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 D6.1

EVALUATION FRAMEWORK AND DEMONSTRATORS PLANNING

Due Date of Delivery:	M15 (31/08/2018 *dd/mm/yyyy*)
Actual Date of Delivery:	18/10/2018 *dd/mm/yyyy*
Workpackage:	**WP6 – MATILDA Industrial Demonstrators and Performance Evaluation**
Type of the Deliverable:	R
Dissemination level:	PU
Editors:	ORO, CNIT
Version:	1.0

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
Roberto Bruschi, Olga Jaramillo, Chiara Lombardo	
COSM	COSMOTE KINITES TILEPIKOINONIES AE
Eleni Theodoropoulou, Ioanna Mesogiti, Dimitris Polydorou	
ORO	ORANGE ROMANIA SA
Horia Stefanescu, Marius Iordache, Bogdan Rusti, Jean Ghenta, Cristian Patachia	
EXXPERT	EXXPERTSYSTEMS GMBH
Earon Beckman, Uday Adusumilli, Daniel Jansen, Akhil Maganti, Jörn Albrecht	
ININ	INTERNET INSTITUTE,COMMUNICATIONS SOLUTIONS AND CONSULTING LTD
Luka Koršič, Dušan Mulac, Jaka Cijan, Janez Sterle	
INC	INCELLIGENT IDIOTIKI KEFALAIOUCHIKI ETAIREIA
Konstantinos Tsagkaris, Panagiotis Demestichas, Athina Ropodi, Nikolaos Stasinopoulos, Marinos Galiatsatos, Dimitris Cardaris, Aristotelis Margaris, Vassilios Bourdouvalis, Nikolaos Gavriil, Konstantinos Bitsakos	
UNIVBRIS	UNIVERSITY OF BRISTOL
Borja Otura, Hamid Falaki, Ben Thomas, John Harris, Anderson Bravaleri, Reza Nejabati, Dimitra Simeonidou	
ITL	ITALTEL SPA
Pietro Paglierani	
BIBA	BIBA - BREMER INSTITUT FUER PRODUKTION UND LOGISTIK GMBH
Kay Burow, Zied Ghrairi	

Disclaimer

Copyright

Table of Contents

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.

This deliverable provides the evaluation framework and the planning for the execution of demonstrators in a coordinated and unified manner, in order to lead to valuable remarks and conclusions about the viability and the sustainability of the MATILDA framework.

Using this framework five demonstrators are implemented over three different test beds, each representing a key 5G vertical. Each demonstrator will evolve in two or three releases, each of the releases having a well-defined set of functionalities to be demoed at international conferences or project review meetings. The first release of the demonstrators is meant to show the interoperability of the different components of the framework. However, these components are not all deployed in their final location or do not have full capabilities regarding 5G application graph deployment, slice adaptation, horizontal/vertical scaling of virtual resources, etc. Such capabilities will be enabled during the second and third (where available) releases of the demonstrators.

In order to be able to assess the different releases of the demonstrators in a consistent manner, the deliverable introduces an evaluation framework with relevant test procedures related to evaluating specific applications' QoS KPIs, the performance of the functionalities/capabilities to be delivered by a single or by multiple MATILDA components, and the solution as a whole for the development and definition of 5G-ready applications over a sliced network infrastructure. A preliminary list of tests is also proposed which will be further refined within the upcoming deliverables from WP6.

In order to be able to schedule all demonstrator releases, and to be consistent with the components or test beds availability, a complete and detailed plan was developed within this deliverable, to be followed within the duration of the project.

2 Introduction

One of the main 5G objectives is to "create an ecosystem for technical and business innovation involving vertical markets, which will serve a larger portfolio of applications with a corresponding multiplicity of requirements ranging from high reliability to ultra-low latency going through high bandwidth and mobility". [5GPP-2015]

The MATILDA project will endorse this commitment by unifying network slicing, edge computing and multi-tenancy abstractions into an integrated framework 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 5G-ready Application Orchestrator, 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, developed based on the requirements identified in the previous work packages of the project with an extended review methodology of test cases planned to be implemented during the execution phase of the demonstrators.

The scope of this deliverable is to provide the evaluation framework along with the planning and coordination steps of the demonstrators' execution, offering the appropriate guidelines and documentation to support the demonstrations' integration in the MATILDA envisaged framework. The evaluation framework is detailed in Section 3, which covers the structure and description of the testing procedures to guide the validation and performance evaluation activities, spanning from end-user experienced performance to specific component testing level, by checking the delivery of the expected functionalities and the communication process through the specified interfaces. To evaluate the risks impacting the ability to meet the expected solution outputs, a Risk and Quality Management evaluation is proposed addressing quantitative and qualitative aspects by taking into account the peculiarities of each use case developed inside the project.

Five demonstrators are implemented over three different test bed platforms described in Section 4. The Orange test bed solution architecture is one of them, and represents a common effort deployment inside MATILDA and SLICENET 5G-PPP EU projects. It is being deployed as a small-scale network following an ETSI MANO architectural framework for virtualization, and combines several network components: device sensors, radio access part VNFs, virtualized Core (vEPC, vHSS), Cloud Enterprise IoT platform and apps creating a logical infrastructure with dedicated resources and topology to sustain vertical application functionality. The University of Bristol's 5G, the second test bed proposal, comprises several computing nodes, control plane/data plane switches, mmWave SDN mesh and radio access components interconnecting different areas, such as Millennium Square (MS), We The Curious (WTC) and the Smart Internet Lab (HPN) in the city of Bristol (United Kingdom), with the capability of providing different levels of virtualized infrastructure. CNIT, UBITECH and COSM integrated test bed infrastructure is the third one, and is composed of two main parts: the radio access part containing Amarisoft eNodeBs with support for NB-IoT and 4G/5G user equipment devices, residential gateway boxes and NB-IoT sensors, and the virtualized infrastructure part composed of a set of high-performance equipment supporting OpenFlow, SDN and the OpenStack Project.

