements in different 5G use cases, this document along with other Annexes function as the guideline of the test execution for all facility sites.

1.2 Scope

The present document provides the overall definition of test methodologies, focused KPIs, and test specifications. The test specifications are divided to two test phases: pre-qualification test and integration test. The integration test includes unit tests, sub-module test, and E2E tests.

The Annex document is organized as follows:

- Annex A describes the general purpose of this document, as well as the requirements on the test equipment and provides the guidelines of test methodology and test specification definition.
- Annex B defines the interested KPIs in the whole project.
- Annex C and Annex D include the detailed test specifications for pre-qualifications, unit tests, sub-module tests, and E2E tests.

1.3 Test Equipment Requirements

The test tools for layer 2 to layer 7 testing including network performance, application performance and security testing are needed. Both physical and virtual test tools are acceptable. However, hardware test equipment solutions are recommended for high performance and scalability tests.

Regarding NFVI/MANO functionality and interoperability test, we suggest using virtualised and automated test tools.

1.3.1 Test Equipment for interoperability tests

For NFVI interoperability, test equipment will be used to generate low rate of traffic in order to verify the correct functionality of the interoperability features. Therefore, hardware based high-performance commercial test equipment is not necessary.

Since interoperability test concept consist too many test scenarios, the test execution should be automated mainly due to time effort.

The NFVI/MANO interoperability tests are fully automated and remotely accessible, so that we can perform the tests remotely.

1.3.2 Test Equipment for performance and scalability tests

In performance and scalability tests, high traffic volume and large amount of connections are expected. It brings higher requirements on the test equipment. In this case, physical traffic generator and test emulator are needed.

Considering different use cases, the traffic generator must support generation of traffic from 1 Kbit/s up to 20Gbit/s, the UE emulator must support emulation of up to 1 million active UEs at the same time.

In unit test and sub-module test, the test emulator needs to emulate all network units or sub-modules. For example, in Transport sub-module test, the test emulator needs to emulate the RAN and mobile core network, to complete the test set up.

Test equipment selection for performance and scalability test areas is highly depend on VNF application use cases.

The table below describes some possible VNF use cases for customers and the recommended test equipment to conduct the test.

Table 1: Test equipment for performance tests

Area	Types	Test equipment type	Possible test equipment	Required test equipment capability		
				For Small size	For Medium size	For Large size
Routing and switch	switch, router	Layer 2-3 traffic generator	Ixia IxNetwork Spirent test center VeeX	5Gbit/s throughput, 10,000 flows	10Gbit/s throughput, 1 Million flow	100Gbit/s throughput, 10million flow
Access, Customer edge	Broadband Network Gateway (BNG),	Layer 2-3 traffic generator with access protocol emulation	Ixia IxNetwork Spirent test center	0.1M sessions 100sessions/s 10Gbit/s throughput	0.5M sessions 200 sessions/s 50Gbit/s throughput	1M sessions 500 sessions/s 100Gbit/s throughput
	Customer Premises Equipment (CPE)	Layer3-7 traffic generator with application traffic emulation	Ixia IxNetwork and IxLoad Spirent test center and Spirant avalanche/cyberflood	Throughput Sessions Session rate TBD	Throughput Sessions Session rate TBD	Throughput Sessions Session rate TBD
	DHCP	Layer 3 traffic generator with access protocol emulation	Ixia IxNetwork Spirent test center	500 sessions/s	1,000 sessions/s	2,000 sessions/s
	DNS	Layer 4-7 traffic generator	Ixia IxLoad Spirant avalanche	20,000 queries/s	100,000 queries/s	200,000 queries/s
Core	Route reflector (RR)	Layer 2-3 traffic generator with BGP emulation	Ixia IxNetwork Spirent test center	1M routes	3M routes	5M routers

Mobile network	Packet and Serving Gateway (PGW), Evolved Packet Data Gateway (ePDG), IMS, HSS	Layer 2-3 traffic generator with following interface emulations: S1-U, S6a, Gx, SGi and S11	Ixia IxLoad NG4T Spirent Landslide TeraVM-vRAN	1M PDN Sessions 500 PDN attachment rate 10Gbit/s throughput	3M PDN Sessions 1000 PDN attachment rate 25Gbit/s throughput	5M PDN Sessions 2000 PDN attachment rate 50Gbit/s throughput
Policy management	AAA server	HSS client emulation (SWx interface)	Ixia IxLoad Spirent Landslide NG4T	20,000 messages/s	100,000 messages/s	200,000 messages/s

1.3.3 Test Equipment for layer 7 performance and security test areas

Layer 7 performance and security tests are not in scope of Release 0 test and will be updated later.

The table below describes some examples on the test equipment requirement for layer7 and security tests.

Table 2: Test equipment for layer 7 and security tests

Types	Test equipment type	Possible test equipment	Required test equipment capability		
			For Small size	For Medium size	For Large size
NGFW	Stateful TCP traffic generator and Application traffic emulation with encrypted und unencrypted traffic	Ixia IxLoad Ixia Breaking point system Spirent avalanche/cyberflood TeraVM	5Gbit/s throughput 1M concurrent connection 50,000 connections/sec Features: CVE, fuzzing techniques, evasion techniques, application emulation	10 Gbit/s throughput 3M concurrent connection 100,000 connections/sec Features: CVE, fuzzing	30 Gbit/s throughput 8M concurrent connection 300,000 connections/sec Features: CVE, fuzzing techniques, evasion techniques, application emulation

				techniques, evasion techniques, application emulation	
IPS/IDS			Features: CVE, fuzzing techniques, evasion techniques, application emulation	Features: CVE, fuzzing techniques, evasion techniques, application emulation	Features: CVE, fuzzing techniques, evasion techniques, application emulation
Load balancer/ADS			5Gbit/s throughput 1M concurrent connection 50,000 connections/sec	10 Gbit/s throughput 3M concurrent connection 100,000 connections/sec	30 Gbit/s throughput 8M concurrent connection 300,000 connections/sec
DPI					
SSL proxy					

2 Testing Methodology Guidelines

In this section, we will describe the general guidelines in the testing methodology for 5G-VINNI project tests. The basic concepts and definitions of each test area are explained, as well as the requirements in test specifications and test process.

2.1 Basic concepts for testing

2.1.1 Overview

The test in each facility site is divided to two test phases: 1) site acceptance test (pre-qualification tests, executed by each facility site in WP2); 2) integration test (unit test, sub-module tests, E2E tests, executed by WP4).

The pre-qualification test functions as site acceptance test, the purpose is to verify the result of the site installation and commissioning. It helps to make sure the facility site is fully ready in both physical set-up and service deployment for WP4 to starting the testing.

The integration test consists of three type of tests: unit test, sub-module test, and E2E test. By dividing the network components to different level of modules, it allows each facility site to start evaluation and verification of the deployment from the beginning. It can also help to identify the bottle neck in the performance test of each network component and provide more detailed performance information of the system under test.

In this section, we will provide the basic concept for each test phase, which functions as the general guideline for all the test description developments and test executions.

2.1.2 Pre-qualification Testing

Pre-qualification tests are executed by the facility owners. The successful results trigger the engagement of WP4 for benchmarking & performance testing.

Test case details are provided in Annex C [4].

2.1.3 Unit Test

Unit test is the smallest test unit for the performance and interoperability tests. Each unit under test represents a separate component of the specific network domain. For example, the unit test for 5G mobile core (5GC) might include the unit test for AMF, SMF, UPF, UDM, etc. so that we can verify the performance of each component in the mobile core network.

The actual test unit is decided according to the test scenario and can be selected based on the implementation of the site and the KPI requirement.

2.1.4 Sub-module Test

The sub-module test is the intermediate level of test unit for the performance tests. Each sub-module under test represents a specific network domain or module. In 5G-VINNI project, the End-to-End(E2E) testing network can be divided to several sub-modules: RAN, Transport, Mobile core, and NFV.

By performing the sub-module performance tests, we can collect the detailed KPIs of each network domain. It can help to identify the bottle neck of the network performance and provide reference data for improvement and optimization of the facility site.

2.1.5 E2E Test

The End-to-End(E2E) test is the highest level of test unit for the performance tests. Each system under test represents a E2E network service, from End User to the Data Network outside of the testing system, or to another End User. In 5G-VINNI, each E2E system under test is a network slice. For example, E2E test for eMBB network slice will focus on the high bandwidth provided by the network for End User to access the Internet.

Actual use cases are taken into consideration while planning the E2E network performance tests. The test results show the real service KPIs in each scenario, and provides real data of the network performance, which can be used as a reference for the capacity of the facility site.

2.1.6 Test Environment

All tests are executed in a stable environment where the SUT and test system setups are pre-defined in test planning phase. The pre-condition of starting the test must be clearly defined, and procedures of the test execution must be strictly followed, to ensure the validity of the results of the test.

Automation testing tools are introduced to perform the test to reduce time consumption and improve testing efficiency. Remote testing can be applied for certain test areas, such as NFVI/MANO interoperability test, where on-site operation is not mandatory, and testing tools are pre-deployed for remote access and operation.

Each facility is responsible in setting up the testing environment and execute the pre-qualification testing, to ensure the readiness of the SUT for the performance test execution.

2.1.7 Test Descriptions

A test description provides the detailed set of instructions (or steps) that need to be followed in order to perform a test. Most often, interoperability tests are described in terms of actions that can be performed by the user(s) of the endpoint device(s).

In the case where the test is executed by a human operator, test will be described in natural language. In the case where the tests are automated, a programming or test language will be used to implement the test descriptions.

The steps in the test description can be of different nature, depending on the kind of action required: trigger a behaviour on one FUT, verify the functional response on another FUT, configure the SUT (add/remove a FUT), check a log, etc. Each step identifies the FUT and/or the interface targeted by the action.

2.2 Test Specifications

The requirements of the test specifications and a template of the test case is listed in D4.1 [2] section 2.3.4.

2.3 Testing Process

The testing process of each facility site for each release in the project can be divided to four phases:

- **Site implementation and preparation**

  Each facility site owner is responsible to implement the site as designed and planned in WP1 and WP2 and prepare the site in physical connections and service configuration to be ready for the execution of all interested tests.

  WP4 is responsible to develop the test plan and test automation tools to execute the test. Each facility site will implement the test tools and test equipment, and in some cases remote access of the test environment, to prepare the site for the testing.

- **Pre-qualification test**

  The facility site owner is responsible to execute the pre-qualification test as defined in D4.1 Annex C [4], and provide the test results as an input for WP4 to take over the site facility for performance tests.

  Not all pre-qualification tests are required to be tested, only the ones mentioned in the test plan preambles are mandatory.

- **Integration test**

  For all unit test, sub-module test, and E2E test, the testing process is as follows:

  1. Check the preambles of the test area if all conditions are met, for example, if the test setup is correct, or if the test environment is configured with required parameters, or if the required pre-qualification tests are tested and passed, etc.
  2. Verify the initial conditions of the SUT and setup the test tools to execute the test.
  3. Execute the test following the test procedures and record the result of each step. Verify if the actual result is same with the expected result.
  4. Calculate the KPIs based on the measurements taken from the test executions.
  5. Perform the postamble as stated in the test plan.

- **Test report**

  Generate test report with the KPIs and results collected during the test execution period. The reporting procedure and report format is out of scope of this document.

References

[1] https://www.5G-VINNI.eu/ © 5G-VINNI consortium 2018

[2] https://onlyoffice.eurescom.eu/products/files/doceditor.aspx?fileid=21137, 5G-VINNI Deliverable D4.1, "Initial report on test-plan creation and methodology, and development of test orchestration framework"

[3] https://onlyoffice.eurescom.eu/products/files/doceditor.aspx?fileid=21352, 5G-VINNI Deliverable D4.1 Annex B, Key Performance Indicators Formalization

[4] https://onlyoffice.eurescom.eu/products/files/doceditor.aspx?fileid=20281, 5G-VINNI Deliverable D4.1 Annex C, 5G-VINNI Pre-qualification Test Cases

[5] https://onlyoffice.eurescom.eu/products/files/doceditor.aspx?fileid=21353, 5G-VINNI Deliverable D4.1 Annex D, 5G-VINNI Test Cases Handbook

[end of document]

Deliverable D4.1 Annex B

Key Performance Indicators Formalization

Editor:	EANTC
Deliverable nature:	Report (R)
Dissemination level: (Confidentiality)	Public (PU)
Contractual delivery date:	31st February 2021
Actual delivery date:	
Suggested readers:	ICT-19 projects, industry verticals seeking to conduct Testing on 5G systems
Version:	0.0
Total number of pages:	15
Keywords:	5G, KPI, Use Cases, Test Plan, Test Framework,

Abstract

[End of abstract]

Disclaimer

This document contains material, which is the copyright of certain 5G-VINNI consortium parties, and may not be reproduced or copied without permission.

In case of Public (PU): All 5G-VINNI consortium parties have agreed to full publication of this document.

In case of Restricted to Programme (PP): All 5G-VINNI consortium parties have agreed to make this document available on request to other framework programme participants.

In case of Restricted to Group (RE): All 5G-VINNI consortium parties have agreed to full publication of this document. However this document is written for being used by <organisation / other project / company etc.> as <a contribution to standardisation / material for consideration in product development etc.>.

In case of Consortium confidential (CO): The information contained in this document is the proprietary confidential information of the 5G-VINNI consortium and may not be disclosed except in accordance with the consortium agreement.

The commercial use of any information contained in this document may require a license from the proprietor of that information.

Neither the 5G-VINNI consortium as a whole, nor a certain part of the 5G-VINNI consortium, warrant that the information contained in this document is capable of use, nor that use of the information is free from risk, accepting no liability for loss or damage suffered by any person using this information.

The EC flag in this document is owned by the European Commission and the 5G PPP logo is owned by the 5G PPP initiative. The use of the flag and the 5G PPP logo reflects that 5G-VINNI receives funding from the European Commission, integrated in its 5G PPP initiative. Apart from this, the European Commission or the 5G PPP initiative have no responsibility for the content.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 815279.

Impressum

Full project title	5G Verticals Innovation Infrastructure
Project acronym	5G-VINNI
Number and title of work-package	WP4 Execution and Validation of Use Cases and KPIs on 5GVINNI End-to-End Facility
Number and title of task(s)	T4.1 Definition of Test Methods and Test Cases for validation of network service-level KPIs and use case-level KPIs
Document title	KPIs Definitions
Editor: Name, company	
Work-package leader: Name, company	Andrea Cattoni, Keysight

Copyright notice

List of authors

Company	Author	Contribution
Keysight Technologies (KEYS)	Andrea Cattoni	

Table of Contents

List of authors 3
Table of Contents 4
List of Figures 5
List of Tables 6
Abbreviations 7
Definitions 8
1 Introduction 9
1.1 5G PPP Vision on KPIs 9
1.2 Refined KPIs 9
1.2.1 Data Rate 9
1.2.2 Mobility (speed) 9
1.2.3 E2E Latency 9
1.2.4 Density (number of devices) 9
1.2.5 Reliability 10
1.2.6 Position Accuracy (Location) 10
1.2.7 Coverage 10
1.2.8 Service Deployment Time 10
1.2.9 Data Volume 10
1.2.10 Autonomy 10
1.2.11 Security 10
1.2.12 Identity 10
2 Key Performance Indicators List 11
3 Key Performance Indicator Formalization Template 12
3.1 3GPP-based KPI Formalization Template 12
4 Key Performance Indicator Definitions 14
4.1 Data Rate 14
4.2 Latency 14
4.3 Reliability and Availability 14
4.4 Quality 14
References 15

List of Figures

N/A

List of Tables

Table 1 – List of Defined KPIs 11

Abbreviations

5G	Fifth Generation (mobile/cellular networks)
5G PPP	5G Infrastructure Public Private Partnership
ITU	International Telecommunications Union
KPI	Key Performance Indicator
E2E	End-to-End
QoE	Quality of Experience
TMV-WG	Test, Measurements, and KPIs validation Working Group

Definitions

This document contains specific terms to identify elements and functions that are considered to be mandatory, strongly recommended or optional. These terms have been adopted for use similar to that in IETF RFC2119, and have the following definitions.

MUST This word, or the terms "REQUIRED" or "SHALL", mean that the definition is an absolute requirement of the specification.

MUST NOT This phrase, or the phrase "SHALL NOT", mean that the definition is an absolute prohibition of the specification.

SHOULD This word, or the adjective "RECOMMENDED", mean that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.

SHOULD NOT This phrase, or the phrase "NOT RECOMMENDED" mean that there may exist valid reasons in particular circumstances when the particular behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.

MAY This word, or the adjective "OPTIONAL", mean that an item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because the vendor feels that it enhances the product while another vendor may omit the same item. An implementation which does not include a particular option MUST be prepared to interoperate with another implementation which does include the option, though perhaps with reduced functionality. In the same vein an implementation which does include a particular option MUST be prepared to interoperate with another implementation which does not include the option (except, of course, for the feature the option provides.)

1 Introduction

This section leverages on the work done by the 5G PPP Technology Board task force focused on the collection of the KPIs identified by the different Phase II Projects.

The main motivation for such a choice is to provide a direct alignment between all the projects, making sure a coherent vision on how to measure network performance.

1.1 5G PPP Vision on KPIs

The technical Annex to the 5G PPP contractual arrangement [1] defines the following KPIs:

P1: Providing 1000 times higher wireless area capacity and more varied service capabilities compared to 2010.

P2: Saving up to 90% of energy per service provided.

P3: Reducing the average service creation time cycle from 90 hours to 90 minutes.

P4: Creating a secure, reliable and dependable Internet with a "zero perceived" downtime for services provision.

P5: Facilitating very dense deployments of wireless communication links to connect over 7 trillion wireless devices serving over 7 billion people.

1.2 Refined KPIs

These have been refined and extend through subsequent work, among others through the brochure "5G empowering vertical industries" [2] and are defined as follows.

1.2.1 Data Rate

Data Rate: Required bit rate for the application to function correctly. It corresponds to the user experienced data rate as defined by ITU. The most demanding vertical use cases are related to Media & Entertainment with maximum values in the order of Gb/s

1.2.2 Mobility (speed)

Maximum relative speed under which the specified reliability should be achieved. The most demanding vertical use cases are related to Automotive and eHealth with maximum value in the order of 500 km/h

1.2.3 E2E Latency

Maximum tolerable elapsed time from the instant a data packet is generated at the source application to the instant it is received by the destination application. If direct mode is used, this is essentially the maximum tolerable air interface latency. If infrastructure mode is used, this includes the time needed for uplink, any necessary routing in the infrastructure, and downlink. The most demanding vertical use cases are related to Factories with minimum values of 100 µs to 10 ms.

1.2.4 Density (number of devices)

Maximum number of devices (vehicles in the case of Automotive) per unit area that are 5G capable, although they might not all be generating traffic simultaneously for the specified application. The most demanding vertical use cases are related to Factories with up to 100/m^2

1.2.5 Reliability

Maximum tolerable packet loss rate at the application layer within the maximum tolerable end-to-end latency for that application. The most demanding vertical use cases are related to eHealth with values up to 99.99999%

1.2.6 Position Accuracy (Location)

Maximum positioning error tolerated by the application. The most demanding vertical use cases are related to Automotive with minimum values in the order of 0.3 m

1.2.7 Coverage

Area within which or population for which the application should function correctly, i.e. the specified requirements (latency, reliability and data rate) are achieved. Most of the vertical sectors have strong requirements on geographic and/or population coverage

1.2.8 Service Deployment Time

Duration required for setting up end-to-end logical network slices characterised by respective network level guarantees, such as bandwidth guarantees, End-to-End (E2E) latency, or reliability, required for supporting services of that particular vertical sector. Programmable networks and multi-tenant capability in 5G will ensure speedy deployment of services (e.g. 5G Infrastructure PPP targets 90 minutes for service deployment).

1.2.9 Data Volume

Quantity of information transferred (downlink and uplink) per time interval over a dedicated area (e.g. 5G Infrastructure PPP targets a maximum of 10 Tb/s/km2).

1.2.10 Autonomy

Time duration for a component to be operational without power being supplied. It relates to battery lifetime, battery load capacity and energy efficiency.

1.2.11 Security

System characteristic ensuring globally the protection of resources and encompassing several dimensions such as authentication, data confidentiality, data integrity, access control, non-repudiation...

1.2.12 Identity

Characteristic to identify sources of content and recognise entities in the system.

2 Key Performance Indicators List

In this Section are listed all the KPIs formalized within this document.

For June 2019 a tentative Table containing the 5G PPP Technical KPIs is to be expected. The document will be updated in future releases and considered a living document together with the other Annexes to D4.1.

Table 1 – List of Defined KPIs

Category	KPI name	Type	5G-PPP KPI
Data Rate	Minimum Expected Upstream Throughput	SLA	P1
Data Rate	Minimum Expected Downstream Throughput	SLA	P1
Latency	Maximum Expected Latency	SLA	P1, P4
Reliability and Availability	Network Reliability	SLA	P4
Quality	Quality of Experience	SLA, Tech. Validation	P1, P4
Data Rate	UL Peak Throughput	Tech. Validation	P1
Data Rate	DL Peak Throughput	Tech. Validation	P1

3 Key Performance Indicator Formalization Template

This Section describes the formal template that has been agreed within the 5G PPP TMV-WG as base for formalizing the KPIs to be used for the validation process. In 5G-VINNI it has been decided to adopt such a template and rely on the joint work between all the projects contributing to the 5G PPP TMV-WG.

