Co-funded by
the Horizon 2020
Framework Programme
of the European Union

MATILDA

A Holistic, Innovative Framework for the Design, Development and Orchestration of 5G-ready Applications and Network Services over Sliced Programmable Infrastructure

DELIVERABLE D6.7

Validation Results, Performance Evaluation and Adoption Guidelines – First demonstration and evaluation phase

Due Date of Delivery:	M21 *Mx* (28/02/2019 *dd/mm/yyyy*)
Actual Date of Delivery:	09/05/2019 *dd/mm/yyyy*
Date of Revision Delivery:	11/02/2020 *dd/mm/yyyy*
Workpackage:	**WP6 – MATILDA Industrial Demonstrators and Performance Evaluation**
Type of the Deliverable:	R
Dissemination level:	PU
Editors:	COSM, CNIT, UBI
Version:	2.0

Call:	H2020-ICT-2016-2
Type of Action:	IA
Project Acronym:	MATILDA
Project ID:	761898
Duration:	35 months
Start Date:	01/06/2017 *dd/mm/yyyy*

Project Coordinator:
Name: Franco Davoli
Phone: +39 010 353 2732
Fax: +39 010 353 2154
e-mail: franco.davoli@cnit.it

Technical Coordinator:
Name: Panagiotis Gouvas
Phone: +30 216 5000 503
Fax: +30 216 5000 599
e-mail: pgouvas@ubitech.eu

List of Authors	
CNIT	CONSORZIO NAZIONALE INTERUNIVERSITARIO PER LE TELECOMUNICAZIONI
R. Bruschi, C. Lombardo	
COSM	COSMOTE KINITES TILEPIKOINONIES AE
G. Lyberopoulos, E. Theodoropoulou, I. Mesogiti, M. Kaisarlis	
UBI	GIOUMPITEK MELETI SCHEDIASMOS YLOPOIISI KAI POLISI ERGON PLIROFORIKIS ETAIREIA PERIORISMENIS EFTHYNIS
P. Gouvas, A. Zafeiropoulos, T. Xirofotos	
INC	INCELLIGENT IDIOTIKI KEFALAIOUCHIKI ETAIREIA
N. Stasinopoulos, A. Ropodi, K. Tsagkaris, P. Demestichas	
ATOS	ATOS SPAIN SA
J. Melian Hernandez	
ERICSSON	ERICSSON TELECOMUNICAZIONI
O. Toscano	
EXXPERT	EXXPERT SYSTEMS GMBH
E. Beckmann, J. Albrecht, U. K. Adusumilli	
ORO	ORANGE ROMANIA
J. Ghenta, H. Stefanescu, M. Iordache	
ININ	INTERNET INSTITUTE Ltd
L. Koršič, D. Mulac, J. Cijan, J. Sterle	
ITA	ITALTEL
A. Albanese	
BIBA	BREMEN INSTITUT FUER PRODUKTION UND LOGISTIK GMBH
K. Burow, Z. Ghrairi	

Disclaimer

The information, documentation and figures available in this deliverable are written by the MATILDA Consortium partners under EC co-financing (project H2020-ICT-761898) and do not necessarily reflect the view of the European Commission.

The information in this document is provided "as is", and no guarantee or warranty is given that the information is fit for any particular purpose. The reader uses the information at his/her sole risk and liability.

Copyright

The MATILDA Consortium consists of:

CONSORZIO NAZIONALE INTERUNIVERSITARIO PER LE TELECOMUNICAZIONI (CNIT)

ATOS SPAIN SA (ATOS)

ERICSSON TELECOMUNICAZIONI (ERICSSON)

INTRASOFT INTERNATIONAL SA (INTRA)

COSMOTE KINITES TILEPIKOINONIES AE (COSM)

ORANGE ROMANIA SA (ORO)

EXXPERTSYSTEMS GMBH (EXXPERT)

GIOUMPITEK MELETI SCHEDIASMOS YLOPOIISI KAI POLISI ERGON PLIROFORIKIS ETAIREIA PERIORISMENIS EFTHYNIS (UBITECH)

INTERNET INSTITUTE, COMMUNICATIONS SOLUTIONS AND CONSULTING LTD (ININ)

INCELLIGENT IDIOTIKI KEFALAIOUCHIKI ETAIREIA (INC)

NATIONAL CENTER FOR SCIENTIFIC RESEARCH "DEMOKRITOS" (NCSRD)

UNIVERSITY OF BRISTOL (UNIVBRIS)

AALTO-KORKEAKOULUSAATIO (AALTO)

UNIVERSITY OF PIRAEUS RESEARCH CENTER (UPRC)

ITALTEL SPA (ITL)

BIBA - BREMER INSTITUT FUER PRODUKTION UND LOGISTIK GMBH (BIBA)

SUITE5 DATA INTELLIGENCE SOLUTIONS LIMITED (S5).

Table of Contents

Table of Acronyms

Acronym	Definition
API	Application Programming Interface
BER	Bit-Error Rate
CSM	Computing Slice Manager
eNB	Evolved Node-B
EPC	Evolved Packet Core
IaaS	Infrastructure as a Service
ICMP	Internet Control Message Protocol
IOPS	Input/Output Operations per second
KPI	Key Performance Indicator
MEC	Mobile Edge Computing
NFV	Network Functions Virtualization
NFVO	Network Functions Virtualization Orchestrator
NS	Network Service
OSS/BSS	Operations and Business Support Systems
PaaS	Platform as a Service
PHP BL	PHP: Hypertext Preprocessor Business Logic
PNF	Physical Network Function
PoP	*Point of Presence*
PPDR	*Public Protection and Disaster Relief*
RAN	Radio Access Network
RSO	Resource Selection Optimizer
SLA	Service Layer Aggreement
SW	Software
UC	Use Case
UE	User Equipment
UI	User Interface

Acronym	Definition
VAO	Vertical Application Orchestrator
VIM	Virtual Infrastructure Manager
VNF	Virtual Network Function
WAN	Wide-Area Network
WIM	Wide-Area Infrastructure Manager

1 Executive Summary

The aim of MATILDA is to deliver "a holistic, innovative framework for design, development and orchestration of 5G-ready applications and network services over sliced programmable infrastructure". The MATILDA project aims to provide a solution as realisation of this framework by unifying network slicing, edge computing and multi-tenancy abstractions into an integrated system, by methodically following the lifecycle process of development, deployment and operation of 5G verticals' use cases. The entire vision will be demonstrated through a set of test cases chosen to highlight different verticals.

In this document, the MATILDA evaluation framework is detailed as flows of validation and evaluation processes, spanning from **MATILDA Solution Components and Functionality Validation** to **General (as a whole) Solution Validation and Evaluation** and further to **Performance Evaluation on the basis of specific KPIs** of MATILDA specific functions and of the whole solution. Validation testing and evaluation flows are addressed at various completion degrees at various project stages, namely: at MATILDA component development phases, at MATILDA components' integration phases, at vertical application on-boarding phases, at MATILDA solution operational phases, at vertical application full deployment phases, and finally at vertical application operational phases. As planned, the list of test objectives and procedures are being refined throughout the project lifetime to better suit implementation specificities that emerge in these project stages, along with testbed specific features, environment setup/tools, etc.

Currently, having finalised the development of a significant number of MATILDA components and having progressed with the partial integration of some of the MATILDA components, the Solution Components and Functionality Validation-related objectives have been refined and specified at the level of specific tests' and success criteria. At this stage of the project, the majority of Solution Components and Functionality Validation tests have been performed - at component development stage, as well as using the MATILDA Demonstrators' Applications during the MATILDA applications' wrapping and on-boarding phase. Preliminary validation results retrieved are the following:

- The Application development and wrapping functionalities (to make an application 5G-ready) have been completely developed and tested (at UBI/CNIT testbeds, using all MATILDA Demonstrators' Applications).
- The lifecycle management (insertion, modification/update, selection, deletion) of applications/application components/VNFs and their metadata in the associated repositories has been tested and successfully validated.
- The Vertical Applications' orchestration and lifecycle management has been tested (at UBI/CNIT testbeds), and successfully validated in terms of enabling Real-Time deployment of an application in various PoPs, including enforcement of specific run-time policies (resource utilisation and security-related).
- A number of Vertical Applications deployment monitoring functionalities have been tested (at UBI/CNIT testbeds and with the PPDR Demonstrator application) and the capability of monitoring compute/network resources utilisation and application behaviour from multiple sources and of extracting Analytics and advanced insights has been successfully validated.

- The lifecycle management of NSs is being finalised and tests (CNIT testbed) have focused on 3GPP network services slice provisioning, and on MEC capabilities using a “Bypass VNF” enabling traffic offloading at edge PoPs; WAN–specific functionality related to the management of network resources on a per-slice basis is also under testing.

Moreover, with the transformation of the MATILDA vertical applications into 5G-ready applications and the on-boarding process of the application graphs having been completed, the MATILDA performance evaluation tests related to the on-boarding process have been refined and specified at the level of specific tests’ and success criteria. Preliminary results have been obtained from the verticals/end-users’ perspective, where it was shown that the on-boarding process of vertical applications (of course after a number of development and feedback cycles between the MATILDA Demonstrators’ and Development teams) is considered to have been user-friendly, understandable in terms of steps to be followed, and the required application descriptive information completely/correctly reflects the applications’ performance/resources requirements.

This is the second version of the document after being reviewed by the EC. In this version an additional section has been included, which provides refined/updated lists of KPIs for the functional and performance evaluation of MATILDA on a per demonstrator basis, as evolved throughout the course of the project until the date of resubmission of document. This version is further complemented with an updated list of risks related to each demonstrator as evolved/revealed also until the date of resubmission of document. Moreover, the actions/ways used in order to address GDPR issues according to EC guidelines in the case of the 5GPACE application are specified.

The final version of the document will include the results related to (1) the MATILDA components integrated solution testing, not only on CNIT/UBI, but also (2) on a number of MATILDA Demonstrators’ testbeds, related to (3) Vertical Applications Deployment and Network Slice Lifecycle Management over a completely integrated MATILDA infrastructure.

2 Introduction

The aim of MATILDA is to deliver “a holistic, innovative framework for design, development and orchestration of 5G-ready applications and network services over sliced programmable infrastructure”.

The MATILDA project will provide a solution as realisation of this framework by unifying network slicing, edge computing and multi-tenancy abstractions into an integrated system by methodically following the lifecycle process of development, deployment and operation of 5G use case verticals. The proposed MATILDA architecture comprises three distinct layers: the Development and Marketplace Environment, which supports all pre-deployment steps of a 5G-enabled application, including the vertical application development and wrapping and the application service graph creation, along with a set of runtime policies used during deployment; the Vertical Application Orchestrator (VAO), in charge of slice intent deployment delivery over the programmable infrastructure; and the Slicing and Management Programmable infrastructure, which is responsible for lifecycle management of the application graph deployment, using network and computing resources from the underlying infrastructure.

The entire vision will be demonstrated through a set of test cases chosen to highlight different verticals. The MATILDA evaluation framework has already been described in Deliverable D6.1 [MATILDA-D6.1], where the different validation and evaluation phases (prior to demonstration) have been defined, along with the preliminary identification of a number of specific test objectives, the associated KPIs and the generic validation method to be followed, including the MATILDA components involved. According to the evaluation framework, tests will span from component validation to functionality validation and evaluation and further to performance evaluation.

The test objectives defined in the framework are being addressed throughout the course of the project at various stages, namely: at MATILDA component development phases, at MATILDA components’ integration phases, at vertical application on-boarding phases, at MATILDA solution operational phases, at vertical application full deployment phases, and finally at vertical application operational phases. As planned, this initial list of test objectives and procedures are being refined throughout the project lifetime to better suit implementation specificities that emerge in these project stages, along with testbed specific features, environment setup/tools, etc.

This deliverable provides an overview of the work done and interim results obtained in the context of Task 6.7 to be finalised towards the project end, and completed in the second revision of this Deliverable. More specifically:

Section 3 provides an overview of the MATILDA Evaluation framework and a mapping between the objectives and the different project stages.

Section 4 provides a refinement in the Solution Components and Functionality Validation-related objectives and specifies them at the level of tests’ and success criteria. It also summarises the retrieved preliminary results from these validation activities.

Section 5 provides a refinement of the Solution Performance evaluation objectives in relation to the vertical applications’ on-boarding process, and further specifies these objectives at the level of tests’ and success criteria, as well. It also summarises the retrieved preliminary evaluation results.

Section 6 includes aspects identified throughout the vertical applications' on-boarding process over all demonstrators to be considered as initial adoption guidelines.

Finally, **Section 7** provides a summary and conclusions of the document.

In this second version of the document, three new annexes have been included as follows:

Following Annex 1 which provides details on the Bypass VNF developed in the context of MATILDA, **Annex 2** provides refined/updated lists of KPIs to be used for the functional and performance evaluation of the solution from each demonstrator perspective.

Annex 3 provides an updated list of risks on a per demonstrator basis.

Finally, **Annex 4** describes the methods/procedures followed in order to address GDPR issues according to EC guidelines in the case of the 5GPACE demonstrator.

3 Evaluation Framework Overview

The initial MATILDA evaluation framework has already been provided in [MATILDA-D6.1]. More specifically, the evaluation framework is structured around a number of test objectives, which reflect complete stakeholders' operational procedures (consisting of one or more MATILDA components/functionalities) or/and complete infrastructure operations (consisting of one or more MATILDA components/functionalities), as well as users'/stakeholders' performance and miscellaneous requirements to be satisfied. These test objectives will be finally evaluated against their associated high-level KPIs.

Complete testing associated with each test objective comprises a number of different validation and evaluation phases and testing procedures spanning from component to functionality validation and evaluation, and further to performance evaluation, specifically targeting:

- **Solution Components and Functionality Validation** aiming at verifying the operation and evaluating the performance of the functionalities/capabilities to be provided by a single (Component Functionality) or by multiple (Complex Functionality) components of the MATILDA solution [MATILDA-D1.1].
- **General Solution Validation** aiming at the evaluation of the solution as a whole for the development and definition of 5G-ready (vertical) applications and NSs and their deployment over a sliced network infrastructure.
- **Performance Evaluation** of specific functions, as well as of the whole MATILDA solution, on the basis of specific applications' KPIs defined in the MATILDA use cases or/and by the MATILDA end-users, as well as towards the 5G-PPP KPIs [MATILDA-D1.1].

Therefore, the test objectives defined in the framework will be addressed throughout the course of the project at various stages, in terms of being:

- refined throughout the project lifetime to better suit implementation specificities that emerge in the various stages,
- elaborated at specific test levels with specific success criteria, even
- tailored to specific testbed features, environment setup/tools, vertical application specificities, etc.

To define the workflow to realise the evaluation framework, we shall consider the MATILDA project implementation phases. In this respect:

- the **MATILDA component development** phases, in which "Solution Components and Functionality Validation" will be performed;
- the **MATILDA components' integration** phases, in which "more complex Functionalities' Validation" and specific "Performance Evaluation" will be performed;
- the **MATILDA solution operational phase**, in which "General Solution Validation" and "Performance Evaluation" will be performed;
- the **vertical application development** (including transformation of application to become "5G-ready") phase, in which "Solution Components and Functionality Validation" and part of "Performance Evaluation" will be performed;

- the **5G-ready (vertical) application on-boarding** phase, in which "Solution Components and Functionality Validation" and part of "Performance Evaluation" will be performed;
- the **5G-ready (vertical) application deployment** phase, in which "Solution Components and Functionality Validation" and part of "Performance Evaluation" will be performed;
- and, finally, the **5G-ready (vertical) application operational phase**, in which "Solution Components and Functionality Validation" and general "Performance Evaluation" will be performed.

These project implementation phases, hence the related validation and evaluation activities, are not followed in a strictly sequential order. The following Figure 1 illustrates the generic approach and a mapping between the MATILDA implementation and the validation and evaluation activities.

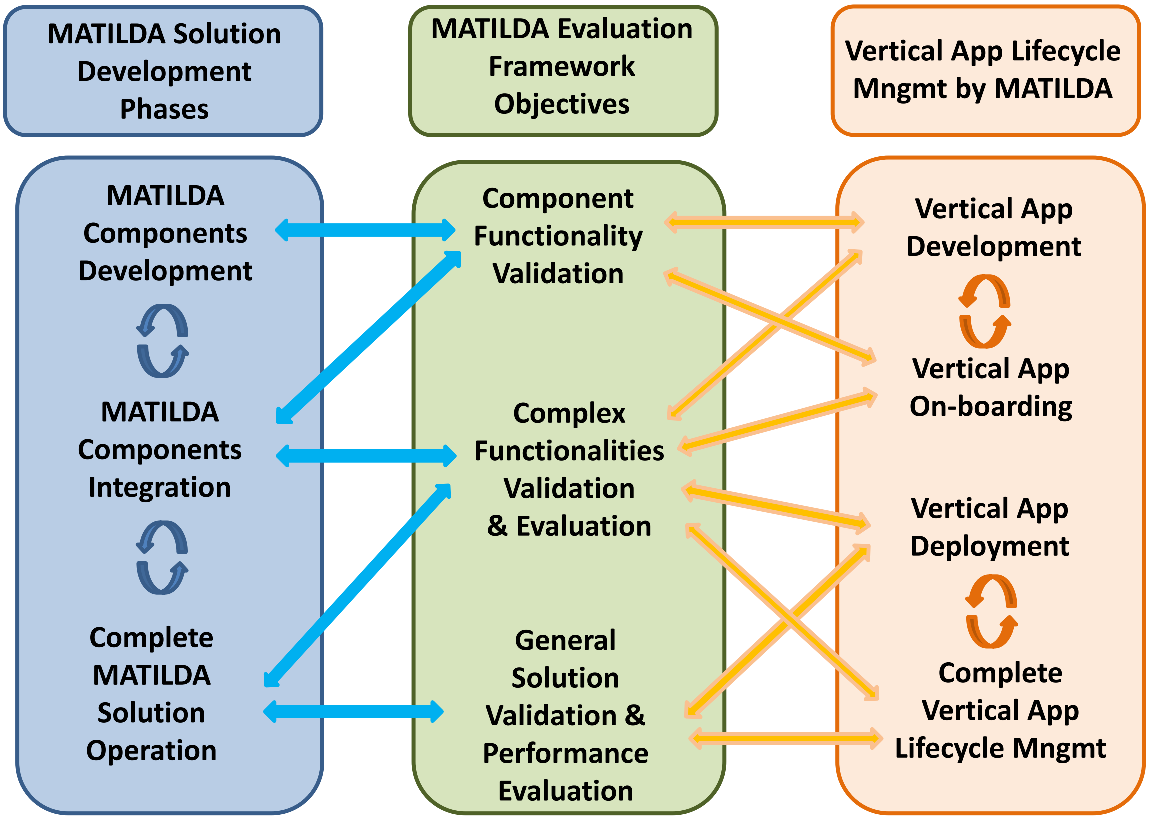

MATILDA Solution Development Phases
MATILDA Evaluation Framework Objectives
Vertical App Lifecycle Mngmt by MATILDA
MATILDA Components Development
MATILDA Components Integration
Complete MATILDA Solution Operation
Component Functionality Validation
Complex Functionalities Validation & Evaluation
General Solution Validation & Performance Evaluation
Vertical App Development
Vertical App On-boarding
Vertical App Deployment
Complete Vertical App Lifecycle Mngmt

Figure 1: MATILDA Evaluation Framework Overview.

3.1 Elaboration of Test Objectives

The MATILDA test objectives have been initially defined in [MATILDA-D6.1], and include the following complete operations:

- 5G-ready Applications Development
- 5G-ready Applications Lifecycle Management in MATILDA repositories[1]
- Vertical Applications Orchestration including Deployment and Monitoring
- Lifecycle Management of a Service Request
- Lifecycle Management of Slices (including slice negotiation and orchestration of network and compute resources)
- Lifecycle Management of NSs (including VNFs)
- Management of Infrastructure Resources (including Multi-site Resource Management)
- Management of Wide-area Network Resources (including Multi-site Resource Management)

The MATILDA solution supports these operations through its main solution components as defined in [MATILDA-D1.1]:

- *The 5G-ready applications development toolkit*, providing support for:
    a. the application/component development and wrapping
    b. the various applications' service graphs' definition/creation/edition
    c. the runtime policies creation/edition.
- *The MATILDA Marketplace*, providing the interface to end users/application owners/verticals and supporting:
    a. the lifecycle management of applications/application components' in the repository
    b. the lifecycle management of VNFs in the repository
    c. the handling of various, different profiles/operations for different users/ stakeholders/roles.
- *The VAO*, enabling:
    a. real-time vertical (5G-ready) application deployment planning; extraction of the slice intent on the basis of the MATILDA metamodels and negotiation of its properties taking under consideration the available programmable resources and the running infrastructure/resources status
    b. enforcement of specific execution policies over the deployed vertical application following a continuous match-resolve-act approach

[1] In the context of this document, the term "vertical application" refers to an application owned/maintained by a vertical industry, and "5G-ready application" to an application that adheres to the MATILDA wrapping principles and metamodels. In some cases, these terms are used interchangeably, because in the context of the MATILDA project all applications used/tested/demonstrated are transformed to 5G-ready version, while representing/belonging to a vertical industry/partner.

   c. monitoring and management of applications/application components through Monitoring and Data fusion mechanisms, and
   d. extraction of advanced insights and events from the analytics data of the Monitoring process, for support of re-active reconfigurations (manually or automatically) of application deployment,
   e. the lifecycle management of the applications (application components) deployment.

- *The NFVO*, supporting the lifecycle management of VNFs (and in some cases PNFs), including configuration and deployment of VNFs by multiple tenants.
- The *OSS/BSS*, providing:
   a. the interface between the VAO and the underlying infrastructure (network and compute resources) domains
   b. the management of network resources within a domain
   c. the monitoring of network nodes/resources within a domain
   d. the creation of the network slices within a domain
   e. the incorporation of VNFs in the network slices within a domain.
- *The Slice manager*, supporting the lifecycle management/high level orchestration of slices.
- *The Virtual Infrastructure Manager (VIM),* exposing the resources of data centers to the NFVO and VAO, supporting multi-tenancy on infrastructure resources. In case of *a multi-domain environment, Multi-Site Resource Management functionalities* are included in the VIM to:
   a. manage resources at diverse cloud facilities,
   b. including the *Computing Slice Manager (CSM)* supporting the deployment of applications/application components at the Network Service Provider's edge facilities.
- *The Wide-area Infrastructure Manager (WIM)*, providing the logical interconnectivity among sets of service/application components instantiated in different PoPs.

Currently, having finalised the development of a significant number of MATILDA components and having progressed with partial integration between some of the MATILDA components, the Solution Components and Functionality Validation-related objectives have been refined and elaborated at the level of specific component functionality and complex functionalities validation tests' and success criteria. Preliminary results have been also obtained for the components that have been developed, as well as for the functionalities that have been completed (delivered through partial integration of a number of components) (see Chapter 4).

The test objectives related to the users'/stakeholders' performance and miscellaneous requirements have been also defined initially in [MATILDA-D6.1] and address aspects such as end-to-end performance evaluation, user friendliness, speed of application deployment, expandability of the solution, scalability, reliability, and so on. These objectives are associated to the General Solution testing; thus, they will be refined and specified at test level at the next stages of the project.

At this point, however, with the transformation of the MATILDA vertical applications into 5G-ready applications and the on-boarding process of these applications' graphs having been completed, the MATILDA general solution evaluation tests related to the on-boarding process have been refined and specified at the level of specific tests' and success criteria. Preliminary results have been also obtained from the verticals'/end-users' perspective.