The first demonstrator proposal is 5GPACE- High Resolution Media on Demand & Banking on the Cloud designed to provide high value services to end users during crowded events, combining the i-EVS framework by Italtel for immersive video services [IARIA17] with the Incelligent system that delivers real time personalized recommendations to end users, from the analysis of historical and mobility data using machine learning techniques. Testing 4.0- Distributed System Testing, the second demonstrator, is addressing the data acquisition and remote testing scenario of Mobile System under Test (SUT) units using the FastWAN solution by ExxpertSystems over a 5G network. 5G Emergency Infrastructure with SLA Enforcement is the third demonstrator, revealing a 5G-enabled emergency response capabilities solution through the iMON product suite for real time intervention monitoring supporting active and passive SLA monitoring and SLS enforcement. The fourth demonstrator, Industry 4.0 Smart Factory – Inter and Intra-Enterprise Integration, addresses the concept of robot farming, in which the collaborative process components (robots, cameras, laser scanners) will collaborate through the vertical by sharing and analyzing time critical data to ensure productivity, flexibility and safety. The last demonstrator, Smart City Intelligent Lighting System, optimized for LoRaWAN and LTE-M networks, is designed to validate several requirements for Smart Lighting vertical deployment, such as automated end to end instantiation as a vertical slice; full management of lighting devices with high availability, and resilience in the case failure scenario; fast scalability and dynamic QoS provisioning and enforcement according to specific slice metrics and real-time network conditions.

For each demonstrator, a set of test scenarios were developed describing evaluation steps and associated roles. The demonstrator release evolution is depicted, summarizing the capability/features of the demo together with test bed requirements to emphasize the evolution among releases in terms of requirements. The acceptance criteria for each of the releases are set as guidance for final validation.

Section 5 describes the master plan proposal, a unique view of all the project phases, highlighting the planning to be followed for the test beds' development, each of them with a relevant set of capabilities to be demoed during international conferences or project review meetings and demonstrators' deployment strategy over the selected test bed.

The document ends with the Conclusions summarizing the contribution of the deliverable.

3 Evaluation Framework Definition

This section describes the initial activities undertaken in the scope of the Task 6.2. In more details, at this stage of the Project, the focus has been on the identification of the most significant outcomes of the other WPs worth evaluating, both singularly and in the context of the MATILDA framework, as well as performance and general solutions testing to provide a whole characterization of the applicability and potential of the innovations brought forth by the MATILDA Project.

The following sub-sections cover the structure and description of the testing procedures. An initial list of tests, as identified at this stage, is provided in Annex 2. Such list will be refined throughout the project lifetime to better suit implementation specificities that might emerge in the development phase, along with test beds specific features, environment setup/tools, etc.

3.1 Structuring of Testing Procedures

At the MATILDA testing phase, a number of tests will be performed spanning from end-user experienced performance to specific solution/component testing level, in order to cover a wide range of aspects that will determine the adoption of the solution. Likewise, the tests to be performed are classified in the following major categories:

- **Performance** testing aiming at evaluating the whole MATILDA solution on the basis of specific applications' QoS KPIs defined in the MATILDA use cases or/and by the MATILDA end-users, as well as towards the 5G-PPP KPIs [MATILDA-D.1.1].
- **Solution Components and Functionality** testing aiming at verifying the operation and evaluating the performance of the functionalities/capabilities to be delivered by a single or by multiple components of the MATILDA solution [MATILDA-D.1.1].
- **General Solution** testing aiming at the evaluation of the solution as a whole for the development and definition of 5G-ready applications and network services and their deployment over a sliced network infrastructure.

More specifically, **Performance** testing will focus on the end-to-end QoS that can be delivered by the MATILDA solution and that is experienced at the application layer. In practice, this will include tests related to:

- *end-to-end bandwidth allocation*, including measuring the achieved maximum and guaranteed data rates required by each use case, verification of the dynamic bandwidth allocation capability relevant to use cases, etc.
- *end-to-end latency*, including verification of meeting the latency requirements of each use case, as well as measuring control and data plane latency and assessing the solution against the 5G-PPP latency KPIs, to the extent that is feasible with the test bed infrastructures
- *services prioritisation,* in terms of QoS adjustments, pre-emption enforcement, etc., especially required by Use Case 3
- *devices' density support*, especially defined by Use Case 5 (for IoT), and by the relevant 5G-PPP KPIs, to the extent that is feasible with the test bed infrastructures.

Solution Components and Functionality testing will aim at testing the MATILDA solution components in a stand-alone –unit testing- and integrated fashion in order to verify that they deliver the expected functionalities/capabilities and communicate through the specified interfaces. In practice, this will include tests related to the following main solution components [MATILDA-D.1.1]:

- *The 5G-ready applications development toolkit*, including tests related to:
    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*, including tests related to:
    a. the interface to end users/application owners/verticals
    b. the lifecycle management of applications/application components' in the repository
    c. the lifecycle management of VNFs in the repository (to be accessed by end-users)
    d. the runtime policies assignment to applications' service graphs
    e. the handling of various, different profiles/functions for different users/ stakeholders/roles.
- *The 5G-ready Application Orchestrator,* including tests related to:
    a. Delivery of real-time deployment planning, taking under consideration the available programmable resources and the current situation in the infrastructure where these resources reside through *the Optimization Engine and the Policy Engine*
    b. Enforcement of specific execution policies over the deployed 5G-enabled application following a continuous match-resolve-act approach
    c. *Monitoring and management* of applications/application components through *Monitoring and Data fusion* mechanisms handling multiple parallel data loads from multiple sources
    d. Extraction of advanced insights and events from the Monitoring process
    e. Support of *Real-Time Analytics* of multiple contexts
    f. Support of re-active reconfigurations of application deployment based on monitoring and analytics data.
- *The Vertical Application Orchestrator (VAO),* including tests related to the extraction of the slice intent on the basis of the MATILDA metamodels upon request, as well as the lifecycle management of application components' and their chains' graph instantiation.
- The *OSS/BSS*, including tests related to:
    a. the interface between the Vertical Application Orchestrator and the underlying 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*, including tests related to the lifecycle management/support of slices.
- *The Network Function Virtualization Orchestrator (NFVO)*, including tests related to the lifecycle management/support of VNFs/PNFs as well as the re-use and parameterization of VNFs by multiple tenants.
- *The Multi-Site Resource Manager,* including tests related to:
    a. managing resources at diverse facilities, like central/remote public/private/hybrid cloud facilities or at the mobile network edge
    b. the *Computing Slice Manager (CSM),* with regard to the deployment of applications/application components at Network Service Provider's edge facilities using:
        a. end-user location and mobility information
        b. locality of computing resources
    c. lifecycle management of an application component deployed at the Telecom Service Provider's facilities through (evolved) IaaS/PaaS APIs.
- *The Virtual Infrastructure Manager (VIM),* including tests related to exposing the resources of data centers to the NFVO and CSM/VAO, supporting multi-tenancy on infrastructure resources.
- *The Wide-area Infrastructure Manager (WIM),* including tests related to the logical interconnectivity among sets of service/application components instantiated in different PoPs.