The template has been inherited from 3GPP, in particular the TS 28.554 and it has been slightly modified to exclude the 3GPP specific fields.

3.1 3GPP-based KPI Formalization Template

a) Long name (Mandatory): This field shall contain the long and descriptive name of the KPI.

b) Description (Mandatory): This field shall contain the description of the KPI.

   Within this field it should be given if the KPI is focusing on network or user view.

c) Logical formula definition (Mandatory):

   The logical formula should describe what the KPI formula is in logical way. The description of the formula is given in a written textual format without any measurement or counter names. E.g. a success rate KPI’s logical formula is the successful event divided by all event.

d) Physical formula definition (Optional):

   This field should contain the KPI formula description using the 3GPP defined counter names.

   This field can be used only if the counters needed for the KPI formula is defined in any of the 3GPP TS for performance measurements (TS 28.552 [1], TS 28.553 [2]).

e) Measurement names used for the KPI (deprecated: 3GPP-specifc):

   This clause should list the measurement names used for the KPI.

   This clause can be filled out only when the underlying measurements for the KPI formula can be defined,

   i.e. physical formula definition is available.

f) KPI Object (deprecated: 3GPP-specifc)

   This clause shall describe the object of the KPI. The object of the KPI is one or some of the following:

   - NR and NG-RAN;
   - 5GC
   - 5GS

g) KPI category (deprecated: 3GPP-specifc)

   This clause contains the classification of the KPI into one of the KPI categories listed in clause 4.

h) Unit of the KPI (mandatory)

   This clause describes the unit of the KPI. The unit can be one of the following:

   - percentage;
   - time interval (second or millisecond or microsecond);
   - Erlang;

- kbit/s.

i) Type of the KPI (Mandatory: modified)

This clause describes the type of the KPI. The KPI type can be one of the following:

- MEAN: This KPI is produced to reflect a mean measurement value based on a number of sample results.
- RATIO: KPI is produced to reflect the percentage of a specific case occurrence to all the cases.
- CUM: This KPI is produced to reflect a cumulative measurement which is always increasing.
- MIN: This KPI is produced as the minimum measurement value amongst all the samples.
- MAX: This KPI is produced as the maximum measurement value amongst all the occurrences.

j) Remark: (Optional)

This field is for any further information that is needed for the KPI definition.

Here it is proposed to define any additional information that would be needed for the KPI definition; e.g. the definition of a call in UTRAN.

4 Key Performance Indicator Definitions

This Section will contain the formalization of the Technical KPIs coming from both 5G PPP TMV-WG and the internal work carried out by the 5G-VINNI partners.

4.1 Data Rate

4.2 Latency

4.3 Reliability and Availability

4.4 Quality

References

[1] 5G PPP, Contractual Arrangement Setting up a Public Private Partnership in the Area of Advanced 5G Network Infrastructure for the Future Internet between the European Union and the 5G Infrastructure Association, Available online: https://5g-ppp.eu/contract/

[2] 5G PPP, 5G empowering vertical industries, brochure, Available online: https://5g-ppp.eu/wp-content/uploads/2016/02/BROCHURE_5PPP_BAT2_PL.pdf

[end of document]

Deliverable D4.1 Annex C

5GVINNI Pre-qualification Test Cases

Editor:	EANTC
Deliverable nature:	Report (R)
Dissemination level: (Confidentiality)	Public (PU)
Contractual delivery date:	31st February 2021
Actual delivery date:	
Suggested readers:	ICT-19 projects, industry verticals seeking to conduct Testing on 5G systems
Version:	0.1
Total number of pages:	19
Keywords:	5G, KPI, Use Cases, Test Plan, Test Framework,

Abstract

[End of abstract]

Disclaimer

This document contains material, which is the copyright of certain 5G-VINNI consortium parties, and may not be reproduced or copied without permission.

In case of Public (PU): All 5G-VINNI consortium parties have agreed to full publication of this document.

In case of Restricted to Programme (PP): All 5G-VINNI consortium parties have agreed to make this document available on request to other framework programme participants.

In case of Restricted to Group (RE): All 5G-VINNI consortium parties have agreed to full publication of this document. However this document is written for being used by <organisation / other project / company etc.> as <a contribution to standardisation / material for consideration in product development etc.>.

In case of Consortium confidential (CO): The information contained in this document is the proprietary confidential information of the 5G-VINNI consortium and may not be disclosed except in accordance with the consortium agreement.

The commercial use of any information contained in this document may require a license from the proprietor of that information.

Neither the 5G-VINNI consortium as a whole, nor a certain part of the 5G-VINNI consortium, warrant that the information contained in this document is capable of use, nor that use of the information is free from risk, accepting no liability for loss or damage suffered by any person using this information.

The EC flag in this document is owned by the European Commission and the 5G PPP logo is owned by the 5G PPP initiative. The use of the flag and the 5G PPP logo reflects that 5G-VINNI receives funding from the European Commission, integrated in its 5G PPP initiative. Apart from this, the European Commission or the 5G PPP initiative have no responsibility for the content.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 815279.

Impressum

Full project title	5G Verticals Innovation Infrastructure
Project acronym	5G-VINNI
Number and title of work-package	WP4 Execution and Validation of Use Cases and KPIs on 5GVINNI End-to-End Facility
Number and title of task(s)	T4.1 Definition of Test Methods and Test Cases for validation of network service-level KPIs and use case-level KPIs
Document title	5GVINNI Test Specification
Editor: Name, company	Mohammad Othman, EANTC
Work-package leader: Name, company	Andrea Cattoni, Keysight

Copyright notice

List of authors

Company	Author	Contribution
EANTC	Mohammad Othman	

Table of Contents

List of authors ... 3
Table of Contents ... 4
List of Figures ... 5
List of Tables ... 6
Abbreviations ... 7
Definitions ... 8
1 Introduction ... 9
2 Pre-qualification Test Cases ... 10
2.1 Transport Network ... 10
2.2 5G New Radio ... 12
2.3 Core Network ... 14
2.4 MANO & NFVI ... 17
References ... 19

List of Figures

Figure 1 – Transport Network Test Architecture 11
Figure 2 – 5G New Radio Test Architecture 13
Figure 3 – E2E NSA Testing Architecture... 16
Figure 4 – NSA Core Wrap-around Testing Architecture 16
Figure 5 – MANO & NFVI Testing Architecture 18

List of Tables

Table 1 – Transport Network Pre-qualification Test Cases 10
Table 2 – New Radio Pre-qualification Test Cases 12
Table 3 – NSA Core Network Pre-qualification Test Cases 14
Table 4 – MANO & NFVI Pre-qualification Test Cases 17

Abbreviations

5G	Fifth Generation (mobile/cellular networks)
5G PPP	5G Infrastructure Public Private Partnership
IoT	Internet of Things
KPI	Key Performance Indicator
OPEX	Operational Expenditure
QoE	Quality of Experience
OPEX	Operational Expenditure
QoS	Quality of Service
R&D	Research and Development
SDN	Software Defined Networks
NFV	Network Functions Virtialization
VNF	Virtualized Network Function
PPDR	
MEC	Mobile Edge Computing
URLLC	
E2E	End to End
RAN	Radio Access Network

Definitions

This document contains specific terms to identify elements and functions that are considered to be mandatory, strongly recommended or optional. These terms have been adopted for use similar to that in IETF RFC2119, and have the following definitions.

MUST This word, or the terms "REQUIRED" or "SHALL", mean that the definition is an absolute requirement of the specification.

MUST NOT This phrase, or the phrase "SHALL NOT", mean that the definition is an absolute prohibition of the specification.

SHOULD This word, or the adjective "RECOMMENDED", mean that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.

SHOULD NOT This phrase, or the phrase "NOT RECOMMENDED" mean that there may exist valid reasons in particular circumstances when the particular behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.

MAY This word, or the adjective "OPTIONAL", mean that an item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because the vendor feels that it enhances the product while another vendor may omit the same item. An implementation which does not include a particular option MUST be prepared to interoperate with another implementation which does include the option, though perhaps with reduced functionality. In the same vein an implementation which does include a particular option MUST be prepared to interoperate with another implementation which does not include the option (except, of course, for the feature the option provides.)

1 Introduction

This document provides the details of the acceptance test cases. These test cases are defined to verify that whatever is handed over from facility sites to WP4 is commonly functioning.

2 Pre-qualification Test Cases

Pre-qualification tests are executed by the facility owners. Then the successful results trigger the engagement of WP4 for benchmarking & performance testing.

2.1 Transport Network

The pre-qualification testing for the transport network is focusing on the network reachability and the link specifications between the Radio Access Site & the facility Data Center; where the other control plane functions and the mobile core network are hosted.

Table 1 – Transport Network Pre-qualification Test Cases

Domain		**Transport Network**		
Enrolled Units		**Cell Site Router (CSR); Data Center Gateway (DC-GW)**		
Reference Documents		Deliverable D1.1 (Design of infrastructure architecture and subsystems v1); D2.1 (5G-VINNI Solution facility sites High Level Design (HLD) - v1)		
Illustration		Figure 1.1		
Aspect	**ID**	**Test Description**	**Test Conditions**	**Result**
Connectivity & Reachability	1.1.1	Physical connectivity between the Cell Site Routers and the DC Gateway exists		
	1.1.2	L2/L3 connectivity between the Cell Site Routers and the DC Gateway (i.e. ICMP Ping) exists		
Link Throughput	1.2.1	Link between CSR & DC-GW can support minimum 10Gbps traffic throughput with max 1% packet loss *(the pass/fail parameters are discussable)*	The throughput depends on: (1) The number of CSRs (2) the expected uplink/ downlink traffic passing through each CSR or DC-GW	
Link Latency	1.3.1	Link latency between CSR & DC-GW is less than 50ms. *(the pass/fail parameters are discussable)*	The technical requirements of the use cases has to be considered	
Link Redundancy &	1.4.1	The connectivity between CSR & DC-GW is tolerant for		

Reliability		at least one physical link failure. *(the pass/fail parameters are discussable)*		
Physical ports availability	1.5.1	CSR & DC-GW have at least one free physical port for tester connectivity purposes (Media & connector type will be determined by KEYS)		
Management network access	1.6.1	Management access to each CSR & DC-GW routers including: • SSH CLI • syslog • SNMP • telemetry gRPC		

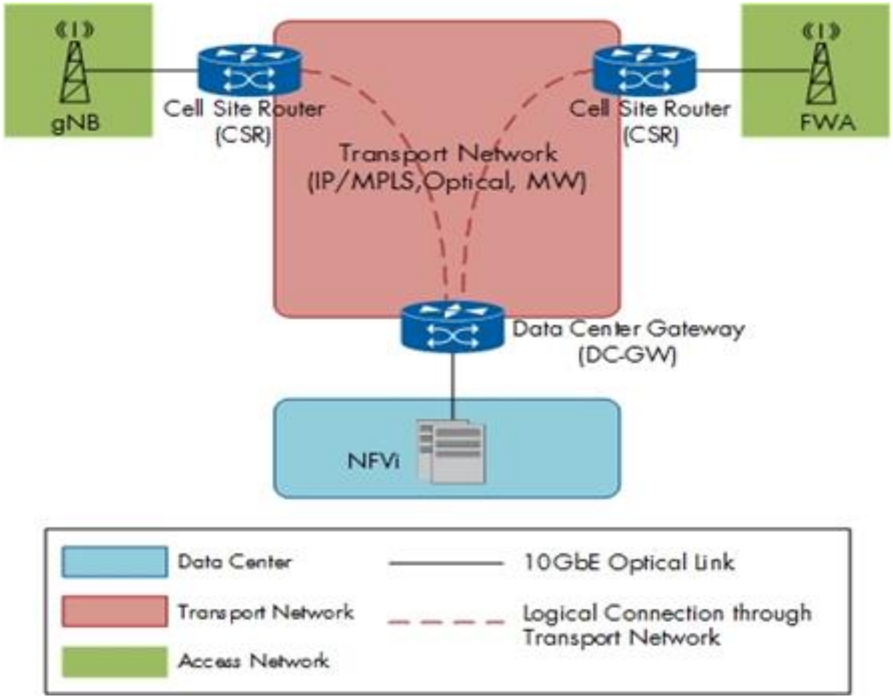


Figure 1 – Transport Network Test Architecture

2.2 5G New Radio

Table 2 – New Radio Pre-qualification Test Cases

Domain		**5G New Radio (NR)**		
Enrolled Units		**5G gNB Cell Site**		
Reference Documents		Deliverable D1.1 (Design of infrastructure architecture and subsystems v1); D2.1 (5G-VINNI Solution facility sites High Level Design (HLD) - v1)		
Illustration		Figure 2.1		
Aspect	**ID**	**Test Description**	**Test Conditions**	**Result**
Connectivity & Reachability (Option 3/3a/3x)	2.1.1	Network reachability is verified on (S1-U) Interface	gNB existence	
	2.1.2	Network reachability is verified on (X2-U) Interface	gNB existence	
	2.1.3	Network reachability is verified on (X2-C) Interface	gNB existence	
	2.1.4	Network reachability is verified on (F1-U) Interface	Radio Split (DU-CU)	
	2.1.5	Network reachability is verified on (F1-C) Interface	Radio Split (DU-CU)	
	2.1.6	Network reachability is verified on (E1) Interface	Central Unit (CU) UP/CP Decoupling	
Radio Network Performance	2.2.1	Access radio network supports minimum 1Gbps DL/UP for at least one connected UE *(the pass/fail parameters are discussable)*	Testing device is capable to support 1Gbps	
	2.2.2	Access radio network supports maximum 10msec latency DL/UP for at least one connected UE *(the pass/fail parameters are discussable)*		
5G NR Features ***(Using EN-DC Test Device)***	2.3.1	UE supports EN Dual Connectivity (EN-DC)		
	2.3.2	gNB support UL/DL and signal that with the UE devices		

	2.3.3	Handover between gNBs is signalled based on 3GPP standard *(the pass/fail parameters are discussable and depends on the use case)*		

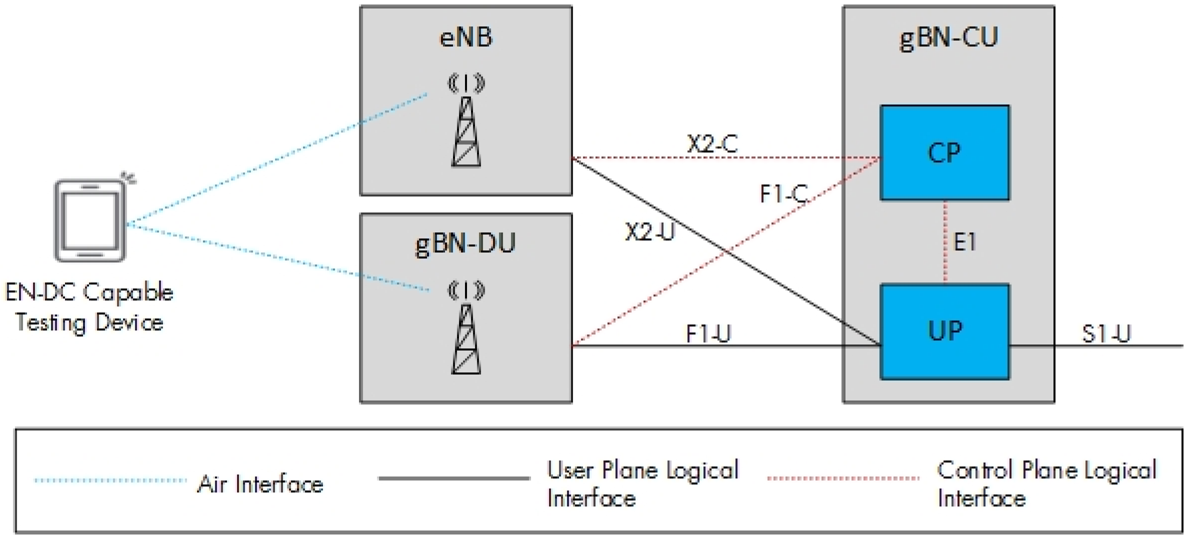


Figure 2 – 5G New Radio Test Architecture

2.3 Core Network

Table 3 – NSA Core Network Pre-qualification Test Cases

Domain		**Core Network (NSA EPC)**		
Enrolled Units		**Virtual EPC**		
Reference Documents		Deliverable D1.1 (Design of infrastructure architecture and subsystems v1); D2.1 (5G-VINNI Solution facility sites High Level Design (HLD) - v1)		
Illustration		Figure 3.1; 3.2		
Aspect	**ID**	**Test Description**	**Test Conditions**	**Result**
Network Slicing (DECOR)	3.1.1	Dedicated MME/S-GW/P-GW virtual instances are instantiated, each to serve one of the 3 network slices provided (eMMB, mMTC,uRLLC)	Depends on the use cases that are provided by each facility	
	3.1.2	S-GW & P-GW virtual instances are instantiated based on Control User Plane Separation (CUPS) for uRLLC slice		
	3.1.3	User plane path (SGW-U & PGW-U) supports minimum 1Gbps traffic throughput for eMMB slice dedicated bearer, including (S1-U, S5 & SGi interfaces) *(the pass/fail parameters are discussable)*		
	3.1.4	User plane path (SGW-U & PGW-U) guarantees less than 10msec for uRLL slice dedicated bearer, including (S1-U, S5 & SGi interfaces) *(the pass/fail parameters are discussable)*		
Connectivity & Reachability	3.2.1	L3/L7 reachability is verified on (S1-MME) interface		
	3.2.2	L3/L7 reachability is verified on (S6a) interface		
	3.2.3	L3/L7 reachability is verified on (S11) interface		

	3.2.4	L3/L7 reachability is verified on (GTP S5) interface		
	3.2.5	L3/L7 reachability is verified on (S1-U) interface between gNB & S-GW		
	3.2.6	L3/L7 reachability is verified on (Sx) interface for S/PGW CP-UP	(Mandatory) for uLLRC slice. (Optional) for the other Slices	
	3.2.7	L3/L7 reachability is verified on (Gx) interface		
	3.2.8	L3/L7 reachability is verified on (SGi) interface		
NSA Core Support	3.3.1	MME supports attach procedures with set DCNR bit		
	3.3.2	Verify E2E EPS bearer procedures (UE-gNB-MME-PDN) for each network slice		
Testing Tools (Emulated RAN) Integration	3.4.1	Satisfy the NFVi requirements for the virtual testing tools (vCPUs, RAM, vNIC) which will be specified by KEYS		
	3.4.2	Virtual testing tools (emulated RAN) have network reachability (Inter-VM) to the vEPC through (S1-MME & S1-U) interfaces		
	3.4.3	Physical testing tools (emulated RAN) have network reachability to the vEPC through (S1-MME & S1-U) interfaces		
	3.4.4	Management access to testing tools including: • HTTPs • SSH CLI • syslog • SNMP		