3.2 Tests Definition Templates

For the purpose of having a homogeneous description of the test objectives, tests and results to be performed in the context of MATILDA have been specified by the contents of the following (Table 1) fixed format tables (the first one also defined in [MATILDA-D6.1], but repeated in this section to facilitate the reading of the rest of the document).

Table 1: Tests Definition in Tabular Format.

Test Objective	<#>	**Type**	<End User Performance/ Functionality / Solution Components / General Solution>
Title	<Title of the Tests.>		
Relevant UCs	<UC #> (applicable only for the End-User Performance tests)		
Validation method – Tests	<Description of the validation method and definition of tests.>		
KPIs	<KPIs and success criteria.>		
Components	<MATILDA solution components; where applicable.>		
Test bed	<Test bed to perform the tests> (if known at this stage)		

Detailed Tests Description: <Title of Test Objective>	
Test #	**Description:** <Elaboration on the tests to be performed towards validating or/and evaluating the associated test objective (described in the previous table).> **Success Criteria:** <Success criteria, against which results will be evaluated.> **Testbed:** <Test bed to perform the tests.>
Results/ Comments	<Description of the expected or obtained results from the associated test, and other relevant comments.>

The following information is associated with the fields of the tables:

- **Number:** This field provides an increasing number to exclusively identify each individual test/set of tests with a specific scope, to ease tracking of its fulfilment in the next steps of the project.
- **Type:** Indicates the category of the test.
- **Title:** The title of the test practically corresponds to the testing purpose.
- **Relevant UCs:** Identifies the Use Case (UC) to which this test is related, and is applicable only to the end-user performance tests.
- **Validation method – Tests:** Provides a brief description of the validation method to be followed and the tests to be performed.
- **KPIs:** Defines the KPIs and the criteria or/and values to evaluate the success of the tests.
- **Components:** Defines the components of the MATILDA solution that are involved or which will be tested. This field is mainly applicable to MATILDA solution and functionality testing.

- **Test bed:** Defines the test bed in which these tests will be performed.

A second table is associated with each test objective (table), which includes the list of elaborated tests (**Test #** - field) to be performed towards validating or/and evaluating the objective against specific success criteria. This table is also used to collect the obtained results of the tests (**Results/Comments** - field).

4 Validation of MATILDA at Component level

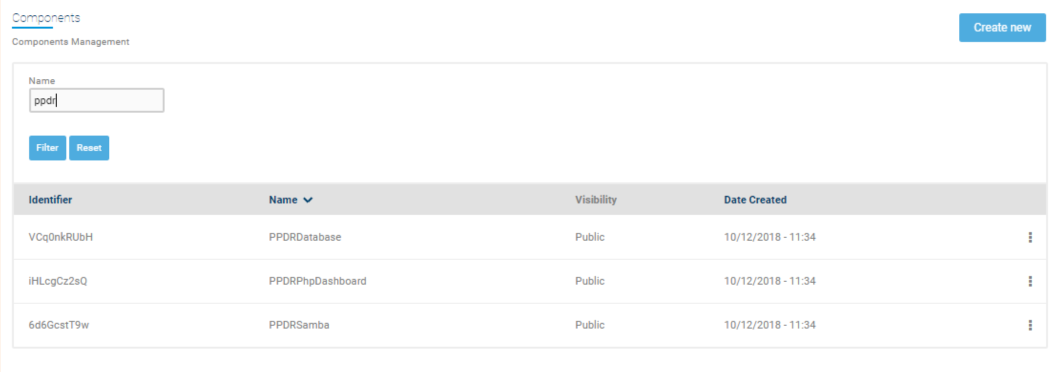
This section provides a refinement of the Solution Components and Functionality Validation-related objectives, as identified and numbered in [MATILDA-D6.1], and an elaboration of these objectives at the level of specific tests' and success criteria. Given the fact that, currently, the development of a number of MATILDA components has been finalised, and some of them have been partially integrated, preliminary test results have been obtained. The latter are summarised in the following tables and figures.

Table 2: 5G-ready Applications Development.

Test Objective	5	**Type**	Functional
Title	5G-ready applications development using MATILDA Toolkit		
Validation method – Tests	Tests related to the 5G-ready applications' development, in particular to: • application component development and wrapping to transform it to cloud-native, including code/wrapping verification, and assessment of the MATILDA Development and Wrapping Toolkit • creation/edition of application service graphs adhering to the MATILDA metamodels to transform it to 5G-ready, including verification of understandability, completeness, assessment of metamodels' and MATILDA Application graph editor • creation/edition of runtime policies at application component level, through the MATILDA Policy Editor.		
KPIs	**Success Criteria:** Successful migration of an on-premises developed application to a 5G-ready version by using the MATILDA 5G-ready Application Development Toolkit.		
Components	Development and Wrapping Toolkit, Application Graph Editor, Policy Editor		
Testbed	UBITECH, CNIT		

Detailed Tests Description: 5G-ready applications development using the MATILDA Toolkit	
Test 1	**Description:** Design and development of components of a 5G-ready application by using the MATILDA toolkit. Application/component development and wrapping so that it becomes cloud-native, including code/wrapping verification, and assessment of the MATILDA Development and Wrapping Toolkit. **Success Criteria:** Error-free wrapping of application components' SW code. Availability of the developed application components in the associated MATILDA components' repository. **Testbed:** UBI – verification (1) at MATILDA Development and Wrapping Toolkit at development and (2) at MATILDA Demonstrators' applications development phases.
Result/ Comments	Tests have been performed focusing on the design, development and registration in the MATILDA Repositories of the components of the MATILDA demonstrators' applications. Indicatively, for the "Emergency Infrastructure with SLA Enforcement" demonstrator, three

components, namely “PPDRDatabase”, “PPDRPhpDashboard” and “PPDRSamba” have been designed and made available in the Components Repository, as depicted in the following figure.



Components

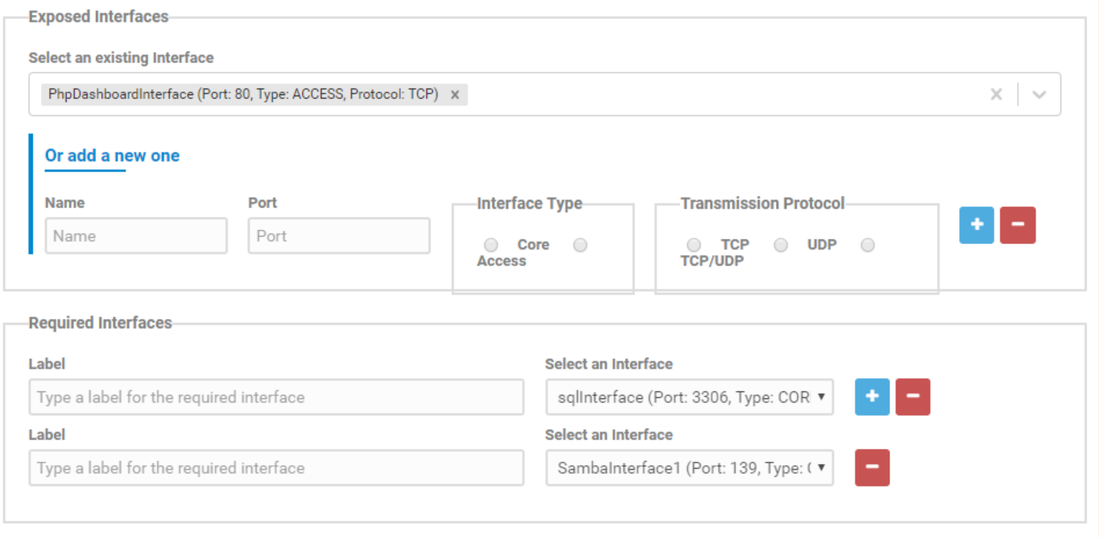
Components Management

Name: ppdr

Identifier	Name	Visibility	Date Created
VCq0nkRUbH	PPDRDatabase	Public	10/12/2018 - 11:34
iHLcgCz2sQ	PPDRPhpDashboard	Public	10/12/2018 - 11:34
6d6GcstT9w	PPDRSamba	Public	10/12/2018 - 11:34

Figure 2: PPDR Application Components in Components Repository.

On a per-component basis, a set of characteristics related to: (a) required and exposed interfaces, (b) minimum execution requirements, (c) elasticity capabilities and (d) environmental variables have been declared and validated in terms of conformance with the MATILDA metamodels (see the following figures). Details on these can be found in [MATILDA-D6.2].



Exposed Interfaces

Select an existing Interface: PhpDashboardInterface (Port: 80, Type: ACCESS, Protocol: TCP)

Or add a new one

Name: Name

Port: Port

Interface Type: ○ Core ○ Access

Transmission Protocol: ○ TCP ○ UDP ○ TCP/UDP

Required Interfaces

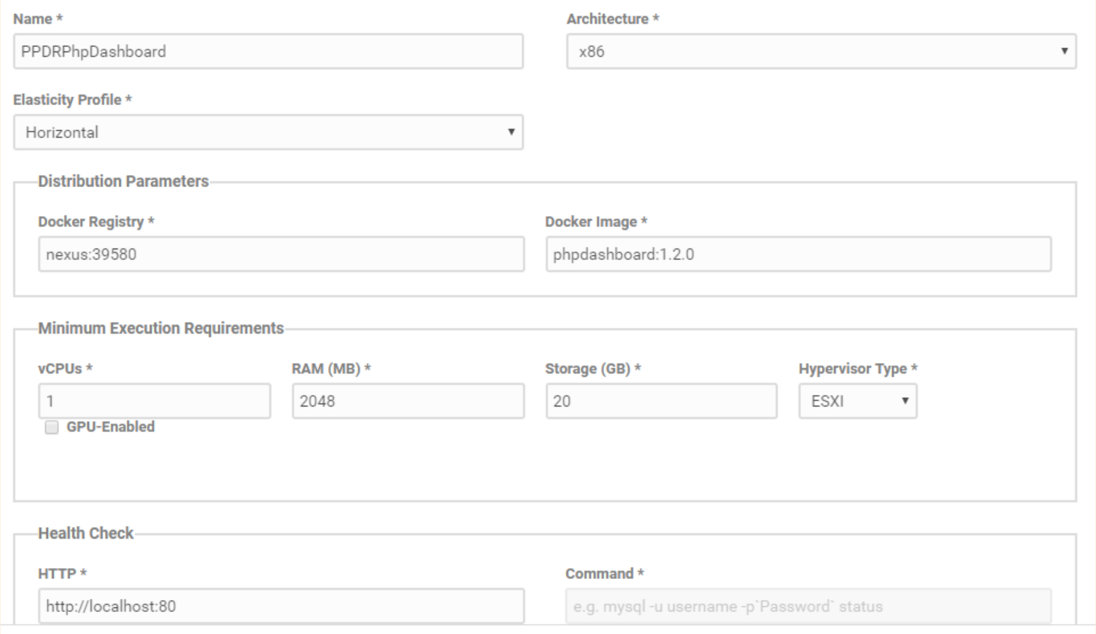
Label: Type a label for the required interface

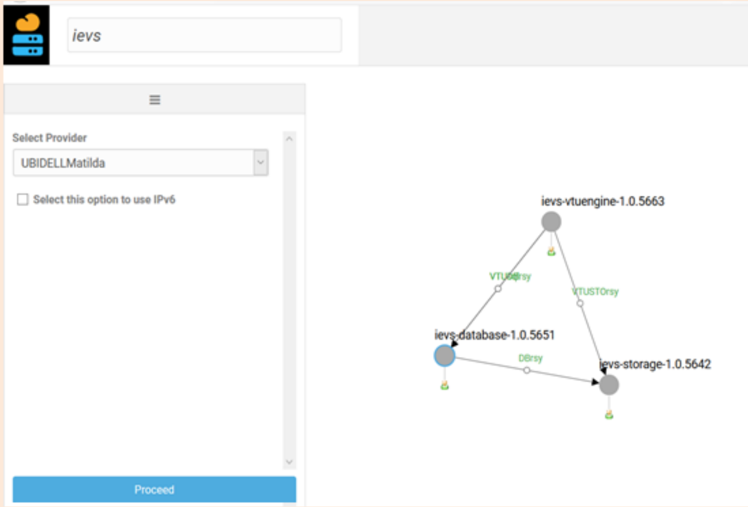
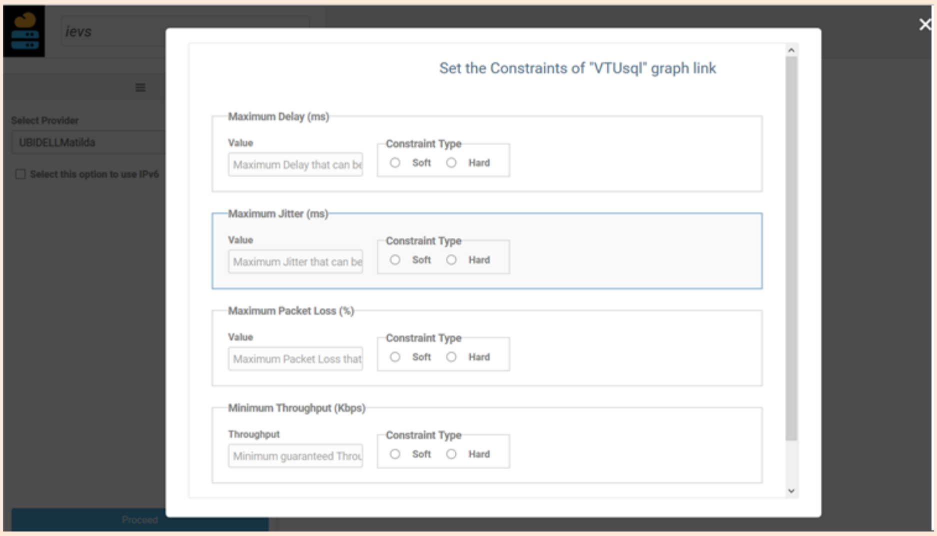
Select an Interface: sqlInterface (Port: 3306, Type: COR

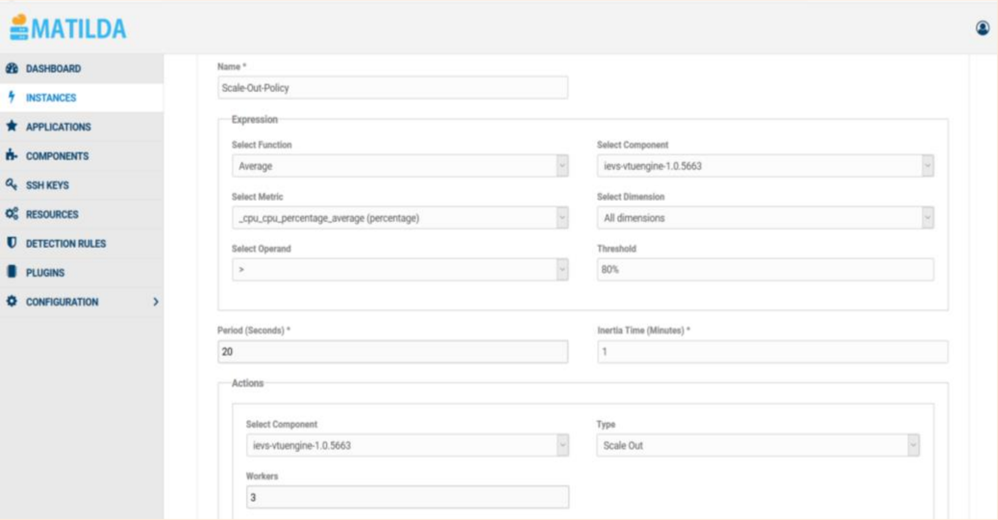
Label: Type a label for the required interface

Select an Interface: SambaInterface1 (Port: 139, Type: (

Figure 3: Application Component Interfaces Specification.

	 Name * PPDRPhpDashboard Architecture * x86 Elasticity Profile * Horizontal Distribution Parameters Docker Registry * nexus:39580 Docker Image * phpdashboard:1.2.0 Minimum Execution Requirements vCPUs * 1 ☐ GPU-Enabled RAM (MB) * 2048 Storage (GB) * 20 Hypervisor Type * ESXI Health Check HTTP * http://localhost:80 Command * e.g. mysql -u username -p`Password` status **Figure 4: Application Component Characteristics' Specification.** Similarly, details on these processes performed with other MATILDA demonstrator applications can be found in [MATILDA-D6.3]-[MATILDA-D6.6].
Test 2	**Description:** Creation/edition of application service graphs adhering to the MATILDA metamodels. Verification/validation of the Application Graph Editor functionality, in terms of: • Capability to select a number of application components from the MATIDLA components' repository to form the application • Capability to define the links/communication interfaces between them • Capability to define/edit the application components' characteristics and requirements (incl. execution requirements, interfaces, etc.) as defined in the Chainable Application Component & 5G-ready Application graph metamodels • Capability to define all the network resource requirements as defined in the Network-aware Application Graph Metamodel. **Success Criteria**: Support of the aforementioned functionalities/capabilities. **Testbed**: UBI – verification (1) at Application graph Editor development phase, and (2) at MATILDA Demonstrators' applications on-boarding phase.
Result/ Comments	Tests have been performed with the first version of the Graph Composer. The latter has been used for the composition of application graphs, incorporating and interconnecting application components that are available in the relevant repository. The following steps have been successfully realised/tested (shown also in the following figures): • Selection of a number of application components from the associated repository in order to create an application graph. • Linking of the application components based on their required and exposed interfaces.

	- Specification of the network resource requirements per link, in particular with regard to link guaranteed and/or desirable data rate, jitter, delay, packet loss (specification of the slice intent).  ievs Select Provider UBIDELLMatilda ☐ Select this option to use IPv6 Proceed ievs-vtuengine-1.0.5663 VTUDBrsy VTUSTOrsy ievs-database-1.0.5651 DBrsy ievs-storage-1.0.5642 **Figure 5: Application Graph Creation.**  Set the Constraints of "VTUsql" graph link Maximum Delay (ms) Value: Maximum Delay that can be Constraint Type: ○ Soft ○ Hard Maximum Jitter (ms) Value: Maximum Jitter that can be Constraint Type: ○ Soft ○ Hard Maximum Packet Loss (%) Value: Maximum Packet Loss that Constraint Type: ○ Soft ○ Hard Minimum Throughput (Kbps) Throughput: Minimum guaranteed Throu Constraint Type: ○ Soft ○ Hard **Figure 6: Network Resource Requirements Specification per Link.** As verified, the application graph was successfully created and validated towards being ready for instantiation. This process/testing steps were followed and the functionality was verified with all MATILDA demonstrator applications (see [MATILDA-D6.2]-[MATILDA-D6.6]).
Test 3	**Description:** Creation/Editing of application/components' elasticity runtime policies' definition during the design time through the MATILDA Policy Editor. **Success Criteria**: Capability to specify valid runtime policies based on the usage of the Policy Editor. **Testbed**: UBI/CNIT – verification (1) at MATILDA Policy Editor development phase, and (2) at PPDR Demonstrator on-boarding and deployment phases.

<table>
<tr>
<td><b>Result/ Comments</b></td>
<td>
<p>As defined in [MATILDA-D1.5], a set of policies should be specified using the Policy Editor, and the expressions are validated immediately upon saving. At the time of writing, this functionality has been tested for a number of policies triggering scaling-out actions on the basis of compute resources utilisation thresholds.</p>
<p>Indicatively, the following figure shows the definition of an elasticity policy triggering a scaling action when the CPU load is above 80%. The rule was validated automatically upon saving during the MATILDA Policy Editor development phase.</p>


<p>Name * Scale-Out-Policy</p>
<p>Expression
Select Function: Average
Select Component: ievs-vtuengine-1.0.5663
Select Metric: _cpu_cpu_percentage_average (percentage)
Select Dimension: All dimensions
Select Operand: &gt;
Threshold: 80%</p>
<p>Period (Seconds) * 20
Inertia Time (Minutes) * 1</p>
<p>Actions
Select Component: ievs-vtuengine-1.0.5663
Type: Scale Out
Workers: 3</p>
<p><b>Figure 7: Scale-out Policy Definition.</b></p>
<p>Scaling out policies’ definition and validation on the basis of custom application component metrics was also verified with the PPDR Demonstrator application (at UBI and CNIT testbeds by the selection of average http requests rate of the PPDR Dashboard component), while similar tests will be performed with the rest of MATILDA demonstrator applications over the declared horizontally scalable components.</p>
</td>
</tr>
<tr>
<td><b>Test 4</b></td>
<td>
<p><b>Description:</b> Creation/Editing of application/components’ security runtime policies definition during the design time through the MATILDA Policy Editor.</p>
<p><b>Success Criteria</b>: Capability to define the aforementioned runtime policies based on security criteria.</p>
<p><b>Testbed</b>: UBI – Initial verification at MATILDA Policy Editor development phase.</p>
</td>
</tr>
<tr>
<td><b>Result/ Comments</b></td>
<td>
<p>A set of security policies can be specified by using the Policy Editor. In the following figure, a security policy is defined to “alert host on ICMP package arrival”. The rule was validated automatically upon saving during the MATILDA Policy Editor development phase.</p>
</td>
</tr>
</table>