Moreover, a set of functionalities (delivered by one or more solution components) will be tested such as:

- The end-to-end Slice definition and realization, including tests related to the time to define, time to realize, time to release a slice.
- The VNF/PNF selection and configuration on a per instance basis
- The runtime policies definition and operation verification
- The multi-tenancy capability on a per component basis.

Finally, General Solution testing will include tests related to the evaluation of the solution as a whole in terms of the following aspects:

- User Friendliness of all end-user interfaces
- Speed of application deployment
- Expandability of the solution, e.g., in terms of expanding the underlying network and compute resources
- Scalability of each component and end-to-end
- Reliability, and so on.

3.2 Tests Definition

For the purpose of having a homogeneous description of the tests to be performed in the context of MATILDA, the tests have been specified by the contents of the following fixed format table:

Table 1: Tests Definition in Tabular Format.

Number	<#>	**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)		

The fields of the table are filled with the following information:

- **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.

An initial list of tests has been identified and is provided in Annex 2. This tests' list will steer the validation and performance evaluation activities of the project (Task 6.9), and will be refined at next stage in order to address better (a) the solution implementation specificities that will emerge throughout the development phase, (b) the demonstrating sites' specific features that are not foreseen prior to deployment, (c) the testing environment setup/tools etc.

3.3 Risk and Quality Management for Framework Evaluation

A proactive approach is envisaged in the MATILDA project to prevent or reduce undesired effects on the deployed solutions. Risks will be constantly assessed and evaluated within the whole project duration. To achieve the proposed objectives, an evaluation is planned in order to evaluate the risks that can affect the ability to meet the expected solution outputs. The risk evaluation would cover both quantitative and qualitative aspects, taking into account the particularities of each use case developed inside the project.

The methodology to be followed for risk management consists of four steps: a) Risk identification – to identify and classify the risk, b) Risk quantification – determine the probability of events and examines the consequences associated with their occurrence, c) Risk response - methods will be produced in order to reduce or control the risk, and d) Risk control and report – identified risks are documented.

3.3.1 Risk

A test bed solution will be deployed for each stage of project, for each component and also for entire MATILDA framework, with the purpose of testing and discovering if there is any potential risk in source code, before entering the environment production.

The assessment obtained from risk analysis contains the probability and the impact of solution inconsistencies that can be discovered, correlated also with the quality characteristics used by the tested system. In order to create test specifications, test cases and sequences of tests, multiple inputs are necessary: the measures of impact errors, probability errors and weighted quality characteristics.

3.3.2 Quality

The quality of the product is measured against documentation of solution, initial requirements/expectations, source code. A centralised data base is used to log all defects found during the tests.

Control points for quality will be used by testing the system to monitor and control the performance of source code for each component and also for the entire framework, to ensure that a defect is detected from the beginning of the lifecycle and not passed to the next phase without agreement based on defect severity.

In that respect, documentation to be provided in the process will be:

- List of test cases which are integrated, as detailed in Annex 2
- Parameters details used for integration, as detailed in Annex 2
- Test stubs, necessary for troubleshooting
- System transition state, for all components as a part and as a framework
- MATILDA solutions software

Using these control points, each partner can be compliant with the integration solution criteria.

3.3.3 Testing Criteria for Integration

Multiple criteria can be defined for integration testing, for Entry and for Exit. Also intermediate criteria are defined in case of discover of a defect, Suspend and Resume.

The Entry Criteria, used at the beginning of tests for each phase of the project, must be fulfilled:

- Hardware, software for test bed
- Integration parameters details
- System transition state
- Integration test case list
- Integration stubs
- Test products prepared for the Test Phase.

The Exit Criteria, which are the final decision of compliance in the tested phase:

- All test scripts executed at least once
- No severity Critical or High defects carried forward
- Only remaining severity Medium and Low defects can be solved later, but before final integration.
- A timeframe for fixing all remained defects must be agreed with the module owner
- Report produced for every completed test
- All deliverables must be delivered.

At the end, the final integration test must be passed with no defects remaining.

For description of defect severity please see Section 3.3.4.

If the result of test phase is fail (no quality pass checked), the test report will contain an explanation of error – what is the behaviour, what is the stub, if any. If the root cause is discovered and there is time to fix it and to re-execute the test, then that will be done.

In case that the defect cannot be fixed in a reasonable time, no matter the cause, then the test report will reflect this.

The fault is passed through a success criterion in order to decide if is a blocking point or not of the test phase. There are three possibilities:

- The test phase is failed – it is a blocking point. The defect obstructs the good functionality of the component or of the entire framework. It is mandatory to be fixed and the test phase repeated.
- The test phase is conditionally passed – it is not a blocking point. The test can continue to the next phase, in an agreed period the defect will be solved and then the failed test repeated.

The test phase is passed – can continue to the next stage.

3.3.4 Fault Management Process

In this sub-section we define a process about how to deal with a fault, how can categorize them from severity point of view and which is the priority to solve it.

We can consider it as defect when a test is failed three times consecutively. A document is fulfilled with an explanation, logged to data base and sent to developer for analysis/ solving.