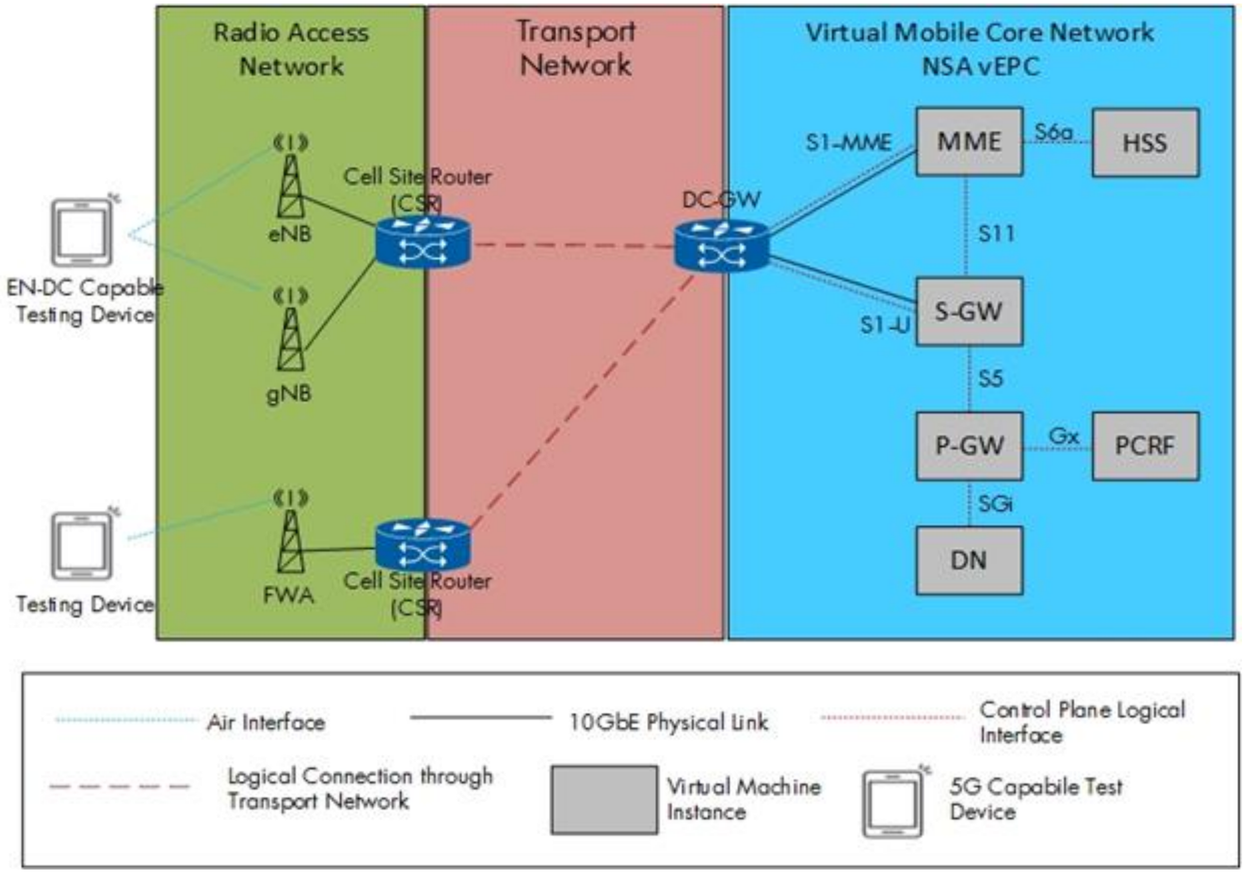


Figure 3 – E2E NSA Testing Architecture

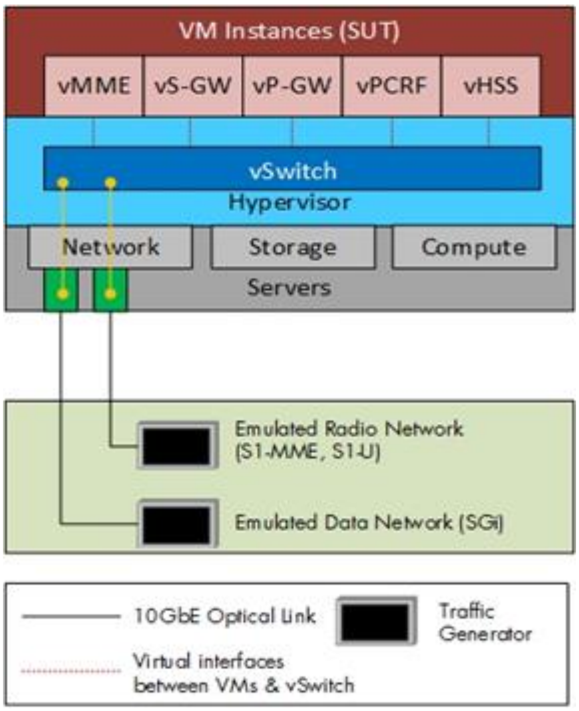


Figure 4 – NSA Core Wrap-around Testing Architecture

2.4 MANO & NFVI

Table 4 – MANO & NFVI Pre-qualification Test Cases

Domain		**MANO & NFVI**		
Enrolled Units		**NFVO, NFVM, VIM, EM, Physical Servers**		
Reference Documents		Deliverable D1.1 (Design of infrastructure architecture and subsystems v1); D2.1 (5G-VINNI Solution facility sites High Level Design (HLD) - v1)		
Illustration		Figure 4.1		
Aspect	**ID**	**Test Description**	**Test Conditions**	**Result**
API Reachability	4.1.1	All MANO components have HTTP/HTTPS through the management network including (NFVO, VNFMs, VIM)		
	4.1.2	Verify the MANO interfaces API reachability		
	4.1.3	Verify the network reachability of the API testing server to the facility management network through IPSec VPN Tunnel		
	4.1.4	Verify the API testing server HTTP/HTTPS reachability to the MANO components		
NFVO Templates	4.2.1	Sharing the VNFD templates for all the NFVs with EANTC		
	4.2.2	Sharing the NSD templates with EANTC		
Infrastructure Inspection & VNF Creation	4.3.1	Verify VIM inspection for the server's hardware components (CPU, RAM, Storage, NIC, …)		
	4.3.2	Verify VNF creation and instantiation through VIM		
	4.3.3	Verify VIM Communication with SDN controller for the VNF's network creation		

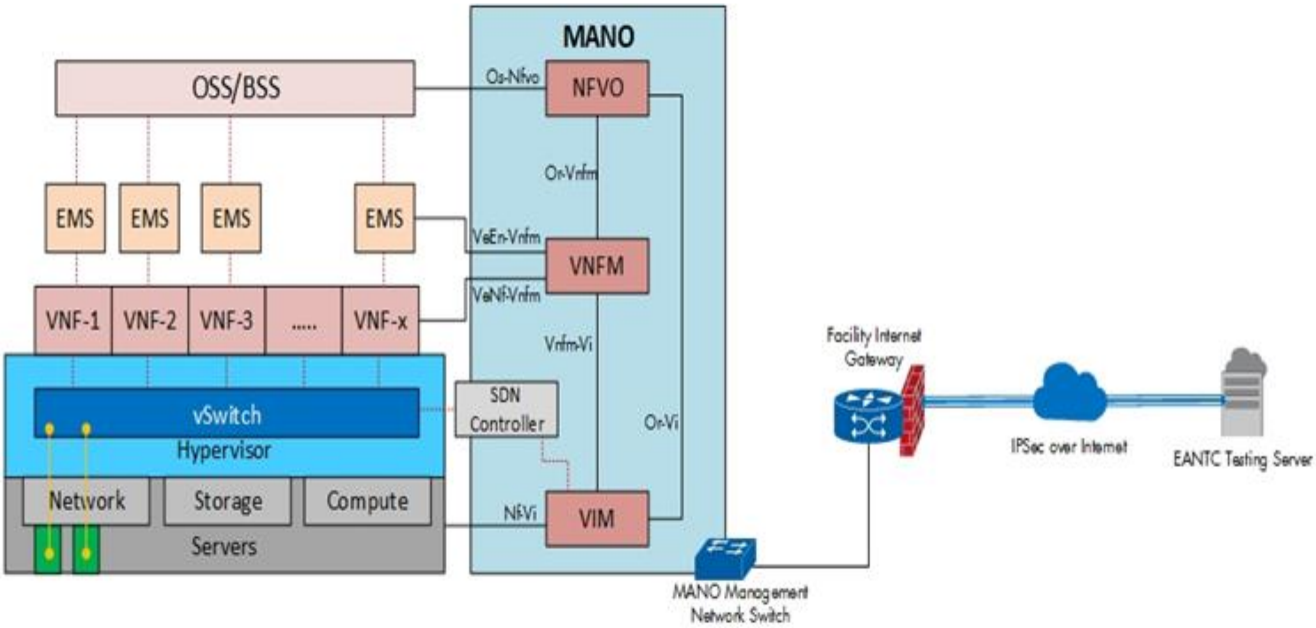


Figure 5 – MANO & NFVI Testing Architecture

References

[1] https://www.5G-VINNI.eu/ © 5G-VINNI consortium 2018

[2] "Study on New Radio (NR) access technology", 3GPP TR 38.912 Release 15;

[3] "Design of network slicing and supporting systems v1", 5GVINNI Deliverable D1.2;

[4] "Design for systems and interfaces for slice operation v1", 5GVINNI Deliverable D1.3;

[5] "Specification of services delivered by each of the 5G-VINNI facilities", 5GVINNI Deliverable D3.1;

[6] "5GVINNI Test Methodologies and Test Cases", 5GVINNI Deliverable D4.1 Annex A - Test Specification;

[7] "Benchmarking Methodology for Network Interconnect Devices", RFC 2544;

[8] "Ethernet service activation test methodology", ITU-T Y.1564;

[end of document]

Deliverable D4.1 Annex D

5G-VINNI Test Cases Handbook

Editor:	EANTC
Deliverable nature:	Report (R)
Dissemination level: (Confidentiality)	Public (PU)
Contractual delivery date:	31st February 2021
Actual delivery date:	
Suggested readers:	ICT-19 projects, industry verticals seeking to conduct Testing on 5G systems
Version:	0.1
Total number of pages:	
Keywords:	5G, KPI, Use Cases, Test Plan, Test Framework,

Abstract

[End of abstract]

Disclaimer

This document contains material, which is the copyright of certain 5G-VINNI consortium parties, and may not be reproduced or copied without permission.

In case of Public (PU): All 5G-VINNI consortium parties have agreed to full publication of this document.

In case of Restricted to Programme (PP): All 5G-VINNI consortium parties have agreed to make this document available on request to other framework programme participants.

In case of Restricted to Group (RE): All 5G-VINNI consortium parties have agreed to full publication of this document. However this document is written for being used by <organisation / other project / company etc.> as <a contribution to standardisation / material for consideration in product development etc.>.

In case of Consortium confidential (CO): The information contained in this document is the proprietary confidential information of the 5G-VINNI consortium and may not be disclosed except in accordance with the consortium agreement.

The commercial use of any information contained in this document may require a license from the proprietor of that information.

Neither the 5G-VINNI consortium as a whole, nor a certain part of the 5G-VINNI consortium, warrant that the information contained in this document is capable of use, nor that use of the information is free from risk, accepting no liability for loss or damage suffered by any person using this information.

The EC flag in this document is owned by the European Commission and the 5G PPP logo is owned by the 5G PPP initiative. The use of the flag and the 5G PPP logo reflects that 5G-VINNI receives funding from the European Commission, integrated in its 5G PPP initiative. Apart from this, the European Commission or the 5G PPP initiative have no responsibility for the content.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 815279.

Impressum

Full project title	5G Verticals Innovation Infrastructure
Project acronym	5G-VINNI
Number and title of work-package	WP4 Execution and Validation of Use Cases and KPIs on 5GVINNI End-to-End Facility
Number and title of task(s)	T4.1 Definition of Test Methods and Test Cases for validation of network service-level KPIs and use case-level KPIs
Document title	5GVINNI Test Specification
Editor: Name, company	Andrea Cattoni, Keysight
Work-package leader: Name, company	Andrea Cattoni, Keysight

Copyright notice

List of authors

Company	Author	Contribution
EANTC		

Table of Contents

List of authors ... 3
Table of Contents ... 4
List of figures ... 8
List of tables ... 9
Abbreviations ... 11
Definitions ... 12
1 Introduction ... 13
2 Unit Tests for AAA ... 14
2.1 Radius ... 14
2.2 Diameter ... 14
2.3 HSS ... 14
2.4 Policy Management ... 14
3 Unit Tests for Inter-Facility Connectivity ... 15
4 Unit Tests for Backhaul ... 16
5 Unit Tests for gNodeB ... 17
5.1 Average Downlink Latency in gNodeB ... 17
5.2 Average delay DL air-interface ... 17
5.3 DL Total PRB usage ... 17
5.4 UL Total PRB usage ... 17
5.5 Average DL UE throughput in gNB ... 17
5.6 Average UL UE throughput in gNB ... 17
5.7 Volume of unrestricted DL UE data in gNB ... 17
5.8 Volume of unrestricted UL UE data in gNB ... 17
5.9 Mean number of RRC Connections ... 18
5.10 Max number of RRC Connections ... 18
5.11 UL Packet Loss Rate ... 18
5.12 UL F1-U Packet Loss Rate ... 18
5.13 DL F1-U Packet Loss Rate ... 18
5.14 DL Packet Drop Rate in gNB-CU-UP ... 18
5.15 DL Packet Drop Rate in gNB-DU ... 18
5.16 Average delay DL in CU-UP ... 19
5.17 Average delay on F1-U ... 19
5.18 Average delay DL in gNB-DU ... 19
5.19 IP Latency DL in gNB-DU ... 19
5.20 UE Context Release Request (gNB-DU initiated) ... 19
5.21 Number of UE Context Release Requests (gNB-CU initiated) ... 19
6 Unit Tests for 5GC Network Functions ... 20
6.1 Unit tests for AMF ... 20
6.1.1 Mean/Max number of registered subscribers ... 20
6.1.2 Number of initial registration requests ... 20

6.1.3 Number of successful initial registrations 20
6.1.4 Number of mobility registration update requests 20
6.1.5 Number of successful mobility registration updates 20
6.1.6 Number of periodic registration update requests 20
6.1.7 Number of successful periodic registration updates 20
6.1.8 Number of emergency registration requests 20
6.1.9 Number of successful emergency registrations 20
6.2 Unit Tests for SMF 20
6.2.1 Mean/Max number of PDU sessions 20
6.2.2 Number of PDU session creation requests 20
6.2.3 Number of successful PDU session creations 20
6.2.4 Number of failed PDU session creations 20
6.3 Unit Tests for UPF 20
6.3.1 Number of incoming GTP data packets on the N3 interface, from (R)AN to UPF 20
6.3.2 Number of outgoing GTP data packets of on the N3 interface, from UPF to (R)AN 20
6.3.3 Number of octets of incoming GTP data packets on the N3 interface, from (R)AN to UP21
6.3.4 Number of octets of outgoing GTP data packets on the N3 interface, from UPF to (R)A21
6.3.5 N6 incoming link usage 21
6.3.6 N6 outgoing link usage 21
6.4 Unit Tests for PCF 21
6.4.1 Number of AM policy association requests 21
6.4.2 Number of successful AM policy associations 21
6.4.3 Number of SM policy association requests 21
6.4.4 Number of successful SM policy associations 21
6.5 Unit Tests for UDM 21
6.5.1 Mean/Max number of registered subscribers through UDM 21
7 Unit Tests for NFVI/MANO 22
7.1 Preambles 22
7.1.1 Pre-Conditions 22
7.1.2 Test Environment Setup 22
7.1.3 Pre-Qualification Test Cases 23
7.2 Test Parameters 23
7.3 Postambles 24
7.4 Test Cases 24
7.4.1 Software Image Management 24
7.4.2 Fault Management 29
7.4.3 Virtualised Resource Performance Management 34
8 Sub-module Tests for NFV Performance Test 40
8.1 Preambles 40
8.1.1 Pre-Conditions 40
8.1.2 Test Environment Setup 41
8.1.3 Pre-Qualification Test Cases 42

8.2 Test Parameters 42
8.3 Measurements 42
8.4 Key Performance Indicators (KPIs) 43
8.5 Postambles 43
8.6 Test Cases 43
8.6.1 NFVI Network Performance Measurement........ 43
8.6.2 NFVI Virtual Resource Performance Measurement 45
8.6.3 Network Service Autoscaling Validation 46
8.6.4 Network Service Speed of Activation Test 49
9 Sub-module Tests for RAN........ 52
9.1 Preambles........ 52
9.1.1 Pre-Conditions........ 52
9.1.2 Test Environment Setup 53
9.1.3 Pre-Qualification Test Cases........ 53
9.2 Test Parameters 53
9.3 Measurements 54
9.4 Key Performance Indicators (KPIs) 54
9.5 Postambles 55
9.6 Test Cases 55
9.6.1 RAN Network Performance Measurement 55
10 Sub-module Tests for vEPC NSA 58
10.1 Preambles........ 58
10.1.1 Pre-Conditions........ 58
10.1.2 Test Environment Setup 58
10.1.3 Pre-Qualification Test Cases........ 58
10.2 Test Parameters 59
10.3 Measurements 59
10.4 Key Performance Indicators (KPIs) 60
10.5 Postambles 60
10.6 Test Cases 60
10.6.1 vEPC NSA Network Performance Measurement........ 60
10.6.2 vEPC NSA Network Capacity Measurement 62
11 E2E Tests........ 65
11.1 Preambles........ 65
11.1.1 Pre-Conditions........ 65
11.1.2 Test Environment Setup 65
11.1.3 Pre-Qualification Test Cases........ 66
11.2 Test Parameters 66
11.3 Measurements 67
11.4 Key Performance Indicators (KPIs) 67
11.5 Postambles 68
11.6 Test Cases 68

11.6.1 E2E eMBB Performance Measurement....68
11.6.2 E2E uRLLC Performance Measurement....70
11.6.3 E2E mMTC Performance Measurement....71
11.6.4 E2E eMBB Scalability Measurement....73
11.6.5 E2E uRLLC Scalability Measurement....74
11.6.6 E2E mMTC Scalability Measurement....76
References....78

List of figures

Figure 2: NFVI/MANO Interoperability test setup 23
Figure 3: NFVI performance test setup 41
Figure 4: Network Service performance test setup 41
Figure 5: End-To-End test setup 66

List of tables

Table 1: NFVI/MANO Interoperability Test Conditions 22
Table 3: Pre-qualification test cases for NFVI/MANO Interoperability Tests 23
Table 4: NFVI/MANO Interoperability Test Parameters 23
Table 5: Parameters for Add software image test 24
Table 6: Procedures & Expected Results for Add software image test 24
Table 7: Parameters for query software image by NFVO test 25
Table 8: Procedures & Expected Results for query software image by NFVO test 25
Table 9: Parameters for query software image by VNFM test 26
Table 10: Procedures & Expected Results for query software image by VNFM test 26
Table 11: Parameters for update software image test 27
Table 12: Procedures & Expected Results for update software image test 28
Table 13: Parameters for delete software image test 29
Table 14: Procedures & Expected Results for delete software image test 29
Table 15: Parameters for virtualised resource fault alarm notification test 30
Table 16: Procedures & Expected Results for virtualised resource fault alarm notification test 30
Table 17: Parameters for virtualised resource fault alarm clearance test 31
Table 18: Procedures & Expected Results for virtualised resource fault alarm clearance test 31
Table 19: Parameters for VNF fault alarm notification test 32
Table 20: Procedures & Expected Results for VNF fault alarm notification test 32
Table 21: Parameters for VNF fault alarm clearance test 33
Table 22: Procedures & Expected Results for VNF fault alarm clearance test 33
Table 23: Parameters for VR PM Job Creation and Notification Monitoring test 34
Table 24: Procedures & Expected Results for VR PM Job Creation and Notification Monitoring test . 34
Table 25: Parameters for VR PM Job Creation and Threshold Monitoring test 35
Table 26: Procedures & Expected Results for VR PM Job Creation and Threshold Monitoring test 36
Table 27: Parameters for VR PM Job Deletion test 37
Table 28: Procedures & Expected Results for VR PM Job Deletion test 37
Table 29: Parameters for VR PM Threshold Deletion test 38
Table 30: Procedures & Expected Results for VR PM Threshold Deletion test 38
Table 31: NFV Test Conditions on Network Components and Services 40
Table 32: Pre-qualification test cases for NFV Performance Tests 42
Table 33: NFV Performance Test Parameters 42
Table 34: NFV Performance Test Measurements and KPI Computation Procedures 42
Table 35: NFV Performance Test KPIs 43
Table 36: Parameters for NFVI network performance measurement 44
Table 37: Procedures & Expected Results for NFVI network performance measurement 44
Table 38: Parameters for NFVI virtual resource performance measurement 45
Table 39: Procedures & Expected Results for NFVI virtual resource performance measurement 46
Table 40: Parameters for network service autoscaling validation 47
Table 41: Procedures & Expected Results for network service autoscaling validation 47

Table 42: Parameters for network service Speed of Activation test 50
Table 43: Procedures & Expected Results for network service Speed of Activation test 50
Table 44: End-To-End Test Conditions on Network Components and Services 65
Table 45: Pre-qualification test cases for End-To-End Tests 66
Table 46: End-To-End Test Parameters 67
Table 47: Measurements and KPI Computation Procedures 67
Table 48: End-To-End Test KPIs 67
Table 49: Parameters for E2E eMBB performance measurement 69
Table 50: Procedures & Expected Results for E2E eMBB performance measurement 69
Table 51: Parameters for E2E uRLLC performance measurement 70
Table 52: Procedures & Expected Results for E2E uRLLC performance measurement 70
Table 53: Parameters for E2E mMTC performance measurement 72
Table 54: Procedures & Expected Results for E2E mMTC performance measurement 72
Table 49: Parameters for E2E eMBB scalability measurement 74
Table 50: Procedures & Expected Results for E2E eMBB scalability measurement 74
Table 51: Parameters for E2E uRLLC scalability measurement 75
Table 52: Procedures & Expected Results for E2E uRLLC scalability measurement 75
Table 53: Parameters for E2E mMTC scalability measurement 76
Table 54: Procedures & Expected Results for E2E mMTC scalability measurement 76

Abbreviations

A list of abbreviations is strongly recommended

5G	Fifth Generation (mobile/cellular networks)
5G PPP	5G Infrastructure Public Private Partnership
E2E	End-to-End
IoT	Internet of Things
KPI	Key Performance Indicator
MEC	Multi-Access Edge Computing
NFV	Network Function Virtualization
OPEX	Operational Expenditure
OPEX	Operational Expenditure
PPDR	Public Protection Disaster Relief
QoE	Quality of Experience
QoS	Quality of Service
R&D	Research and Development
RAN	Radio Access Network
SDN	Software Defined Networks
URLLC	Ultra-Reliable Low Latency Communications
VNF	Virtual Network Function

Definitions

This document contains specific terms to identify elements and functions that are considered to be mandatory, strongly recommended or optional. These terms have been adopted for use similar to that in IETF RFC2119, and have the following definitions.

MUST This word, or the terms "REQUIRED" or "SHALL", mean that the definition is an absolute requirement of the specification.