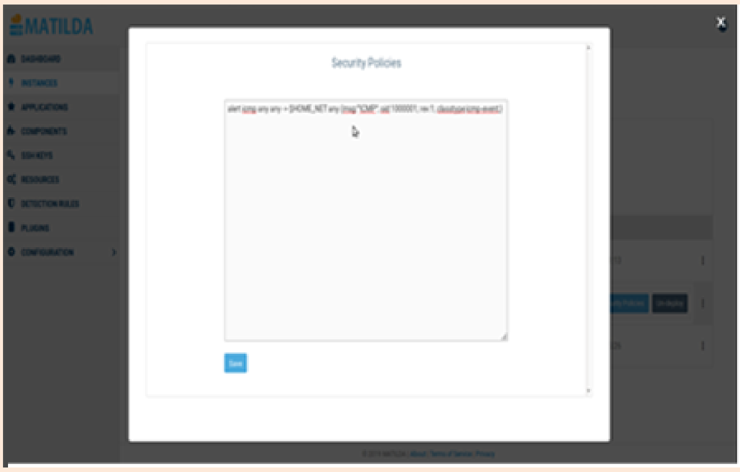
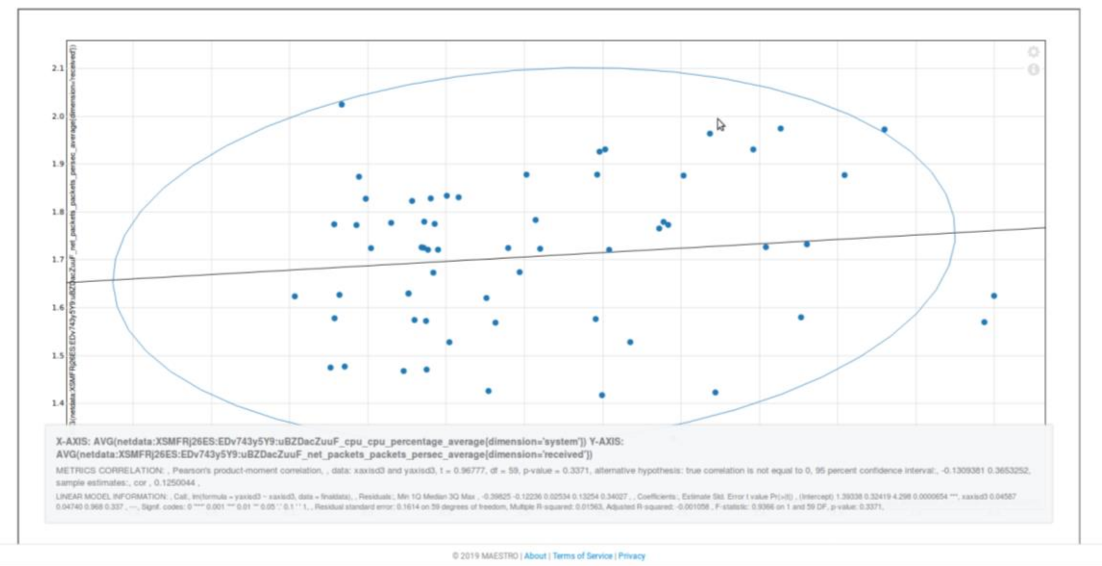
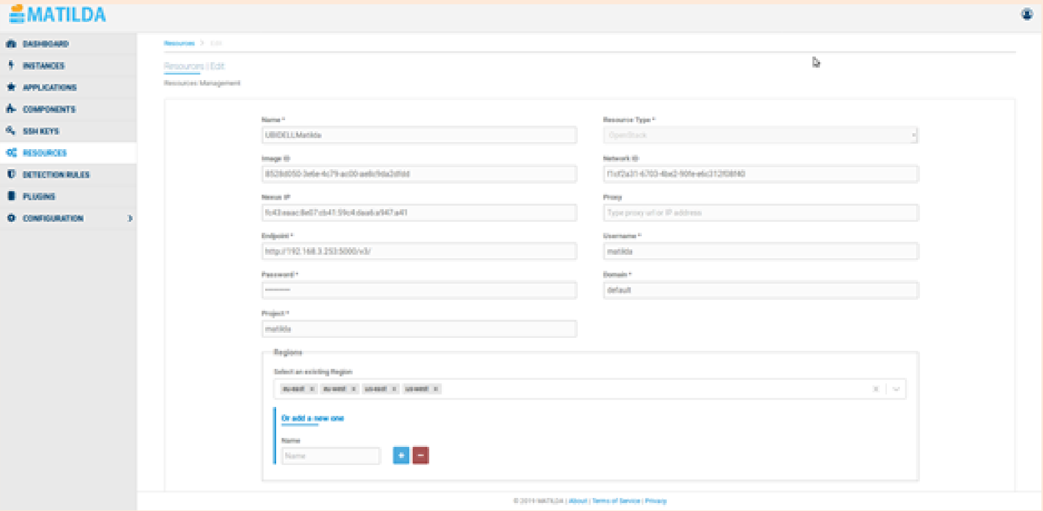
	 MATILDA DASHBOARD INSTANCES APPLICATIONS COMPONENTS SSH KEYS RESOURCES DETECTION RULES PLUGINS CONFIGURATION Security Policies Save alert icmp any any -> $HOME_NET any (msg:"ICMP"; sid:1000001; rev:1; classtype:icmp-event;) **Figure 8: Security Policy Definition.**
Test 5	**Description:** Extraction of Slice attributes from the MATILDA application components' profiling functionality. **Success Criteria**: Capability to define the aforementioned runtime policies based on application components' profiling. **Testbed**: UBI – Initial verification at MATILDA , profiling functionality development phase.
Result/ Comments	The Slice intent attributes comprise a set of compute and network resource requirements that have to be considered during the deployment of an application. The definition of these attributes can be performed based on existing information and expertise of the software developer/DevOps MATILDA user or based on the results produced by resource efficiency and elasticity efficiency profiling mechanisms (an indicative linear regression result is shown in the following figure 9). Test have been performed at MATILDA profiling functionality development phase using internal applications. The profiling results are used for the final definition of the deployment and operational requirements of an application that are included in the slice intent description. Similar tests will be performed with the rest of MATILDA demonstrator applications.  X-AXIS: AVG(netdata:XSMFRj26ES:EDv743y5Y9:uBZDacZuuF_cpu_cpu_percentage_average(dimension='system')) Y-AXIS: AVG(netdata:XSMFRj26ES:EDv743y5Y9:uBZDacZuuF_net_packets_packets_persec_average(dimension='received')) METRICS CORRELATION: , Pearson's product-moment correlation, , data: xaxisd3 and yaxisd3, t = 0.96777, df = 59, p-value = 0.3371, alternative hypothesis: true correlation is not equal to 0, 95 percent confidence interval:, -0.1309381 0.3653252, sample estimates:, cor , 0.1250044 , LINEAR MODEL INFORMATION: , Call:, lm(formula = yaxisd3 ~ xaxisd3, data = finaldata), , Residuals:, Min 1Q Median 3Q Max , -0.39825 -0.12236 0.02534 0.13254 0.34027 , , Coefficients:, Estimate Std. Error t value Pr(>|t|) , (Intercept) 1.39338 0.32419 4.298 0.0000654 ***, xaxisd3 0.04587 0.04740 0.968 0.337 , ---, Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1, , Residual standard error: 0.1614 on 59 degrees of freedom, Multiple R-squared: 0.01563, Adjusted R-squared: -0.001058 , F-statistic: 0.9366 on 1 and 59 DF, p-value: 0.3371, **Figure 9: MATILDA Profiling Functionality Results.**

Table 3: 5G-Ready Applications' Lifecycle Management in the MATILDA Marketplace.

Test Objective	6	**Type**	Functional
Title	5G-Ready Applications' Lifecycle Management in the MATILDA Marketplace		
Validation method – Tests	Tests to be performed are related to the lifecycle management of 5G-ready applications/ components/ VNFs through the MATILDA Marketplace, including: • verification of the interface to end-users/application owners/verticals in terms of including all necessary functionality for these stakeholders • the lifecycle management of applications/application components/VNFs modules and their metadata in the repository, including: ○ insertion ○ modification/update ○ selection ○ deletion ○ users' access rights definition/alteration • the handling of the user rights for various, different profiles/functions for different users/stakeholders/roles.		
KPIs	**Success Criteria:** Successful performance of the functionalities that have been specified to be performed through the MATILDA marketplace interface on a per user/role basis. Consistency maintained between the information shown through GUIs with the actual repository information, and the specified rules on a per user/role basis.		
Components	MATILDA Marketplace interface, Component Repository, VNF repository, Application Graph Repository		
Testbed	All		

Detailed Tests Description: 5G-Ready Applications' Lifecycle Management Testing	
Test 1	**Description:** Registration of Infrastructure Resources/Domains on which a 5G-ready application can be deployed. **Success Criteria**: Capability to register different types of resources/domains to enable the design/deployment of components/graphs. **Testbed**: UBI - verification (1) at MATILDA Marketplace development and (2) at MATILDA Demonstrators' applications on-boarding phases.
Result/ Comments	A set of resources are registered and made available for deployment. The registration of resources has been tested under various infrastructure types like: Amazon Web Services, Google Cloud and OpenStack.

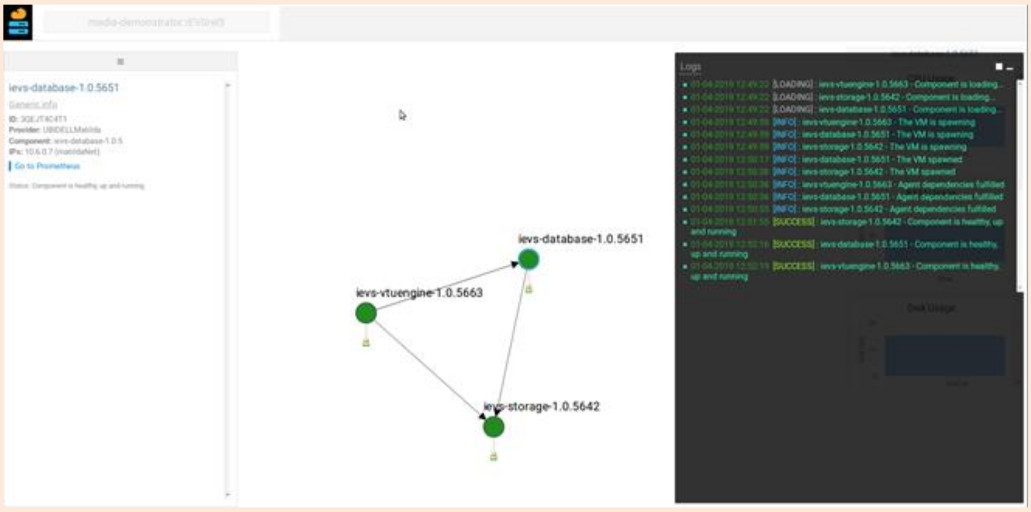
	 MATILDA DASHBOARD INSTANCES APPLICATIONS COMPONENTS SSH KEYS RESOURCES DETECTION RULES PLUGINS CONFIGURATION Resources > Edit Resources | Edit Resources Management Or add a new one + − © 2019 MATILDA | About | Terms of Service | Privacy Name * UBIDELLMatilda Resource Type * OpenStack Image ID 8528d050-3e6e-4c79-ac00-ae8c9da2dfdd Network ID f1cf2a31-6703-4be2-90fe-e6c312f08f40 Nexus IP fc43:eeac:8e07:cb41:59c4:daa6:a947:a41 Proxy Type proxy url or IP address Endpoint * http://192.168.3.253:5000/v3/ Username * matilda Password * •••••••• Domain * default Project * matilda Regions Select an existing Region eu-east × eu-west × us-east × us-west × Name Name **Figure 10: Registration of PoPs' Resources.**
Test 2	**Description:** Verification/validation of performing 5G-ready applications' Lifecycle Management procedures through the MATILDA Marketplace, including: • insertion of a new application/ application component • modification/update • selection/query • deletion **Success Criteria**: Capability to perform the aforementioned 5G-ready applications' Lifecycle Management procedures through the MATILDA Marketplace. **Testbed**: UBI – verification (1) at MATILDA Marketplace development phase and (2) at MATILDA Demonstrators' applications on-boarding phases.
Result/ Comments	Insertion, editing, modification and removal of application components, applications and graphs in the associated MATILDA repositories was heavily tested and the functionalities were successfully verified throughout the MATILDA Demonstrators' applications' on-boarding phases; detailed descriptions can be found in [MATILDA-D6.2]-[MATILDA-D6.6]
Test 3	**Description:** VNFs Lifecycle Management in the MATILDA Marketplace, including: • insertion of a new VNF/NS • modification/update • selection/query • deletion **Success Criteria**: Capability to perform the aforementioned VNFs' Lifecycle Management procedures in the MATILDA Marketplace. **Testbed**: UBI/ATOS - Initial verification at MATILDA Marketplace development phase.
Result/ Comments	Insertion, editing, modification and removal of VNFs in the associated MATILDA repository was initially tested at ATOS testbed and the functionalities were successfully verified.

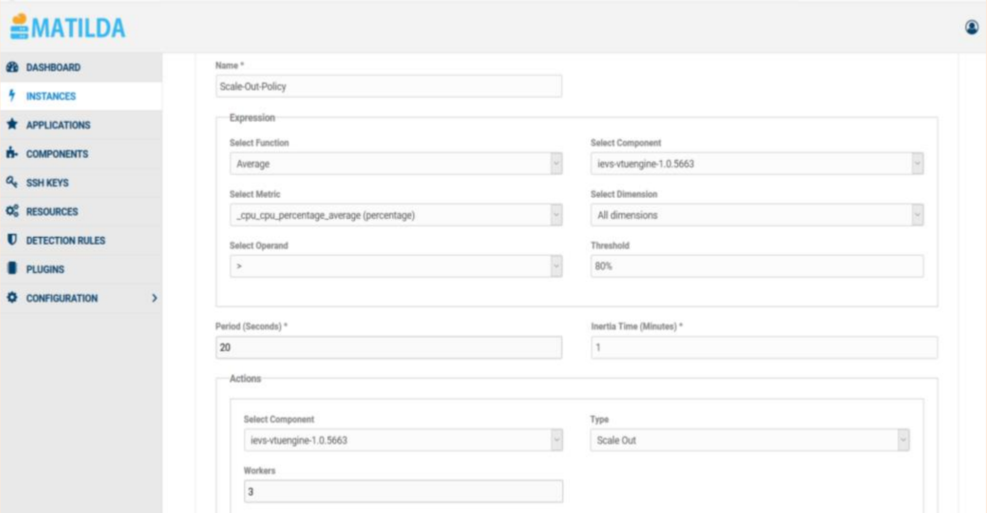
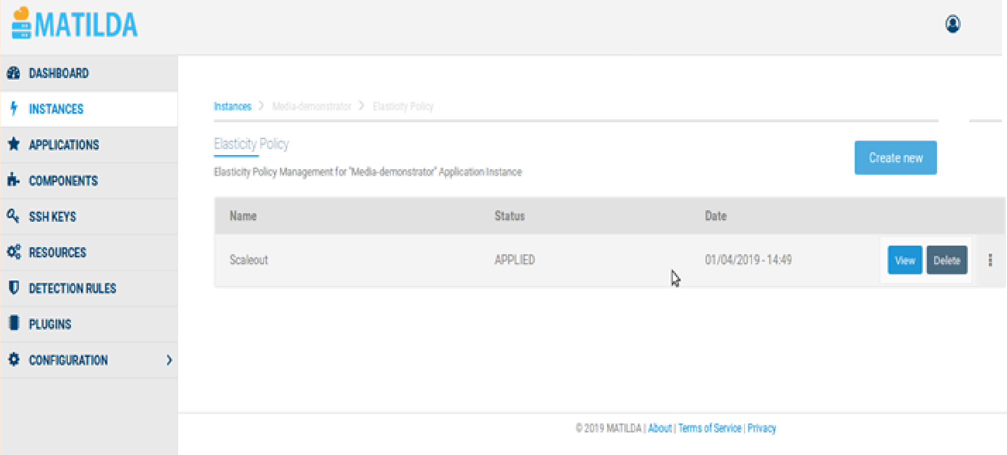
	Further tests will be performed at the MATILDA demonstrators' application deployment phases, after the complete integration of the MATILDA Marketplace VNFs repository with the WIM and VIMs in the next project period.

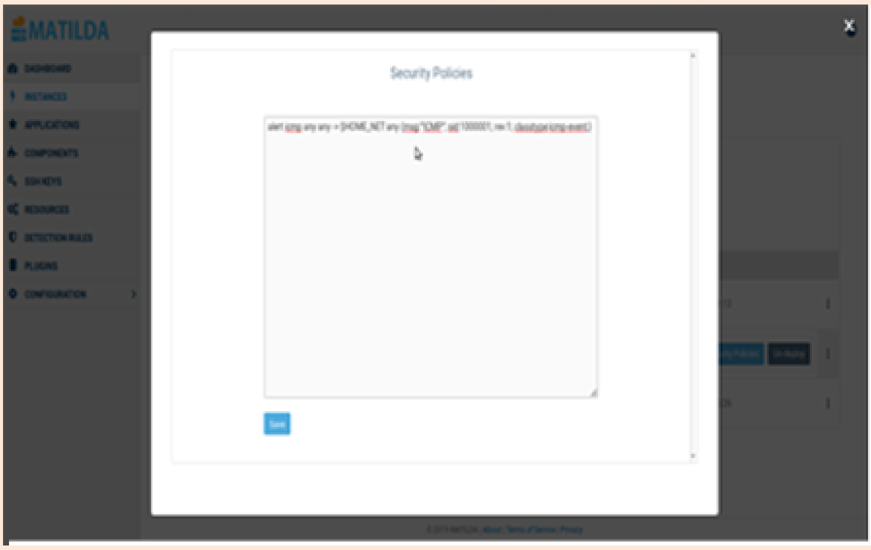
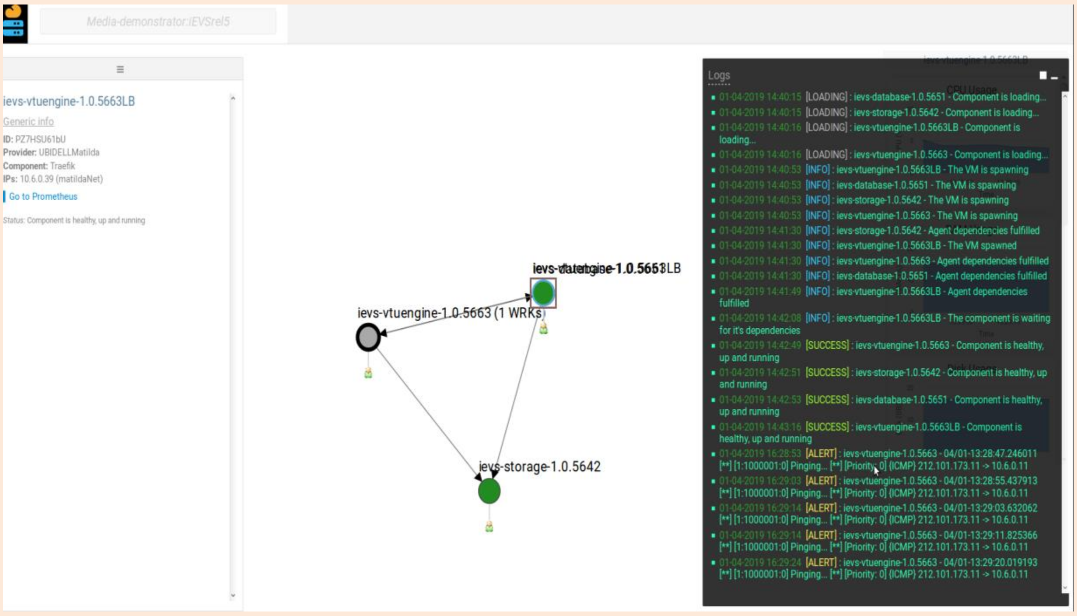
Table 4: Vertical Applications' Orchestration and Lifecycle Management.

Test Objective	7 & 9 (unified)	**Type**	Functional
Title	Vertical Applications' Orchestration		
Validation method – Tests	Tests to be performed are related to the real-time deployment of a 5G-Ready, Vertical Application through MATILDA, including: • the extraction of the slice intent from the service graphs definitions on the basis of the MATILDA metamodels • delivery of Real-Time deployment planning of the vertical application components optimized by taking into account: the application service graph, the relevant execution policies, the programmable resources availability in various PoPs and the network resources' availability • enforcement of specific execution policies over the deployed vertical application following a continuous match-resolve-act approach, based on monitoring data and analytics • termination of application instance operation upon request.		
KPIs	**Success Criteria:** Successful performance of the functionalities related to the optimized, real-time deployment of a (5G-ready) Vertical Application, in terms of requested resources and provisioned ones taking into account the infrastructure capabilities. Successful performance of the functionalities related to the re-active reconfiguration of a (5G-ready) Vertical Application deployment. Successful performance of the functionalities related to Vertical Application termination.		
Components	VAO, Optimisation Engine, Policy Engine, Intelligent Proxy, Execution Manager		
Testbed	All		

Detailed Tests Description: Vertical Applications' Orchestration Testing	
Test 1	Lifecycle management of a Vertical Application, including: • instantiation of Application Graph • modification (if needed) of Application Graph • termination/deletion of Application Graph **Success Criteria**: Capability to apply and monitor lifecycle management functions during the overall lifecycle of a Vertical Application. **Testbed**: UBI –verification (1) at MATILDA VAO development phase and, at the next stage, (2) at MATILDA demonstrators' application deployment phases.

Result/ Comments	The Vertical Application lifecycle management includes operations that span from the application graph instantiation, where each application component is initially loaded, up to the termination of the application provisioning, where all application components are terminated/deleted. In these tests, it was verified that upon request the VAO can spawn successfully a VM per component and that the component's dependencies are fulfilled. Afterwards, each component enters an operational phase until the termination of application provisioning is decided. The following figure 11 demonstrates an application graph that has been instantiated by the VAO (along with the logs during the initialization process).  media-demonstratorE5thl0 ievs-database-1.0.5651 Generic info ID: 3QEJT4C4T1 Provider: UBIDELLMatilda Component: ievs-database-1.0.5 IPs: 10.6.0.7 (matildaNet) Go to Prometheus Status: Component is healthy up and running ievs-database-1.0.5651 ievs-vtuengine-1.0.5663 ievs-storage-1.0.5642 Logs 01-04-2019 12:49:22 [LOADING] : ievs-vtuengine-1.0.5663 - Component is loading... 01-04-2019 12:49:22 [LOADING] : ievs-storage-1.0.5642 - Component is loading... 01-04-2019 12:49:22 [LOADING] : ievs-database-1.0.5651 - Component is loading... 01-04-2019 12:49:59 [INFO] : ievs-vtuengine-1.0.5663 - The VM is spawning 01-04-2019 12:49:59 [INFO] : ievs-database-1.0.5651 - The VM is spawning 01-04-2019 12:49:59 [INFO] : ievs-storage-1.0.5642 - The VM is spawning 01-04-2019 12:50:17 [INFO] : ievs-database-1.0.5651 - The VM spawned 01-04-2019 12:50:38 [INFO] : ievs-storage-1.0.5642 - The VM spawned 01-04-2019 12:50:36 [INFO] : ievs-vtuengine-1.0.5663 - Agent dependencies fulfilled 01-04-2019 12:50:36 [INFO] : ievs-database-1.0.5651 - Agent dependencies fulfilled 01-04-2019 12:50:55 [INFO] : ievs-storage-1.0.5642 - Agent dependencies fulfilled 01-04-2019 12:51:55 [SUCCESS] : ievs-storage-1.0.5642 - Component is healthy, up and running 01-04-2019 12:52:16 [SUCCESS] : ievs-database-1.0.5651 - Component is healthy, up and running 01-04-2019 12:52:19 [SUCCESS] : ievs-vtuengine-1.0.5663 - Component is healthy, up and running **Figure 11: Instantiation of an Application Graph by the VAO.**
Test 2	**Description**: Verification that the slice provisioning (resources (slice) that are initially allocated) to an application/application graph upon its deployment/instantiation on infrastructure is in accordance with the slice intent. **Success Criteria**: Correct deployment of application components, so that the application is functional upon instantiation triggered by the VAO. Consistency maintained between the VAO information and the actual application/application graph state. **Testbed**: CNIT/UBI - Initial verification at VAO development phase.
Result/ Comments	Proper slice provisioning demonstrated during the demo in November: the VAO is able to launch the VMs composing the application (with the correct resources as defined in the Slice Intent) and they communicate properly among each other (as specified in the connectivity links defined in the slice intent).
Test 3	**Description:** Enforcement of elasticity (runtime) policies at component/application level, defined through the MATILDA Policy Editor and verification of the deriving actions' triggering, in particular for the following runtime policies: • Scale out in case of exceeding resources utilisation: definition of threshold and margin (time - resources) for various parameters (e.g.), verification of scale out performance. **Success Criteria**: Capability to enforce the aforementioned runtime policies based on resource criteria.