The person which are doing the test will indicate the priority and also the severity of this issue, based on criteria that will be detailed later on. Using these characteristics the tester determines maximum time period allocated for resolution, solving and re-test scheduling.

There are two types of faults, categorised through nature of the source: specifications or implementation. This information must to be indicated in the report. If is related to specifications, then the corresponding requirement must to be re-analyse in order to avoid any ambiguities.

A good practice is to create a Test Failure report from the start of Test Phase in order to annotate all steps passed through the test's processes: the observation, the integration and the fault management.

Priority is used in order to prioritize the defect resolution. It is assigned to help designate the order in which defects should be fixed. If a blocking point of one test is or if is a pre-condition which must to be fulfilled for another test, depending on the impact on the business, priorities are shown in Table 2.

Table 2: Priority definition.

Level	Priority	Business Impact Level	Impact to Business
1	Critical	Severe Impact	The system/release cannot 'go live' with this defect outstanding as it would cause severe or total loss of business. Severe impact to plan, cannot continue until fix supplied. Must be fixed now (usually for blocking defect).
2	High	Significant Impact	Significant impact. For example - many test cases are blocked. The system/release could be prevented from 'going live' by such a defect. A defect is high-priority if one or more of the following is true: • material impact on customer service that could cause attrition • significant operations impact • breakdown in function causing client attrition in short term (measured in weeks)
3	Medium	Moderate impact	Only moderate or slight impact to plan. Only single test case affected and awaiting fix to continue. Not necessarily enough to prevent 'going live', but impacts non-core system functionality. Medium priority defect should have a resolution plan. If possible fix before final release.
4	Low	Minor Impact	No significant impact on the customer. No impact on business processes. Minor or no impact to plan. For example, test can continue without fix. Fix could be left until after go-live.

Severity is a measure of the technical impact on the system availability, also considering if a workaround is feasible or not. The severity definition is depicted in Table 3.

Table 3: Severity definition.

Level	Severity	System available	Severity of Function Degrade	Workaround Available	Impact
1	Critical	No	Severe	No	System unavailable/ workaround unavailable/ the function is inoperable and/ or the defect blocks execution of all or >75% of remaining tests.
2	High	Yes	High	No	Unacceptable. Not a fatal problem, but severe enough to cause substantial loss of function or business. System is available, but the function is inoperable for the majority of cases (or the function is inoperable for some cases but no workaround is available).
3	Medium	Yes	Medium	Yes	Survivable. Minor loss of functionality in a non- critical area or that would cause some inconvenience or small amount of embarrassment if promoted to a live system. The function is inoperable for the majority of cases but a reasonable workaround is available (or functions inoperable for a minority of cases but with a more difficult workaround).
4	Low	Yes	Low	Yes	Easy workaround available. Also covers cosmetic problems. Examples include: wrong font or text size; Minor spelling or grammatical mistake; incorrect positioning on screen used for internal use only.

At the end of tests, after all components were tested separately, the entire framework tests will be performed in order to check if all requirements are met. Tests are to ascertain that the required and specified functionalities are actually present, and that they function as specified.

3.3.5 Tests Readiness Review

All the applications composing the system will be deployed in a controlled integration environment.

Before running the pilot, a hands-on system test, named Test Readiness review, is performed in order to confirm that the system is ready for further tests. At that stage we will follow the specific documentation for assessment against this quality gate to proceed further in small or even large pilots.

4 Test Beds and Demonstrators

4.1 *Orange Smart City Test Bed*

The Orange test bed solution architecture, for MATILDA Smart City use case testing and validation, will be deployed as a small-scale network which will combine several network components with the aim of creating a logical infrastructure with appropriate isolation, dedicated resources and topology to sustain Smart City application vertical functionality.

Orange Romania Smart City test bed integrates the device sensors (lighting poles equipped with radio modules) with all associated RAN access part VNF, virtualized Core (vEPC; vHSS) and a Cloud Enterprise IoT platform and apps.

4.1.1 MATILDA SliceNet Collaboration

Orange Romania is one of the main contributors and the use-case leader for the Smart City vertical for the two 5G-PPP EU projects SliceNet and MATILDA. The test bed and the Smart Lighting demonstrator are built as a collaborative work among these two projects and are depicted in Figure 1. The SliceNet project is focused on the 5G infrastructure layer, while the MATILDA project focuses more on the 5G-ready applications to be deployed over this 5G infrastructure layer.

SliceNet defines an E2E slicing-friendly 5G network architecture, vertical-oriented, with cognitive network management, control and orchestration, with an innovative one-stop-solution to meet the service requirements, enabling the verticals to Plug & Play (P&P) their use cases, as an integrated FCAPS (Fault-management, Configuration, Accounting,