MUST NOT This phrase, or the phrase "SHALL NOT", mean that the definition is an absolute prohibition of the specification.

SHOULD This word, or the adjective "RECOMMENDED", mean that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.

SHOULD NOT This phrase, or the phrase "NOT RECOMMENDED" mean that there may exist valid reasons in particular circumstances when the particular behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.

MAY This word, or the adjective "OPTIONAL", mean that an item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because the vendor feels that it enhances the product while another vendor may omit the same item. An implementation which does not include a particular option MUST be prepared to interoperate with another implementation which does include the option, though perhaps with reduced functionality. In the same vein an implementation which does include a particular option MUST be prepared to interoperate with another implementation which does not include the option (except, of course, for the feature the option provides.)

1 Introduction

This document will contain the specifications for all the test cases developed within the scope of the 5G-VINNI project.

2 Unit Tests for AAA

N/A in Release 0.

2.1 Radius

2.2 Diameter

2.3 HSS

2.4 Policy Management

3 Unit Tests for Inter-Facility Connectivity

N/A in Release 0.

Interconnection between 5GVINNI Facility Sites

D1.1 Section 5 “Interconnection and Interoperability of 5G-VINNI sites”

Test:

- Throughput with RFC2544
- QoS differentiation (prioritization of traffic, this will depend on the solution used, e.g. SD-WAN, Internet, MPLS-VPN)

4 Unit Tests for Backhaul

N/A in Release 0.

5 Unit Tests for gNodeB

N/A in Release 0.

3GPP TS 28.552 V15.0.0 (2018-09) section 5.1

5.1 Average Downlink Latency in gNodeB

DU (Distributed Unit)

5.2 Average delay DL air-interface

This measurement provides the average (arithmetic mean) time it takes to get a response back on a HARQ transmission in the downlink direction. The measurement is optionally split into subcounters per QoS level (mapped 5QI or QCI in NR option 3).

5.3 DL Total PRB usage

This measurement provides the total usage (in percentage) of physical resource blocks (PRBs) on the downlink for any purpose.

5.4 UL Total PRB usage

This measurement provides the total usage (in percentage) of physical resource blocks (PRBs) on the uplink for any purpose.

5.5 Average DL UE throughput in gNB

This measurement provides the average UE throughput in downlink. This measurement is intended for data bursts that are large enough to require transmissions to be split across multiple slots. The UE data volume refers to the total volume scheduled for each UE regardless if using only primary- or also supplemental aggregated carriers. The measurement is optionally split into subcounters per QoS level (mapped 5QI or QCI in NR option 3)

5.6 Average UL UE throughput in gNB

This measurement provides the average UE throughput in uplink. This measurement is intended for data bursts that are large enough to require transmissions to be split across multiple slots. The UE data volume refers to the total volume scheduled for each UE regardless if using only primary- or also supplemental aggregated carriers. The measurement is optionally split into subcounters per QoS level (mapped 5QI or QCI in NR option 3).

5.7 Volume of unrestricted DL UE data in gNB

This measurement provides the percentage of DL data volume for UEs in the cell that is classified as unrestricted, i.e., when the volume is so low that all data can be transferred in one slot and no UE throughput sample could be calculated. The UE data volume refers to the total volume scheduled for each UE regardless if using only primary- or also supplemental aggregated carriers. The measurement is optionally split into subcounters per QoS level (mapped 5QI or QCI in NR option 3).

5.8 Volume of unrestricted UL UE data in gNB

This measurement provides the percentage of UL data volume for UEs in the cell that is classified as unrestricted, i.e., when the volume is so low that all data can be transferred in one slot and no UE throughput sample could be calculated. The UE data volume refers to the total volume scheduled for each UE regardless if using only primary- or also supplemental aggregated carriers. The measurement is optionally split into subcounters per QoS level (mapped 5QI or QCI in NR option 3).

5.9 Mean number of RRC Connections

This measurement provides the mean number of users in RRC connected mode during each granularity period

5.10 Max number of RRC Connections

This measurement provides the maximum number of users in RRC connected mode during each granularity period

5.11 UL Packet Loss Rate

Split gNB Performance Measurement. This measurement provides the fraction of PDCP SDU packets which are not successfully received at gNB-CU-UP. It is a measure of the UL packet loss including any packet losses in the air interface, in the gNB-CU and on the F1-U interface. Only user-plane traffic (DTCH) and only PDCP SDUs that have entered PDCP (and given a PDCP sequence number) are considered. The measurement is optionally split into subcounters per QoS level (mapped 5QI or QCI in NR option 3).

5.12 UL F1-U Packet Loss Rate

Split gNB Performance Measurement. This measurement provides the fraction of PDCP SDU packets which are not successfully received at gNB-CU-UP. It is a measure of the UL packet loss on the F1-U interface. The measurement is optionally split into subcounters per QoS level (mapped 5QI or QCI in NR option 3)

5.13 DL F1-U Packet Loss Rate

Split gNB Performance Measurement. This measurement provides the fraction of PDCP SDU packets which are not successfully received at the gNB-CU). It is a measure of the DL packet loss on the F1-U interface. The measurement is optionally split into subcounters per QoS level (mapped 5QI or QCI in NR option 3).

5.14 DL Packet Drop Rate in gNB-CU-UP

Split gNB Performance Measurement. This measurement provides the fraction of PDCP SDU packets which are dropped on the downlink, due to congestion, traffic management etc in the gNB-CU-UP. Only user-plane traffic (DTCH) is considered. A dropped packet is one whose context is removed from the gNB-CU-UP without any part of it having been transmitted on the F1-U or Xn-U or X2-U interface. The measurement is optionally split into subcounters per QoS level (mapped 5QI or QCI in NR option 3).

5.15 DL Packet Drop Rate in gNB-DU

Split gNB Performance Measurement. This measurement provides the fraction of RLC SDU packets which are dropped on the downlink, due to congestion, traffic management etc in the gNB-DU. Only user-plane traffic (DTCH) is considered. A dropped packet is one whose context is removed from the gNB-DU without any part of it having been transmitted on the air interface. The measurement is optionally split into subcounters per QoS level (mapped 5QI or QCI in NR option 3)

5.16 Average delay DL in CU-UP

Split gNB Performance Measurement. This measurement provides the average (arithmetic mean) PDCP SDU delay on the downlink within the gNB-CU-UP, for all PDCP packets. The measurement is optionally split into subcounters per QoS level (mapped 5QI or QCI in NR option 3).

5.17 Average delay on F1-U

Split gNB Performance Measurement. This measurement provides the average (arithmetic mean) GTP packet delay on the F1-U interface. The measurement is optionally split into subcounters per QoS level (mapped 5QI or QCI in NR option 3).

5.18 Average delay DL in gNB-DU

Split gNB Performance Measurement. This measurement provides the average (arithmetic mean) RLC SDU delay on the downlink within the gNB-DU, for initial transmission of all RLC packets. The measurement is optionally split into subcounters per QoS level (mapped 5QI or QCI in NR option 3).

5.19 IP Latency DL in gNB-DU

Split gNB Performance Measurement. This measurement provides the average IP Latency in DL (arithmetic mean) within the gNB-DU, when there is no other prior data to be transmitted to the same UE in the gNB-DU. The measurement is optionally split into subcounters per QoS level

5.20 UE Context Release Request (gNB-DU initiated)

Split gNB Performance Measurement. This measurement provides the number of UE CONTEXT Release initiated by gNB-DU for each release cause.

5.21 Number of UE Context Release Requests (gNB-CU initiated)

Split gNB Performance Measurement. This measurement provides the number of UE CONTEXT RELEASE initiated by gNB-CU for each release cause.

6 Unit Tests for 5GC Network Functions

UE user equipment

6.1 Unit tests for AMF

Access and Mobility management function

3GPP TS 28.552 V15.0.0 (2018-09) section 5.2

6.1.1 Mean/Max number of registered subscribers

6.1.2 Number of initial registration requests

6.1.3 Number of successful initial registrations

6.1.4 Number of mobility registration update requests

6.1.5 Number of successful mobility registration updates

6.1.6 Number of periodic registration update requests

6.1.7 Number of successful periodic registration updates

6.1.8 Number of emergency registration requests

6.1.9 Number of successful emergency registrations

6.2 Unit Tests for SMF

Session Management Function

3GPP TS 28.552 V15.0.0 (2018-09) section 5.3

6.2.1 Mean/Max number of PDU sessions

6.2.2 Number of PDU session creation requests

6.2.3 Number of successful PDU session creations

6.2.4 Number of failed PDU session creations

6.3 Unit Tests for UPF

User Plane Function

3GPP TS 28.552 V15.0.0 (2018-09) section 5.4

6.3.1 Number of incoming GTP data packets on the N3 interface, from (R)AN to UPF

This measurement provides the number of GTP data PDUs on the N3 interface which have been accepted and processed by the GTP-U protocol entity on the N3 interface

6.3.2 Number of outgoing GTP data packets of on the N3 interface, from UPF to (R)AN

This measurement provides the number of GTP data PDUs on the N3 interface which have been generated by the GTP-U protocol entity on the N3 interface

6.3.3 Number of octets of incoming GTP data packets on the N3 interface, from (R)AN to UP

This measurement provides the number of octets of incoming GTP data packets on the N3 interface which have been generated by the GTP-U protocol entity on the N3 interface

6.3.4 Number of octets of outgoing GTP data packets on the N3 interface, from UPF to (R)A

This measurement provides the number of octets of outgoing GTP data packets on the N3 interface which have been generated by the GTP-U protocol entity on the N3 interface

6.3.5 N6 incoming link usage

This measurement provides the PDU-layer incoming link usage of N6 interface

6.3.6 N6 outgoing link usage

This measurement provides the PDU-layer outcoming link usage of N6 interface

6.4 Unit Tests for PCF

Policy Control Function

3GPP TS 28.552 V15.0.0 (2018-09) section 5.5

6.4.1 Number of AM policy association requests

6.4.2 Number of successful AM policy associations

6.4.3 Number of SM policy association requests

6.4.4 Number of successful SM policy associations

6.5 Unit Tests for UDM

Unified Data Management

3GPP TS 28.552 V15.0.0 (2018-09) section 5.6

6.5.1 Mean/Max number of registered subscribers through UDM

This measurement provides the mean/max number of registered state subscribers to UDM per network slice instance

7 Unit Tests for NFVI/MANO

7.1 Preambles

7.1.1 Pre-Conditions

The NFVI and MANO must be deployed as designed in 5G VINNI Deliverables [9][10]. The test controlling system and test automation tools are installed and connected to the System Under Test (SUT).

The test conditions on network components and services are listed in Table 1.

Table 1: NFVI/MANO Interoperability Test Conditions

Components	Optional Conditions	Test Condition 1	Test Condition 2
Service Orchestration	Nokia Flowone	✓	✓
EMS	Ericssion	✓	
VNF	Ericsson	✓	
	Samsung		✓
NFVI Hyperviser	Nokia NCIR	✓	
NFVO	Nokia CBND	✓	
	Samsung CMS		✓
VNFM	Ericsson ENM /Nokia CBAM	✓	
	Samsung CMS		✓
VIM	Nokia NCIR	✓	
	HP Openstack		✓
SDN Controller	Nuage	✓	✓
Physical Devices	HP		✓
	Nokia Airframe OR	✓	
	Nuage WBX	✓	

7.1.2 Test Environment Setup

The NFVI/MANO Interoperability test setup is shown in Figure 1.

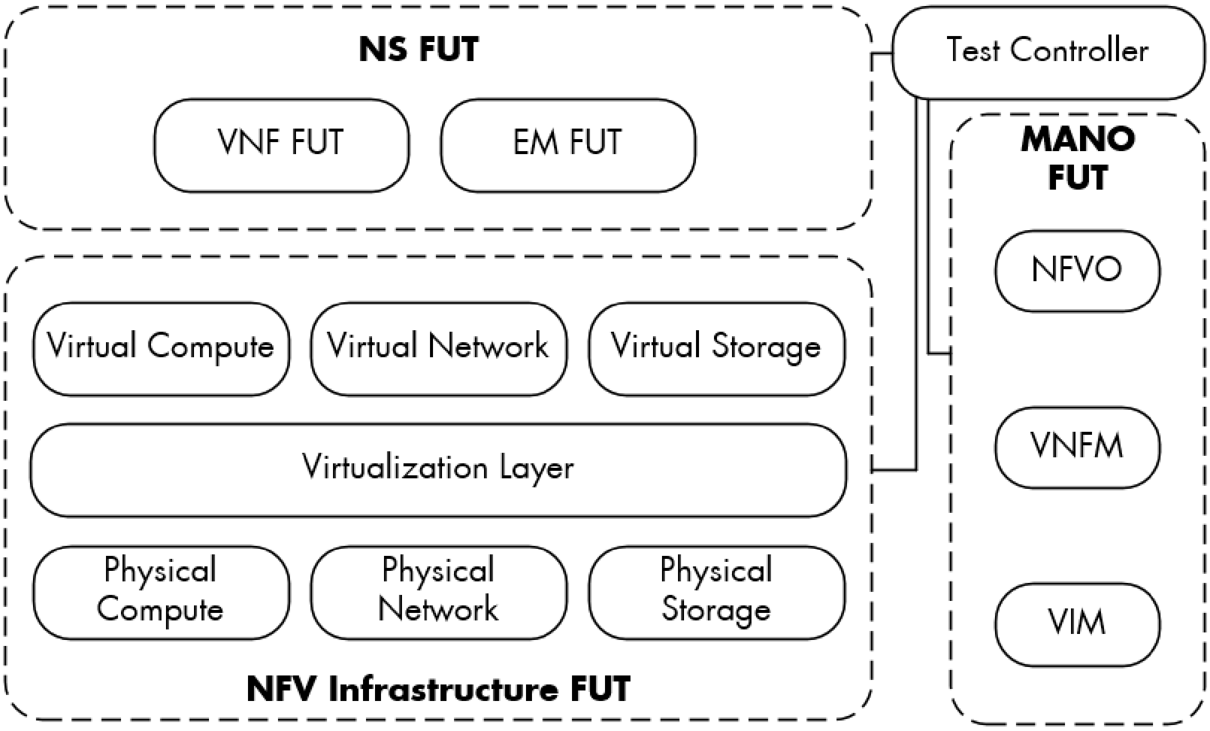


Figure 1: NFVI/MANO Interoperability test setup

7.1.3 Pre-Qualification Test Cases

All Pre-Qualification test cases, as defined in 5GVINNI Deliverable [6] section 2.1.2, shown in Table 2, under 'Mandatory Tests' and 'Conditional Tests' must be passed.

- Each facility site must comply with one of the conditions defined in Table 1.
- Conditional Tests must also be tested and passed for the sites complying with the condition.
- Optional Tests can be optionally tested based on the capability of the site.

Table 2: Pre-qualification test cases for NFVI/MANO Interoperability Tests

Domain	Mandatory Tests	Conditional Tests (Condition 1)	Conditional Tests (Condition 2)	Optional Tests
MANO & NFVI	4.1.1-4.1.4, 4.2.1, 4.2.2, 4.3.1-4.3.3			

7.2 Test Parameters

The following parameters are standard test parameters applying to all test cases in this test area. All tests should be executed with parameters defined in Table 3 unless it's stated differently in the specific test case.

Table 3: NFVI/MANO Interoperability Test Parameters

Parameters derived from KPIs	Value
Test runs	1

7.3 Postambles

Unless stated otherwise in the test case description, the following procedures should be executed after the test case steps:

- Stop generating traffic from the Traffic Generator (TG)
- Remove all Test NS and VNF
- Collect all logs and result files, upload to the log server

7.4 Test Cases

7.4.1 Software Image Management

7.4.1.1 Add Software Image

Purpose	Verify that the NFVO can add a software image to the image repository managed by the VIM
Description	In this test, we will add a software image to the image repository managed by the VIM and verify that there's no error message after adding the image.
Initial Conditions	The following initial conditions are applied for this section: 1. NFVO can add software images on the VIM 2. VIM supports "add image" operations by the NFVO 3. VIM supports software image information queries by the NFVO Please refer to Figure 1 for the Test Setup diagram.
Parameters	Use the specific test case parameters described below.

Table 4: Parameters for Add software image test

Parameters	**Values**
Software image information	name, metadata, location

Procedures & Expected Results	The following table describes the Procedures & Expected Results in detail.

Table 5: Procedures & Expected Results for Add software image test

Steps	**Procedures**	**Expected Results**
1.	Trigger an "add image" operation on the NFVO.	The image is added successfully
2.	Query the list of images on the image repository managed by the VIM.	The image information is listed
3.	Verify the NFVO operation	NFVO shows no "add image"

<table>
<tr><td></td><td><table><tr><td></td><td>errors</td><td>operation errors</td></tr></table></td></tr>
<tr><td><b>Measurements</b></td><td>If all steps are executed successfully, this test is considered as passed.</td></tr>
<tr><td><b>Postambles</b></td><td>1. Delete the added software image
2. Execute the postamble procedure defined in section 0.</td></tr>
</table>

7.4.1.2 Query Software Image by NFVO

<table>
<tr><td><b>Purpose</b></td><td>Verify that the NFVO can retrieve the information of a software image from the image repository managed by the VIM.</td></tr>
<tr><td><b>Description</b></td><td>In this test, we will use the NFVO to retrieve the information of a software image from the image repository managed by the VIM, and compare it with the information collected by VIM, to verify that the information retrieved by NFVO is correct.</td></tr>
<tr><td><b>Initial Conditions</b></td><td>The following initial conditions are applied for this section:
1. The software image to be queried is added to the image repository managed by the VIM
2. NFVO can query software image information on the VIM
3. VIM supports software image information queries by the NFVO
Please refer to Figure 1 for the Test Setup diagram.</td></tr>
<tr><td><b>Parameters</b></td><td>Use the specific test case parameters described below.</td></tr>
<tr><td></td><td><b>Table 6: Parameters for query software image by NFVO test</b><table><tr><th>Parameters</th><th>Values</th></tr><tr><td>Software image information</td><td>name, metadata, location</td></tr></table></td></tr>
<tr><td><b>Procedures & Expected Results</b></td><td>The following table describes the Procedures & Expected Results in detail.</td></tr>
<tr><td></td><td><b>Table 7: Procedures & Expected Results for query software image by NFVO test</b><table><tr><th>Steps</th><th>Procedures</th><th>Expected Results</th></tr><tr><td>1.</td><td>Trigger the query of the relevant image information on the NFVO using the unique image identifier.</td><td>The image information is retrieved successfully.</td></tr><tr><td>2.</td><td>Query the list of images on the</td><td>The image information is listed</td></tr></table></td></tr>
</table>

	image repository managed by the VIM.	
3.	Compare the image information collected in step 1 and 2.	The information obtained by NFVO in step 1 matches the information from step 2.

Measurements	If all steps are executed successfully, this test is considered as passed.
Postambles	Execute the postamble procedure defined in section 0.

7.4.1.3 Query Software Image by VNFM

Purpose	Verify that the VNFM can retrieve the information of a software image from the image repository managed by the VIM.
Description	In this test, we will use the VNFM to retrieve the information of a software image from the image repository managed by the VIM, and compare it with the information collected by VIM, to verify that the information retrieved by VNFM is correct.
Initial Conditions	The following initial conditions are applied for this section: 1. The software image to be queried is added to the image repository managed by the VIM 2. VNFM can query software image information on the VIM 3. VIM supports software image information queries by the VNFM Please refer to Figure 1 for the Test Setup diagram.
Parameters	Use the specific test case parameters described below.

Table 8: Parameters for query software image by VNFM test

Parameters	Values
Software image information	name, metadata, location

Procedures & Expected Results	The following table describes the Procedures & Expected Results in detail.

Table 9: Procedures & Expected Results for query software image by VNFM test

Steps	Procedures	Expected Results
1.	Trigger the query of the relevant image information on	The image information is retrieved successfully.

		the VNFM using the unique image identifier.	
	2.	Query the list of images on the image repository managed by the VIM.	The image information is listed
	3.	Compare the image information collected in step 1 and 2.	The information obtained by VNFM in step 1 matches the information from step 2.
Measurements	If all steps are executed successfully, this test is considered as passed.		
Postambles	Execute the postamble procedure defined in section 0.		

7.4.1.4 Update Software Image

Purpose	Verify that the NFVO can update the metadata of a software image in the image repository managed by the VIM
Description	In this test, we will use the NFVO to update the metadata of a software image from the image repository managed by the VIM, and compare it with the information collected by VIM, to verify that the information updated by NFVO is correct.
Initial Conditions	The following initial conditions are applied for this section: 1. The software image to be updated is added to the image repository managed by the VIM 2. NFVO can update software image information on the VIM 3. VIM supports "update image" operations by the NFVO 4. VIM supports software image information queries by the NFVO Please refer to Figure 1 for the Test Setup diagram.
Parameters	Use the specific test case parameters described below.