	Testbed: UBI –verification (1) at MATILDA VAO development phase and, at the next stage, (2) at MATILDA demonstrators' application deployment phases.
Result/ Comments	The actual enforcement of the specified policies was tested over a running instance of an application and the triggering and enforcement of the specified rules and actions was successful. The following figures show the designed elasticity policy, information regarding the firing up of a rule and the successful realisation of a scaling action. The scaling out policy enforcement in the PPDR Dashboard (scale out an instance in case of high CPU usage and high avg http requests rate) was also verified with the PPDR Demonstrator application at the PPDR Demonstrator deployment phases (at UBI and CNIT testbeds). Similar tests will be performed at the next stage with other MATILDA vertical applications.  MATILDA DASHBOARD INSTANCES APPLICATIONS COMPONENTS SSH KEYS RESOURCES DETECTION RULES PLUGINS CONFIGURATION Name * Scale-Out-Policy Expression Select Function: Average Select Component: ievs-vtuengine-1.0.5663 Select Metric: _cpu_cpu_percentage_average (percentage) Select Dimension: All dimensions Select Operand: > Threshold: 80% Period (Seconds) * 20 Inertia Time (Minutes) * 1 Actions Select Component: ievs-vtuengine-1.0.5663 Type: Scale Out Workers: 3 **Figure 12: Scaling out Policy Definition - PPDR Application.**  MATILDA DASHBOARD INSTANCES APPLICATIONS COMPONENTS SSH KEYS RESOURCES DETECTION RULES PLUGINS CONFIGURATION Create new View Delete Instances > Media-demonstrator > Elasticity Policy Elasticity Policy Elasticity Policy Management for "Media-demonstrator" Application Instance Name Status Date Scaleout APPLIED 01/04/2019 - 14:49 © 2019 MATILDA | About | Terms of Service | Privacy **Figure 13: Scaling out Policy Enforcement - PPDR Application.**
Test 4	**Description:** Enforcement of security (runtime) policies at component/application level, defined through the MATILDA Policy Editor, and verification of the deriving actions' triggering, in particular for runtime policies based on **detection of alert or intrusion mechanisms.**

	Success Criteria: Capability to enforce the aforementioned runtime policies based on security related criteria. **Testbed**: UBI –verification (1) at MATILDA VAO development phase and at the next stage (2) at MATILDA demonstrators' application deployment phases.
Result/ Comments	The actual enforcement of a security policy -defined to "alert host on ICMP package arrival"- was tested over a running instance of an application and the triggering and enforcement of the specified rules and actions was successful.  MATILDA DASHBOARD INSTANCES APPLICATIONS COMPONENTS SSH KEYS RESOURCES DETECTION RULES PLUGINS CONFIGURATION Save Security Policies alert icmp any any -> $HOME_NET any (msg:"ICMP" sid:1000001; rev:1; classtype:icmp-event;) **Figure 14: Security Policy Definition.**  ievs-vtuengine-1.0.5663 (1 WRKs) ievs-vtuengine-1.0.5663LB ievs-database-1.0.5651 ievs-storage-1.0.5642 Media-demonstrator:IEVSrel5 ievs-vtuengine-1.0.5663LB Generic info **ID:** PZ7HSU61bU **Provider:** UBIDELLMatilda **Component:** Traefik **IPs:** 10.6.0.39 (matildaNet) Go to Prometheus Status: Component is healthy, up and running Logs ▪ 01-04-2019 14:40:15 [LOADING] : ievs-database-1.0.5651 - Component is loading... ▪ 01-04-2019 14:40:15 [LOADING] : ievs-storage-1.0.5642 - Component is loading... ▪ 01-04-2019 14:40:16 [LOADING] : ievs-vtuengine-1.0.5663LB - Component is loading... ▪ 01-04-2019 14:40:16 [LOADING] : ievs-vtuengine-1.0.5663 - Component is loading... ▪ 01-04-2019 14:40:53 [INFO] : ievs-vtuengine-1.0.5663LB - The VM is spawning ▪ 01-04-2019 14:40:53 [INFO] : ievs-database-1.0.5651 - The VM is spawning ▪ 01-04-2019 14:40:53 [INFO] : ievs-storage-1.0.5642 - The VM is spawning ▪ 01-04-2019 14:40:53 [INFO] : ievs-vtuengine-1.0.5663 - The VM is spawning ▪ 01-04-2019 14:41:30 [INFO] : ievs-storage-1.0.5642 - Agent dependencies fulfilled ▪ 01-04-2019 14:41:30 [INFO] : ievs-vtuengine-1.0.5663LB - The VM spawned ▪ 01-04-2019 14:41:30 [INFO] : ievs-vtuengine-1.0.5663 - Agent dependencies fulfilled ▪ 01-04-2019 14:41:30 [INFO] : ievs-database-1.0.5651 - Agent dependencies fulfilled ▪ 01-04-2019 14:41:49 [INFO] : ievs-vtuengine-1.0.5663LB - Agent dependencies fulfilled ▪ 01-04-2019 14:42:08 [INFO] : ievs-vtuengine-1.0.5663LB - The component is waiting for it's dependencies ▪ 01-04-2019 14:42:49 [SUCCESS] : ievs-vtuengine-1.0.5663 - Component is healthy, up and running ▪ 01-04-2019 14:42:51 [SUCCESS] : ievs-storage-1.0.5642 - Component is healthy, up and running ▪ 01-04-2019 14:42:53 [SUCCESS] : ievs-database-1.0.5651 - Component is healthy, up and running ▪ 01-04-2019 14:43:16 [SUCCESS] : ievs-vtuengine-1.0.5663LB - Component is healthy, up and running ▪ 01-04-2019 16:28:53 [ALERT] : ievs-vtuengine-1.0.5663 - 04/01-13:28:47.246011 [**] [1:1000001:0] Pinging... [**] [Priority: 0] {ICMP} 212.101.173.11 -> 10.6.0.11 ▪ 01-04-2019 16:29:03 [ALERT] : ievs-vtuengine-1.0.5663 - 04/01-13:28:55.437913 [**] [1:1000001:0] Pinging... [**] [Priority: 0] {ICMP} 212.101.173.11 -> 10.6.0.11 ▪ 01-04-2019 16:29:14 [ALERT] : ievs-vtuengine-1.0.5663 - 04/01-13:29:03.632062 [**] [1:1000001:0] Pinging... [**] [Priority: 0] {ICMP} 212.101.173.11 -> 10.6.0.11 ▪ 01-04-2019 16:29:14 [ALERT] : ievs-vtuengine-1.0.5663 - 04/01-13:29:11.825366 [**] [1:1000001:0] Pinging... [**] [Priority: 0] {ICMP} 212.101.173.11 -> 10.6.0.11 ▪ 01-04-2019 16:29:24 [ALERT] : ievs-vtuengine-1.0.5663 - 04/01-13:29:20.019193 [**] [1:1000001:0] Pinging... [**] [Priority: 0] {ICMP} 212.101.173.11 -> 10.6.0.11 **Figure 15: Security Policy Enforcement.**
Test 5	**Description:** Verification of proper termination of application instance operation including the update of VAO with regard to the application status and the release of allocated resources.

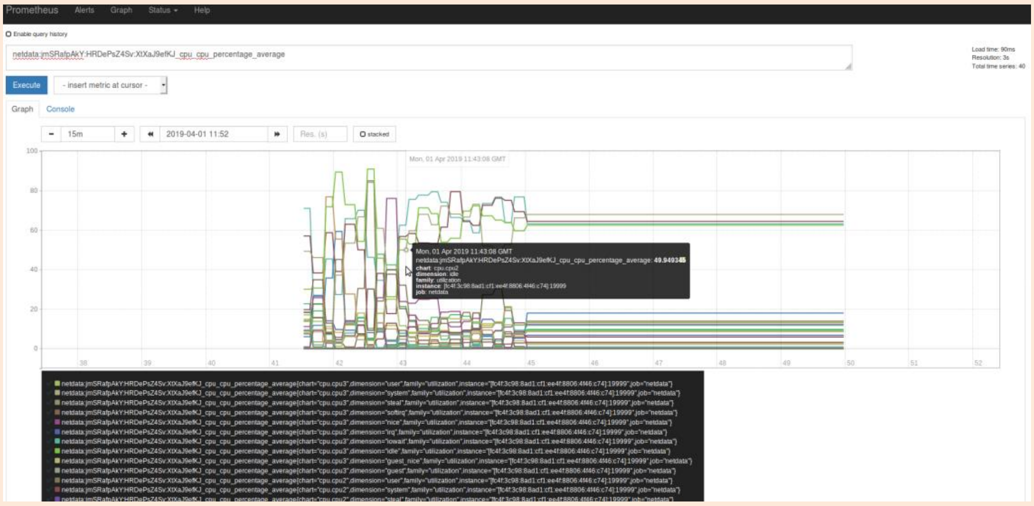
	Success Criteria: Upon trigger (e.g., manually from VAO), correct termination/deletion of the application instance and release of the allocated resources. Consistency maintained between the VAO information and the actual application/application graph state. **Testbed**: UBI –verification (1) at MATILDA VAO development phase, (2) at PPDR Demonstrator application deployment phase and at the next stage (3) at other MATILDA demonstrators' application deployment phases.
Result/ Comments	Tests have been performed to verify the correct termination/deletion of the application instance and release of the allocated resources, triggered manually from the VAO, and it was successfully verified that resources were correctly released and consistency was maintained between the information presented in VAO and the actual status of the application.

Table 5: Vertical Applications' Deployment Monitoring.

Test Objective	8	**Type**	Functional
Title	Vertical Applications' Deployment Monitoring		
Validation method – Tests	Tests to be performed are related to the real-time and historical monitoring of a Vertical Application deployment, including: • real-time monitoring of multiple applications/application components through a set of active and passive probes, • incorporation of monitoring processes defined in the application service graphs/metamodels • Fusion of monitoring data coming from multiple parallel data loads from multiple sources • support of Real-Time Analytics of multiple contexts • extraction of advanced insights and events from the monitoring process, e.g. through data mining, as well as predictive and prescriptive analytics mechanisms (i.e. regression, clustering or classification algorithms) • evaluation of the extracted information in terms of validity, usefulness, versatility, effectiveness and sophisticated processing.		
KPIs	**Success Criteria:** Successful performance of real-time monitoring of multiple applications/ application components • Extraction of Real-time Descriptive Analytics for all the application components • Data Fusion of data coming from all application components • Generation of Real-time Predictive Analytics for metrics coming from all application components • Representation of Fused Descriptive and Predictive Analytics on an Analytics Dashboard to support infrastructure DevOps and development decision making		

	The obtained results/information are valid, useful, versatile depending on the nature of the application, effective towards undertaking corrective actions, and processing is sophisticated leading to advanced conclusions.
Components	VAO, Stream Aggregator, Data Fusion and Real-time Analytics
Testbed	All

Detailed Tests Description: 5G-Ready Applications Deployment Monitoring Testing	
Test 1	**Description:** Verification of incorporation of monitoring processes defined in the application service graphs/metamodels. Such monitoring processes can be included either as application components or a VNFs and it shall be suitable to be configured accordingly. **Success Criteria:** Verification of definition/development of monitoring processes to be included either as application components or as VNFs. **Testbed:** UBI – Initial verification at MATILDA VAO development phase.
Result/ Comments	It was verified that the netdata plugins denoted in the application components specification (at the application components' wrapping phase) are activated and are providing relevant monitoring data to Prometheus.
Test 2	**Description:** Verification of real-time monitoring of multiple applications/application components through a set of active and passive probes. 1. Testing of proper initiation of active and passive probes. 2. Testing of configuration of active and passive probes (e.g. in terms of measurements' interval per parameter, measurement window, interface to retrieve measurement, etc.) 3. Testing of retrieval of the following parameters: • Compute Resources utilisation in terms of: CPU, RAM, IOPS, etc. • Network Resources utilisation in terms of: bandwidth, latency, etc. • Application/Application components load in terms of: function calls, APIs' utilization, open connections, database query load, application latency, etc. **Success Criteria:** Correct retrieval of measurements of the aforementioned parameters (correctness in terms of data and in terms of being in accordance with the probes' configuration). **Testbed:** UBI – Initial verification at MATILDA VAO development phase.
Result/ Comments	Monitoring is supported based on a set of netdata plugins, while monitoring data is collected and provided by Prometheus. Indicatively, monitoring metrics for CPU and sent/ received TCP segments collected during the application's operation are shown in figures 16 and 17 below.



○ Enable query history

netdata:jmSRafpAkY:HRDePsZ4Sv:XtXaJ9efKJ_cpu_cpu_percentage_average

Load time: 90ms
Resolution: 3s
Total time series: 40

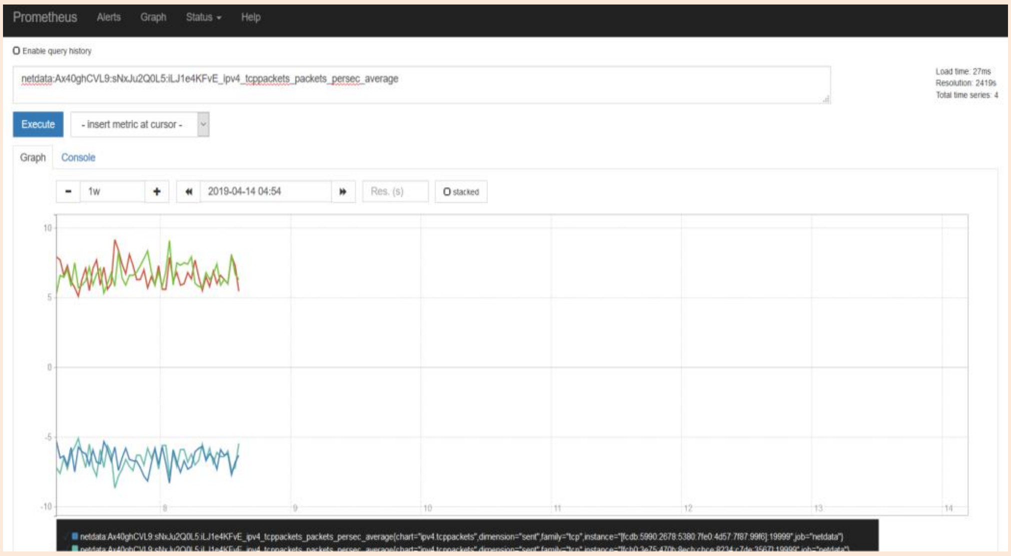
- insert metric at cursor -

15m | 2019-04-01 11:52 | Res. (s) | ○ stacked

Mon, 01 Apr 2019 11:43:08 GMT
netdata:jmSRafpAkY:HRDePsZ4Sv:XtXaJ9efKJ_cpu_cpu_percentage_average: 49.949345
chart: cpu.cpu2
dimension: idle
family: utilization
instance: [fc4f:3c98:8ad1:cf1:ee4f:8806:4f46:c74]:19999
job: netdata

```
netdata:jmSRafpAkY:HRDePsZ4Sv:XtXaJ9efKJ_cpu_cpu_percentage_average{chart="cpu.cpu3",dimension="user",family="utilization",instance="[fc4f:3c98:8ad1:cf1:ee4f:8806:4f46:c74]:19999",job="netdata"}
netdata:jmSRafpAkY:HRDePsZ4Sv:XtXaJ9efKJ_cpu_cpu_percentage_average{chart="cpu.cpu3",dimension="system",family="utilization",instance="[fc4f:3c98:8ad1:cf1:ee4f:8806:4f46:c74]:19999",job="netdata"}
netdata:jmSRafpAkY:HRDePsZ4Sv:XtXaJ9efKJ_cpu_cpu_percentage_average{chart="cpu.cpu3",dimension="steal",family="utilization",instance="[fc4f:3c98:8ad1:cf1:ee4f:8806:4f46:c74]:19999",job="netdata"}
netdata:jmSRafpAkY:HRDePsZ4Sv:XtXaJ9efKJ_cpu_cpu_percentage_average{chart="cpu.cpu3",dimension="softirq",family="utilization",instance="[fc4f:3c98:8ad1:cf1:ee4f:8806:4f46:c74]:19999",job="netdata"}
netdata:jmSRafpAkY:HRDePsZ4Sv:XtXaJ9efKJ_cpu_cpu_percentage_average{chart="cpu.cpu3",dimension="nice",family="utilization",instance="[fc4f:3c98:8ad1:cf1:ee4f:8806:4f46:c74]:19999",job="netdata"}
netdata:jmSRafpAkY:HRDePsZ4Sv:XtXaJ9efKJ_cpu_cpu_percentage_average{chart="cpu.cpu3",dimension="irq",family="utilization",instance="[fc4f:3c98:8ad1:cf1:ee4f:8806:4f46:c74]:19999",job="netdata"}
netdata:jmSRafpAkY:HRDePsZ4Sv:XtXaJ9efKJ_cpu_cpu_percentage_average{chart="cpu.cpu3",dimension="iowait",family="utilization",instance="[fc4f:3c98:8ad1:cf1:ee4f:8806:4f46:c74]:19999",job="netdata"}
netdata:jmSRafpAkY:HRDePsZ4Sv:XtXaJ9efKJ_cpu_cpu_percentage_average{chart="cpu.cpu3",dimension="idle",family="utilization",instance="[fc4f:3c98:8ad1:cf1:ee4f:8806:4f46:c74]:19999",job="netdata"}
netdata:jmSRafpAkY:HRDePsZ4Sv:XtXaJ9efKJ_cpu_cpu_percentage_average{chart="cpu.cpu3",dimension="guest_nice",family="utilization",instance="[fc4f:3c98:8ad1:cf1:ee4f:8806:4f46:c74]:19999",job="netdata"}
netdata:jmSRafpAkY:HRDePsZ4Sv:XtXaJ9efKJ_cpu_cpu_percentage_average{chart="cpu.cpu3",dimension="guest",family="utilization",instance="[fc4f:3c98:8ad1:cf1:ee4f:8806:4f46:c74]:19999",job="netdata"}
netdata:jmSRafpAkY:HRDePsZ4Sv:XtXaJ9efKJ_cpu_cpu_percentage_average{chart="cpu.cpu2",dimension="user",family="utilization",instance="[fc4f:3c98:8ad1:cf1:ee4f:8806:4f46:c74]:19999",job="netdata"}
netdata:jmSRafpAkY:HRDePsZ4Sv:XtXaJ9efKJ_cpu_cpu_percentage_average{chart="cpu.cpu2",dimension="system",family="utilization",instance="[fc4f:3c98:8ad1:cf1:ee4f:8806:4f46:c74]:19999",job="netdata"}
netdata:jmSRafpAkY:HRDePsZ4Sv:XtXaJ9efKJ_cpu_cpu_percentage_average{chart="cpu.cpu2",dimension="steal",family="utilization",instance="[fc4f:3c98:8ad1:cf1:ee4f:8806:4f46:c74]:19999",job="netdata"}
```

Figure 16: CPU–related Monitoring.



○ Enable query history

netdata:Ax40ghCVL9:sNxJu2Q0L5:iLJ1e4KFvE_ipv4_tcppackets_packets_persec_average

Load time: 27ms
Resolution: 2419s
Total time series: 4

- insert metric at cursor -

1w | 2019-04-14 04:54 | Res. (s) | ○ stacked

```
netdata:Ax40ghCVL9:sNxJu2Q0L5:iLJ1e4KFvE_ipv4_tcppackets_packets_persec_average{chart="ipv4.tcppackets",dimension="sent",family="tcp",instance="[fcdb:5990:2678:5380:7fe0:4d57:7f87:99f6]:19999",job="netdata"}
netdata:Ax40ghCVL9:sNxJu2Q0L5:iLJ1e4KFvE_ipv4_tcppackets_packets_persec_average{chart="ipv4.tcppackets",dimension="sent",family="tcp",instance="[fcb0:3e75:470b:8ecb:cbce:8234:c7de:3567]:19999",job="netdata"}
```

Figure 17: Sent/ Received TCP Packets Monitoring.

Similar tests will be performed at the next stage with the MATILDA vertical applications' deployments.

Test 3

Description: Verification of fusion of monitoring data coming from multiple parallel data loads from multiple sources, including:

1. Testing of validity of data and retrieval of a set of listed performance/utilisation/etc. parameters (KPIs) for all application components by the pub-sub mechanism
2. Testing of correlation between performance/ utilisation/etc. parameters (KPIs) in each component and across components
3. Testing the validity of composite/aggregated performance/ utilisation/etc. parameters (KPIs) that feed next stages of descriptive and predictive real-time analysis

4. Testing of the scalability of the fusion and retrieval procedures, as well as the historical persistence of aggregated metrics.

Success Criteria: Correct fusion of monitoring data coming from multiple parallel data loads from multiple sources.

Testbed: UBI - verification (1) at Stream Aggregator development phase, and (2) at MATILDA PPDR demonstrator application's deployment phase.

Result/ Comments

In this phase of the project, having performed the aforementioned tests with the PPDR application, the following results have been achieved:

1. Successful validation of data and verification of correct retrieval of performance/utilisation/etc. parameters (KPIs) (as those defined in Test 2) for the application graph components of the PPDR application coming from the Prometheus monitoring data flows.
2. Successful build-up of correlation of parameters (KPIs) and complex composite parameters (KPIs) for the application graph components of the PPDR application.
3. Successful validation of the composite/aggregate parameters (KPIs) were used for the build-up of real-time descriptive and predictive time-series modelling.
4. Successful scaling up of the fusion mechanism with multiple workers which aggregate data and train the predictive model.

In the following figure 18, the metrics selected to be visualised are: new and predicted average disk size required per operation, new and predicted average system iops (Input/Output per second), new and predicted average applications' CPU, CPU time and CPU percentage.

For the training of the models and feature selections, all available metrics of the PPDR deployed instance were used.

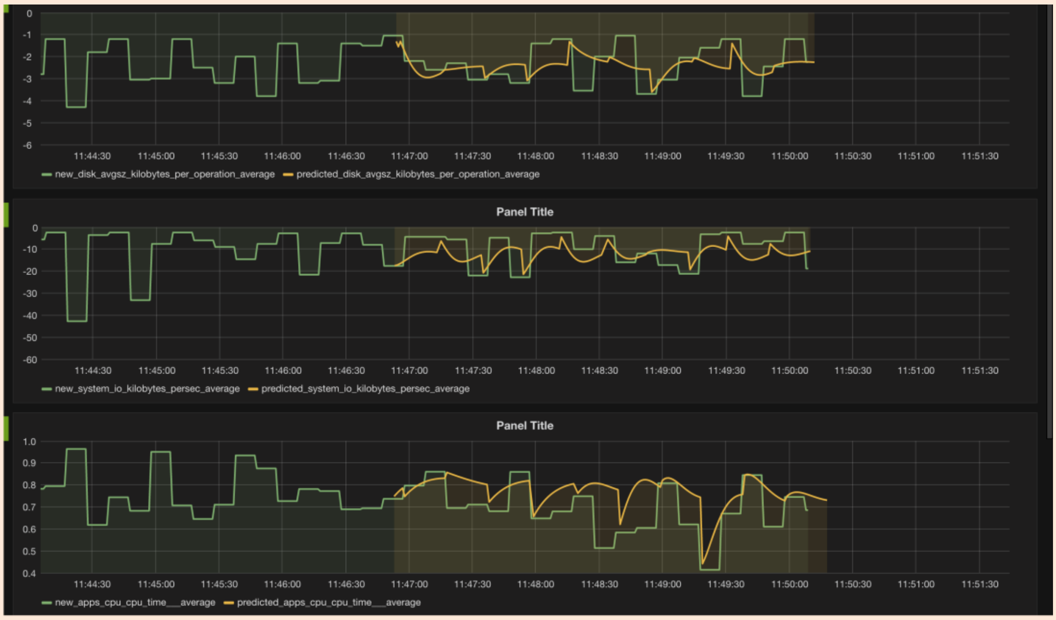


Figure 18: Visualisation of Performance Metrics for the PPDR Application.