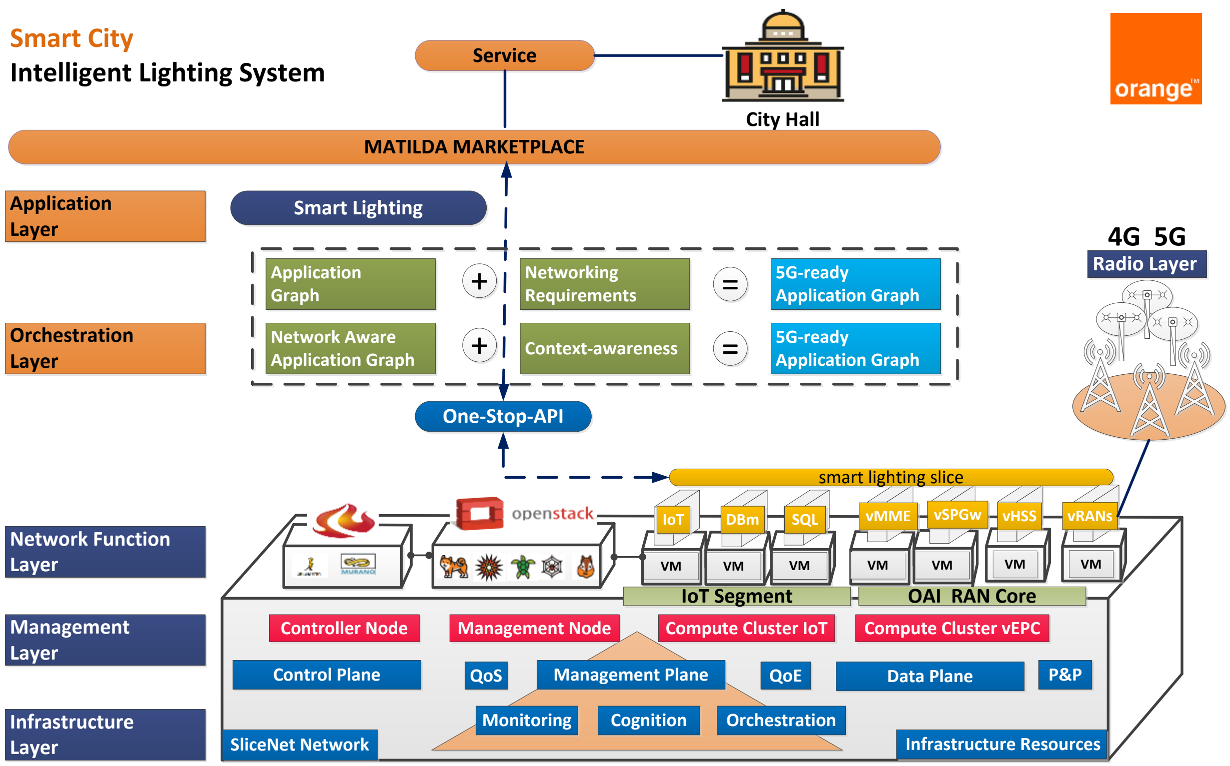

Figure 1: 5GPPP MATILDA SLICENET Collaboration.

Performance, and Security) framework, which provides QoS and QoE service assurance, analytics, Artificial Intelligence/Machine Learning (AI/ML) and enhanced control [SLICENET–D.3.1]. MATILDA defines the 5G marketplace, from where the applications/services are instantiated, together with the application orchestration layer, where the requirements of the application are abstracted towards the 5G network infrastructure as a Slice Intent metamodel [MATILDA-D.1.1]. The point of interconnection among the two projects is represented by the Open Stop API, defined by SLICENET [SLICENET–D.3.1], which receives the Slice Intent meta model and orchestrates its demands/requirements over the 5G network architecture.

In addition, the Smart City use case capabilities are expected to be extended with specific components developed in SliceNet, such as P&P function, QoE optimizer, QoS CP control, monitoring metrics, sensing and actuation through cognition, as a collaborative activity between the two 5G-PPP research projects.

4.1.2 Test bed architecture

The Orange Smart City test bed architecture is composed of the four main components depicted in Figure 2:

1. RAN and Core network part
2. IoT Aggregator (IoT connector) platform
3. IoT Middleware (Enterprise) part, including IoT connectivity
4. Resource and Service Orchestration.

The RAN part is composed of:

- UE/devices: Lighting poles/lamps and smart sensors with different radio type capabilities (LoRA/LTE/LTE-M/Nb-IoT)
  - The test bed provides also the smooth transition from LoRa approach to LTE/LTE-M and Nb-IoT access network
- OpenAirInterface (OAI) solution [OpenAirInterface]:
  - E-UTRAN: USRP x310, vRAN.

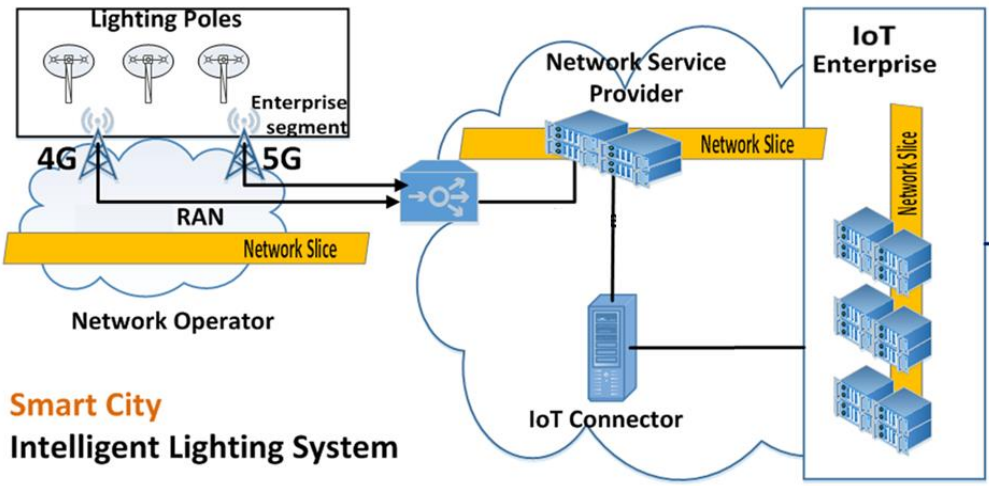

Figure 2: ORO test bed infrastructure.

OpenAirInterface is an open-source software-based implementation of 3GPP LTE Release 8/Release 9 including features from LTE-Advanced (Rel 10 to 14) and 5G under development Release. Its capabilities span across the full 3GPPP protocol stack: E-UTRAN (eNB, UE), EPC (MME, S+P-GW, HSS), and it provides real-time RF and scalable emulation platforms. It is compatible with the available Software Defined Radio [SDR] platforms (ExpressMIMO2, USRP, Lime SDR), and allows for the deployment of a fully-compliant 4G eNodeB and EPC in a x86-based computer or into a more enriched capacity data centre.

The OAI fully virtualized features are supported by the 3GPP implemented architecture software stack, as depicted in Figure 3.

USRP X310 [USRP X310] is a high-performance, scalable software defined radio (SDR) platform that combines two extended-bandwidth daughterboard slots, covering DC – 6 GHz with up to 160 MHz of baseband bandwidth, multiple high-speed interface options (PCIe, dual 10 GigE, dual 1 GigE). The open source software architecture of X310 provides cross-platform UHD driver support, making it compatible with a large number of supported development frameworks, reference architectures, and open source projects.