Table 10: Parameters for update software image test

Parameters	Values
Software image information (old)	name, metadata, location
Updated software image information	name, metadata, location

<table>
<tr><td><b>Procedures & Expected Results</b></td><td>The following table describes the Procedures & Expected Results in detail.</td></tr>
<tr><td></td><td>
<b>Table 11: Procedures & Expected Results for update software image test</b>
<table>
<tr><th>Steps</th><th>Procedures</th><th>Expected Results</th></tr>
<tr><td>1.</td><td>Trigger an "update image" operation on the NFVO using the unique image identifier and including the updated metadata.</td><td>The image information is updated successfully.</td></tr>
<tr><td>2.</td><td>Query the list of images on the image repository managed by the VIM.</td><td>The image information is listed</td></tr>
<tr><td>3.</td><td>Compare the image information collected in step 1 and 2.</td><td>The information obtained by NFVO in step 1 matches the information from step 2.</td></tr>
<tr><td>4.</td><td>Verify the NFVO operation errors</td><td>NFVO shows no "update image" operation errors</td></tr>
</table>
</td></tr>
<tr><td><b>Measurements</b></td><td>If all steps are executed successfully, this test is considered as passed.</td></tr>
<tr><td><b>Postambles</b></td><td>1. Change the software image information back to the original version.
2. Execute the postamble procedure defined in section 0.</td></tr>
</table>

7.4.1.5 Delete Software Image

Purpose	Verify that the NFVO can delete a software image from the image repository managed by the VIM
Description	In this test, we will use the NFVO to delete a software image from the image repository managed by the VIM, and compare it with the information collected by VIM, to verify that the software image is deleted by NFVO successfully.
Initial Conditions	The following initial conditions are applied for this section: 1. The software image to be deleted is added to the image repository managed by the VIM 2. NFVO can delete software image information on the VIM 3. VIM supports "delete image" operations by the NFVO 4. VIM supports software image information queries by the NFVO Please refer to Figure 1 for the Test Setup diagram.

Parameters	Use the specific test case parameters described below.

Table 12: Parameters for delete software image test

Parameters	Values
Software image information	name, metadata, location

Procedures & Expected Results	The following table describes the Procedures & Expected Results in detail.

Table 13: Procedures & Expected Results for delete software image test

Steps	Procedures	Expected Results
1.	Trigger an "delete image" operation on the NFVO using the unique image identifier.	The image is deleted successfully.
2.	Query the list of images on the image repository managed by the VIM.	The affected image is deleted from the image repository managed by the VIM.
3.	Verify the NFVO operation errors	NFVO shows no "delete image" operation errors

Measurements	If all steps are executed successfully, this test is considered as passed.
Postambles	1. Add the software image back the image repository managed by the VIM. 2. Execute the postamble procedure defined in section 0.

7.4.2 Fault Management

7.4.2.1 Virtualised Resource Fault Alarm Notification

Purpose	Verify that a fault alarm notification propagates to the NFVO when a virtualised resource that is required for the NS connectivity fails
Description	In this test, we will trigger a fault on a virtualised resource and verify that the fault alarm notification is created by NFVO.
Initial Conditions	The following initial conditions are applied for this section: 1. The test NS is instantiated 2. NFVO is subscribed to virtualised resources fault alarms on the VIM

	Please refer to Figure 1 for the Test Setup diagram.
Parameters	Use the specific test case parameters described below.

Table 14: Parameters for virtualised resource fault alarm notification test

Parameters	Values
Virtualised Resource Fault Type	NIC failure

Procedures & Expected Results	The following table describes the Procedures & Expected Results in detail.

Table 15: Procedures & Expected Results for virtualised resource fault alarm notification test

Steps	Procedures	Expected Results
1.	Trigger a fault on a virtualised resource that is required for the NS instance connectivity on the NFVI (e.g. disable the NIC allocated to a network resource)	The virtualised resource fault is triggered.
2.	Query the list of virtualised resource fault alarms on VIM.	A virtualised resource fault alarm is created.
3.	Query the list of NS fault alarms on NFVO.	A NS fault alarm is created.

Measurements	If all steps are executed successfully, this test is considered as passed.
Postambles	1. Restore the triggered fault on the virtualised resource. 2. Execute the postamble procedure defined in section 0.

7.4.2.2 Virtualised Resource Fault Alarm Clearance Notification

Purpose	Verify that a fault clearance notification propagates to the NFVO when a failed virtualised resource that is required for the NS connectivity is recovered
Description	In this test, we will recover a virtualised resource fault and verify that the fault alarm is cleared on NFVO.
Initial Conditions	The following initial conditions are applied for this section:

	1. The test NS is instantiated 2. NFVO is subscribed to virtualised resources fault alarms on the VIM 3. NS fault alarm is created on the NFVO by failing a virtualised resource that is required for the NS connectivity Please refer to Figure 1 for the Test Setup diagram.
Parameters	Use the specific test case parameters described below.

Table 16: Parameters for virtualised resource fault alarm clearance test

Parameters	**Values**
Virtualised Resource Fault Type	NIC failure

Procedures & Expected Results	The following table describes the Procedures & Expected Results in detail.

Table 17: Procedures & Expected Results for virtualised resource fault alarm clearance test

Steps	**Procedures**	**Expected Results**
1.	Resolve the failure of the virtualised resource that is required for the NS instance connectivity on the NFVI (e.g. disable the NIC allocated to a network resource)	The virtualised resource fault is resolved.
2.	Query the list of virtualised resource fault alarms on VIM.	The virtualised resource fault alarm is cleared.
3.	Query the list of NS fault alarms on NFVO.	The NS fault alarm is cleared.

Measurements	If all steps are executed successfully, this test is considered as passed.
Postambles	1. Trigger the same fault on the virtualised resource. 2. Execute the postamble procedure defined in section 0.

7.4.2.3 VNF Fault Alarm Notification

Purpose	Verify that a VNF fault alarm notification propagates via the VNFM to the NFVO when a VNF fault is triggered by a failed virtualised resource

<table>
<tr><td><b>Description</b></td><td>In this test, we will trigger a virtualised resource fault and verify that the fault alarm notification is created on VIM, VNFM, and NFVO.</td></tr>
<tr><td><b>Initial Conditions</b></td><td>The following initial conditions are applied for this section:
1. The test NS is instantiated
2. NFVO is subscribed to VNF fault alarms on the VNFM
3. VNFM is subscribed to virtualised resources fault alarms on the VIM
Please refer to Figure 1 for the Test Setup diagram.</td></tr>
<tr><td><b>Parameters</b></td><td>Use the specific test case parameters described below.</td></tr>
<tr><td></td><td><b>Table 18: Parameters for VNF fault alarm notification test</b>
<table>
<tr><th>Parameters</th><th>Values</th></tr>
<tr><td>Virtualised Resource Fault Type</td><td>NIC failure</td></tr>
</table>
</td></tr>
<tr><td><b>Procedures & Expected Results</b></td><td>The following table describes the Procedures & Expected Results in detail.</td></tr>
<tr><td></td><td><b>Table 19: Procedures & Expected Results for VNF fault alarm notification test</b>
<table>
<tr><th>Steps</th><th>Procedures</th><th>Expected Results</th></tr>
<tr><td>1.</td><td>Trigger a failure on a virtualised resource allocated to the relevant VNF instance (e.g. terminate the virtualised resource directly on the VIM)</td><td>The virtualised resource fault is created.</td></tr>
<tr><td>2.</td><td>Query the list of virtualised resource fault alarms on VIM.</td><td>The virtualised resource fault alarm is created.</td></tr>
<tr><td>3.</td><td>Query the list of VNF fault alarms on VNFM.</td><td>The VNF fault alarm is created.</td></tr>
<tr><td>3.</td><td>Query the list of NS fault alarms on NFVO.</td><td>The NS fault alarm is cleared.</td></tr>
</table>
</td></tr>
<tr><td><b>Measurements</b></td><td>If all steps are executed successfully, this test is considered as passed.</td></tr>
<tr><td><b>Postambles</b></td><td>1. Restore the triggered fault on the virtualised resource.
2. Execute the postamble procedure defined in section 0.</td></tr>
</table>

7.4.2.4 VNF Fault Alarm Clearance Notification

<table>
<tr><td><b>Purpose</b></td><td>Verify that a VNF fault alarm clearance notification propagates via the VNFM to the NFVO when a VNF fault is cleared by resolving a failed virtualised resource.</td></tr>
<tr><td><b>Description</b></td><td>In this test, we will recover a virtualised resource fault and verify that the fault alarm is cleared on VIM, VNFM, and NFVO.</td></tr>
<tr><td><b>Initial Conditions</b></td><td>The following initial conditions are applied for this section:
1. The test NS is instantiated
2. NFVO is subscribed to VNF fault alarms on the VNFM
3. VNFM is subscribed to virtualised resources fault alarms on the VIM
4. NS fault alarm is created on the NFVO by failing a virtualised resource that is allocated to the relevant VNF instance.
Please refer to Figure 1 for the Test Setup diagram.</td></tr>
<tr><td><b>Parameters</b></td><td>Use the specific test case parameters described below.</td></tr>
<tr><td></td><td><b>Table 20: Parameters for VNF fault alarm clearance test</b>
<table>
<tr><th>Parameters</th><th>Values</th></tr>
<tr><td>Virtualised Resource Fault Type</td><td>NIC failure</td></tr>
</table></td></tr>
<tr><td><b>Procedures & Expected Results</b></td><td>The following table describes the Procedures & Expected Results in detail.</td></tr>
<tr><td></td><td><b>Table 21: Procedures & Expected Results for VNF fault alarm clearance test</b>
<table>
<tr><th>Steps</th><th>Procedures</th><th>Expected Results</th></tr>
<tr><td>1.</td><td>Resolve the failure of the virtualised resource allocated to the relevant VNF (e.g. restart the virtualised resource directly on the VIM)</td><td>The virtualised resource fault is resolved.</td></tr>
<tr><td>2.</td><td>Query the list of virtualised resource fault alarms on VIM.</td><td>The virtualised resource fault alarm is cleared.</td></tr>
<tr><td>3.</td><td>Query the list of VNF fault alarms on VNFM.</td><td>The VNF fault alarm is cleared.</td></tr>
<tr><td>4.</td><td>Query the list of NS fault alarms on NFVO.</td><td>The NS fault alarm is cleared.</td></tr>
</table></td></tr>
</table>

Measurements	If all steps are executed successfully, this test is considered as passed.
Postambles	1. Trigger the same fault on the virtualised resource. 2. Execute the postamble procedure defined in section 0.

7.4.3 Virtualised Resource Performance Management

7.4.3.1 VR PM Job Creation and Notification Monitoring

<table>
<tr><td><b>Purpose</b></td><td>Verify that the performance metrics of a virtualised resource that is required for a NS instance connectivity can be monitored using PM jobs and notifications.</td></tr>
<tr><td><b>Description</b></td><td>In this test, we will create a VR PM job for a virtualized resource on NFVO and verify that the performance metrics of a virtualised resource that is required for a NS instance connectivity can be monitored using PM jobs and notifications.</td></tr>
<tr><td><b>Initial Conditions</b></td><td>The following initial conditions are applied for this section:
1. The test NS is instantiated
2. Monitoring parameters (e.g. metrics, metric groups, collection and reporting periods) are defined
Please refer to Figure 1 for the Test Setup diagram.</td></tr>
<tr><td><b>Parameters</b></td><td>Use the specific test case parameters described below.</td></tr>
<tr><td></td><td><b>Table 22: Parameters for VR PM Job Creation and Notification Monitoring test</b>
<table>
<tr><th>Parameters</th><th>Values</th></tr>
<tr><td>Monitoring parameters</td><td>metrics, metric groups, collection and reporting periods</td></tr>
</table>
</td></tr>
<tr><td><b>Procedures & Expected Results</b></td><td>The following table describes the Procedures & Expected Results in detail.</td></tr>
<tr><td></td><td><b>Table 23: Procedures & Expected Results for VR PM Job Creation and Notification Monitoring test</b>
<table>
<tr><th>Steps</th><th>Procedures</th><th>Expected Results</th></tr>
<tr><td>1.</td><td>Trigger the NFVO to create a VR PM job for a virtualised resource that is allocated to the target NS instance.</td><td>The VR PM job is created.</td></tr>
</table>
</td></tr>
</table>

	<table><tr><td>2.</td><td>Query the VR PM jobs on VIM according to the monitoring parameters.</td><td>The query results show that the VR PM job is created on VIM.</td></tr><tr><td>3.</td><td>Trigger the NFVO to subscribe to the VR PM job created in step 1</td><td>The NFVO is subscribed to the VR PM job successfully.</td></tr><tr><td>4.</td><td>Monitor the VR PM notifications for at least one metric collection and reporting period.</td><td>A "performance information available" notification for the monitored virtualised resource was generated by the VIM to the NFVO.</td></tr></table>
Measurements	If all steps are executed successfully, this test is considered as passed.
Postambles	1. Remove the VR PM job created in step 1. 2. Execute the postamble procedure defined in section 0.

7.4.3.2 VR PM Job Creation and Threshold Monitoring

Purpose	Verify that the performance metrics of a virtualised resource that is required for a NS instance connectivity can be monitored using PM jobs and thresholds.
Description	In this test, we will create a VR PM job with VR PM threshold for a virtualized resource on NFVO and trigger the virtualised resource to cross the specified threshold to verify that the performance metrics of a virtualised resource can be monitored using PM jobs and thresholds.
Initial Conditions	The following initial conditions are applied for this section: 1. The test NS is instantiated 2. Monitoring parameters (e.g. metrics, metric groups, thresholds) are defined Please refer to Figure 1 for the Test Setup diagram.
Parameters	Use the specific test case parameters described below.
	Table 24: Parameters for VR PM Job Creation and Threshold Monitoring test <table><tr><th>Parameters</th><th>Values</th></tr><tr><td>Monitoring parameters</td><td>metrics, metric groups, thresholds</td></tr><tr><td>VR PM Threshold</td><td></td></tr></table>

Procedures & Expected Results	The following table describes the Procedures & Expected Results in detail.

Table 25: Procedures & Expected Results for VR PM Job Creation and Threshold Monitoring test

Steps	**Procedures**	**Expected Results**
1.	Trigger the NFVO to create a VR PM job for a virtualised resource that is allocated to the target NS instance.	The VR PM job is created.
2.	Query the VR PM jobs on VIM according to the monitoring parameters.	The query results shows that The VR PM job is created on VIM.
3.	Trigger the NFVO to create a VR PM threshold for the virtualised resource monitored in step 1.	The VR PM threshold is created.
4.	Trigger the NFVO to subscribe to the threshold crossing notification for the VR PM threshold created in step 3.	The NFVO is subscribed to the threshold crossing notification.
5.	Trigger the virtualised resource to cross the specified threshold (e.g. by increasing resource utilization levels in the virtualisation container).	The virtualised resource crosses the specified threshold.
6.	Monitor the VR PM notifications for at least three metric collection and reporting period.	The "threshold crossed" notification for the monitored virtualised resource is generated by the VIM to the NFVO.

Measurements	If all steps are executed successfully, this test is considered as passed.
Postambles	1. Remove the VR PM job created in step 1, and the VR PM threshold created in step 3. 2. Restore the virtualised resource utilization back to normal status. 3. Execute the postamble procedure defined in section 0.

7.4.3.3 VR PM Job Deletion

<table>
<tr><td><b>Purpose</b></td><td>Verify that the monitoring of performance metrics of a virtualised resource that is required for a NS instance connectivity can be stopped by deleting PM jobs.</td></tr>
<tr><td><b>Description</b></td><td>In this test, we will delete a VR PM job for a virtualized resource on NFVO to verify that the monitoring of performance metrics of a virtualised resource that is required for a NS instance connectivity can be stopped.</td></tr>
<tr><td><b>Initial Conditions</b></td><td>The following initial conditions are applied for this section:
1. The test NS is instantiated
2. A virtualised resource that is required for the NS connectivity is monitored by the NFVO
Please refer to Figure 1 for the Test Setup diagram.</td></tr>
<tr><td><b>Parameters</b></td><td>Use the specific test case parameters described below.</td></tr>
<tr><td></td><td><b>Table 26: Parameters for VR PM Job Deletion test</b>
<table>
<tr><th>Parameters</th><th>Values</th></tr>
<tr><td>Monitoring parameters</td><td>metrics, metric groups, collection and reporting periods</td></tr>
</table></td></tr>
<tr><td><b>Procedures & Expected Results</b></td><td>The following table describes the Procedures & Expected Results in detail.</td></tr>
<tr><td></td><td><b>Table 27: Procedures & Expected Results for VR PM Job Deletion test</b>
<table>
<tr><th>Steps</th><th>Procedures</th><th>Expected Results</th></tr>
<tr><td>1.</td><td>Trigger the NFVO to delete a VR PM job for a virtualised resource that is allocated to the target NS instance.</td><td>The VR PM job is deleted.</td></tr>
<tr><td>2.</td><td>Query the VR PM jobs on VIM according to the monitoring parameters.</td><td>The query results show that the VR PM job is deleted on VIM.</td></tr>
<tr><td>3.</td><td>Monitor the VR PM notifications for at least three metric collection and reporting period.</td><td>NO "performance information available" notification for the monitored virtualised resource is generated by the VIM to the NFVO.</td></tr>
</table></td></tr>
</table>

Measurements	If all steps are executed successfully, this test is considered as passed.
Postambles	1. Add back the VR PM job deleted in step 1. 2. Execute the postamble procedure defined in section 0.

7.4.3.4 VR PM Threshold Deletion

Purpose	Verify that a threshold created for a virtualised resource that is required for a NS instance connectivity can be deleted.
Description	In this test, we will delete a VR PM threshold for a virtualized resource on NFVO and trigger the virtualised resource to cross the specified threshold to verify that the threshold created for a virtualised resource that is required for a NS instance connectivity can be deleted.
Initial Conditions	The following initial conditions are applied for this section: 1. The test NS is instantiated. 2. A threshold for a virtualised resource that is required for the NS connectivity is created by the NFVO. Please refer to Figure 1 for the Test Setup diagram.
Parameters	Use the specific test case parameters described below.

Table 28: Parameters for VR PM Threshold Deletion test

Parameters	Values
Monitoring parameters	metrics, metric groups, thresholds
VR PM Threshold	

Procedures & Expected Results	The following table describes the Procedures & Expected Results in detail.

Table 29: Procedures & Expected Results for VR PM Threshold Deletion test

Steps	Procedures	Expected Results
1.	Trigger the NFVO to delete a VR PM job for a virtualised resource that is allocated to the target NS instance.	The VR PM job is deleted.
2.	Query the VR PM jobs on VIM according to the monitoring parameters.	The query results show that The VR PM job is deleted on VIM.

	3.	Trigger the virtualised resource to cross the specified threshold (e.g. by increasing resource utilization levels in the virtualisation container).	The virtualised resource crosses the specified threshold.
	4.	Monitor the VR PM notifications for at least three metric collection and reporting period.	NO "threshold crossed" notification for the monitored virtualised resource is generated by the VIM to the NFVO.
Measurements	If all steps are executed successfully, this test is considered as passed.		
Postambles	1. Add back the VR PM job deleted in step 1 2. Restore the virtualised resource utilization back to normal status. 3. Execute the postamble procedure defined in section 0.		

8 Sub-module Tests for NFV Performance Test

8.1 Preambles

8.1.1 Pre-Conditions

The NFVI and MANO must be deployed as designed in 5G VINNI Deliverables [9][10]. The test controlling system and test automation tools are installed and connected to the System Under Test (SUT). A virtual machine as test function is deployed on the NFVI to be validated. This test function acts as origin of the test packets to be sent. Dependent on the identified VNF type, a corresponding terminating virtual machine as second test function is deployed on the NFVI according to the deployment constraints of the VNF type.

The test conditions on network components and services are listed in Table 30.

Table 30: NFV Test Conditions on Network Components and Services

Components	Optional Conditions	Test Condition 1	Test Condition 2
Service Orchestration	Nokia Flowone	✓	✓
EMS	Ericssion	✓	
VNF	Ericsson	✓	
	Samsung		✓
NFVI Hyperviser	Nokia NCIR	✓	
NFVO	Nokia CBND	✓	
	Samsung CMS		✓
VNFM	Ericsson ENM /Nokia CBAM	✓	
	Samsung CMS		✓
VIM	Nokia NCIR	✓	
	HP Openstack		✓
SDN Controller	Nuage	✓	✓
Physical Devices	HP		✓
	Nokia Airframe OR	✓	
	Nuage WBX	✓	

8.1.2 Test Environment Setup

The NFV Infrastructure Performance test setup is shown in Figure 2.