Comments/Next steps:

	- For points (1)-(3), future tests will involve the newly on-boarded use-cases, (including cross-models between different use-cases). - For point (4), future tests will follow scalability experiments on the monitoring and application graph side.
Test 4	**Description:** Verification of Real-Time Analytics retrieval of multiple contexts, in terms of: 1. Testing of the ability to process real-time data -whether standalone or aggregated- received from streaming sources and file systems. 2. Testing of deployment of streaming algorithms (e.g. for regression), which can simultaneously learn from the streaming data as well as apply the model on the streaming data. **Success Criteria:** Correct processing of incoming data and implementation of streaming algorithms. **Testbed:** UBI - verification (1) at Data Analytics component development phase, and (2) at MATILDA PPDR demonstrator application deployment phase.
Result/ Comments	Standalone testing of Real-Time Analytics component has been performed with a test use-case (PPDR) data and artificially generated data. As verified, the component is able to: 1. Receive Kafka streams and identify/count “events” within specific timeframes real-time, and 2. Train a streaming model in near real-time (i.e. real-time regression) through Incelligent’s dedicated streaming framework (≤100 time units). Next steps will include the integration with other VAO components, and further testing of the functionality with other MATILDA demonstrators’ applications and testbed data.
Test 5	**Description:** Verification of extraction of advanced insights and events from the monitoring process, e.g. through data mining, as well as predictive and prescriptive analytics mechanisms (i.e. regression, clustering or classification algorithms). **Success Criteria:** Successful deployment of specific KPI prediction models based on historical data and measurement of their performance. **Testbed:** UBI- Initial verification at Data Analytics development phase.
Result/ Comments	At this project phase, the integration of the monitoring process with advanced machine learning processing libraries has been tested for batch analytics, i.e. offline learning using historical data. The capability to measure model performance KPIs, based on the model employed, has been tested as well. The following screenshot shows a correlation graph between monitored attributes of an application graph.

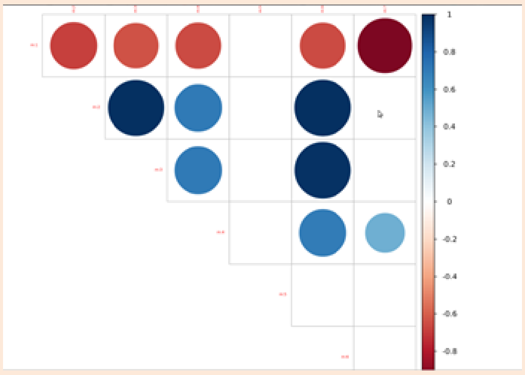


Figure 19: Correlation Graph of an Application Graph.

Standalone testing of Data Fusion and Analytics component with artificially generated data.

As verified, the component is able to:

- Deploy a model based on batch analytics using Spark MLib, specifically for regression and classification using historical (batch) data;
- Collect model training/inference performance metrics for model tuning.

Next steps: Integration with other VAO components, and further testing with more demonstrators' applications and testbed data.

Table 6: Lifecycle Management of a Service Request.

Test Objective	10	**Type**	Functional
Title	Lifecycle management of a service request and high-level orchestration of network and compute resources (OSS/BSS operation)		
Validation method – Tests	Tests to be performed include: • the interface between the VAO and the underlying network and compute resources domains • the high-level orchestration of the creation of the network/compute slices within a domain • the interaction with the NFVO for the incorporation of NSs in the network slices within a domain • the management of resources within a domain • the monitoring of nodes/resources within a domain.		
KPIs	**Success Criteria:** Successful performance of OSS/BSS functionalities related to the lifecycle management of a service request and the high-level orchestration of network and compute resources. Consistency maintained between the OSS/BSS information and the actual provisioning of the requested service.		
Components	OSS/BSS, NFVO, VIMs		
Testbed	All		

Detailed Tests Description: OSS/BSS Operation Testing	
Test 1	**Description**: Verification of the interface between the VAO and the underlying network and compute resources domains in terms of consistency and correctness of mapping/cross-checking of service graphs' resource requirements to resource requests accounts across a single domain or multiple domains. Verification of consistency maintained between the VAO originated application requests and the user account privileges/services (maintained in user SLAs). **Success Criteria:** Correct mapping between the slice intent provided by the VAO and the resources requested to the VIM(s) and NFVO. Consistency maintenance/resolution between the VAO originated application requests and the user account privileges/services (maintained in user SLAs). **Testbed:** CNIT/UBI
Result/ Comments	Verification of consistency between slice intent requirements and selected computing resources demonstrated during the demo in November. Correspondence between resources (in terms of RAM/disk space, etc.) allocated for the VMs and what required in the slice intent. Regarding the NS provisioning, this is planned to be fully tested at next project stage with the integration of the NFVO in the MATILDA framework (at the time of writing in progress); validation results to be included in D6.13.
Test 2	**Description**: Verification of the high-level orchestration of the creation of the network/compute slices within a domain. Verification of keeping track of requests for slices from VAO, of provisioned slices and resources' offers, along with user account privileges/services information. **Success Criteria:** A repository keeps track of the required user and status information. Testbed: CNIT/UBI
Result/ Comments	Creation of the network/compute slices demonstrated during the demo in November. This information is collected in the persistency store that currently is not equipped with a GUI.
Test 3	**Description**: Verification of the interaction of OSS/BSS with the NFVO for the incorporation of VNFs and NSs in the network slices extracted from the slice intent received from the VAO (within a domain). OSS/BSS shall request the instantiation of NSs upon resolution of slice intent. **Success Criteria:** Correct resolution of slice intent in terms of identifying required VNFs. Correct communication with NFVO, and correct instantiation of NSs. **Testbed:** CNIT/UBI
Result/ Comments	Planned to be tested at next project stage with the integration of the NFVO in the MATILDA framework (at the time of writing in progress); validation results to be included in D6.13.

Test 4	**Description**: Verification of the management of resources within a domain including also the monitoring of nodes/resources within this domain. Verification of monitoring of resources in terms of availability of compute nodes and corresponding network links/connectivity. **Success Criteria:** Correct monitoring of compute nodes resources. Correct monitoring of network links/connectivity resources. **Testbed:** CNIT/UBI
Result/ Comments	Planned to be tested at the next project stage as the algorithms in the RSO only provide basic functionalities at this stage and require further extensions to provide more refined outputs according to monitored parameters; validation results to be included in D6.13.
Test 5	**Description**: Verification of the high-level orchestration of the creation of the network/compute slices over multiple domains. Verification of keeping track of requests for slices from VAO, of provisioned slices, and resources' offers along with user account privileges/services information. **Success Criteria:** A repository keeps track of the required user and status information for multiple domains. **Testbed:** CNIT/UBI
Result/ Comments	Planned to be tested at next project stage with the integration of the NFVO in the MATILDA framework (at the time of writing in progress); validation results to be included in D6.13.
Test 6	**Description**: Verification of the management of resources within a domain including also the monitoring of nodes/resources within multiple domains. Verification of monitoring of resources in terms of availability of compute nodes and corresponding network links/connectivity. **Success Criteria:** Correct monitoring of compute nodes resources. Correct monitoring of network links/connectivity resources. **Testbed:** CNIT/UBI
Result/ Comments	Planned to be tested at the next project stage, because the monitoring of resources across multiple domains is bound to the integration of the WIM within the MATILDA framework, which is still in progress; validation results to be included in D6.13.

Table 7: Lifecycle Management of Slices.

Test Objective	11	**Type**	Functional
Title	Lifecycle Management of Slices		
Validation method – Tests	Tests to be performed are related to the lifecycle management of slices on the infrastructure, and include: • the initial provisioning of the slice (in terms of compute/network resources and QoS) requested from the slice intent for a specific application deployment		

	- the deletion of the slice (release of resources) upon application instance termination.
KPIs	**Success Criteria:** Successful performance of functionalities related to the lifecycle management of a slice.[2] Consistency maintained between the Slice Manager information and the actual slice state.
Components	Slice Manager (as part of the OSS, and Resource Selector Optimizer)
Testbed	All

Detailed Tests Description: Lifecycle Management of Slices Testing	
Test 1	**Description:** Verification that upon instantiation of a slice the relevant information is maintained at the Slice Manager side. **Success Criteria:** Consistency maintained between the Slice Manager information and the actual slice state. **Testbed:** CNIT/UBI
Result/ Comments	Planned to be tested at final project stage with the complete integration of WIM and NFVO in the MATILDA framework (at the time of writing in progress); validation results to be included in D6.13.
Test 2	**Description:** Verification that upon deletion of a slice the relevant information is maintained at the Slice Manager side. **Success Criteria:** Consistency maintained between the Slice Manager information and the actual slice state. **Testbed:** CNIT/UBI
Result/ Comments	Upon the termination of the provision of an application, the reserved resources are released and made available for upcoming deployments. The functionality is tested and validated in the VAO that includes the Resources Manager. Further testing (with regard to the NSs part of the slice) is planned at next project stage with the complete integration of WIM and NFVO in the MATILDA framework (at the time of writing in progress); validation results to be included in D6.13.

Table 8: Management of VNFs/NSs.

Test Objective	12	**Type**	Functional
Title	Management of NSs (mainly with regard to VNFs)		

[2] Tests related to the successful performance of functionalities referring to the lifecycle management of a slice are covered in the tests defined in Table 4, Table 8, Table 9, Table 10, and Table 11. So, tests related to this objective will focus on testing the consistency maintained between the Slice Manager information and the actual slice's state.

Validation method – Tests	Tests to be performed are related to the lifecycle management/support of network functions, including: • the mapping of the slice intent to specific network resources and VNFs • the re-use and configuration of VNFs from multiple tenants / in multiple slices • the instantiation of VNFs for a specific slice • the termination of the slice and associated VNF instances' operation.
KPIs	**Success Criteria:** Successful performance of the functionalities that are related to the VNFs lifecycle. Consistency maintained between the NFVO information and the actual instantiated VNFs' state.
Components	NFVO
Testbed	All

Detailed Tests Description: Management of NSs	
Test 1	**Description:** Verification of the mapping of the slice intent to specific, suitable 3GPP NSs, to support the 5G-ready applications' deployment. 3GPP network services are mainly related to RAN, composed of the PNFs constituting the eNBs, the EPC functional entities, and the additional VNF providing MEC functionality, for defining bearers on a per-UE basis. **Success Criteria:** Correct mapping of the slice intent to specific, suitable 3GPP network services (bearer). **Testbed:** CNIT (and at the next stage Demonstrators' testbeds)
Result/ Comments	Tests on the RAN have been performed in the CNIT testbed to assess the performance of the bypass VNF described in [MATILDA-D4.1]. Detailed description and results can be found in Annex 1: Validation of the Bypass VNF.
Test 2	**Description:** Verification of the mapping of the slice intent to specific, suitable NSs, including non-3GPP network VNFs. Non-3GPP VNFs can include security-related, traffic handling, and monitoring VNFs. **Success Criteria:** Correct mapping of the slice intent to specific, suitable VNFs. **Testbed:** CNIT (and at the next stage Demonstrators' testbeds)
Result/ Comments	Planned to be tested at the next project stage with the integration of NFVO in the MATILDA framework (at the time of writing in progress); validation results to be included in D6.13.
Test 3	**Description:** Verification of the re-use, configuration and instantiation of VNFs from multiple tenants / in multiple slices. **Success Criteria:** Correct initial configuration and instantiation of VNFs from multiple tenants / in multiple slices. **Testbed:** CNIT (and at the next stage Demonstrators' testbeds)

Result/ Comments	Planned to be tested at the next project stage with the integration of NFVO in MATILDA framework (at the time of writing in progress); validation results to be included in D6.13.
Test 3	**Description:** Verification of termination of VNFs once the relevant network slice is deleted. **Success Criteria:** Termination of VNFs and release of resources once the relevant network slice is deleted. **Testbed:** CNIT (and at the next stage Demonstrators' testbeds)
Result/ Comments	Planned to be tested at the next project stage with the integration of NFVO in the MATILDA framework (at the time of writing in progress); validation results to be included in D6.13.

Table 9: Management of Infrastructure Resources.

Test Objective	13	**Type**	Functional
Title	Management of Infrastructure Resources		
Validation method – Tests	Tests to be performed are related to management of infrastructure resources, including: • exposing of infrastructure resources' availability to the necessary entities including the CSM/VAO and NFVO, • instantiation of infrastructure resources upon request after resolution of resources' availability, • multi-tenancy.		
KPIs	**Success Criteria:** Successful performance of the functionalities that are related to the infrastructure resources' management on PoPs.		
Components	VAO, VIM		
Testbed	All		

Detailed Tests Description: Management of Infrastructure Resources	
Test 1	**Description:** Verification of exposure of infrastructure resources' availability to the necessary entities including the CSM/VAO and NFVO. **Success Criteria:** Infrastructure resources' availability information exchanged between VIM and NFVO and CSM/VAO is correct. **Testbed:** CNIT/UBI
Result/ Comments	Information exchange between OSS and VIM has been successfully tested and demonstrated during the demo in November. Information exchange between OSS and NFVO is planned to be tested at the next project stage with the integration of NFVO in MATILDA framework (at the time of writing in progress); validation results to be included in D6.13.

Test 2	**Description:** Verification of instantiation of appropriate infrastructure resources upon request after resolution of slice intent and resources availability. **Success Criteria:** Correct instantiation of appropriate infrastructure resources upon request. **Testbed:** CNIT/UBI
Result/ Comments	The manual instantiation and release of infrastructure resources has been demonstrated during the demo in November. Further tests related to the instantiation of appropriate infrastructure resources based on resources' availability are planned at the next project stage with the integration of the NFVO in the MATILDA framework (at the time of writing in progress); validation results to be included in D6.13.
Test 3	**Description:** Verification of multi-tenancy support, isolation and sharing of infrastructure resources. **Success Criteria:** Multiple tenants can share infrastructure resources. **Testbed:** CNIT/UBI
Result/ Comments	Sharing of infrastructure resources among multiple tenants has been tested and demonstrated during the demo in November.

Table 10: Management of Wide-area Network Resources.

Test Objective	14	**Type**	Functional
Title	Management of Wide-area Network Resources		
Validation method – Tests	Tests to be performed are related to the logical interconnectivity among sets of service/application components instantiated in different PoPs, including: • the network resources allocation and QoS provisioning for each link defined in the application service graph • the maintenance and modification of the network resources based on runtime policies and general status of the WAN • the release of network resources upon termination of the application instance operation.		
KPIs	**Success Criteria:** Successful performance of the functionalities that are related to the network resources provisioning lifecycle.		
Components	WIM		
Testbed	All		

Detailed Tests Description: Management of Wide-area Network Resources

Test 1	**Description:** Verification that the VIM has calculated a list of candidate sets of PoPs for deployment of the groups of service/application obtained from the reduced service graph. **Success Criteria:** The ENM GUI shows the list of candidate VIM(s) fulfilling the proximity requirements. **Testbed:** CNIT
Result/ Comments	Initial testing of the Ericsson WIM solution has been performed. Further testing is planned for the next project stage with the complete integration of all components into the MATILDA framework; validation results to be included in D6.13.
Test 2	**Description:** Verification of establishment of interconnectivity among sets of service/application components instantiated in different PoPs, by provisioning the correct network resources and QoS for each link defined in the application service graph. **Success Criteria:** Connectivity achieved as slice instantiation based on the application slice intent, and network resources management/availability resolution. **Testbed:** CNIT
Result/ Comments	Initial testing of the Ericsson WIM solution has been performed. Further testing is planned for the next project stage with the complete integration of all components into the MATILDA framework; validation results to be included in D6.13.
Test 3	**Description:** Verification of the maintenance and modification of the network resources based on runtime policies and general status of the WAN while ensuring interconnectivity among sets of service/application components instantiated in different PoPs. **Success Criteria:** Connectivity modification based on runtime policies and WAN resources' availability. **Testbed:** CNIT
Result/ Comments	Initial testing of the Ericsson WIM solution has been performed. Further testing is planned for the next project stage with the complete integration of all components into the MATILDA framework; validation results to be included in D6.13.
Test 4	**Description:** Verification of the release of network resources upon termination of the application instance operation. **Success Criteria:** Complete release of network resources upon termination of application instance. **Testbed:** CNIT
Result/ Comments	Initial testing of the Ericsson WIM solution has been performed. Further testing is planned for the next project stage with the complete integration of all components into the MATILDA framework; validation results to be included in D6.13.

Table 11: Multi-site Resource Management.

Test Objective	15	**Type**	Functional
Title	Multi-site Resource Management		
Validation method – Tests	Tests to be performed are related to managing resources at diverse facilities, like central/remote public/private/hybrid cloud facilities or at the mobile network edge, more specifically including: • information exchange between the VAO (Execution Manager), the Computing Slice Broker and the VIMs regarding their availability of resources • deployment of applications/application components at Network Service Provider's edge facilities using information related to: ◦ end-user location and locality of computing resources ◦ availability of resources of various PoPs • lifecycle management of an application component deployed at the Telecom Service Provider's facilities through (evolved) IaaS/PaaS APIs.		
KPIs	**Success Criteria:** Successful performance of the functionalities that are related to managing resources at diverse PoPs. Consistency maintained between the information at the Multi-Site Resource Manager, and the actual PoPs and Network Infrastructures.		
Components	VAO, Multi-site Resource Management functionality, VIM, WIM		
Testbed	CNIT		

Detailed Tests Description: Multi-site Resource Management	
Test 1	**Description:** Verification that the correct information is exchanged between the VAO's Execution Manager, the Computing Slice Broker and a number of VIMs (of various PoPs, like central/remote public/private/hybrid cloud facilities or at the mobile network edge), regarding their availability of resources. **Success Criteria:** Exchange of the necessary and correct information regarding the resources' request and availability. **Testbed:** CNIT, UBI and/or Demonstrators' facilities
Result/ Comments	Planned to be tested at the next project stage with the complete integration of all MATILDA subsystems framework; validation results to be included in D6.13.
Test 2	**Description:** Verification of deployment of applications/application components at Network Service Provider's edge facilities using information related to locality of computing resources and availability of resources of various PoPs. **Success Criteria:** Correct placement of compute resources to the appropriate PoP after resolution of slice intent information. **Testbed:** CNIT, UBI and/or Demonstrators' facilities

Result/ Comments	Planned to be tested at the next project stage with the complete integration of all MATILDA subsystems framework; validation results to be included in D6.13.

5 Evaluation of MATILDA Solution On-boarding Process

Currently, the transformation of the stand-alone vertical applications (that are available in the context of the MATILDA project) into 5G-ready applications and the on-boarding process of these applications' graphs has been completed. At this point, significant hands-on experience has been already acquired by the verticals as MATILDA end-users with regard to the on-boarding process. Given this experience, the MATILDA performance evaluation tests related to the on-boarding process have been refined and specified at the level of specific tests' and success criteria, in the tables of this section. Preliminary results obtained from the verticals'/end-users' perspective are also summarised in these tables.

Table 12: User Friendliness Evaluation.

Test Objective	16	**Type**	Other
Title	User Friendliness		
Relevant UCs	All		
Validation method – Tests	User friendliness of the MATILDA solution interfaces to users/stakeholders can be evaluated at MATILDA system design and development phases by the end-users represented by the relevant consortium partners.		
KPIs	**Relevant KPIs**: Evaluation feedback collected by various end users/stakeholders of the MATILDA solution.		
Components	General Solution		
Testbed	All		

Detailed Tests Description: User Friendliness	
Test 1	**Description:** Evaluation of User Friendliness of the on-boarding process of the vertical applications. **Success Criteria:** >80% of end-users find the on-boarding process user-friendly (feedback to be received over the project period through discussions between partners as well as on the basis of a questionnaire with regard to quality of graphics, responsiveness of interface, identification of steps and navigation through screens, etc.).
Result/ Comments	Interim results show that the on-boarding process of vertical applications is considered as user-friendly for the partners/personnel involved in the project. A recursive Software Engineering process is already adopted so that any comments/remarks from verticals is taken into account and addressed so that the user interface meets their requirements.

	Further assessment will be performed on the basis of end-user questionnaires towards the project end.
Test 2	**Description:** Evaluation of understandability of requested input for the description of resources/performance requirements, etc., during the on-boarding process of the vertical applications. **Success Criteria:** >80% of end-users can clearly understand the input that is needed during the on-boarding process (feedback to be received over the project period through discussions between partners, as well as on the basis of a questionnaire).
Result/ Comments	Interim results show that the parameters requested during the on-boarding process of vertical applications is easy to understand. A recursive Software Engineering process is already adopted so that any questions/clarifications asked from verticals is taken into account and addressed, so that the user interface becomes more understandable. Further assessment will be performed on the basis of end-user questionnaires towards the project end.
Test 3	**Description:** Evaluation of completeness/correctness of description of required performance/ resources. **Success Criteria:** • 100% of end-users find the description of required performance/ resources in terms of parameters, target values, etc., requested during the on-boarding process correct in terms of defining their vertical application. • >80% of end-users find the description of required performance/ resources in terms of parameters, target values, etc., requested during the on-boarding process complete in terms of defining their vertical application. (feedback to be received over the project period through discussions between partners, as well as on the basis of a questionnaire).
Result/ Comments	Interim results show that the parameters requested during the on-boarding process describe the partners' vertical applications in a complete and correct way. A recursive Software Engineering process is already adopted so that any addition requested from verticals is taken into account and added to the MATILDA VAO. Further assessment will be performed on the basis of end-user questionnaires towards the project end.

Table 13: Speed of Application Deployment Evaluation.

Test Objective	17	**Type**	Other
Title	Speed of Application Deployment		
Relevant UCs	All		

Validation method – Tests	Tests to be performed will include measurement and evaluation of the time required for an application deployment for the various UCs' applications, at various infrastructures, with various deployment parameters to be defined at run-time (service graphs and run-time policies). The factors affecting the speed of deployment in each case will be identified, analysed and evaluated.
KPIs	**Relevant KPIs**: Speed of deployment from the initial application selection from the MATILDA Application Repository to the completion of the Application initial deployment on a selected infrastructure. The speed of deployment will be assessed against the relevant 5G-PPP KPI.
Components	General Solution
Testbed	All

Detailed Tests Description: Speed of Application Deployment	
Test 1	**Description:** Evaluation of speed of on-boarding process per component and for the whole graph, etc. **Success Criteria:** The speed of the on-boarding process highly depends on the vertical application, with respect to the number of components and their complexity in the resources' definition and handling. The vertical application on-boarding process through the MATILDA interface should take less time than that needed to deploy manually the application graph by using the common open source cloud Infrastructure APIs (i.e. directly through OpenStack APIs).
Result/ Comments	Interim results show that the on-boarding process of vertical applications is considered significantly faster than using the common open source cloud Infrastructure API of OpenStack, since for the non-cloud experts these would require to build expertise in defining interfaces between components, defining the resource requirements, while not having a common view (GUI) of the deployed application graph. As quantified for the PPDR and 5GPACE application ([MATILDA-D6.2], [MATILDA–D6.3]), the average components' on-boarding time is about 15 min. Further assessment will be performed on the basis of end-user questionnaires towards the project end.
Test 1	**Description:** Evaluation of speed of application component/application graph, etc., deployment. **Success Criteria:** The speed of the application component/application graph, etc., deployment should be only restricted by the time needed to deploy application component images on cloud infrastructure and by the time needed from the WIM domain to provision the network resources.
Result/ Comments	In principle, the speed of application component/application graph, etc., deployment depends on multiple factors including the time to resolve the slice intent, PoPs resources availability, the specific cloud infrastructure Virtual Machines' spawning time, and so on. Therefore, the complete assessment of this aspect will be performed during the next period with the integration of all MATILDA components on a single infrastructure.

Preliminarily, as quantified for the PPDR and 5GPACE application ([MATILDA-D6.2], [MATILDA–D6.3]), the average deployment time for a single component is ~3 min (capped to the cloud infrastructure Virtual Machines' spawning time).