In the Smart City test bed architecture, the eNodeB functionality will be deployed through the USRP X310 or a Huawei commercial eNodeB to connect the lighting pools over the LTE network and the LTE-M/Nb-IoT to the IoT Enterprise infrastructure. The eNodeB will use the virtual Evolved Packet Core network which is hosted on one of the Orange test bed virtualized infrastructures.

The Core solution- vEPC (vMME, vHSS, vSPGW) [OpenAirInterface] instances deployed and up & running – provides the connection of the lighting sensors through RAN components to the Enterprise Infrastructure where the ThingsBoard IoT [ThingsBoard] middleware

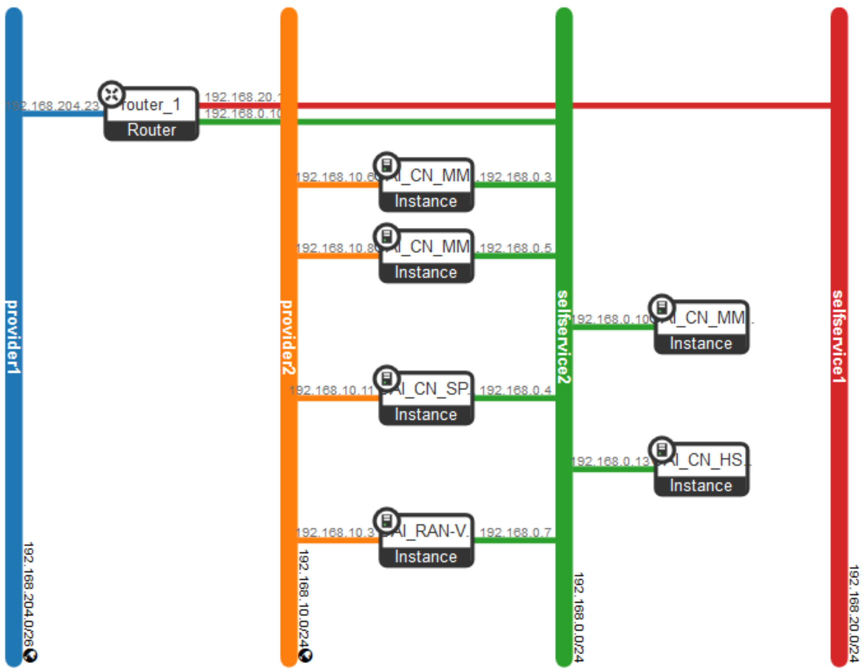

Figure 4: OpenStack logical infrastructure instantiation.

component and other applications are deployed.

The deployed logical infrastructure and functions (VNFs) through the OpenStack are instantiated as presented in Figure 4, with the specification that the orange self-service network provides local connectivity within the same domain between virtualized 4G network (eNB, vEPC), the green self-service provides global connectivity for vEPC to internet (SGi interface) as well as access to the vHSS (S6a interface), and the red self-service connects the IoT platform and other apps as micro-services apps to the global network.

Networks: Green network for S11; S6a; S5S8, SGi interfaces (MME/SP-GW/HSS), also provided internet access to the PGW; Orange network for S1-C/S1-U, connecting also the OAI RAN and the USRP into a L2 domain, including physical switches.

The IoT Aggregator platform consists of:

- Load balancer BigIP – Cisco F5 Load-Balancer BigIP [Big-IP F5]
- IoT connector platform [FLASHNET].

The IoT connector platform, key component in the IoT solution, provides the secure transmission of the raw data information from the device sensors to the IoT Enterprise application (IoT platform, dashboard, command and control). The IoT connector platform integrates a two-way communication between IoT Enterprise platform apps and the devices sensors; assuring the interoperability and scalability and simplified communication, it distributes the messages of different services and exposes multiple IoT protocols. The IoT connector provides connectivity from the use-case IoT devices to the network elements deployed in the infrastructure. It is a fixed component which is not intended to be instantiated or dynamically provisioned and orchestrated.

The explicit IoT connector roles and functionality, linked to the Apps, are in the architecture described in Figure 5, based on [SLICENET-D.3.1], including series of exposed REST API functions, as:

- Device management
- API Key
- Queues management
- Alerts
- Web

The software package is installed on a virtual machine with specific resources, supporting several IoT communication protocols, as MqTT [MQTT], CoAP [COAP], REST [REST], LoRa

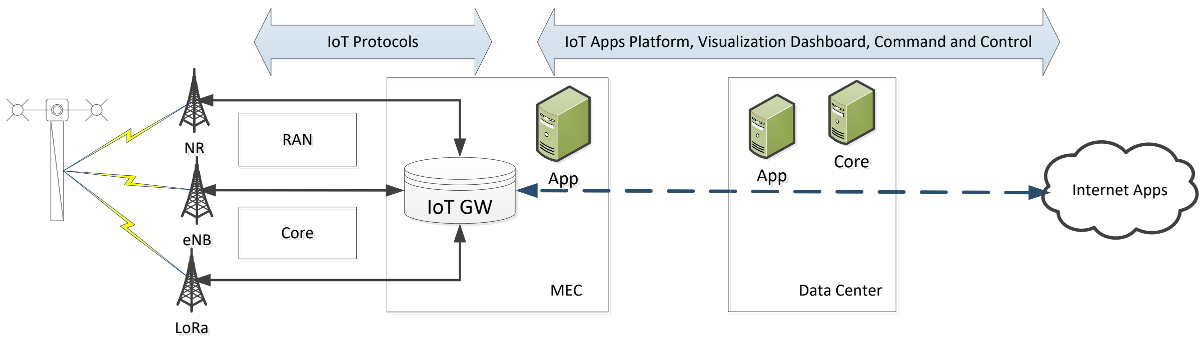

Figure 5: IoT connector architecture integration.

[LORA], using a machine with:

- CPU: 8 Cores
- RAM: 8GB
- Storage: 500GB (databased used for caching and indexing)
- IoT Middleware / IoT Enterprise is composed by several servers with the role of demonstrator components hosting.
- Resource and Service Orchestration.