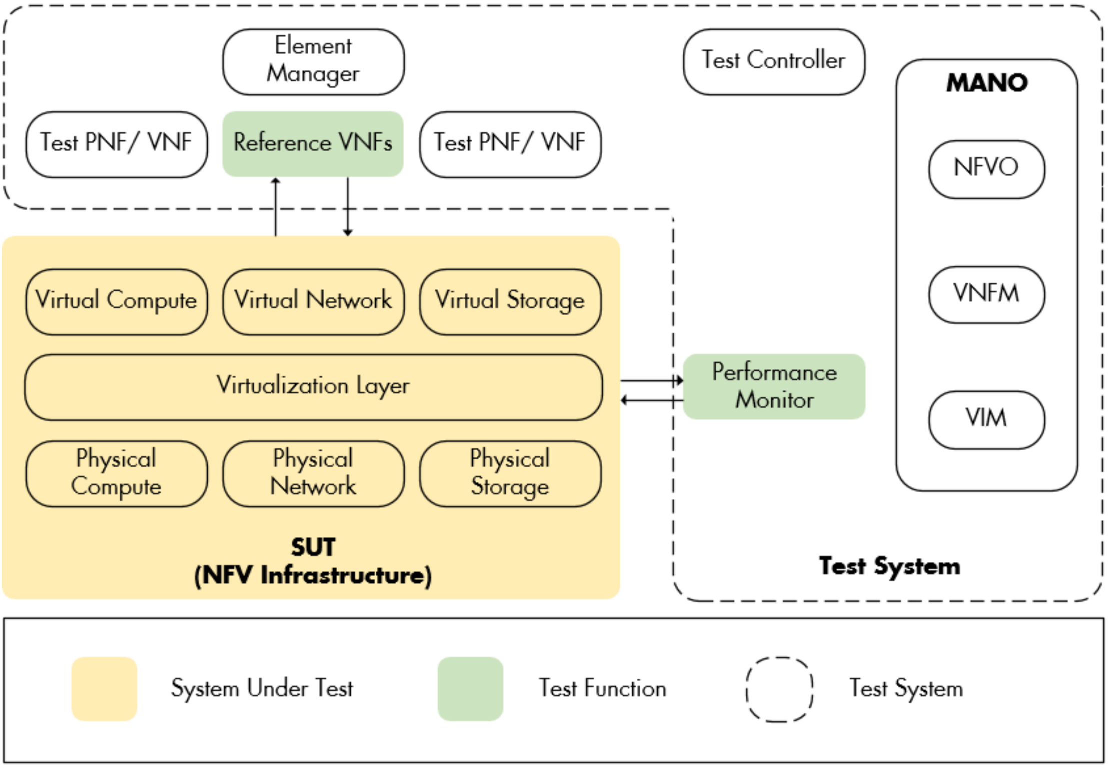


Figure 2: NFVI performance test setup

The Network Service Performance test setup is shown in Figure 3.

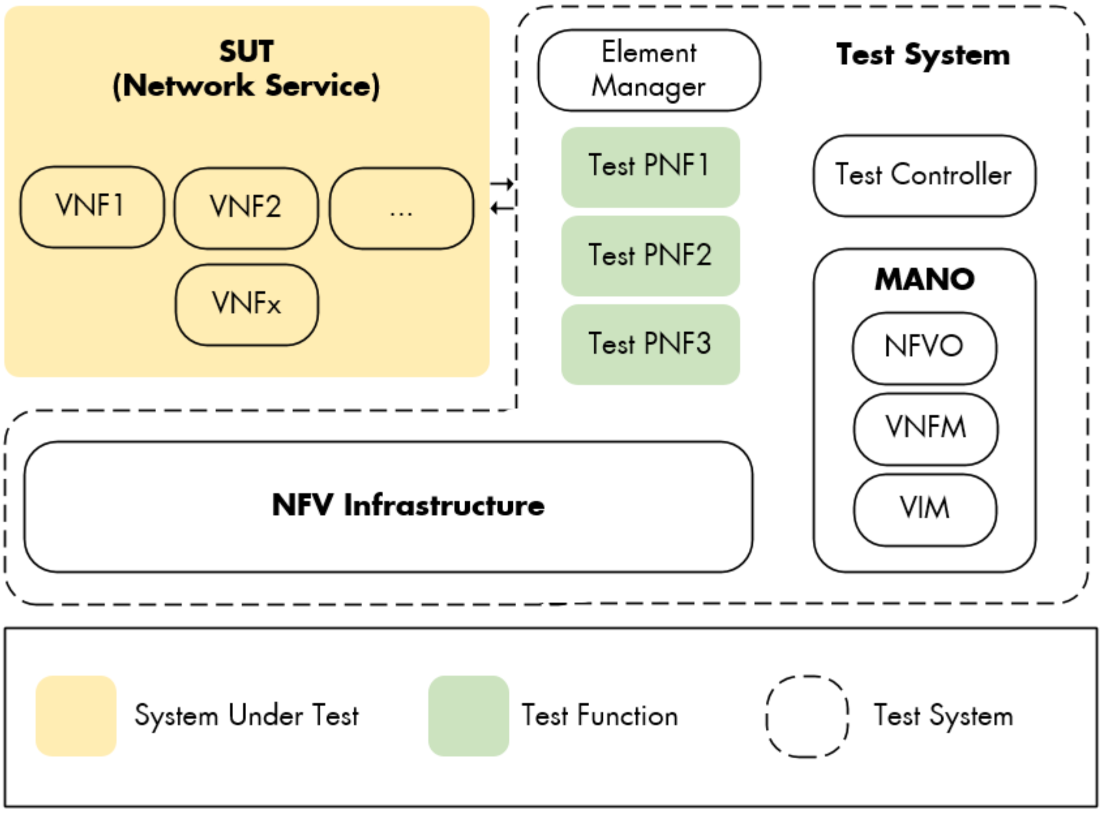


Figure 3: Network Service performance test setup

8.1.3 Pre-Qualification Test Cases

All Pre-Qualification test cases, as defined in 5GVINNI Deliverable [6] section 2.1.2, shown in Table 31: Pre-qualification test cases for NFV Performance Tests, under 'Mandatory Tests' and 'Conditional Tests' must be passed.

- Each facility site must comply with one of the conditions defined in Table 30: NFV Test Conditions on Network Components and Services.
- Conditional Tests must also be tested and passed for the sites complying with the condition.
- Optional Tests can be optionally tested based on the capability of the site.

Table 31: Pre-qualification test cases for NFV Performance Tests

Domain	Mandatory Tests	Conditional Tests (Condition 1)	Conditional Tests (Condition 2)	Optional Tests
MANO & NFVI	4.1.1-4.1.4, 4.2.1, 4.2.2, 4.3.1-4.3.3			

8.2 Test Parameters

The following parameters are standard test parameters applying to all test cases in this test area. All tests should be executed with parameters defined in Table 32 unless it's stated differently in the specific test case.

Table 32: NFV Performance Test Parameters

Parameters derived from KPIs	Value
Packet size (Bytes)	64, 128, 256, 512, 1024, 1280, 1518
Traffic type	UDP
Test duration for 1 run	15mins
Test runs	3

8.3 Measurements

The KPI Computation Procedures for each of the measurements are described in Table 33.

Table 33: NFV Performance Test Measurements and KPI Computation Procedures

Measurements	KPI Computation Procedures
Network Latency	The average value of measured Latency in all test iterations
Network Throughput	The average value of measured Throughput in all test iterations
Network Packet loss	The average value of measured Packet loss in all test iterations
Storage Performance	The average value of measured Storage Performance in all test iterations

Processor Utilization	The average value of measured Processor Utilization in all test iterations
Memory Latency	The average value of measured Memory Latency in all test iterations
NS Scale out time	The average value of measured NS Scale out time in all test iterations
Speed of Activation	The average value of measured time needed to activate a Network Service that comprises multiple VNFs in all test iterations

8.4 Key Performance Indicators (KPIs)

Based on the Service Level Agreement (SLA) of different vertical industry scenarios, Table 34 depicts the KPI requirement of the 5GVINNI project for NFV Performance Tests.

Table 34: NFV Performance Test KPIs

KPIs	Values
Network Latency	
Network Throughput	
Network Packet loss	
Storage Performance (Input/Output Operations Per Second, storage throughput, storage latency)	
Processor Utilization	
Memory Latency	
NS Scale out time	
Speed of Activation	

8.5 Postambles

Unless stated otherwise in the test case description, the following procedures should be executed after the test case steps:

- Stop generating traffic from the Traffic Generator (TG)
- Remove all Test NS and VNF
- Collect all logs and result files, upload to the log server

8.6 Test Cases

8.6.1 NFVI Network Performance Measurement

Purpose	Measure the NFVI Network performance, including network latency,

<table>
<tr><td></td><td>throughput, and packet loss.</td></tr>
<tr><td><b>Description</b></td><td>In this test, we will validate the NFVI network performance, including network latency, throughput, and packet loss, by measuring the L3 network latency, throughput, and packet loss between an originating virtual machine and a target entity which can be another virtual machine or a destination outside of the NFVI.</td></tr>
<tr><td><b>Initial Conditions</b></td><td>The following initial conditions are applied for this section:
1. The test controller configures the network between the test functions (e.g. using floating IP addresses) in the NFVI to be validated.
2. The Test VNF as origin of the test packets to be sent is deployed as specified in Table 35.
3. The second terminating Test Virtual Machine is deployed on the NFVI according to the deployment constraints of the VNF type.
4. TG is setup to run RFC 2544 [7] / ITU-T Y.1564 [8] test to measure the NFVI network performance.
Please refer to Figure 2 for the Test Setup diagram.</td></tr>
<tr><td><b>Parameters</b></td><td>Use the specific test case parameters described below.</td></tr>
<tr><td></td><td><b>Table 35: Parameters for NFVI network performance measurement</b>
<table>
<tr><th>Parameters derived from KPIs</th><th>Values</th></tr>
<tr><td>VNF Type 1</td><td>vEPC</td></tr>
<tr><td>VNF Type 2</td><td>?</td></tr>
</table></td></tr>
<tr><td><b>Procedures & Expected Results</b></td><td>The following table describes the Procedures & Expected Results in detail.
For each test step, all packet sizes should be tested separately.</td></tr>
<tr><td></td><td><b>Table 36: Procedures & Expected Results for NFVI network performance measurement</b>
<table>
<tr><th>Steps</th><th>Procedures</th><th>Expected Results</th></tr>
<tr><td>1.</td><td>Run RFC 2544 / ITU-T Y.1564 test on TG between the Test Functions to measure the latency with each packet size defined in Table 32.</td><td>The network latency for NFVI matches the network latency requirement stated in Table 34.</td></tr>
<tr><td>2.</td><td>Run RFC 2544 / ITU-T Y.1564 test on TG between the Test Functions to measure the throughput with each packet size defined in Table 32.</td><td>The network throughput for NFVI matches the network throughput requirement stated in Table 34.</td></tr>
</table></td></tr>
</table>

3.	Run RFC 2544 / ITU-T Y.1564 test on TG between the Test Functions to measure the packet loss with each packet size defined in Table 32.	The network packet loss for NFVI matches the network packet loss requirement stated in Table 34.
4.	Repeat step 1-3 for specific Test runs as stated in Table 32, collect the results.	Record test results and compute the KPIs.

Measurements	Calculate the results for each KPI according to the computation procedure defined in section 0.
Postambles	Execute the postamble procedure defined in section 0.

8.6.2 NFVI Virtual Resource Performance Measurement

Purpose	Measure the NFVI virtual resource performance, including storage, processor, and memory.
Description	In this test, we will validate the storage performance in a virtual machine on the NFVI, including storage, processor, and memory. We'll deploy a virtual machine that contains tools to measure the storage I/O, processor utilization, and latency for random memory access.
Initial Conditions	The following initial conditions are applied for this section: 1. The test controller configures the test function in the NFVI to be validated. 2. The Test VNF with the test tools is deployed and configured as specified in Table 37. Please refer to Figure 2 for the Test Setup diagram.
Parameters	Use the specific test case parameters described below.

Table 37: Parameters for NFVI virtual resource performance measurement

Parameters derived from KPIs	Values
Storage I/O Type	Random Read and Write
Block size for the IO units	?
I/O Test File name and size	?

Procedures &	The following table describes the Procedures & Expected Results in detail.

Expected Results	For each test step, all packet sizes should be tested separately.

Table 38: Procedures & Expected Results for NFVI virtual resource performance measurement

Steps	Procedures	Expected Results
1.	The test controller instructs the test function to start the storage performance test according to parameters defined in Table 32 and Table 37.	The storage performance for NFVI matches the requirement stated in Table 34.
2.	The test controller instructs the test function to start the processor utilization test according to parameters defined in Table 32.	The processor utilization for NFVI matches the requirement stated in Table 34.
3.	The test controller instructs the test function to start the memory latency test according to parameters defined in Table 32.	The memory latency for NFVI matches the requirement stated in Table 34.
4.	Repeat step 1-3 for specific Test runs as stated in Table 32, collect the results.	Record test results and compute the KPIs.

Measurements	Calculate the results for each KPI according to the computation procedure defined in section 0.
Postambles	Execute the postamble procedure defined in section 0.

8.6.3 Network Service Autoscaling Validation

Purpose	To verify the successful completion of NS autoscaling in response to auto scale stimuli, and measure the NS scale out time.
Description	NS and VNF scaling can be triggered in 3 ways: a) Autoscaling, b) On-demand scaling and c) Manually triggered scaling. A Network Service can dynamically react to a sustained spike in customer traffic by scaling out; similarly, during periods of reduced customer traffic, it can scale in. In this test, we'll test autoscaling only and highlights a use case that employs NS scale out. The NS scale out is triggered by an increasing the traffic load generated by the Test Tools. The scaling procedures are initiated by the NFV Orchestrator and the VNF Manager after they detect the increased traffic

<table>
<tr><td></td><td>load.</td></tr>
<tr><td><b>Initial Conditions</b></td><td>The following initial conditions are applied for this section:
1. The test controller configures the test function in the NFVI to be validated.
2. The user has defined multiple flavours for the Network Service Under Test (NSUT). The test starts with Flavour A of the NSUT. The user has defined the performance target levels for both Flavours A and B of the NSUT.
3. The NSUT has been assigned the necessary NFVI resources to perform at its performance target for Flavour A.
4. The Test PNFs 1and 2 have the needed resources and capabilities to exchange control and user plane traffic at the performance target levels of the NSUT Flavour A.
5. Test Controller is able to access the NSD and access its fields related to autoscale policy and the stimuli that are needed to cause the autoscaling.
6. The Test PNFs 1, 2 and 3 have the needed resources and capabilities to stimulate the NSUT to scale out to Flavour B (increased traffic or CPU load)
7. The test PNFs 1, 2 and 3 are time synchronized.
Please refer to Figure 3 for the Test Setup diagram.</td></tr>
<tr><td><b>Parameters</b></td><td>Use the specific test case parameters described below.</td></tr>
<tr><td></td><td><b>Table 39: Parameters for network service autoscaling validation</b>
<table>
<tr><th>Parameters derived from KPIs</th><th>Values</th></tr>
<tr><td>Test NS Type</td><td>vFW</td></tr>
<tr><td>NSUT Flavours</td><td>Flavour A, Flavour B</td></tr>
<tr><td>Flavour A Traffic load limit</td><td>?</td></tr>
</table></td></tr>
<tr><td><b>Procedures & Expected Results</b></td><td>The following table describes the Procedures & Expected Results in detail.
For each test step, all packet sizes should be tested separately.</td></tr>
<tr><td></td><td><b>Table 40: Procedures & Expected Results for network service autoscaling validation</b>
<table>
<tr><th>Steps</th><th>Procedures</th><th>Expected Results</th></tr>
<tr><td>1.</td><td>Test Controller instructs the Test PNFs to initiate the test.
Test PNFs 1 and 2 establish the necessary connections with the Network Service endpoints, prior to exchanging traffic with</td><td>Validate that the necessary user plane or control plane connections between Test PNFs and NSUT have been established.</td></tr>
</table></td></tr>
</table>

		the NSUT Flavour A.	
	2.	The Test PNFs 1 & 2 originate bi-directional traffic toward the NSUT Flavour A at its performance target level.	The bi-directional traffic is sent and received successfully between Test PNFs 1 & 2.
	3.	The Test PNFs dial up the traffic and load toward the NSUT Flavour A, to a level that will trigger a scale out to Flavour B. Record the time T1, when the traffic and load reach the level that will trigger autoscaling.	Autoscaling is triggered and starting time T1 is recorded.
	4.	Starting at time T1, assess the performance 'P' of the NSUT periodically by making traffic measurements at Test PNFs 2 and 3. The exact metrics are NSUT dependent. The polling interval for the NSUT performance measurements are user defined but it is recommended the polling is done once every second. The polling is done for a user defined maximum of Tmax seconds.	/
	5.	Measure the first instance time t = T2, when traffic is observed on PNF3. Log the value of T2. Do not repeat this step after the T2 has been logged the first time.	The First instance time is recorded
	6.	Compare the measured performance 'P' of the NSUT to the performance target for NSUT Flavour B. If 'P' is lower than the performance target and time t < Tmax, go back to Step 4 and continue polling.	/
	7.	Measure the received traffic at Test PNFs 2 and 3 and ensure that the distribution (load balancing) of traffic across PNF2 and 3 meets the user	/

	expectations. If the load balancing is improper and time t < Tmax, go back to Step 4 and continue polling.	
8.	If time t > Tmax, Stop the traffic and go to step 9. Else, log time t = T3, as time at which scale out is completed and [T3-T1] as the time needed to complete scale out. Stop the traffic and trigger the scale in of NSUT. NSUT returns to Flavour A state.	Scaling out time is recorded
9.	Repeat step 1-8 for specific Test runs as stated in Table 32, collect the results.	Record test results and compute the KPIs.

Measurements	Calculate the results for each KPI according to the computation procedure defined in section 0.
Postambles	Execute the postamble procedure defined in section 0.

8.6.4 Network Service Speed of Activation Test

Purpose	To measure the time needed to activate a Network Service that comprises multiple VNFs in a service chain.
Description	In this test, we'll validate the speed of activation of services in highly dynamic NFV environments. The test setup consists of an NFV server hosting a Network Service under Test (NSUT). The NSUT is comprised of 3 VNFs in a service chain. An example of such a Network service would be a vCPE service chain consisting of a vFirewall, vCE device and a vRouter. Physical test devices are connected to the NSUT and will be used to originate and terminate traffic.
Initial Conditions	The following initial conditions are applied for this section: 1. The user has defined the criteria (Ps) for deeming the NS instantiation as a success. Ps can either be a single metric or a matrix of metrics (for e.g. required goodput rate, connections per second etc.). The exact set of criteria is dependent on the Network Service under consideration. 2. The user has defined Tmax as the maximum time allowed for the completion of Network Service activation. 3. Physical test devices are used, to ensure microsecond accuracy in timing measurements and to eliminate any influence that the

<table>
<tr><td></td><td>presence of a test VNF will have on the shared NFV environment.
4. The test PNFs 1 and 2 are time synchronized.
Please refer to Figure 3 for the Test Setup diagram.</td></tr>
<tr><td><b>Parameters</b></td><td>Use the specific test case parameters described below.</td></tr>
<tr><td></td><td><b>Table 41: Parameters for network service Speed of Activation test</b>
<table>
<tr><th>Parameters derived from KPIs</th><th>Values</th></tr>
<tr><td>Test NS Type</td><td>vEPC</td></tr>
<tr><td>NSUT Performance Target (Ps)</td><td>?</td></tr>
<tr><td>Tmax</td><td>?</td></tr>
</table>
</td></tr>
<tr><td><b>Procedures & Expected Results</b></td><td>The following table describes the Procedures & Expected Results in detail.
For each test step, all packet sizes should be tested separately.</td></tr>
<tr><td></td><td><b>Table 42: Procedures & Expected Results for network service Speed of Activation test</b>
<table>
<tr><th>Steps</th><th>Procedures</th><th>Expected Results</th></tr>
<tr><td>1.</td><td>The Test Controller instructs the NFV Orchestrator to instantiate the Network Service at time T = T1.</td><td>Verify that the NFV Orchestrator notifies the test Controller after it completes the NS instantiation.</td></tr>
<tr><td>2.</td><td>At time T = T1, at the same time that the Network Service instantiation is requested, the Test Controller instructs the test devices to start exchanging appropriate bi-directional L2-L7 traffic with the Network Service under Test.
Traffic is originated at a rate that matches the performance target (Ps) of the NSUT.</td><td>Measure the service performance (QoS) metrics periodically (recommended once every 100 ms) until the time when all the QoS metrics meet or exceed the target performance (Ps) of the Network Service. Log the time T = T2 when service performance ≥ Ps.</td></tr>
<tr><td>3.</td><td>Repeat step 1-2 for specific Test runs as stated in Table 32, collect the results.</td><td>Record test results and compute the KPIs.</td></tr>
</table>
</td></tr>
<tr><td><b>Measurements</b></td><td>Calculate the results for each KPI according to the computation procedure defined in section 0.</td></tr>
</table>

Postambles	Execute the postamble procedure defined in section 0.

9 Sub-module Tests for RAN

9.1 Preambles

9.1.1 Pre-Conditions

Radio Access Network (RAN) will be tested based on 5G NSA option 3x. The DU-CU splitting for gNB is implemented by hosting the radio central unit (CU) on x86 based servers.

The end user device is expected to support dual-connectivity for both E-UTRA and new radio (NR) air interfaces.

Due to the restricted number of UE devices that is enrolled in the test bed. NO capacity test cases will be performed for the radio network (i.e. Max number of attached subscribers).

The test conditions on network components and services are listed in 44.