The refinement of the objectives, the specification of tests and the collection of results regarding the rest of the performance aspects defined in MATILDA D6.1 will be provided with the next deliverable version after the finalisation of the validation and evaluation activities.

6 Adoption Guidelines

In this section, the main adoption guidelines of the first release of the MATILDA framework are presented, considering the feedback received from the first round of evaluation. Such guidelines are going to be further specified and documented towards the final release of the MATILDA framework.

Having finalised the onboarding of all the MATILDA demonstrators' applications in the MATILDA installation in the testbed maintained by UBITECH, the overall application components and graphs registration and composition processes were considered simple by the users. At this phase, an important step is the proper specification of the application graph, including the set of application components and their required and exposed interfaces.

The application developer interested to upload components in the available repositories can easily provide the developed software in a containerized format and declare in the UI the set of deployment constraints, environmental variables and interconnection interfaces of the components. As next step, by usage of the Application Graph Composer, applications can be made available for deployment over a telco infrastructure. It should be noted that, over the whole onboarding period, feedback was collected and addressed, while further minor improvements suggested by users in the UI and the overall registration process will be considered in the upcoming period.

Importance has to be given on the correct specification of the deployment requirements -at component and link level- since they drive the relevant resources' reservation in the telco infrastructure and are highly related to the provision of resource guarantees to achieve the required application performance. In addition to the deployment requirements, specification of runtime policies has to be well thought. For instance, the exact rules and approach for scaling in and out per horizontally scalable component have to be specified considering the resource bottlenecks affecting a component's performance, as well as the observed elasticity efficiency (e.g. considering the time required for the provision of a new instance). In this process, the consultation obtained from the results of the profiling mechanisms can prove very helpful.

Furthermore, in all MATILDA demonstrator cases, the validation of components' proper operation, performed through the examination of visualisations of sets of time series data collected by the Prometheus monitoring engine, was critical for the identification of potential issues and required fixes in the way that the applications and network infrastructure were provisioned.

In the upcoming period, activities will focus on incorporating and testing more advanced network slice lifecycle management functionalities, aiming at the validation of the provision of the envisaged functionalities and network services in accordance to each vertical industry needs. Furthermore, through the instantiation of the MATILDA framework in the MATILDA testbeds, useful insights with regard to the way that the framework can be replicated and interconnected with heterogeneous infrastructure are going to be made available.

7 Conclusions

The MATILDA evaluation framework has been specified as flows of validation and evaluation processes spanning from **MATILDA Solution Components and Functionality Validation** to **General (as a whole) Solution Validation and Evaluation** and further to **Performance Evaluation on the basis of specific KPIs** of MATILDA specific functions and of the whole solution. Validation testing and evaluation flows are addressed at various completion degrees at various project stages; namely: at MATILDA component development phases, at MATILDA components' integration phases, at vertical application on-boarding phases, at MATILDA solution operational phases, at vertical application full deployment phases, and finally at vertical application operational phases. As planned, the list of test objectives and procedures are being refined throughout the project lifetime to better suit implementation specificities that emerge in these project stages, along with testbed specific features, environment setup/tools, etc.

Currently, having finalised the development of a significant number of MATILDA components and having progressed with the partial integration of some of the MATILDA components, the Solution Components and Functionality Validation-related objectives have been refined and specified at the level of specific tests' and success criteria.

At this project stage, the majority of Solution Components and Functionality Validation tests have been performed - at component development stage, as well as using the MATILDA Demonstrators' Applications during the MATILDA applications' wrapping and on-boarding phase.

To this end, testing and validation activities have been performed for the following MATILDA components/functionalities:

1. The Application Development and Wrapping Toolkit,
2. The MATILDA Marketplace; in particular, the Application and VNF repositories,
3. The VAO,
4. The NFVO, with special focus on 3GPP network services,
5. The VIM, and
6. The WIM.

At the time of writing, partial integration of these components has been performed, allowing further for:

- testing of the application orchestration functionality ranging from definition to deployment on various PoPs (integration of components 1, 2 & 3)
- testing of the application deployment functionality with 3GPP network service provisioning (integration of components 3 & 4 (3GPP-part))
- testing of multisite resource management functionality (integration of components 3 & 5)

Following an iterative development process consisting of cycles of component testing, as well as of cycles of complete functionalities testing performed by MATILDA vertical application teams –e.g. throughout the course of the MATILDA Demonstrators' vertical applications wrapping/transformation into 5G-ready, and their on-boarding process- providing feedback to the solution development teams, it has been ensured that stakeholders' clarifications/suggestions/changes are addressed. At the time of writing, considering the validation and evaluation activities the results retrieved are the following:

- The Application development and wrapping functionalities (to make an application 5G-ready) have been completely developed and tested (at UBI/CNIT testbeds, using all MATILDA Demonstrators' Applications), in terms of successfully enabling:
    - creation/edition/modification of application service graphs adhering to the MATILDA metamodels
    - creation/edition of runtime policies (in particular resource utilization and security – related ones) at application component level.
- The lifecycle management (insertion, modification/update, selection, deletion) of applications/application components/VNFs and their metadata in the associated repositories has been tested and successfully validated.
- The Vertical Applications' orchestration and lifecycle management has been tested (at UBI/CNIT testbeds), in terms of successfully enabling:
    - extraction of the slice intent from the service graphs definitions on the basis of the MATILDA metamodels,
    - Real-Time deployment of an application in various PoPs,
    - enforcement of specific run-time policies (resource utilisation, and security-related),
    - termination of application instance operation upon request.
- A number of the Vertical Applications deployment monitoring functionalities have been tested (at UBI/CNIT testbeds and the PPDR Demonstrator application), and the following have been successfully validated:
    - Activation/configuration of active and passive probes monitoring compute/ network/ application resources utilisation/behaviour.
    - Fusion of monitoring data coming from multiple parallel data loads from multiple sources.
    - Extraction of Real-Time Analytics (tests initially using the PPDR application data and artificially generated data) and advanced insights.
- The lifecycle management of NSs has been completely developed and tested (at CNIT testbeds), in terms of successfully enabling:
    - Lifecycle management of a 3GPP network services slice
    - MEC capabilities using a "Bypass VNF" enabling traffic offloading at edge PoPs.
    - WAN – specific functionality related to the management of network resources on a per slice basis.

Considering the performance evaluation of the MATILDA solution on-boarding process, from the preliminary results obtained from the verticals'/end-users' perspective, it was shown that the on-boarding process of vertical applications is considered to have been user-friendly, understandable in terms of steps to be followed, and the required application descriptive information completely/correctly reflects the applications' performance / resources requirements.

In the next project period, testing will focus (1) on the MATILDA components integrated solution testing, not only on CNIT/UBI but also (2) on a number of MATILDA Demonstrators' testbeds, related to (3) Vertical Applications Deployment and Network Slice Lifecycle Management over a completely integrated MATILDA infrastructure.

References

[5GPP-2015]	5GPPP, "5G Vision: The next generation of communication networks and services". URL: https://5g-ppp.eu/wp-content/uploads/2015/02/5G-Vision-Brochure-v1.pdf
[MATILDA-D1.1]	MATILDA project deliverable, D1.1 - MATILDA Framework and Reference Architecture, available online at: http://www.MATILDA-5g.eu/index.php/outcomes
[MATILDA-D1.5]	MATILDA project deliverable, D1.5 - Deployment and Runtime Policy Metamodel, available online at: http://www.MATILDA-5g.eu/index.php/outcomes
[MATILDA-D4.1]	MATILDA project deliverable, D4.1 – Multi-site Resources Management and Execution Mechanisms, available online at: http://www.MATILDA-5g.eu/index.php/outcomes
[MATILDA-D6.1]	MATILDA project deliverable, D6.1 - Evaluation Framework and Demonstrators' Planning, available online at: http://www.MATILDA-5g.eu/index.php/outcomes
[MATILDA-D6.2]	MATILDA project deliverable, D6.2 - Emergency Infrastructure with SLA Enforcement Implementation Report, available online at: http://www.MATILDA-5g.eu/index.php/outcomes
[MATILDA-D6.3]	MATILDA project deliverable, D6.3 - High Resolution Media on Demand Implementation Report, available online at: http://www.MATILDA-5g.eu/index.php/outcomes
[MATILDA-D6.4]	MATILDA project deliverable, D6.4 - Smart City Intelligent Lighting System Implementation Report, available online at: http://www.MATILDA-5g.eu/index.php/outcomes
[MATILDA-D6.5]	MATILDA project deliverable, D6.5 - Industry 4.0 Smart Factory Implementation Report, available online at: http://www.MATILDA-5g.eu/index.php/outcomes
[MATILDA-D6.6]	MATILDA project deliverable, D6.6 - Automobile Electrical Systems Remote Control Implementation Report, available online at: http://www.MATILDA-5g.eu/index.php/outcomes

Annex 1: Validation of the Bypass VNF

The following tests have been performed to assess the performance achieved by realizing the MEC attach point by means of the bypass VNF described in the D4.1 report. Since the current 3GPP 4G architectural specification does not allow exposing its reference points externally, additional functionalities are required to expose S1-AP and S1-U protocol interfaces externally in order to define and manage bearers on a per-UE basis and handle their traffic accordingly.

In this respect, the Bypass VNF provides the functionalities enabling traffic to be intercepted and directed to the application of interest, located in a specific VIM, before being processed by the EPC. This operation clearly allows saving processing times; however, care must be taken to avoid that the additional functionalities' overhead overcomes the processing times due to the EPC.

In order to assess the potential overhead due to the additional functionalities designed for the realization of the MEC attach points in the MATILDA framework, tests have been performed to characterize the delay between the UE and the destination of its traffic ascribable to the presence of the Bypass VNF. As shown in Figure 20-Figure 22, the testbed consists of a router tester and two servers. The transmitting port of the router tester behaves as a UE, and the receiving one represents the endpoint of the traffic. The first server provides the GTP encapsulation required to emulate the behaviour of the S1-U protocol, which is not available in the router tester, and is realized by using a Linux VM. This configuration is the same for all the tests.

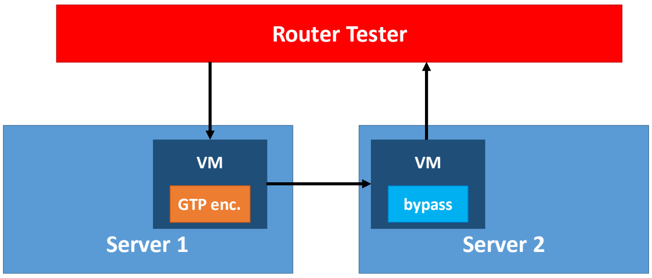


Figure 20: Test1 Configuration.

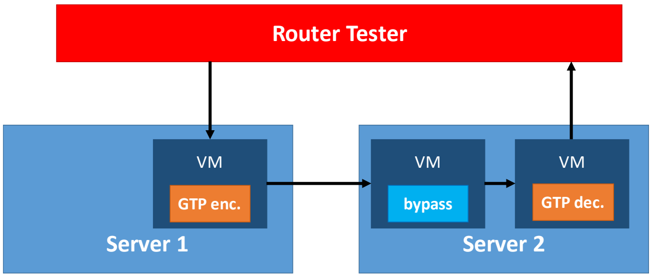


Figure 21: Test2 Configuration.

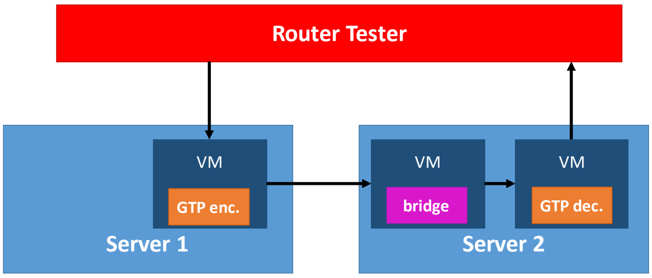


Figure 22: Test3 Configuration.

In Test1 (Figure 20), the second server hosts a VM that contains the Bypass VNF, which inspects the incoming traffic and, when packets belonging to the service of interest are identified, removes their GTP-U and adds a VLAN tag before sending the packet to the end-point (e.g., the receiving port of the router tester). In Test2 (Figure 21), instead, packets entering the Bypass VNF are not identified as belonging to the service of interest and, as such, they are passed to another VM to be decapsulated and forwarded to the end-point. Finally, in Test3 (Figure 22) the Bypass VNF is replaced with a simple Linux bridge and traffic processing proceeds as in Test3.

For all three test cases, traffic has been transmitted at increasing rates from 10000 to 20000 pkt/s. Tested packet sizes have been set to 74, 740 and 1440 Bytes and both Constant Bit Rate (CBR) and bursty traffic have been employed.

The average latency obtained for the above-mentioned packet sizes, at varying offered load, is shown in Figure 25. It can be noticed that, for all results, the performance obtained with the Bypass VNF in Test1 are slightly better with respect to Test2 and Test3. Although the latency reduction may not seem remarkable, it should be noticed that the presence of the VNF entails the presence of a number of additional processing operations (described in detail in the D4.1 report, Section 8.1).

The low overhead due to the bypass operations can be further analysed by considering the profiling results obtained on the VNF by using the perf tool and reported in Figure 26. It can be noticed that most of the overhead is ascribable to the *virtio_recv_mergeable_pkts* function, which is related to VM I/O processes and does not depend on bypass.

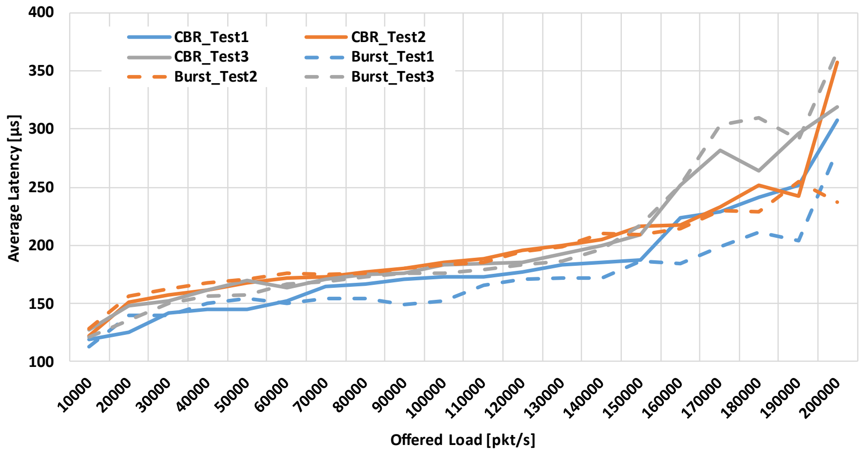


Figure 23: Results obtained for packets of 74 Bytes.

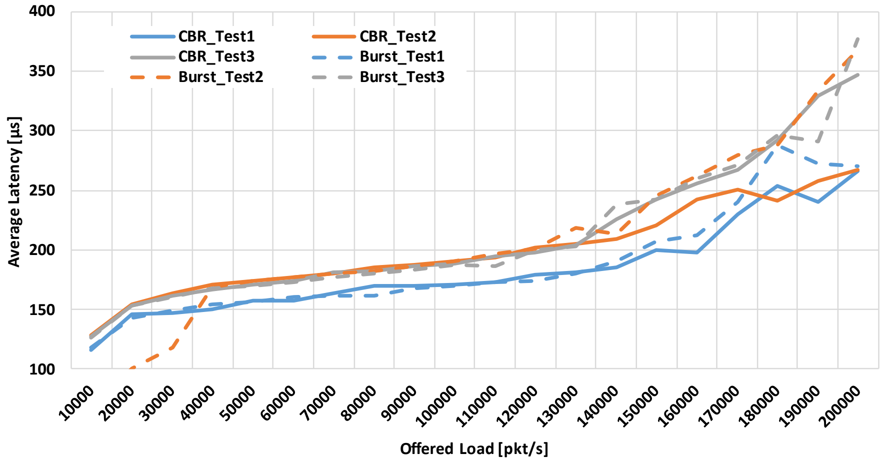


Figure 24: Results obtained for packets of 740 Bytes.

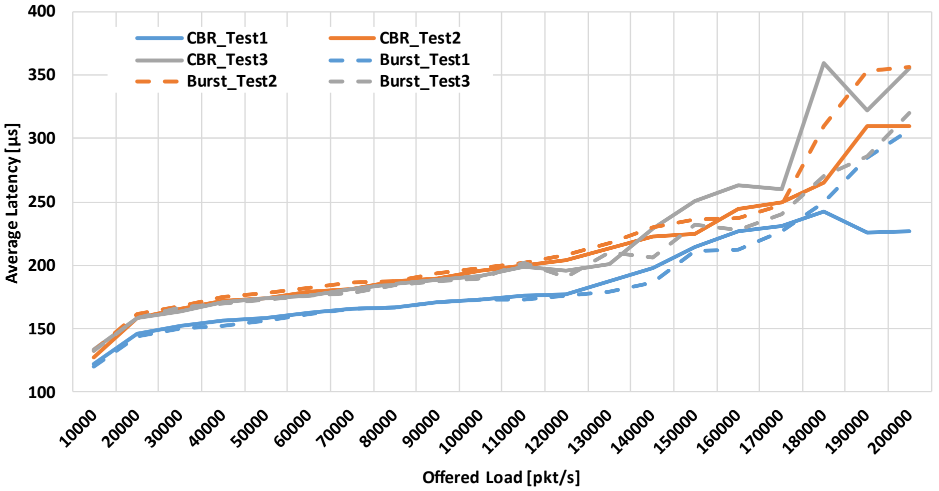
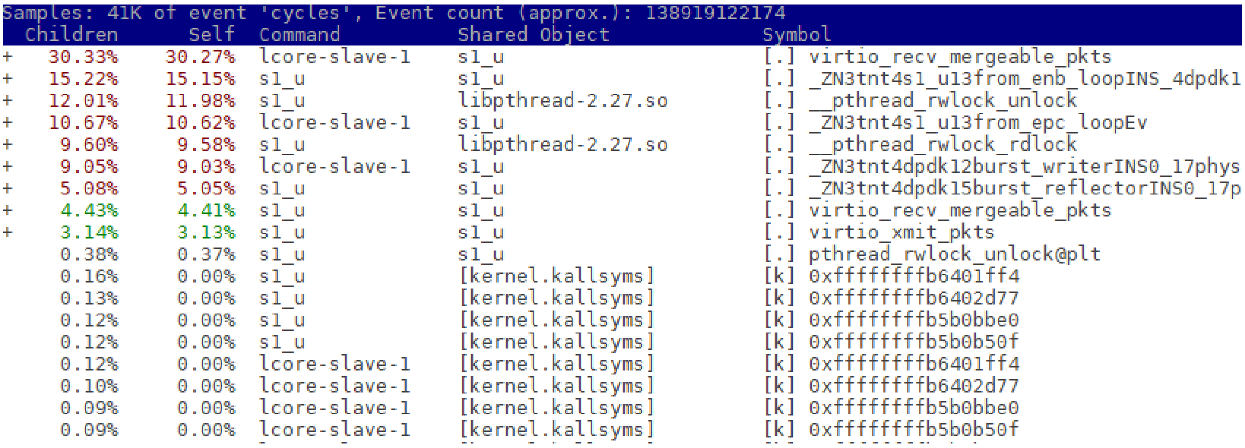


Figure 25: Results obtained for packets of 1440 Bytes.



```
Samples: 41K of event 'cycles', Event count (approx.): 138919122174
  Children      Self  Command          Shared Object                Symbol
+   30.33%    30.27%  lcore-slave-1    s1_u                         [.] virtio_recv_mergeable_pkts
+   15.22%    15.15%  s1_u             s1_u                         [.] _ZN3tnt4s1_u13from_enb_loopINS_4dpdk1
+   12.01%    11.98%  s1_u             libpthread-2.27.so           [.] __pthread_rwlock_unlock
+   10.67%    10.62%  lcore-slave-1    s1_u                         [.] _ZN3tnt4s1_u13from_epc_loopEv
+    9.60%     9.58%  s1_u             libpthread-2.27.so           [.] __pthread_rwlock_rdlock
+    9.05%     9.03%  lcore-slave-1    s1_u                         [.] _ZN3tnt4dpdk12burst_writerINS0_17phys
+    5.08%     5.05%  s1_u             s1_u                         [.] _ZN3tnt4dpdk15burst_reflectorINS0_17p
+    4.43%     4.41%  s1_u             s1_u                         [.] virtio_recv_mergeable_pkts
+    3.14%     3.13%  s1_u             s1_u                         [.] virtio_xmit_pkts
     0.38%     0.37%  s1_u             s1_u                         [.] pthread_rwlock_unlock@plt
     0.16%     0.00%  s1_u             [kernel.kallsyms]            [k] 0xffffffffb6401ff4
     0.13%     0.00%  s1_u             [kernel.kallsyms]            [k] 0xffffffffb6402d77
     0.12%     0.00%  s1_u             [kernel.kallsyms]            [k] 0xffffffffb5b0bbe0
     0.12%     0.00%  s1_u             [kernel.kallsyms]            [k] 0xffffffffb5b0b50f
     0.12%     0.00%  lcore-slave-1    [kernel.kallsyms]            [k] 0xffffffffb6401ff4
     0.10%     0.00%  lcore-slave-1    [kernel.kallsyms]            [k] 0xffffffffb6402d77
     0.09%     0.00%  lcore-slave-1    [kernel.kallsyms]            [k] 0xffffffffb5b0bbe0
     0.09%     0.00%  lcore-slave-1    [kernel.kallsyms]            [k] 0xffffffffb5b0b50f
```

Figure 26: Results of profiling performed on the Bypass VNF.

Annex 2: Refined Demonstrators' KPIs

In line with the MATILDA evaluation framework described in this document, in parallel with the iterative testing, validation and evaluation phases, and along the evolution of originally stand-alone applications to cloud-native ones, the KPIs related to the applications and demonstrators have been further refined from the initially specified ones. These refinements, and in specific cases changes, in the KPIs definition and/or target values compared to the initially reported ones in [MATILDA-D1.6], are attributed to the following:

- At initial stages, KPIs were defined considering the requirements of general applications falling in the category of each demonstrator; as the project progressed, the applications to be finally used were further elaborated, thus KPIs (especially network ones) were further nailed down to the performance requirements of the specific applications to be tested.
- The initial KPIs referred to traditional applications instantiation (e.g. in a single server, single client mode), while in the context of the project the applications were also modified to include a number of cloud-native components; thus, in many cases, it was required to move from single-link KPIs to application graph KPIs (also reflected in the metamodels).
- Throughout the course of the project, even the expertise of partners has advanced to obtain a better understanding of the requirements and KPIs that they shall expect to be achieved by the solution. To this end, even new KPIs, especially related to the operation of the MATILDA solution (below defined as operational KPIs) and the rapid deployment and instantiation of an application, have been considered and added in the initial list - mainly where it has been considered a critical parameter for the demonstrator. Indicatively, the deployment time (operational KPI) has been added in the case of the 5G PPDR application that is usually deployed on demand, contrary to the smart lightning application that does not require rapid on demand deployment.
- On the other hand, in certain cases, specific KPIs were defined, which however cannot be demonstrated by the project test facilities and/or are out of scope of the core of the MATILDA solution, but rather need to be evaluated in larger and completely operational environments. These KPIs were removed or their values have been changed to be more applicable to the specific, available test facilities, features, environment setup/tools, etc.