The entire resource system test bed is orchestrated by the Heat project [HEAT], with support of templates that are under the developing phase. The Heat orchestrator may interact with OpenStack Telemetry [OST] services to provide resources auto scaling.

The implementation of Open Source Mano (OSM) [OSM], a quality open source Management and Orchestration stack in line with ETSI NFV, is under discussion. OSM provides NFV management and orchestration, covers the life-cycle-management (design, run) and provides plugins for integrating with OpenStack.

The services creation, as an answer for the vertical's communication request, requires the integration and coexistence of several developments and innovation functions, as One-Stop-API (OSA), Market Place and Services Orchestrator [MATILDA-D.1.1]. The OSA is the vertical enabler for designing and provisioning of the services through a pool of selectable features by allowing the proper selection of applications characteristics [SLICENET-D.2.2]. The OSA is expected to be integrated with the Marketplace.

4.1.3 Test bed HW infrastructure

The entirely virtualized network elements are deployed over a compute cluster, hosted on a dedicated HPE server:

- ProLiant DL360 Gen10(1) [HP DL360] composed of:2 processors, 12 core/processors, 512GB RAM, 2.4 TB HD, Network adapter 1Gbps/2ports, Ubuntu 16.04 Server LTS
- HP Servers - ProLiant DL380 Gen9(3) [HP DL380] composed of: 2 processors, 12 core/processors, 128GB RAM, 600/1.2T GB HDD, Network adapter 1Gbps/6ports, Ubuntu 16.04 Server LTS.

The deployment scenario for physical resources includes 2 Cisco Switches NX-OS 9K [NEXUS 9K] used only for networking purpose and a network segment (Fortinet appliance [Fortinet Appliance]) that provides secured management access to the infrastructure and internet accessibility through a router Cisco ASR9k [ASR 9K] (SGi interface) for service consumers.

4.1.4 Test bed virtualized infrastructure

The test bed virtualized infrastructure is depicted in Figure 6. The deployed virtual servers act as controlling node (h2020-server1), orchestration and management node (h2020-server2) and the last two (h2020-server3 and h2020-server4) are computing nodes hosting all the presented VMs for the NFV/VNF deployment, including all the developments linked to the end-to-end scenario and the virtual environment of the Thingboards platform:

- h2020-server 1 - Controller node containing VNFO (Heat), VNFM (Murano [MURANO] – Ceilometer [CEILOMETER]) and Tacker module (Tacker [TACKER]) with addressing role to the orchestration module
- h2020-server 2 – OpenStack Management node containing VIM (Nova [NOVA] – Neutron [NEUTRON] – Keystone [KEYSTONE] – Glance [GLANCE]) and OSS/BSS (Horizon [HORIZON])
- h2020-server 3 - computing node using KVM [KVM] virtualization capacity to instantiate several virtual machines for all Smart City application components, having the role of processing and storage of the information provided by Smart City sensors and actuators
- h2020-server4 is a computing note hosting the vEPC.

4.1.5 OpenStack Architecture Overview

The NFV/VNF implementation deployed in Orange use case is based on the ETSI MANO architectural framework for virtualization (Figure 7). The Orange test bed virtualization implementation is using an open source framework to control the pool of compute, storage and networking resources, OpenStack based on Ubuntu implementation, high level presented in Figure 8 and OpenStack [OpenStack]components detailed on the following tables.

The OpenStack implementation provides to the use case a private cloud service model, as Infrastructure as a Service, where the required use case platforms and network elements are deployed, then through the SDN controller [SDN controller] a communication service is created and delivered to the consumers. The service is then exposed to the consumer, through the scalable platform, as SaaS (Software as a Service [SaaS]).

OpenStack software infrastructure is depicted on each of the 4 servers in Table 4.

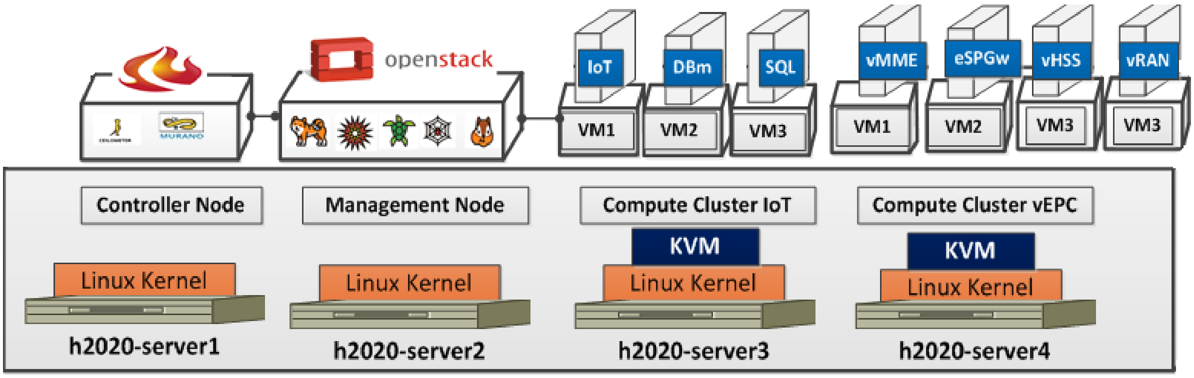

Figure 6: Orange Test bed virtualized infrastructure.