Table 44: RAN Test Conditions on Network Components and Services

Components	Optional Conditions	Test Condition 1	Test Condition 2
LTE eNB	Samsung	✓	
	Ericsson		✓
5G gNB	Samsung	✓	
	Ericsson		✓
LTE Radio Frequency	TBD		
	TBD		
Frequency BW per User	TBD		
5G Radio Frequency	TBD		
Frequency BW per User	TBD		
MIMO Configuration	TBD		

9.1.2 Test Environment Setup

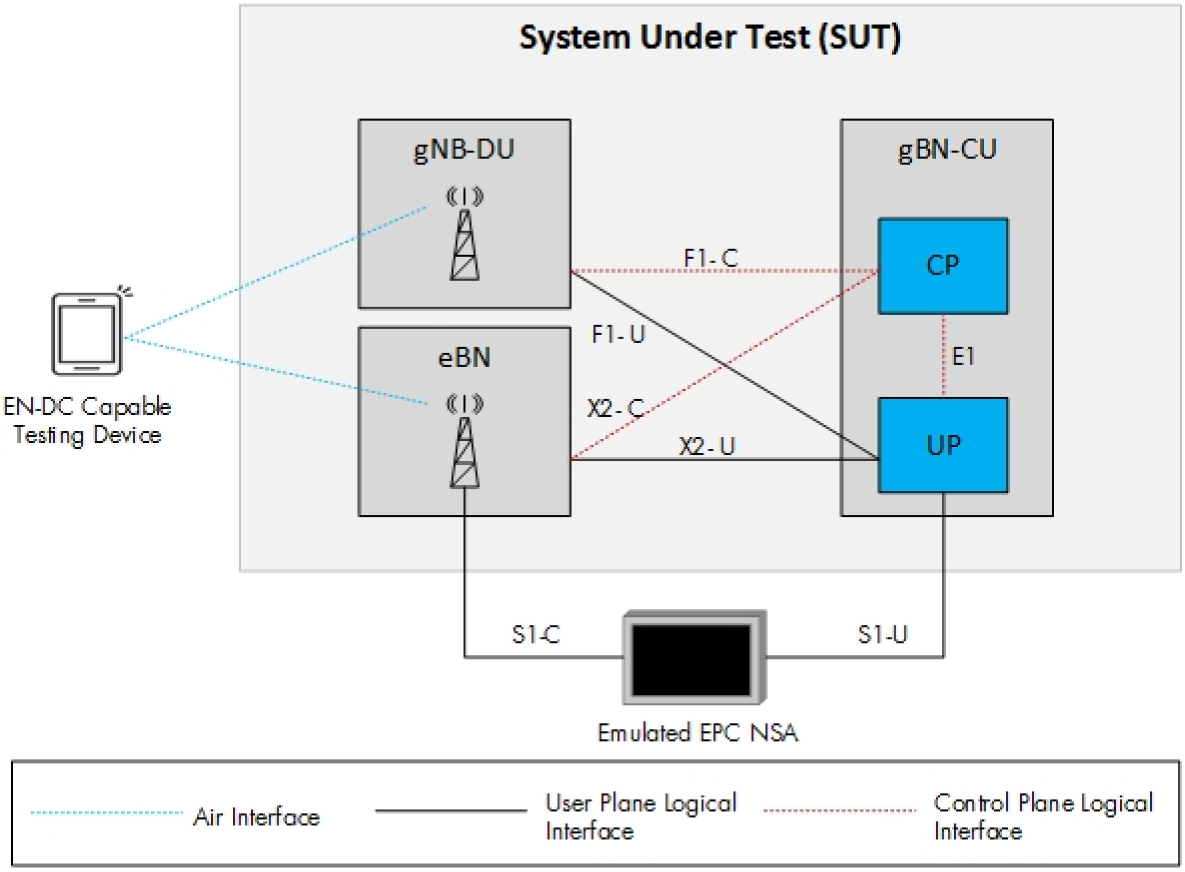


Figure 5: RAN performance test setup

9.1.3 Pre-Qualification Test Cases

All Pre-Qualification test cases, as defined in 5GVINNI Deliverable [6], shown in Table 31: Pre-qualification test cases for NFV Performance Tests, under 'Mandatory Tests' and 'Conditional Tests' must be passed.

- Each facility site must comply with one of the conditions defined in Table 30: NFV Test Conditions on Network Components and Services.
- Conditional Tests must also be tested and passed for the sites complying with the condition.
- Optional Tests can be optionally tested based on the capability of the site.

Table 45: Pre-qualification test cases for NFV Performance Tests

Domain	Mandatory Tests	Conditional Tests (Condition 1)	Conditional Tests (Condition 2)	Optional Tests
5G New Radio (NR)	2.1.1-2.1.6, 2.2.1, 2.2.2, 2.3.1, 2.3.2, 2.3.3			

9.2 Test Parameters

The following parameters are standard test parameters applying to all test cases in this test area. All tests should be executed with parameters defined in 46 unless it's stated differently in the specific test case.

Table 46: RAN Performance Test Parameters

Parameters derived from KPIs	Value
Control plane packet size (Bytes)	512
User pane packet size (Bytes)	IMIX
Traffic type	GTPoSCTP, GTPoUDP
Test duration for 1 run	15mins
Test runs	3

9.3 Measurements

The KPI Computation Procedures for each of the measurements are described in 47.

Table 47: NFV Performance Test Measurements and KPI Computation Procedures

Measurements	KPI Computation Procedures
Network Latency	The average value of measured Latency in all test iterations
Network Throughput	The average value of measured Throughput in all test iterations
Network Packet loss	The average value of measured Packet loss in all test iterations
Processor Utilization	The average value of measured Processor Utilization in all test iterations
Memory Latency	The average value of measured Memory Latency in all test iterations

9.4 Key Performance Indicators (KPIs)

Based on the Service Level Agreement (SLA) of different vertical industry scenarios, 48 depicts the KPI requirement of the 5GVINNI project for NFV Performance Tests.

Table 48: NFV Performance Test KPIs

KPIs	Values
Control Plane Traffic Latency (mean)	<10ms
User Plane Traffic Latency (mean)	<10ms
Upstream Throughput (Peak data rate)	UL 10 Gbit/.
Downstream Throughput (Peak data rate)	DL 20 Gbit/s
User Plane Upstream Packet Loss (max)	$<10^{-5}$

User Plane Downstream Packet Loss (max)	<10^{-5}
User Plane Upstream Jitter	/
User Plane Downstream Jitter	
Virtualized Resources Utilization	?

9.5 Postambles

Unless stated otherwise in the test case description, the following procedures should be executed after the test case steps:

- Stop generating traffic from the Traffic Generator (TG)
- Collect all logs and result files, upload to the log server

9.6 Test Cases

9.6.1 RAN Network Performance Measurement

Purpose	Measure the end to end performance of RAN based on 3GPP Rel15 NSA system architecture.
Description	These test procedures verify the performance of the 5G NR in NSA deployment. The traffic generators will emulate the mobile core network with single UE .
Initial Conditions	The following initial conditions are applied for this section: 1. Setup TG to run RFC 2544 / ITU-T Y.1564 test to measure the network performance.
Parameters	Use the specific test case parameters described below.
	Table 48: Parameters for 5G NR NSA network performance measurement Parameters derived from KPIs | Values (two empty rows)
Procedures & Expected Results	The following table describes the Procedures & Expected Results in detail. For each test step, all packet sizes should be tested separately.

Table 49: Procedures & Expected Results for RAN network performance measurement

Steps	Procedures	Expected Results
1.	Verify TG IP connectivity from UE emulator to the emulated mobile core & data network.	TG has IP connectivity from UE emulator to the emulated Data Network.
2.	Run RFC 2544 / ITU-T Y.1564 test on UE to measure the upstream traffic latency	matches the upstream traffic latency requirement stated in section 9.4
3.	Run RFC 2544 / ITU-T Y.1564 test on TG to measure the downstream traffic latency	matches the downstream traffic latency requirement stated in section 9.4
4.	Run RFC 2544 / ITU-T Y.1564 test on UE to measure the upstream traffic throughput	matches the upstream traffic throughput requirement stated in section 9.4
5.	Run RFC 2544 / ITU-T Y.1564 test on TG to measure the downstream traffic throughput	matches the downstream traffic throughput requirement stated in section 9.4
6.	Run RFC 2544 / ITU-T Y.1564 test on UE to measure the upstream traffic jitter	matches the upstream traffic jitter requirement stated in section 9.4
7.	Run RFC 2544 / ITU-T Y.1564 test on TG to measure the downstream traffic jitter	matches the downstream traffic jitter requirement stated in section 9.4
8.	Run RFC 2544 / ITU-T Y.1564 test on UE to measure the upstream traffic packet loss	matches the upstream traffic jitter requirement stated in section 9.4
9.	Run RFC 2544 / ITU-T Y.1564 test on TG to measure the downstream traffic packet loss	matches the downstream traffic jitter requirement stated in section 9.4
10.	Repeat step 3-6 for 3 times, collect the results	
11.	Collect all logs and result files, upload to the log server	All logs archived in log server.

Measurements	Calculate the results for each KPI according to the computation procedure defined in section 0.

Postambles	Execute the postamble procedure defined in section 0.

10 Sub-module Tests for vEPC NSA

10.1 Preambles

10.1.1 Pre-Conditions

Mobile core testing is based on 5G NSA core deployment. So, before executing the planned test cases, the following assumptions have to be achieved:

- All the functional components of vEPC are deployed on single/multiple servers
- All the functional components of vEPC are reachable & accessible through management network
- Pre-qualification tests are passed

The test conditions on network components and services are listed in 44.

Table 50: RAN Test Conditions on Network Components and Services

Components	Optional Conditions	Test Condition 1	Test Condition 2
vEPC	Samsung	✓	
	Ericsson		✓

10.1.2 Test Environment Setup

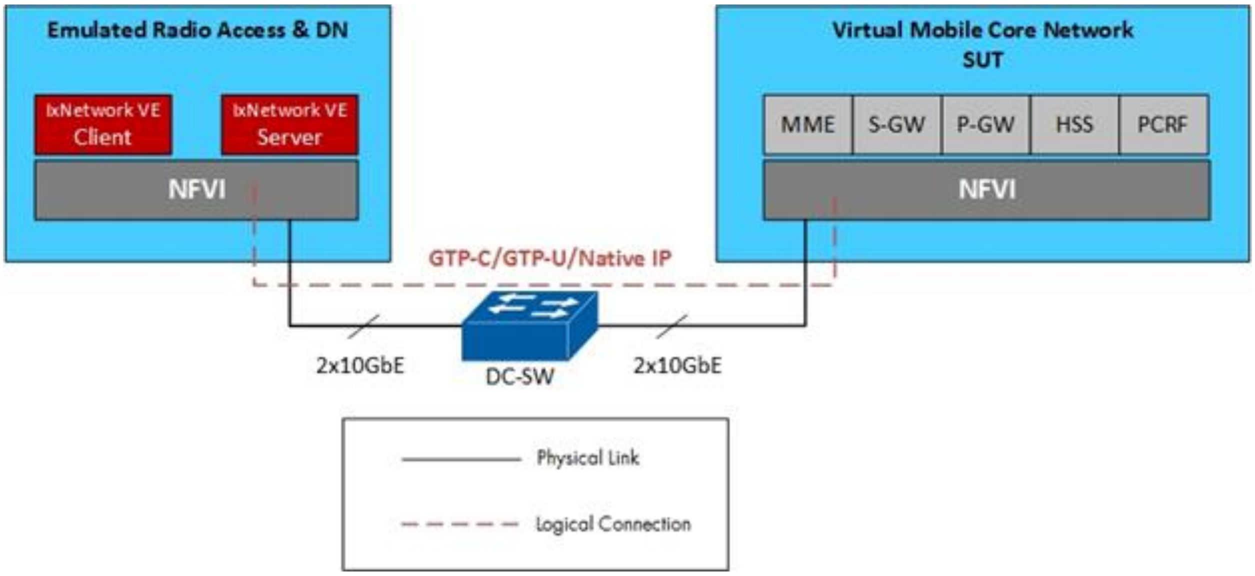


Figure 6: vEPC NSA performance test setup

10.1.3 Pre-Qualification Test Cases

All Pre-Qualification test cases, as defined in 5GVINNI Deliverable [6], shown in Table 31: Pre-qualification test cases for NFV Performance Tests, under 'Mandatory Tests' and 'Conditional Tests' must be passed.

- Each facility site must comply with one of the conditions defined in Table 30: NFV Test Conditions on Network Components and Services.
- Conditional Tests must also be tested and passed for the sites complying with the condition.

- Optional Tests can be optionally tested based on the capability of the site.

Table 51: Pre-qualification test cases for NFV Performance Tests

Domain	Mandatory Tests	Conditional Tests (Condition 1)	Conditional Tests (Condition 2)	Optional Tests
Core Network (NSA EPC)	3.1.1-3.1.4, 3.2.1-3.2.8, 3.3.1, 3.3.2, 3.4.1-3.4.4			

10.2 Test Parameters

The following parameters are standard test parameters applying to all test cases in this test area. All tests should be executed with parameters defined in 52 unless it's stated differently in the specific test case.

Table 52: vEPC NSA Performance Test Parameters

Parameters derived from KPIs	Value
Control plane packet size (Bytes)	512
User pane packet size (Bytes)	IMIX
Traffic type	GTPoSCTP, GTPoUDP
Number of Emulated UEs	1000
Test duration for 1 run	15mins
Test runs	3

10.3 Measurements

The KPI Computation Procedures for each of the measurements are described in 47.

Table 47: NFV Performance Test Measurements and KPI Computation Procedures

Measurements	KPI Computation Procedures
Network Latency	The average value of measured Latency in all test iterations
Network Throughput	The average value of measured Throughput in all test iterations
Network Packet loss	The average value of measured Packet loss in all test iterations
Processor Utilization	The average value of measured Processor Utilization in all test iterations
Memory Latency	The average value of measured Memory Latency in all test iterations

10.4 Key Performance Indicators (KPIs)

Based on the Service Level Agreement (SLA) of different vertical industry scenarios, Table 53 depicts the KPI requirement of the 5GVINNI project for vEPC NSA Performance Tests.

Table 53: NFV Performance Test KPIs

KPIs	Values
UE Attach Procedure Latency (mean)	<10ms
Control Plane Traffic Latency (mean)	<10ms
User Plane Traffic Latency (mean)	<10ms
Upstream Throughput (Peak data rate)	UL 10 Gbit/.
Downstream Throughput (Peak data rate)	DL 20 Gbit/s
User Plane Upstream Packet Loss (max)	<10−5
User Plane Downstream Packet Loss (max)	<10−5
User Plane Upstream Jitter	/
User Plane Downstream Jitter	
Number of active UE (max)	/
UE attachment rate	/
UE detachment rate	/
Virtualized Resources Utilization	?

10.5 Postambles

Unless stated otherwise in the test case description, the following procedures should be executed after the test case steps:

- Stop generating traffic from the Traffic Generator (TG)
- Collect all logs and result files, upload to the log server

10.6 Test Cases

10.6.1 vEPC NSA Network Performance Measurement

Purpose	Measure the end to end performance of vEPC mobile core which supports 5G radio network based on 3GPP Rel15 NSA system architecture.

Description

These test procedures verify the performance of the NSA mobile core. The traffic generators will emulate the radio access network along with the UE traffic and external data networks (i.e. Internet, IoTGW, MCSGW).

Initial Conditions

The following initial conditions are applied for this section:

1. Setup TG to run RFC 2544 / ITU-T Y.1564 test to measure the network performance.
2. Configure the emulated EN-DC UEs based on Table 52
3. Configure 3 emulated PDNs (eMBB, URLL, MTC)
4. Connect Traffic Generator to the physical DC switch, with two VLANs (VLAN_GTPc, VLAN_GTPu)

Parameters

Use the specific test case parameters described below.

Table 54: Parameters for vEPC NSA network performance measurement

Parameters derived from KPIs	Values

Procedures & Expected Results

The following table describes the Procedures & Expected Results in detail.

For each test step, all packet sizes should be tested separately.

Table 55: Procedures & Expected Results for vEPC NSA network performance measurement

Steps	Procedures	Expected Results
1.	Verify TG IP connectivity from UE emulator to the emulated Data Network.	TG has IP connectivity from UE emulator to the emulated Data Network.
2.	Measure the latency to register a single user	Emulated one UE registered successfully which matches the attachment latency requirements in section 9.4
3.	Run RFC 2544 / ITU-T Y.1564 test on TG to measure the upstream traffic latency	matches the upstream traffic latency requirement stated in section 9.4
4.	Run RFC 2544 / ITU-T Y.1564 test on TG to measure the downstream traffic latency	matches the downstream traffic latency requirement stated in section 9.4
5.	Run RFC 2544 / ITU-T Y.1564 test on TG to measure the	matches the upstream traffic throughput requirement stated

	upstream traffic throughput	in section 9.4
6.	Run RFC 2544 / ITU-T Y.1564 test on TG to measure the downstream traffic throughput	matches the downstream traffic throughput requirement stated in section 9.4
7.	Run RFC 2544 / ITU-T Y.1564 test on TG to measure the upstream traffic jitter	matches the upstream traffic jitter requirement stated in section 9.4
8.	Run RFC 2544 / ITU-T Y.1564 test on TG to measure the downstream traffic jitter	matches the downstream traffic jitter requirement stated in section 9.4
9.	Run RFC 2544 / ITU-T Y.1564 test on TG to measure the upstream traffic packet loss	matches the upstream traffic jitter requirement stated in section 9.4
10.	Run RFC 2544 / ITU-T Y.1564 test on TG to measure the downstream traffic packet loss	matches the downstream traffic jitter requirement stated in section 9.4
11.	Repeat step 3-6 for 3 times, collect the results	
12.	Collect all logs and result files, upload to the log server	All logs archived in log server.

Measurements	Calculate the results for each KPI according to the computation procedure defined in section 0.
Postambles	Execute the postamble procedure defined in section 0.

10.6.2 vEPC NSA Network Capacity Measurement

Purpose	Measure the system capacity vEPC that will support 5G network deployments based on NSA architecture.
Description	vEPC 5G NSA system capacity is quantified based on the maximum number of active users that can register in MME & SGW successfully. In addition, the maximum attachment & detachment rates are other KPIs that will be verify the capacity of the system.
Initial Conditions	The following initial conditions are applied for this section: 1. Setup TG with the expected maximum number of users. This includes the definition of IMSI, APN & traffic profile.

Parameters	Use the specific test case parameters described below.

Table 56: Parameters for vEPC NSA network capacity measurement

Parameters derived from KPIs	Values
Emulated UE quantity	Based on the vendor input
APN	Default DN
User Plane Packet Size (Bytes)	700
Bandwidth per user (Kb/s)	1
Service under test	vEPC

Procedures & Expected Results	The following table describes the Procedures & Expected Results in detail. For each test step, all packet sizes should be tested separately.

Table 57: Procedures & Expected Results for vEPC NSA capacity measurement

Steps	Procedures	Expected Results
1.	Verify TG IP connectivity from UE emulator to the emulated Data Network.	TG has IP connectivity from UE emulator to the emulated Data Network.
2.	Run the test on TG by gradually attaching the emulated users and setup the attachment rate (1K/s) with (session life time> test execution period)	matches the maximum number of active user's benchmark stated in section 1.3.
3.	Run the test on TG to hard-detach all the pre-registered users and measure the detachment rate	matches the maximum detach rate of active user's benchmark stated in section 1.3.
4.	Repeat step 2-3 for 3 times, collect the results	
5.	Verify zero users is registered to MME & SGW	Clear flow table in MME, SGW & PGW
6.	Run the test on TG by attaching the all the emulated users (session life time> test execution period)	

	7.	Measure the CPU & Memory utilization for vEPC VMs	
	8.	Collect the results of attachment rate and compute resource utilization	The collected results from TG, vEPC NMS & VIM must be matched
	9.	Repeat step 5-8 for 3 times	
	10.	Collect all logs and result files, upload to the log server	All logs archived in log server.
Measurements	Calculate the results for each KPI according to the computation procedure defined in section 0.		
Postambles	Execute the postamble procedure defined in section 0.		

RAN

Core

Transport

MUST statements in D1.1

LTE ‐ NR Dual Connectivity

LTE ‐ NR Dual Connectivity MUST be supported at facility sites implementing NSA architecture.

Mobility

In 5GVINNI connected and idle mode mobility MUST be supported for both NSA and SA architectures.

NR QoS Framework

For facility sites implementing architecture with 5G Core the NR QoS framework MUST be supported.

11 E2E Tests

11.1 Preambles

11.1.1 Pre-Conditions

End-to-End test network should be installed and configured with three slices, eMBB, uRLLC, and mMTC, as defined in 5GVINNI Deliverables [3][4][5]. For each facility site under test, at least one of the slices must be configured. Depending on the actual deployment of each site, 5G New Radio (NR) and 5G Core (5GC) are optional conditions, but at least one of them must be present for the testing.

The test conditions on network components and services are listed in Table 43.

Table 43: End-To-End Test Conditions on Network Components and Services

Optional Conditions	Test Condition 1	Test Condition 2
eMBB network slice	✓	✓
uRLLC network slice	✓	
mMTC network slice	✓	✓
4G eNB	✓	
5G NR	✓	✓
4G EPC		✓
5GC	✓	
MEC		✓

11.1.2 Test Environment Setup

3GPP specified a set of architecture options for 5G deployment in release 15 [2]. Options 3, 3a and 3x (Non-Standalone) allow NR deployments reusing EPC with the support of LTE eNB. With these options, the LTE eNB is connected to the EPC with Non-standalone NR.