At this point it shall be mentioned that the MATILDA project has closely monitored and contributed significantly to the work performed in the context of 5G-PPP TB regarding the 5G KPIs. In particular, this work included:

1. Early identification, definition, and iterative refinement of the KPIs to be measured and evaluated in 5G infrastructures, along with information regarding the measurement points and methods. Major contribution to this work has been provided by MATILDA and initially published in "K.X. Du, B. Sayadi, G. Carrozzo, F. Lazarakis, A. Kourtis, M.S. Siddiqui, J. Sterle, O. Carrasco, and R. Bruschi, "Definition and Evaluation of Latency in 5G:A Framework Approach", URL: http://www.jkjmanagement.com/5gwf19-4/papers/p135-du.pdf, 2019 IEEE 5G World Forum", and secondly in "5G PPP phase II KPIs – Annex to Programme Management Report", (5G-PPP document not yet accessible on-line).
2. Alignment of this work with MATILDA project testing and evaluation activities, especially mapping it to specific demonstrators, work-packages, and demonstration activities; and communication of this information to the TB KPIs WG. This

information has been reported in PPP Programme Management Report (finalised June 2019) (i.e. 5G-PPP Phase 2 KPIs – Annex to Programme Management Report).

In the opposite direction, the work performed and information obtained in the context of TB KPIs WG, has been fused in the MATILDA demonstrators' and solution validation and evaluation activities. This work (including 5G – infrastructure, and solution related operational KPIs) will be fully provided in the final version of this Deliverable (as MATILDA Deliverable D6.13), while the demonstrators-related KPIs refinement is provided in the following tables.

Table 14. Refined KPIs for Emergency Infrastructure with SLA Enforcement (5G PPDR) Demonstrator.

KPI	Description	Measured (Where/ How)	Acceptance Criteria/ Threshold
Network KPIs			
Availability	Network availability	End-to-end continuous measurements of network connectivity, e.g. qMON Client/Server collecting connectivity KPIs, statistics can be used to calculate availability	>99,999 % (at fully operational/ commercial environment)
Reliability	Network reliability	End-to-end continuous measurements of network connectivity, e.g. qMON Client/Server collecting connectivity KPIs, statistics can be used to calculate reliability	>99,999 % (at fully operational/ commercial environment)
Network Slicing Capability	Network Slice Management	Orchestration framework support slice management when deploying vertical application	Must be available
End-to-end Latency for interactive applications	Connected devices should be able to communicate without significant delay/ latency. Example application: • Real-time queries in transactional databases	End-to-end measurements of network latency, e.g. qMON Client (on end device, UE) and qMON Server (in DC) collecting RTT KPIs	< 20 ms
End-to-end Latency for mission critical applications[3]	Connected devices should be able to communicate without significant delay/ latency. Example application: • Remote control of drones and robots	End-to-end measurements of network latency, e.g. qMON Client (on end device, UE/drone) and qMON Server (in DC) collecting RTT KPIs	< 1 ms

[3] Limited to the availability of 5G NR, 5G NR UE and URLLC capabilities.

Bandwidth	High bandwidth required for: • Data intensive applications for PPDR use • Ultra HD video streaming from disaster site (land and aerial based)	End-to-end measurements of network bandwidth, e.g. qMON Client (on end device, UE) and qMON Server (in DC) collecting bandwidth KPIs	~20 Mbps/user
Jitter	Time-critical communications should be stable and reliable. Timing variation must be minimal	End-to-end measurements of network jitter, e.g. qMON Client (on end device, UE) and qMON Server collecting jitter KPIs	< 1ms
Packet Loss	Reliability and high availability of the services in extreme conditions is essential for emergency systems. Therefore, packet loss should be made as small as possible	End-to-end measurements of network packet loss, e.g. qMON Client (on end device, UE) and qMON Server collecting packet loss KPIs	< 0.01%
Operational KPIs			
iMON Dashboard components on-boarding time	Time required for the App developer to on-board the iMON Dashboard components	Measure time required to onboard iMON Dashboard components, e.g., time collected from VAO orchestrator log	~15 minutes
iMON Dashboard component deployment time	Time needed to deploy an individual component of the application graph	Measure time required to deploy individual component, e.g., time collected from VAO orchestrator log	~3 minutes *(This was targeted at 2 minutes in D1.6 and was changed to 3 minutes in D6.2 due to the project and architecture constraints.)*
iMON Dashboard application graph deployment time	Time to deploy the iMON Dashboard application graph	Measure time required to whole application graph, e.g., time collected from VAO orchestrator log	~5 minutes
Resource Usage Monitoring	Compute/storage/networking resource usage monitoring	UI for monitoring, e.g., Prometheus	Must be available
iMON Dashboard component scalability	PHP BL components of the iMON dashboard must support horizontal scaling	Scaling UI in orchestrator, information collected from VAO orchestrator log	Must be available
Scaling time	Time required to trigger the scaling after a certain threshold was reached	Measure time required to trigger the scaling, e.g. time collected from VAO orchestrator log	~ 30s

Availability	Service availability (e.g. iMON Dashboard)	End-to-end continuous measurements of service availability, e.g. qMON Client/Server collecting web KPIs, statistics can be used to calculate availability	>99,99%
Reliability	Service reliability (e.g. iMON Dashboard)	End-to-end continuous measurements of service reliability, e.g. qMON Client/Server collecting web KPIs, statistics can be used to calculate availability	>99,99%

For the refinement of the KPIs for the 5GPACE demonstrator, KPIs that typically characterize a crowded event have been considered. A list of meaningful operational KPIs, which focus on the capabilities that the MATILDA Framework should make available to handle the overall lifecycle of the 5GPACE App, have been also identified.

Table 15. Refined KPIs for High-Resolution Media on Demand Vertical, with Smart Retail Venues' integration (5GPACE) Demonstrator.

KPI	Description	Measured (Where/ How)	Acceptance Criteria/ Threshold
Network KPIs			
Device Density	The application expects a big number of connected devices	Measured for the entire graph, in ININ/Athens review deployment and be reported in the final review	~32 per Small Cell ~50 per WIFI Hot Spot
Mobility	End-User mobility	Measured for the entire graph, in the testbed deployed for the final review	Static users/low (0-3m/s)
Availability	Network availability	Measured for the entire graph, in the testbed deployed for the final review	>99%
Reliability	Network reliability	Measured for the entire graph, in the testbed deployed for the final review	>99%
User Data Rate	As a video application, high data rates per user are required.	Measured for the entire graph, in the testbed deployed for the final review	~10 Mbps/user, depending on quality
End-to-end Latency	For real-time video sharing, small delays are required	Measured for the entire graph, in the testbed deployed for the final review	Maximum 1 s
Access Interoperability	Interoperability with various access technologies (4G/5G/WAN)	Measured WAN and 4G/5G available for the entire graph, in review deployment, 4G/5G to be measured during final review deployment	Must be available
Edge Computing	Edge computing capabilities for network offloading	Measured for the entire graph, in the testbed deployed for the final review	Must be available

Storage at the Edge	Storage capabilities to save multimedia contents at the network edge	Measured for the entire graph, in the testbed deployed for the final review	Must be available
Computing acceleration at the edge	High resolution video processing requires HW acceleration	Measured for the entire graph, in the testbed deployed for the final review	Must be available
Network Slicing Capability	Network Slice Management	Measured for the entire graph, in the testbed deployed for the final review	Must be available
Operational KPIs			
5GPACE App deployment time	Time to on-board and deploy for the first time the 5GPACE App	Measured for the entire graph in the testbed deployed for the final review, tested already during continuous deployment between ITL (Milan) – INC (Athens) the development testbed (Athens/Genoa)	~90 minutes
5GPACE App on-boarding time	Time required of the App developer to on-board the 5GPACE App	Measured for the entire graph in the testbed deployed for the final review, tested already during continuous deployment between ITL (Milan) – INC (Athens) the development testbed (Athens/Genoa)	~15 minutes
Resource Usage Monitoring	Compute/storage/networking resource usage monitoring	Measured for the entire graph in the testbed deployed for the final review, tested already during continuous deployment between ITL (Milan) – INC (Athens) the development testbed (Athens/Genoa)	Must be available
5GPACE App component scalability	Specific components of the 5GPACE App must be able to scale horizontally	Measured for the entire graph in the testbed deployed for the final review, tested already during continuous deployment between ITL (Milan) – INC (Athens) the development testbed (Athens/Genoa)	Must be available
Scaling time	Time required to start/stop a component once a pre-defined parameter crosses the corresponding threshold	Measured for the entire graph in the testbed deployed for the final review, tested already during continuous deployment between ITL (Milan) – INC (Athens) the development testbed (Athens/Genoa)	~ 20s
Availability	Service availability	Measured for the entire graph in the testbed deployed for the final review, tested already during continuous deployment between ITL (Milan) – INC (Athens) the development testbed (Athens/Genoa)	High >99%

Reliability	Service reliability	Measured for the entire graph in the testbed deployed for the final review, tested already during continuous deployment between ITL (Milan) – INC (Athens) the development testbed (Athens/Genoa)	High 99%
5GPACE App repository	Repository for the on-boarded App	Measured for the entire graph in the testbed deployed for the final review, tested already during continuous deployment between ITL (Milan) – INC (Athens) the development testbed (Athens/Genoa)	Must be available
Locality Awareness	The 5GPACE App requires locality awareness	Measured for the entire graph in the testbed deployed for the final review, tested already during continuous deployment between ITL (Milan) – INC (Athens) the development testbed (Athens/Genoa)	Must be available
HW video acceleration management	Management of HW acceleration resources in the infrastructure	Capability available in MATILDA testbeds (Athens/Genoa/Bristol), PCI passthrough usable	Must be available
Multi-site management	The 5GPACE App can be composed of components instantiated in different sites	Measured for the entire graph in the testbed deployed for the final review, tested already during continuous deployment between ITL (Milan) – INC (Athens) the development testbed (Athens/Genoa)	Must be available

To correctly dimension the needed network, compute and storage resources to achieve the given KPIs, ad hoc modelling tools have been developed and are available for i-EVS. Such tools can express the amount of IT resources as a function of the total number of attendees, and few QoE related parameters.

Table 16. Refined KPIs for Smart City Intelligent Lighting System Demonstrator.

KPI	Description	Measured (Where/How)	Acceptance Criteria/ Threshold
Network KPIs			
Device bandwidth capacity	Evaluates the transfer capacity volume of information collected from sensors to IoT platform.	OAI-RAN resource allocation (component C7)	~0.1 Mbps

Total slice bandwidth	Evaluates the transfer capacity volume of aggregated information from sensors to IoT platform. Calculated as (device number) x (bandwidth/device) (helpful for VNFs system parametrization)	OAI-RAN resource allocation (component C7)	~ 100Mbps
End-to-end Latency	Measures packet round trip time from IoT platform to device sensor.	Ping measurements initiated from one agent of the IoT platform (component C2); the graph is displayed using Grafana.	< 300 msec
Jitter	Evaluates packet delay variation in latency between IoT platform and device sensor.	Ping measurements initiated from one agent of the IoT platform (component C2); the graph is displayed using Grafana.	~100 msec
Availability	Calculated as network up time/ total time, reflects in percentage the availability/ stability performance of Smart City demo platform	Infrastructure availability; measures how long the allocated compute and network resources are up; data retrieved with Prometeus and graphs displayed using Grafana; for relevance, the measurements are made for at least one month.	> 99.99%
Packet Loss	Shows the percentage of packets lost during transfer between sensors and IoT platform. The Smart Lighting service is not critical, therefore retransmission is being allowed, without affecting end-to-end application functionality.	Ping measurements initiated from one agent of the IoT platform (component C2); the graph is displayed using Grafana.	< 0.1%
Operational KPIs			
Device status	Evaluates the number of smart light sensors deployed on testbed platform.	Viewed on application's dashboard	56 Smart Light sensors
Service Availability	Calculated as service up time/ total time, reflects in percentage the availability/ stability performance of Smart City service	Measured with 2 scripts. One script measures the availability of service offered to the service consumer - emulates a user login on dashboard. The second script measures the availability of service provided by application: emulates command send or received to/ from a sensor. The graphs are displayed using Grafana; for relevance, the measurements are made for at least one month.	>99.99%

Table 17. Refined KPIs for Industry 4.0 Demonstrator.

KPI	Description	Measured (Where/How)	Acceptance Criteria/ Threshold
Network KPIs			
Availability	Network availability	End-to-end continuous measurements of network connectivity, statistics can be used to calculate availability	>99,9 % (at fully operational environment)
Reliability	Network availability	End-to-end continuous measurements of network connectivity	>99,9 % (at fully operational environment)
Network Slicing Capability	Network Slice Management	Orchestration framework support slice management when deploying vertical application	Must be available
Low Latency (Production scenario)	Low Latency is required especially between the onsite systems (e.g., production scenario) to ensure the required performance of functional testing	Measured by the components of application graph	Inside test environment (BIBA) Latency <20 ms
Low Latency (Logistics scenario)	Low latency is requires to ensure the required performance of functional testing	Measured by the components of application graph	Inside test environment (BIBA) Latency < 1000 ms
Interoperability with Various Access Networks	The services for testing shall be supported seamlessly over various Access Networks.	WLAN LTE (Ethernet)	Operation and seamless handover of the communication session across the networks
Mobility	Units under test (BIBA truck) should maintain established communication session in mobility	Measured in platform on monitoring in session	Up to 50 km/h
Packet Loss	Shows the percentage of packets lost during transfer between sensors and IoT platform	Ping measurements initiated from one agent of the IoT platform	< 0.1%
Operational KPIs			
Resource Usage Monitoring	Monitoring of the current resource usage is needed to allow preparation of dynamic scaling	Measured in platform on application instantiation as well as monitoring in session	Must be available

Table 18. Refined KPIs for Automobile Electrical Systems Remote Control Demonstrator.

KPI	Description	Measured (Where/ How)	Acceptance Criteria/ Threshold
Network KPIs			
Flexible Bandwidth Allocation at various interfaces	Flexible bandwidth allocation is needed between geographically distributed systems / sub-systems under test to ensure the integrity and required performance of distributed functional and integration testing. Bandwidth allocation based on the demand by the end clients by the means of slice re-negotiations by application graph components.	C2, C3, C4, C5 Measured by the components of application graph	10 Mbit/s (Mbps) between interfaces
Low Delay/Latency	Low Delay is required between geographically distributed systems / sub-systems under test to ensure the integrity and required performance of functional and integration testing.	C2, C3, C4, C5 Measured by the components of application graph	Inside Germany - Approximately 50 ms Latency Inside Europe - Approximately 100 ms Latency Worldwide - Approximately 200 ms Latency
Interoperability with Various Access Networks (WLAN, LTE) **Only LTE and possibly WLAN is used by the devices in use case**	The infrastructure/services for deploying FastWAN Test Systems shall be supported seamlessly over various Access Networks.	WLAN LTE	Operation and seamless handover of the communication session across the networks
Density of Connections (Not relevant)	Number of connections that can be simultaneously made by edge clients without drop in performance	C2, C5 Metric measured on the MATILDA infrastructure and on edge clients	10s to 100s of test equipment, edge databases and HMI devices
Jitter	Variation in the latency of packets on communication channels. This latency shall be minimal especially when doing real time monitoring	C2, C3, C5 Measured by the edge clients, application components as well as on the MATILDA infrastructure	Less than 1ms
Mobility	Units under test and remote test monitoring units should maintain established communication session in mobility	C2, C5 (cannot be tested in the Testbed infrastructure)	Up to 130 Km/h

Memory allocation for components of application graph	Mechanisms for demand based memory allocation as well as scaling in established sessions should be supported	UI, CORE and DB Components Measured in platform on application instantiation as well as monitoring in session	Up to 8GB
Operational KPIs			
High Availability	The test systems interconnection infrastructure/services shall be always available.	Whole graph Measured on MATILDA infrastructure (TBC)	99.99% of operational time
Resource Usage Monitoring	To allow preparation of dynamic scaling the monitoring of the current resource usage is needed	Whole graph Measured on MATILDA infrastructure (TBC)	Must be available
Component scalability	Dynamic scaling is needed to fulfil actual user communication requests	Whole graph Measured on MATILDA infrastructure (TBC)	Must be available
Time to scale	Time required for launching additional instances of application graph components on demand	Core, UI	~ 1 minute
Deployment time	Time needed for first installation and onboarding	All components	~90 minutes
Single component deployment time	Time needed for deployment of individual components of application graph	All components	~30 minutes
Network Service Deployment time	Time need for end to end service deployment	All components	~90 minutes
Onboarding time	Time needed for update and onboarding during development	Single component	~15 minutes
Onboarding time	Time needed for update and onboarding during development	Whole graph	~30 min
Locality Awareness	Locality awareness needed for optimized scalability and routing of data communications over distributed application graph component instances communication.	Whole graph	Expected to be provided by service host Must be available

Multi-site management	Functionality at its core is implemented as a distributed application. The edge application components as well as core components require network monitoring services and management provided by infrastructure to dynamically adapt to demands	Whole graph	Must be available
Recovery on session loss	Test context and session are managed by the components of the application. In case of communication losses, the session should be re-established, if demanded by the client, based on the new slice information. Data shall be cached local to the instance until certain limit and time for test session recovery	Whole graph	Capability to associate between application instance session and slice must be available
Comparison between standalone FastWAN vs FastWAN as microservice on 5G.	Bandwidth allocation, latency and jitter are compared between the non 5G deployment and 5G deployment of FastWAN	Whole graph	Above mentioned thresholds for bandwidth allocation, latency and jitter
Security – Slice Isolation	Multiple instances of application graph should have slice isolation mechanisms to provide data security	Whole graph	Must be available -access to two slices which contain isolated streams
Security – Dynamic Snort Rule Management	Protection against access to services by unauthorized edge devices.	MATILDA Infrastructure A rogue producer spoofing the MAC address sending aberrant data Verifying the capability to identify aberrant behavior and update snort rules for application graph VMs	Must be available – aberrant data not present in: 1 GUI 2 DB 3 Processing Nodes

Annex 2 References

[MATILDA-D1.6] D1.6 – Supported Verticals, Use Cases and Acceptance Criteria. http://www.matilda-5g.eu/index.php/outcomes

Annex 3: Updated Risks Assessment and Contingency Plans

MATILDA pays particular attention to risks management during the execution of the project, enacted through an iterative cycle of: a) identifying risks, b) analyzing risks, c) managing risks, & d) monitoring risks. Following, we provide risk analyses, including the main identified and detected risks and the way they are resolved throughout the course of the project until the date of resubmission of this deliverable. The set of identified risks are summarized in the table below, ranging from scientific and technical to organizational and communicational. Initially, a set of risks associated with the development and operation of each of the MATILDA demonstrators are enlisted as headings to tables and their respective management strategies are detailed in each of the tables presented below.

Table 19. Identified risks and mitigation actions for the Emergency Infrastructure with SLA Enforcement (5G PPDR) Demonstrator.

Risk	Description	Mitigation actions	Risk Assessment and Management
VM instantiation failure	Application component cannot spawn if no VM is available	Multi-host IaaS should be available	Registration of multi-host IaaS infrastructure has taken place in the testbeds in Genoa, Ljubljana and Athens. Minor problems in deployments related to the allocation of computational resources have been identified and fixed. There is no negative impact from these actions in the overall demonstrator planning and operation.
Intra-connectivity failure	Application components are unable to communicate between each other	Reliable IaaS networking	The network connectivity between the IaaS in the demonstrator deployment sites would have been subject to tests before the main demonstration event. The facilities in Genoa, Ljubljana, Athens and Bristol have verified network connections.
Internet connectivity failure	Users are unable to reach web application	Redundant network uplinks	Access to web application is guaranteed through the establishment of the appropriate network connectivity and management mechanisms.
5G-ready slice is not reliable	Mobile devices not violating SLA defined KPIs (e.g. RTT)	qMON-based network monitoring provides detailed monitoring metrics	Based on data provide by qMON, reactive actions for guaranteeing QoS levels and SLAs are applied (e.g. scaling actions).
MATILDA orchestrator integration with the OSM	Networking performance monitoring via VNF-based qMON component is not possible	qMON is implemented as application component and directly instantiated through the MATILDA platform	qMON is integrated within MATILDA for providing QoS monitoring data, while it has also been made available as a VNF for usage by NFVOs.

Table 20. Identified risk and mitigation actions for the High-Resolution Media on Demand Vertical, with Smart Retail Venues' integration (5GPACE) Demonstrator.

Risk	Description	Mitigation actions	Risk Assessment and Management
Difficulty of integrating two separate systems	The combined Italtel and Incelligent systems require a large and heterogeneous set of compute and network resources.	The MATILDA Framework significantly facilitates the creation and deployment of the combined Italtel-Incelligent system for immersive and personalized services, enabling the straightforward integration of the two inter-operating systems through the micro-service approach ("Service Mesh").	Systems are designed based on the "Service Mesh" Approach. Integration has been tested successfully.
Data privacy & security compliance	5GPACE is designed to support a number of end-users through their mobile devices), which involves the gathering and processing of private and sensitive data. User privacy must be respected according to trade ethics and privacy regulations.	Measures are taken during the design, development and deployment of the 5GPACE application, ensuring obtaining/withdrawing user consent, user's right to data access and right to be forgotten. Furthermore, personal data are anonymized through data transmission between microservices.	The 5GPACE App adheres to the measures described. Additionally, it should be noted that the testbed on which the application will be deployed is a controlled environment.
Locality Awareness	The use case requires for the system to be aware of the user's location.	As a minimum requirement, being aware when a user is approaching an area (e.g. a shop within a venue) is enough for the demonstrator. Beacons will be tested.	Beacons have been successfully tested.
Edge Computing capabilities	System originally designed for Edge computing capabilities for network offloading and low latency requirements fulfillment.	Edge components of the combined application graph have been deployed closer to the UE and the backend components remotely to emulate a hybrid Edge deployment.	The initial demonstration in Ljubljana included a cloud-native application deployment over the internet at the UBITECH development lab location with the UEs at the demonstration location. The performance (i.e. video latency and throughput was reasonable over a Wi-Fi connection. For the final demonstration, where LTE Radio access will be used, the Edge components will be located at the demonstration venue, thus maintaining good properties.

Table 21. Identified risk and mitigation actions for the Smart City Intelligent Lighting System Demonstrator.

Risk	Description	Mitigation actions	Risk Assessment and Management
OpenStack framework functionality	Entire or some parts of IaaS is not working properly	Framework configuration file and VM's image back-up. Restore from back-up repository.	A set of test sequences were developed to evaluate the functionality: 1. Component unit test (Neutron, Nova, Keystone, Ceilometer etc) 2. Integration test - test suite for multiple deployment scenarios 3. Functionality test Interoperability test - to verify if a particular functionalities respond as expected.
VM's instantiation failure	Application component cannot spawn or cannot deploy the slice network if no VM available	Instantiate VM using image from back-up	Virtualization tool functionalities evaluation, for OpenStack image creation on VM machine
No connectivity setup between Smart City components	Application components are unable to talk between each other	Redundant connectivity deployment	VNF/PNF deployment tests
No connectivity between device sensors and C2 IoT Platform	Devices are unable to talk with platform	Redundant connectivity using GPRS technology or NB-IOT (due to multiple radio access capabilities of the sensors)	LTE connectivity from IoT Platform to UE to evaluate stability of the system (retainability measurements ~ 30 minutes of traffic)
C2 component instantiation failure	Main Smart City Application component is unavailale	Multiple deployment iterations	Monitoring C2 component functionality through an automated script

Table 22. Identified risk and mitigation actions for the Industry 4.0 Demonstrator.

Risk	Description	Mitigation actions	Risk Assessment and Management
No connectivity setup between the components	Mobile equipment used in demonstration not identifying connecting to infrastructure network	Coordinating with testbed operators for information on equipment requirements for establishing connections. Request for information such as Network carrier information, SIM cards, Band frequency used etc.	Communication modems supporting multiple carrier bands are selected and test to be performed with one common band supported at in-house location as well as the two testbed locations
Connectivity failure between devices and application components	Application components are unable to communicate with each other / Losing internet connectivity	Using multiple network connectivity options such as LTE and WLAN	Demo devices configuration and testing in-house to switch between the available communication networks. The facilities in Bristol and Athens have verified network connections.
Application components not instantiating	Multiple VMs to instantiate application graph components may fail to translate properly from docker compose mechanism	Regular onboarding testing on the final testbed infrastructures	The translation of application graph docker compose is regularly communicated

Table 23. Identified risk and mitigation actions for the Automobile Electrical Systems Remote Control Demonstrator.

Risk	Description	Mitigation actions	Risk Assessment and Management
Mobile devices not registering to test infrastructure	Mobile equipment used in demonstration not identifying connecting to infrastructure network	Coordinating with testbed operators for information on equipment requirements for establishing connections. Request for information such as Network carrier information, SIM cards, Band frequency used etc.	Communication modems supporting multiple carrier bands are selected and test to be performed with one common band supported at in-house location as well as the two testbed locations
Connectivity failure between edge devices and application components	Losing internet connectivity on C2 and C5 when communicating over LTE	Using multiple network connectivity options on the edge devices such as bot LTE and WLAN	Demo devices configuration and testing in-house to switch between the available communication networks

Application components not instantiating	Multiple VMs to instantiate application graph components may fail to translate properly from docker compose mechanism	Regular onboarding testing on the final testbed infrastructures	The translation of application graph docker compose is regularly communicated and discrepancies in application deployment via multiple VMs monitored and corrective actions in translation taken
Shared memory API translation to MATILDA Orchestrator	In house demonstrator's application graph instantiation use single VM shared memory which is in contrast to MATILDA's orchestration mechanism which uses one VM per application component and host shared context. Shared memory API translation could result in unexpected behaviour	Regular onboarding testing on the final testbed infrastructures	Monitoring of memory allocation statistics using MATILDA Prometheus interface to check application components behavior within specification. Having additional monitoring capabilities in application components for observing data corruptions and application behavior violations such as cross communications between multiple instances of same application component
Application component execution on vCPU (virtualized CPU) can lead to timing issues	Application graph components relaying real-time statistics and data may fail to report in time when the host CPU fails to attend the slice of application component in time due to resource overload. In such cases the application can land in undefined states	Regular onboarding testing and load tests on the final testbed infrastructures	Monitoring of CPU usage and process polling statistics using MATILDA Prometheus interface to check application components behavior within specification. Having additional monitoring capabilities in application components for observing timing issues

In sequence, the overall project technical and management risks as identified initially in the MATILDA proposal are further detailed in terms of current status of the risk assessment as well as of situation and management processes adopted to mitigate the situation until the date of this deliverable submission.

Table 24. Overall Project Identified Risks and Risk Assessment & Management

Risk	WP	Proposed Risk-mitigation Measure	Risk Assessment and Management
R1: Insufficient consortium coordination	WP8	The effective management of the consortium will be assured with the appropriate Project Management described in WP8. The roles & responsibilities of each partner are already identified and will be continuously reviewed in order to mitigate the risk of overlapping and implementation of the same activities by multiple partners.	Effective and mainly proactive management of the consortium has been realized. The roles and responsibilities of each partner have been made clear, avoiding overlapping while guaranteeing interoperability of the implemented set of tasks included in a given activity.

R2: Insufficient consortium competence / effectiveness	WP8	The project team is highly complementary and composed of partners that possess the set of skills required for the mainstreams of research and technological development. Moreover, all the technologies that will be used in the implementation of the project will be carefully selected in order to minimize potential risks that could be introduced based on the chosen technologies.	Minor issues in terms of effectiveness have been identified and tackled by the consortium. Continuous technology watch is taking place aiming at the adoption -where feasible- of latest specifications and releases in the various technologies adopted within the project.
R3: Conflicts over ownership	WP7	Disagreements in the consortium over ownership may result in non-agreement on IPR. The principles and the existing assets included in the Consortium Agreement, the continuous Task 7.4 on IPR handling, and the creation of an ongoing IPR inventory will ensure a proper protection of generated and prior IPR.	Agreement on the IPR aspects has been realised and continuously monitored. No conflicts of ownership have been identified.
R4: Shortage of resources and/or change of personnel	WP8	Problems with personnel relate to lack of competencies and withdrawals. However, all the partners have assured that they will choose their best personnel to implement the relevant activities. All partners have the ability to replace any member of their team with another person with comparable competencies, in case of inability to continue. The project partners would make binding agreements on the availability of resources. Keeping close contact with all partners in order to ensure early communication of budget and personnel-related problems.	Minor issues in terms of availability of resources have been identified and resolved within the project. Complementarity and team spirit among the project participants is in a high level, guaranteeing the tackling of any identified inefficiencies or shortage in resources.
R5: Lack of communication among the partners	WP8	Keeping close contact with all partners by conducting regular teleconferences and virtual meetings. Organization of regular plenary and technical meetings at different partners' sites. Consideration of reworking the exploitation plans when needed. Detailed project plan that clearly states goals and responsibilities of each of the project partners.	Regular teleconferences, project meetings and ad-hoc communications among partners are periodically taking place. These regular virtual as well as periodic physical meetings have been very useful in helping to keep track of events as they unfold within the project's development cycle. Exploitation plans per partner are also under revision, towards the release of the final exploitation plans by the end of the project.
R6: Partner withdrawal	WP8	Immediate substitution by another partner, from existing partnerships, through dissemination activities or via the tight relationship and interaction with the 5G-PPP.	No partner withdrawal instance is required.

Table 25. Administrative/Financial Risks and Risk Assessment & Management

Risk	WP	Contingency	Risk Assessment & Management
R1: Tight schedule for Reference Architecture	WP1	WP1 is diverse: it covers stakeholders' identification & requirements analysis (T1.1), verticals definition (T1.7), heavy conceptualization (T1.3, T1.4, T1.5, T1.6), as well as a rigorous start towards the development of the reference architecture (T1.2). By consolidating these listed tasks in a single WP, it ensures a top coordination and execution of tasks equally needed for the project to advance in the right direction, including support from the entire consortium. The workplan needs to start strong. Additional resources will be allocated if needed.	The overall MATILDA reference architecture has been released on time. The different aspects concerned with the realization of the overall project objectives within the WP1 has been actualized and the project is rapidly progressing with a tight coordination with the MATILDA reference architecture.
R2: Conceptual Failure of Reference Architecture	WP1	The architecture (along with the design requirements) will be carefully designed and refined through two tasks, T1.1 & T1.2, based on a thorough review of the relevant 5G-PPP reference architecture results, standardization work in 3GPP SA2, and other 5G architecture activities, as well as on the requirements of verticals.	Validation of the overall approach has taken place based on the acquired project results as well as the progression in the development of different relevant standards specifications.
R3: Limited Functionality or Inadequate Integration of MATILDA Mechanisms	WP2, WP3, WP4, WP5.	The project workplan includes two tight cycles of development, integration and demonstration of the several components (WP2-WP5). The successful integration of these components into the MATILDA framework (WP5) represents a critical chapter in the workplan. An overlap is in place between implementation and integration, as well as the continuous participation of the same partners. Strong horizontal technical coordination of WP2-WP5 will also be in place.	Efficient and successful integration of mechanisms provided by WP2, WP3 and WP4 has taken place in WP5. Over the integrated MATILDA platform, five demonstrators are going to be shown, by using the provided functionality by the mechanisms in the various layers.

R4: Time for Testbed Deployment and Verticals Development is Underestimated	WP6	MATILDA provides a set of operational testbeds in CNIT, UNIVBRIS and UBITECH, while smaller installations are also made available in ININ and ORO. The design and implementation of the verticals has started early in MATILDA, from the in-depth identification of the 5G requirements (T1.1), the supported use cases for each vertical (T1.7), the scope definition and planning (T6.1) and the two development-demonstration cycles (WP5-WP6).	MATILDA testbeds are operational supporting the realization of trials based on the tests conducted before and during preparations for the project review meetings. Appropriate planning and testbeds preparation is taking place, prior to each demonstration phase.
R5: Project propositions too ambitious to work properly in project runtime	WP2, WP3, WP4.	The project will make careful steps towards the realization of its objectives. If needed, the consortium has the experience to adjust these objectives so that they can be achievable and still yield the anticipated results. The project will follow the motto "think big, act small" to produce results that can realistically become exploitable and useful after its completion –in cooperation with the 5G-PPP.	All project objectives are on track and while some have been achieved, others are gradually been realized in a timely manner.
R6: Project facing technology replacement issues	WP2, WP3, WP4.	ICT technologies continue to develop at rocket speed, and it is difficult to foresee their evolution. For this reason, the project will be engaged in a continual technology watch effort, which will last till the very end of the project. The technical management of the project will always be in touch with the scientific community for learning about possible future disruptive technologies relevant to the project activities. The consortium will deliver concepts that are going to be easily adopted and reused by stakeholders and other initiatives, and to be built on existing standards so as to effectively face potential technology replacement issues.	Continuous technology watch is constantly and efficiently taking place. Close monitoring of working groups in various standardization bodies is also taking place, leading to the adoption -where feasible- of recent technological specifications.

R7: Insufficient Project Impact, Stakeholders Reach Out and Low Interest from Verticals	WP8, WP9.	MATILDA has onboarded five demonstration partners, representing the vertical segment (smart cities, connected vehicles, emergency infrastructure, media and industry 4.0). In addition, the consortium consists of a number of vendors, mobile operators and service providers, indicating the interest of industry in MATILDA. The extended community and business network of these industrial partners (ATOS, ERICSSON, INTRASOFT, ORO, COSM), e.g. the extended customer base and business network of TEI (over 1,000 networks in more than 180 countries), will reassure the reach out of a critical mass of stakeholders, service providers, vendors and verticals.	A wide set of trials is taking place and dissemination activities are regularly being undertaken by the MATILDA project partners. MATILDA is also present in a set of conferences and workshops, providing presentations and live demos of the ongoing developments within the project. Strong presence is also realized within various 5G PPP WGs with contributions on behalf of MATILDA in various working items.

Annex 4: GDPR Issues in Demonstrators

The General Data Protection Regulation (GDPR) [1] is a framework for harmonizing data protection rules across the European Union. The regulation "[…] *applies to the processing of personal data wholly or partly by automated means and to the processing other than by automated means of personal data which form part of a filing system or are intended to form part of a filing system* […]" [2]. In this respect, since Demonstrator 1 "High Resolution Media on Demand Vertical, with Smart Retail Venues' Integration" involves both the gathering of private and sensitive data (demographic, location and consumption data points) and the processing of such data, this use case must strictly follow the GDPR principles and rules.

In this respect, a list of the measures undertaken during the design, development and deployment of the 5GPACE application have been reported in Deliverable D1.6 [3]. Such measures have been conceived by taking into consideration the specific functionalities of the 5GPACE application and the corresponding interactions with the user that involve personal data.

In particular, thorough consideration has been put in defining a set of terms of consent in a clear way that can be seen in the first screen of the application and also easily accessed at any time in the application settings: in order to adhere to the trade ethics and privacy regulations, the user must originally give his/her consent before registering and is then able to withdraw it through the 5GPACE App, therefore respecting user's right to be forgotten.

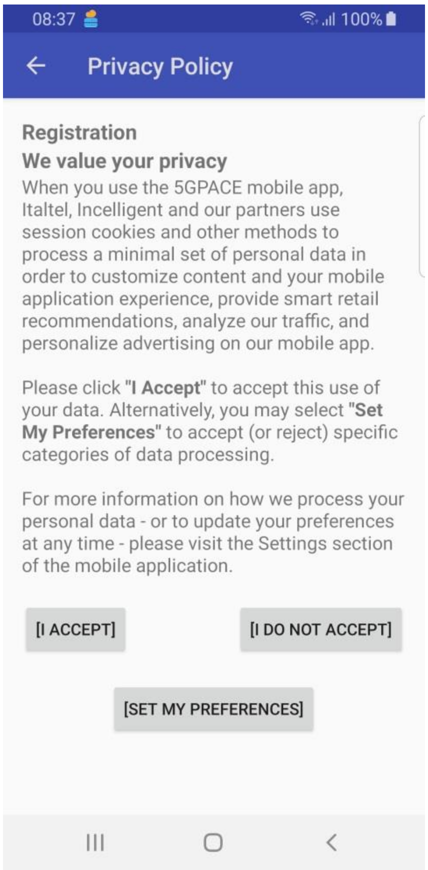
For this reason, before registering to use the application, the users agree to a set of terms of consent which are clear and describe exactly how their data will be stored and processed. The consent form is presented below:

Registration
We value your privacy
When you use the 5GPACE mobile app, Italtel, Incelligent and our partners use session cookies and other methods to process a minimal set of personal data in order to customize content and your mobile application experience, provide smart retail recommendations, analyze our traffic, and personalize advertising on our mobile app.

Please click "**I Accept**" to accept this use of your data. Alternatively, you may select "**Set My Preferences**" to accept (or reject) specific categories of data processing.

For more information on how we process your personal data - or to update your preferences at any time - please visit the Settings section of the mobile application.

three buttons
[I accept] [I do not accept] [Set My Preferences]



Privacy Policy

Registration

We value your privacy

When you use the 5GPACE mobile app, Italtel, Incelligent and our partners use session cookies and other methods to process a minimal set of personal data in order to customize content and your mobile application experience, provide smart retail recommendations, analyze our traffic, and personalize advertising on our mobile app.

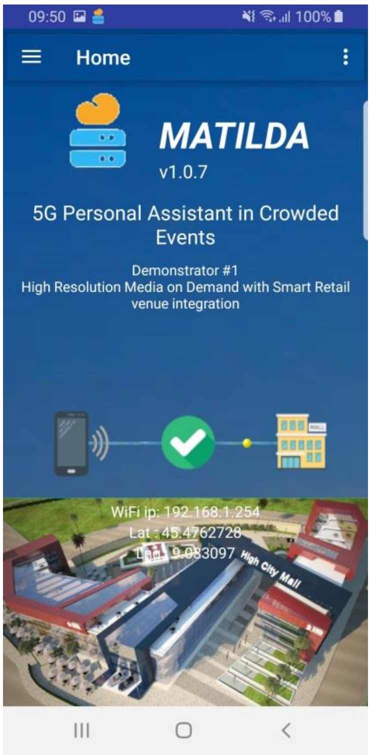
Please click "**I Accept**" to accept this use of your data. Alternatively, you may select "**Set My Preferences**" to accept (or reject) specific categories of data processing.

For more information on how we process your personal data - or to update your preferences at any time - please visit the Settings section of the mobile application.

[I ACCEPT] [I DO NOT ACCEPT]

[SET MY PREFERENCES]

Figure 27: Acceptance of Privacy Policy of 5GPACE Mobile App upon initial registration.



Home

MATILDA

v1.0.7

5G Personal Assistant in Crowded Events

Demonstrator #1
High Resolution Media on Demand with Smart Retail venue integration

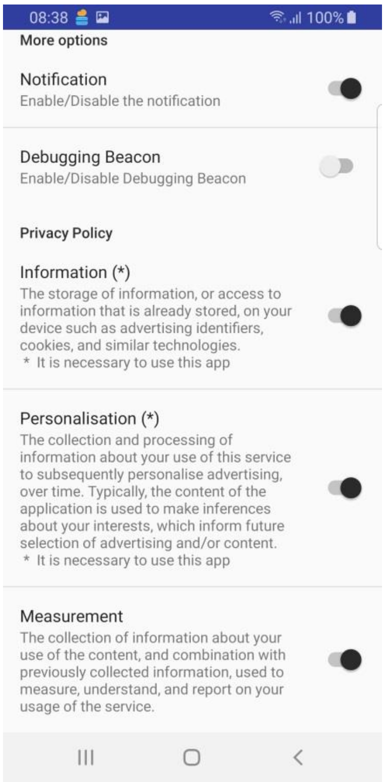
WiFi ip: 192.168.1.254
Lat : 45.4762728
Lon : 9.083097

Figure 28: 5GPACE Mobile App initiation upon acceptance of Privacy Policy.

Additionally, a special User Preferences/ Settings page had been added to the 5GPACE mobile app to allow the user to update/withdraw their privacy preferences. This is backed by the following Operational Procedure which is now implemented manually (via email ticketing) or through a dedicated form at a later development stage.

Settings page

- **Information storage and access**
  The storage of information, or access to information that is already stored, on your device such as advertising identifiers, cookies, and similar technologies. **[Radio Button]**
- **Personalisation**
  The collection and processing of information about your use of this service to subsequently personalise advertising, over time. Typically, the content of the application is used to make inferences about your interests, which inform future selection of advertising and/or content. **[Radio Button]**
- **Measurement**
  The collection of information about your use of the content, and combination with previously collected information, used to measure, understand, and report on your usage of the service. **[Radio Button]**



08:38 100%

More options

Notification
Enable/Disable the notification

Debugging Beacon
Enable/Disable Debugging Beacon

Privacy Policy

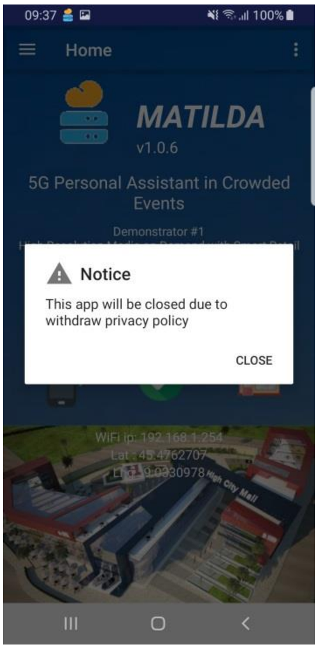
Information (*)
The storage of information, or access to information that is already stored, on your device such as advertising identifiers, cookies, and similar technologies.
* It is necessary to use this app

Personalisation (*)
The collection and processing of information about your use of this service to subsequently personalise advertising, over time. Typically, the content of the application is used to make inferences about your interests, which inform future selection of advertising and/or content.
* It is necessary to use this app

Measurement
The collection of information about your use of the content, and combination with previously collected information, used to measure, understand, and report on your usage of the service.

Figure 29: User Preferences/ Settings page of 5GPACE Mobile App.

Furthermore, if, at a later time, the user desires to update his/preferences, even withdraw his/her consent, Italtel and Incelligent have outlined the process with which this can made possible. In Activity Diagram 1 (see below), the interactions between the various components are presented. In detail, as the user navigates to the privacy settings menu, he/she may request an update through the 5GPACE app component. The latter acknowledges the updated request and updates the user preferences of the UGDM component which is responsible for the collection and monitoring of the relevant information. At the same time, if the user withdraws his/her consent, any past user location data are deleted from the UserLocationDB along with the recommendation cache for the specific user.



Notice

This app will be closed due to withdraw privacy policy

CLOSE

Figure 30: Notification of withdrawal of Privacy Policy consent at 5GPACE Mobile App.

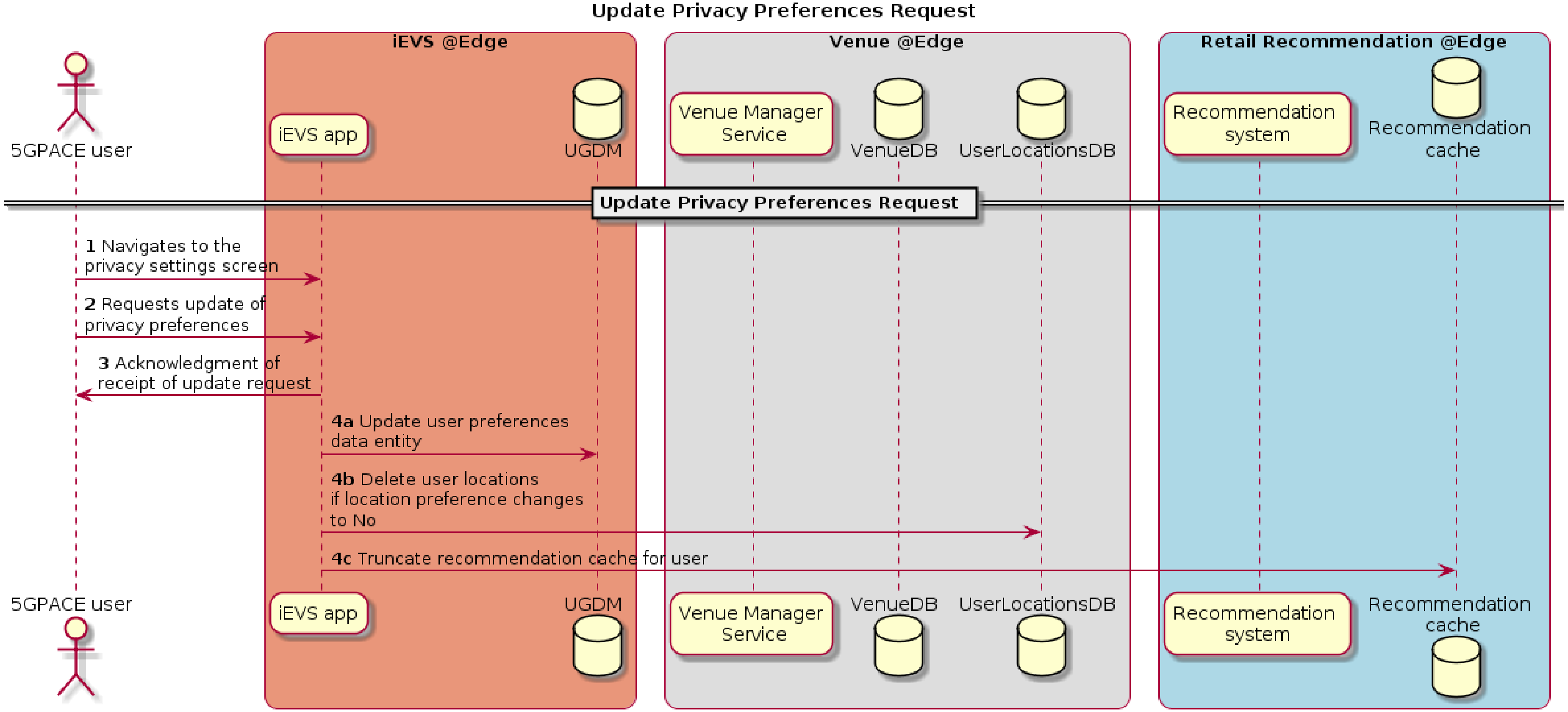
It is worth noting that the remaining demonstrators do not process personal data and, as such, they do not need to comply with the GDPR rules.

Annex 4 References

[1] Data protection Rules for the protection of personal data inside and outside the EU. http://ec.europa.eu/justice/data-protection/reform/index_en.htm

[2] https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:02016R0679-20160504&from=EN

[3] D1.6 – Supported Verticals, Use Cases and Acceptance Criteria. http://www.matilda-5g.eu/index.php/outcomes


Update Privacy Preferences Request
iEVS @Edge
Venue @Edge
Retail Recommendation @Edge
5GPACE user
iEVS app
UGDM
Venue Manager Service
VenueDB
UserLocationsDB
Recommendation system
Recommendation cache
Update Privacy Preferences Request
1 Navigates to the privacy settings screen
2 Requests update of privacy preferences
3 Acknowledgment of receipt of update request
4a Update user preferences data entity
4b Delete user locations if location preference changes to No
4c Truncate recommendation cache for user

Activity Diagram 1: Interactions between components in case of updating privacy preferences.