In this section, we'll test the End-To-End network performance based on the Option 3x(NSA) as shown in Figure 4.

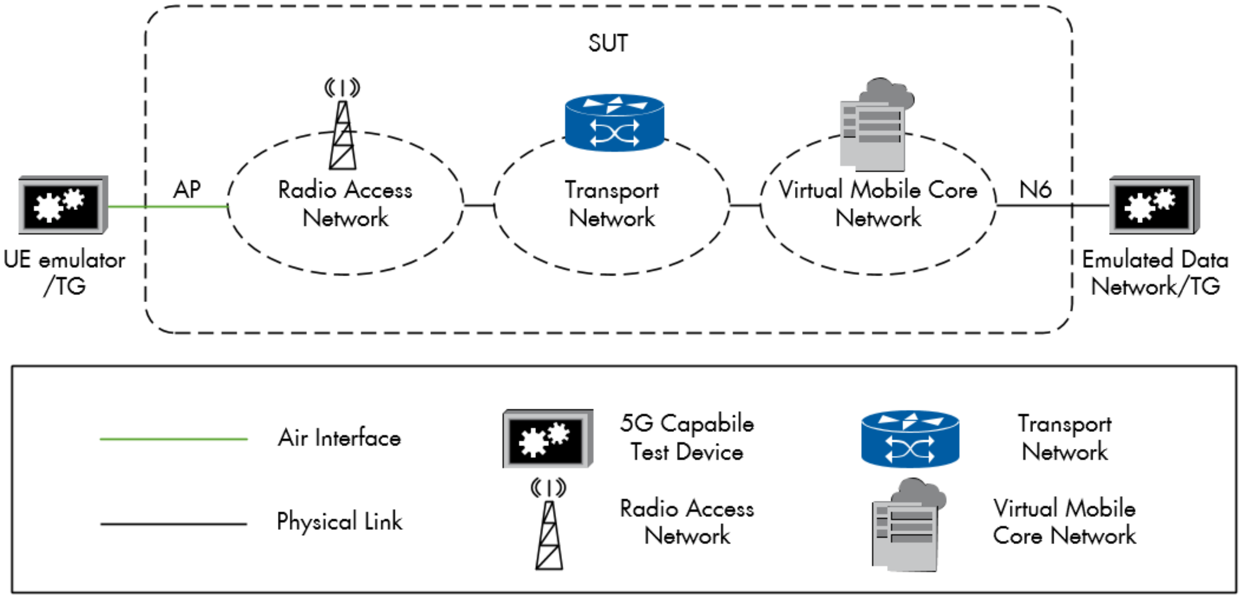


Figure 4: End-To-End test setup

11.1.3 Pre-Qualification Test Cases

All Pre-Qualification test cases, as defined in 5GVINNI Deliverable [6] section 2.1.2, shown in Table 44, under 'Mandatory Tests' and 'Conditional Tests' must be passed.

- Each facility site must comply with one of the conditions defined in Table 43.
- Conditional Tests must also be tested and passed for the sites complying with the condition.
- Optional Tests can be optionally tested based on the capability of the site.

Table 44: Pre-qualification test cases for End-To-End Tests

Domain	Mandatory Tests	Conditional Tests (Condition 1)	Conditional Tests (Condition 2)	Optional Tests
Transport Network	1.1.1, 1.1.2, 1.2.1, 1.3.1, 1.4.1, 1.5.1, 1.6.1			
5G New Radio (NR)	2.1.1-2.1.6, 2.2.1, 2.2.2, 2.3.1, 2.3.2, 2.3.3			
Core Network (NSA EPC)	3.1.1-3.1.4, 3.2.1-3.2.8, 3.3.1, 3.3.2, 3.4.1-3.4.4			
MANO & NFVI	4.1.1-4.1.4, 4.2.1, 4.2.2, 4.3.1-4.3.3			

11.2 Test Parameters

The following parameters are standard test parameters applying to all test cases in this test area. All tests should be executed with parameters defined in Table 45 unless it's stated differently in the specific test case.

Table 45: End-To-End Test Parameters

Parameters derived from KPIs	Value
Packet size (Bytes)	64, 128, 256, 512, 1024, 1280, 1518
Traffic type	UDP
Test duration for 1 run	15mins
Test runs	3

11.3 Measurements

The KPI Computation Procedures for each of the measurements are described in Table 46.

Table 46: Measurements and KPI Computation Procedures

Measurements	KPI Computation Procedures
Latency	The average value of measured Latency in all test iterations
Throughput	The average value of measured Throughput in all test iterations
Packet loss	The average value of measured Packet loss in all test iterations
Jitter	The average value of measured Jitter in all test iterations
UE attachment rate	The average value of the measured UE attachment rate in all test iterations UE attachment rate = $\frac{\text{Successfully attached UE amount}}{\text{Time from the begin to the end of the UE attachment}}$’
UE detachment rate	The average value of the measured UE detachment rate in all test iterations UE detachment rate = $\frac{\text{Successfully detached UE amount}}{\text{Time from the begin to the end of the UE detachment}}$’

11.4 Key Performance Indicators (KPIs)

Based on the Service Level Agreement (SLA) of different vertical industry scenarios, each network slice provides service with different KPIs. Table 47 depicts the KPI requirement of the 5GVINNI project for each network slice.

Table 47: End-To-End Test KPIs

KPIs	eMBB	uRLLC	mMTC
Latency	<5ms	<1ms	/
Throughput (Peak data ate)	DL 20 Gbit/s UL 10 Gbit/.	/	/

Packet loss	<10−5	<10−5	<10−5
Jitter	/	/	/
Number of Emulated Users	/	/	>1M devices / km^2
UE attachment rate	/	/	?
UE detachment rate	/	/	?

11.5 Postambles

Unless stated otherwise in the test case description, the following procedures should be executed after the test case steps:

- Stop generating traffic from the Traffic Generator (TG)
- Detach all emulated users from the network
- Stop the UE emulator and Data Network emulator
- Collect all logs and result files, upload to the log server

11.6 Test Cases

11.6.1 E2E eMBB Performance Measurement

Purpose	Measure the End to End performance of UE IP packets transmitted from UE to the N6 interface in the eMBB slice of the 5G network.
Description	In this test, we will verify the E2E performance of the eMBB slice based on Option 3x (NSA). We will send IP traffic from the 5G UE device to the emulated Data Network through the Non-standalone NR to measure the E2E traffic latency, throughput, jitter, and packet loss. High throughput is expected in the eMBB network slice.
Initial Conditions	The following initial conditions are applied for this section: 1. End-to-end network and slice meet the requirements in section 11.1. 2. End user is attached to the network. 3. End user can reach the emulated Data Network. 4. TG is setup to run RFC 2544 [7] / ITU-T Y.1564 [8] test to measure the E2E network performance. Please refer to Figure 4 for the Test Setup diagram.
Parameters	Use the specific test case parameters described below.

Table 48: Parameters for E2E eMBB performance measurement

Parameters derived from KPIs	Values
UE quantity	1
Service under test	eMBB

Procedures & Expected Results	The following table describes the Procedures & Expected Results in detail. For each test step, all packet sizes should be tested separately.

Table 49: Procedures & Expected Results for E2E eMBB performance measurement

Steps	Procedures	Expected Results
1.	Verify UE device registration status	UE device is registered successfully.
2.	Verify TG IP connectivity from test UE to the emulated Data Network.	TG has IP connectivity from test UE to the emulated Data Network.
3.	Run RFC 2544 / ITU-T Y.1564 test on TG between the UE and the Emulated Data Network to measure the latency with each packet size defined in Table 45.	E2E latency for eMBB slice matches the latency requirement stated in Table 47.
4.	Run RFC 2544 / ITU-T Y.1564 test on TG between the UE and the Emulated Data Network to measure the throughput with each packet size defined in Table 45.	E2E throughput for eMBB slice matches the throughput requirement stated in Table 47.
5.	Run RFC 2544 / ITU-T Y.1564 test on TG between the UE and the Emulated Data Network to measure the jitter with each packet size defined in Table 45.	E2E jitter for eMBB slice matches the jitter requirement stated in Table 47.
6.	Run RFC 2544 / ITU-T Y.1564 test on TG between the UE and the Emulated Data Network to measure the packet loss with each packet size defined in Table 45.	E2E packet loss for eMBB slice matches the packet loss requirement stated in Table 47.

7.	Repeat step 3-6 for specific Test runs as stated in Table 45, collect the results.	Results show high throughput and reasonable latency for the eMBB slice.

Measurements	Calculate the results for each KPI according to the computation procedure defined in section 0.
Postambles	Execute the postamble procedure defined in section 0

11.6.2 E2E uRLLC Performance Measurement

Purpose	Measure the End to End performance of UE IP packets transmitted from UE to the N6 interface in the uRLLC slice of the 5G network.
Description	In this test, we will verify the E2E performance of the uRLLC slice based on Option 3x (NSA). We will send IP traffic from the 5G UE device to the emulated Data Network through the Non-standalone NR to measure the E2E traffic latency, throughput, jitter, and packet loss. Ultra-Low Latency is expected in uRLLC network slice.
Initial Conditions	The following initial conditions are applied for this section: 1. End-to-end network and slice meet the requirements in section 11.1. 2. End user is attached to the network. 3. End user can reach the emulated Data Network. 4. TG is setup to run RFC 2544 [7] / ITU-T Y.1564 [8] test to measure the E2E network performance. Please refer to Figure 4 for the Test Setup diagram.
Parameters	Use the specific test case parameters described below.

Table 50: Parameters for E2E uRLLC performance measurement

Parameters derived from KPIs	Values
UE quantity	1
Service under test	uRLLC

Procedures & Expected Results	The following table describes the Procedures & Expected Results in detail. For each test step, all packet sizes should be tested separately.

Table 51: Procedures & Expected Results for E2E uRLLC performance measurement

Steps	Procedures	Expected Results
1.	Verify UE device registration status	UE device is registered successfully.
2.	Verify TG IP connectivity from UE to the emulated Data Network.	TG has IP connectivity from UE to the emulated Data Network.
3.	Run RFC 2544 / ITU-T Y.1564 test on TG between the UE and the Emulated Data Network to measure the latency with each packet size defined in Table 45.	E2E latency for uRLLC slice matches the latency requirement stated in Table 47.
4.	Run RFC 2544 / ITU-T Y.1564 test on TG between the UE and the Emulated Data Network to measure the throughput with each packet size defined in Table 45.	E2E throughput for uRLLC slice matches the throughput requirement stated in Table 47.
5.	Run RFC 2544 / ITU-T Y.1564 test on TG between the UE and the Emulated Data Network to measure the jitter with each packet size defined in Table 45.	E2E jitter for uRLLC slice matches the jitter requirement stated in Table 47.
6.	Run RFC 2544 / ITU-T Y.1564 test on TG between the UE and the Emulated Data Network to measure the packet loss with each packet size defined in Table 45.	E2E packet loss for uRLLC slice matches the packet loss requirement stated in Table 47.
7.	Repeat step 3-6 for specific Test runs as stated in Table 45, collect the results.	Results show high throughput and reasonable latency for the uRLLC slice.

Measurements	Calculate the results for each KPI according to the computation procedure defined in section 0.
Postambles	Execute the postamble procedure defined in section 0

11.6.3 E2E mMTC Performance Measurement

Purpose	Measure the End to End performance of Machine Type Communication in the mMTC slice of 5G network.

<table>
<tr><td><b>Description</b></td><td>In this test, we will verify E2E performance of the mMTC slice based on Option 3x (NSA). We will send IP traffic from UE to UE, and from UE to Data Network, for mMTC service through the Non-standalone NR to measure the E2E traffic latency, throughput, jitter and packet loss.</td></tr>
<tr><td><b>Initial Conditions</b></td><td>The following initial conditions are applied for this section:
1. End-to-end network and slice meet the requirements in section 11.1.
2. End users are attached to the network.
3. End user can reach the emulated Data Network.
4. TG is setup to run RFC 2544 [7] / ITU-T Y.1564 [8] test to measure the E2E network performance.
Please refer to Figure 4 for the Test Setup diagram.</td></tr>
<tr><td><b>Parameters</b></td><td>Use the specific test case parameters described below.</td></tr>
<tr><td></td><td><b>Table 52: Parameters for E2E mMTC performance measurement</b>
<table>
<tr><th>Parameters derived from KPIs</th><th>Values</th></tr>
<tr><td>UE quantity</td><td>2</td></tr>
<tr><td>Service under test</td><td>mMTC</td></tr>
</table></td></tr>
<tr><td><b>Procedures & Expected Results</b></td><td>The following table describes the Procedures & Expected Results in detail.
For each test step, all packet sizes should be tested separately.</td></tr>
<tr><td></td><td><b>Table 53: Procedures & Expected Results for E2E mMTC performance measurement</b>
<table>
<tr><th>Steps</th><th>Procedures</th><th>Expected Results</th></tr>
<tr><td>1.</td><td>Verify UE device registration status</td><td>UE device is registered successfully.</td></tr>
<tr><td>2.</td><td>Verify TG IP connectivity from one UE to another UE.</td><td>TG has IP connectivity from one UE to another UE.</td></tr>
<tr><td>3.</td><td>Run RFC 2544 / ITU-T Y.1564 test on TG between the UEs to measure the latency with each packet size defined in Table 45.</td><td>E2E latency for mMTC slice matches the latency requirement stated in Table 47.</td></tr>
<tr><td>4.</td><td>Run RFC 2544 / ITU-T Y.1564 test on TG between the UEs to measure the throughput with each packet size defined in Table 45.</td><td>E2E throughput for mMTC slice matches the throughput requirement stated in Table 47.</td></tr>
</table></td></tr>
</table>

5.	Run RFC 2544 / ITU-T Y.1564 test on TG between the UEs to measure the jitter with each packet size defined in Table 45.	E2E jitter for mMTC slice matches the jitter requirement stated in Table 47.
6.	Run RFC 2544 / ITU-T Y.1564 test on TG between the UEs to measure the packet loss with each packet size defined in Table 45.	E2E packet loss for mMTC slice matches the packet loss requirement stated in Table 47.
7.	Verify TG IP connectivity from UE to emulated Data Network. Repeat step 3-6 to test the End-To-End network performance from UE to the Data Network.	E2E network performance matches the KPI requirement stated in Table 47
8.	Repeat step 1-7 for specific Test runs as stated in Table 45, collect the results.	Results shows network performance matches the KPI requirements for mMTC slice.

Measurements	Calculate the results for each KPI according to the computation procedure defined in section 0.
Postambles	Execute the postamble procedure defined in section 0

11.6.4 E2E eMBB Scalability Measurement

Purpose	Measure the scalability for eMBB slice of the 5G network.
Description	In this test, we will verify the scalability of the eMBB slice based on Option 3x (NSA). We will emulate large amount of user connections to test the performance of adding/deleting massive amount of user sessions for eMBB service through the Non-standalone NR.
Initial Conditions	The following initial conditions are applied for this section: 1. End-to-end network and slice meet the requirements in section 11.1. 2. Emulated end users are detached from the network. Please refer to Figure 4 for the Test Setup diagram.
Parameters	Use the specific test case parameters described below.

Table 54: Parameters for E2E eMBB scalability measurement

Parameters derived from KPIs	Values
Emulated UE quantity	1M
Service under test	eMBB

Procedures & Expected Results	The following table describes the Procedures & Expected Results in detail. For each test step, all packet sizes should be tested separately.

Table 55: Procedures & Expected Results for E2E eMBB scalability measurement

Steps	Procedures	Expected Results
1.	Start the UE emulator and attach all emulated UEs to the network, record the time needed to finish all UE attachment.	All emulated UE devices can attach to the network successfully, the attachment rate matched the requirement stated in Table 47
2.	Verify TG IP connectivity from UE emulator to the emulated Data Network.	TG has IP connectivity from UE emulator to the emulated Data Network.
3.	Detach all emulated UEs from the network, record the time needed for all UEs sessions to be cleared.	All emulated UE devices can detach from the network successfully, the detachment rate matched the requirement stated in Table 47.
4.	Repeat step 1-3 for specific Test runs as stated in Table 45, collect the results.	Results show the scalability of eMBB slice matches the requirement stated in Table 47.

Measurements	Calculate the results for each KPI according to the computation procedure defined in section 0.
Postambles	Execute the postamble procedure defined in section 0

11.6.5 E2E uRLLC Scalability Measurement

Purpose	Measure the scalability for uRLLC slice of the 5G network.
Description	In this test, we will verify the scalability of the uRLLC slice based on Option 3x

	(NSA). We will emulate large amount of user connections to test the performance of adding/deleting massive amount of user sessions for uRLLC service through the Non-standalone NR.
Initial Conditions	The following initial conditions are applied for this section: 1. End-to-end network and slice meet the requirements in section 11.1. 2. Emulated end users are detached from the network. Please refer to Figure 4 for the Test Setup diagram.
Parameters	Use the specific test case parameters described below.

Table 56: Parameters for E2E uRLLC scalability measurement

Parameters derived from KPIs	Values
Emulated UE quantity	1M
Service under test	uRLLC

Procedures & Expected Results	The following table describes the Procedures & Expected Results in detail. For each test step, all packet sizes should be tested separately.

Table 57: Procedures & Expected Results for E2E uRLLC scalability measurement

Steps	Procedures	Expected Results
1.	Start the UE emulator and attach all emulated UEs to the network, record the time needed to finish all UE attachment.	All emulated UE devices can attach to the network successfully, the attachment rate matched the requirement stated in Table 47
2.	Verify TG IP connectivity from UE emulator to the emulated Data Network.	TG has IP connectivity from UE emulator to the emulated Data Network.
3.	Detach all emulated UEs from the network, record the time needed for all UEs sessions to be cleared.	All emulated UE devices can detach from the network successfully, the detachment rate matched the requirement stated in Table 47.
4.	Repeat step 1-3 for specific Test runs as stated in Table 45, collect the results.	Results show the scalability of uRLLC slice matches the requirement stated in Table 47.

Measurements	Calculate the results for each KPI according to the computation procedure defined in section 0.
Postambles	Execute the postamble procedure defined in section 0

11.6.6 E2E mMTC Scalability Measurement

Purpose	Measure the scalability for the mMTC slice of 5G network.
Description	In this test, we will verify E2E performance of the mMTC slice based on Option 3x (NSA). We will emulate large amount of Machine Type connections to test the performance of adding/deleting massive amount of user sessions for mMTC service through the Non-standalone NR.
Initial Conditions	The following initial conditions are applied for this section: 1. End-to-end network and slice meet the requirements in section 11.1. 2. Emulated end users are detached from the network. Please refer to Figure 4 for the Test Setup diagram.
Parameters	Use the specific test case parameters described below.

Table 58: Parameters for E2E mMTC scalability measurement

Parameters derived from KPIs	Values
Emulated UE quantity	1M
Service under test	mMTC

Procedures & Expected Results	The following table describes the Procedures & Expected Results in detail. For each test step, all packet sizes should be tested separately.

Table 59: Procedures & Expected Results for E2E mMTC scalability measurement

Steps	Procedures	Expected Results
1.	Start the UE emulator and attach all emulated UEs to the network, record the time needed to finish all UE attachment.	All emulated UE devices can attach to the network successfully, the attachment rate matched the requirement stated in Table 47
2.	Verify TG IP connectivity from UE emulator to the emulated Data Network.	TG has IP connectivity from UE emulator to the emulated Data Network.

<table>
<tr><td></td><td>
<table>
<tr><td>3.</td><td>Detach all emulated UEs from the network, record the time needed for all UEs sessions to be cleared.</td><td>All emulated UE devices can detach from the network successfully, the detachment rate matched the requirement stated in Table 47.</td></tr>
<tr><td>4.</td><td>Repeat step 1-3 for specific Test runs as stated in Table 45, collect the results.</td><td>Results show the scalability of mMTC slice matches the requirement stated in Table 47.</td></tr>
</table>
</td></tr>
<tr><td><b>Measurements</b></td><td>Calculate the results for each KPI according to the computation procedure defined in section 0.</td></tr>
<tr><td><b>Postambles</b></td><td>Execute the postamble procedure defined in section 0</td></tr>
</table>

References

[1] https://www.5G-VINNI.eu/ © 5G-VINNI consortium 2018

[2] "Study on New Radio (NR) access technology", 3GPP TR 38.912 Release 15;

[3] "Design of network slicing and supporting systems v1", 5GVINNI Deliverable D1.2;

[4] "Design for systems and interfaces for slice operation v1", 5GVINNI Deliverable D1.3;

[5] "Specification of services delivered by each of the 5G-VINNI facilities", 5GVINNI Deliverable D3.1;

[6] "5GVINNI Test Methodologies and Test Cases", 5GVINNI Deliverable D4.1 Annex A - Test Specification;

[7] "Benchmarking Methodology for Network Interconnect Devices", RFC 2544;

[8] "Ethernet service activation test methodology", ITU-T Y.1564;

[9] "Design of infrastructure architecture and subsystems v1", 5GVINNI Deliverable D1.1;

[10] "5G-VINNI Solution Facility-sites High Level Design (HLD)", 5GVINNI Deliverable D2.1;

(Find some recommendations and examples how to reference on https://ilrb.cf.ac.uk/citingreferences/tutorial/theexamples6.html)

[end of document]