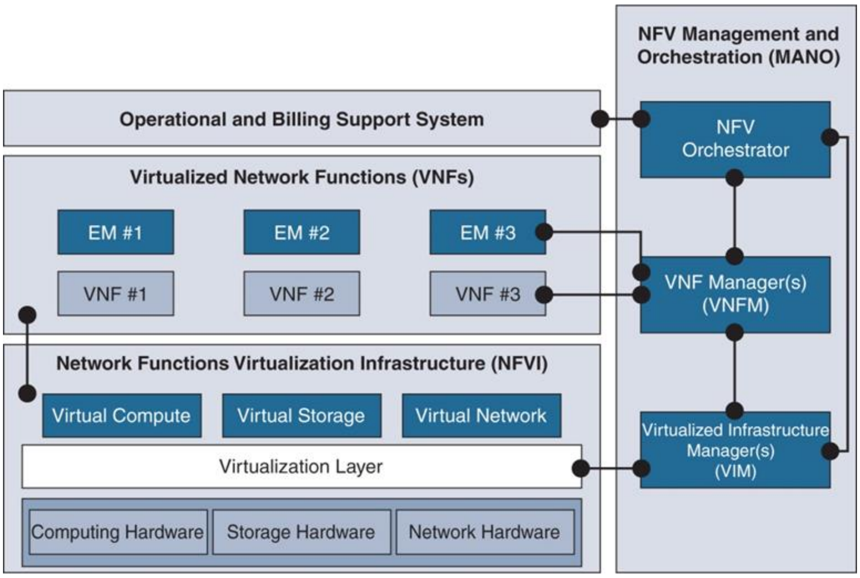

Figure 7: OpenStack architecture.

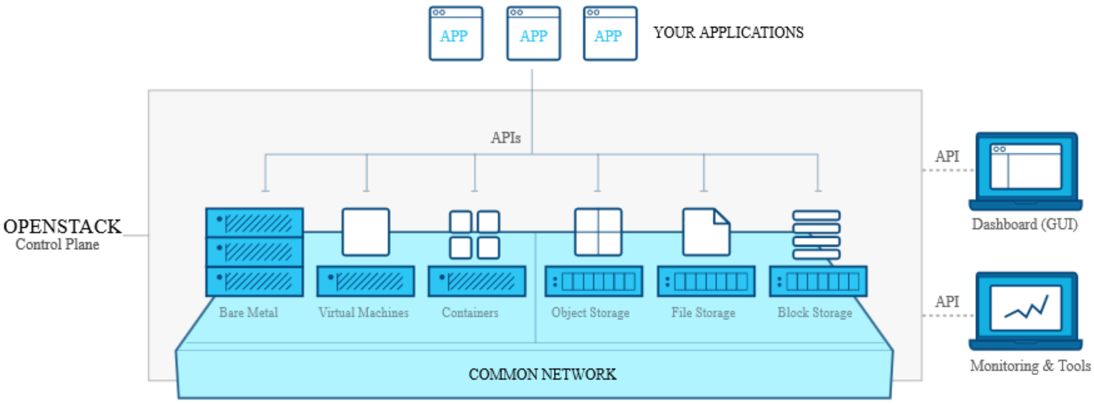

Figure 8: OpenStack components.

Table 4: Servers' software package and version.

h2020-server1 Controller Node		
Software Module	**Software package**	**Version**
Ceilometer	ceilometer-agent-notification	8.1.4
	ceilometer-agent-compute	8.1.4
	ceilometer-collector	8.1.4
Murano	murano-api	3.2.0
	murano-engine	3.2.0

h2020-server1 Controller Node		
Software Module	**Software package**	**Version**
Nova	nova-api	15.1.0
	nova-conductor	15.1.0
	nova-consoleauth	15.1.0
	nova-novncproxy	15.1.0
	nova-scheduler	15.1.0
Cinder	cinder-scheduler	10.0.6
Neutron	neutron-dhcp-agent	10.0.5
	neutron-l3-agent	10.0.5
	neutron-linuxbridge-agent	10.0.5
	neutron-linuxbridge-cleanup	10.0.5
	neutron-metadata-agent	10.0.5
	neutron-server	10.0.5
Glance	glance-api	14.0.1
	glance-registry	14.0.1
Horizon		11.0.0
h2020-server3 - Compute cluster IoT		
Software Module	**Software package**	**Version**
Nova	nova-compute	15.1.0
	nova-manage	15.1.0
Cinder	cinder-volume	10.0.6
Neutron	neutron-linuxbridge-agent	10.0.5
	neutron-linuxbridge-cleanup	10.0.5
Ceilometer	ceilometer-send-sample	8.1.4
	ceilometer-agent-compute	8.1.4
KVM	QEMU emulator	2.8.0

h2020-server4 Compute cluster vEPC		
Software Module	**Software package**	**Version**
Nova	nova-compute	15.1.0
	nova-manage	15.1.0
Cinder	cinder-volume	10.0.6
Neutron	neutron-linuxbridge-agent	10.0.5
	neutron-linuxbridge-cleanup	10.0.5
Ceilometer	ceilometer-send-sample	8.1.4
	ceilometer-agent-compute	8.1.4
KVM	QEMU emulator	2.8.0

4.1.6 Test Bed Network Topology

The test bed interfaces are composed of four connectivity layers:

- physical layer:
  - 2 routers (acting as gateways for demarcation between WAN networks and Core infrastructure)
  - 2 aggregation switches (for future planned expansions)
  - 2 ToR switches [TOR] used for servers' connectivity
- underlay network layer - based on VLAN and created during the infrastructure provisioning
- overlay networks - containing tunnelling mechanisms (VXLAN / MPLSoUDP [MPLSoUDP]) used for tenants' traffic isolation
- the administrative subnet used for remote management access to all nodes of the infrastructure.

4.1.7 Test Bed Monitoring Capabilities

The monitoring functionalities are responsible for collecting and processing information provided by the Telco provider that will be further used by the MATILDA Application Orchestrator to understand the performance and usage of the network instantiated slice and associated services. It gathers information, such as counters, events and/or alarms, from the physical and virtual network deployed resources, slices and services. Monitoring capabilities:

- Service monitoring - end to end service monitoring, Smart Lighting application monitoring
- Slice monitoring - monitoring parameters, data specific to the Smart Lighting slice, including real time monitoring of sliced network activity and health, specific for slice components, outputs to be exposed.

- Traffic monitoring monitor the consumed user data traffic inside the slice (e.g., resources consumed in OAI-RAN smart lighting slice)
- Topology monitoring – provides monitoring information of the slice topology, dynamically discovered or from templates
- VNF & PNF monitoring (keep alive mechanism)
- QoS metrics and Resources metrics.

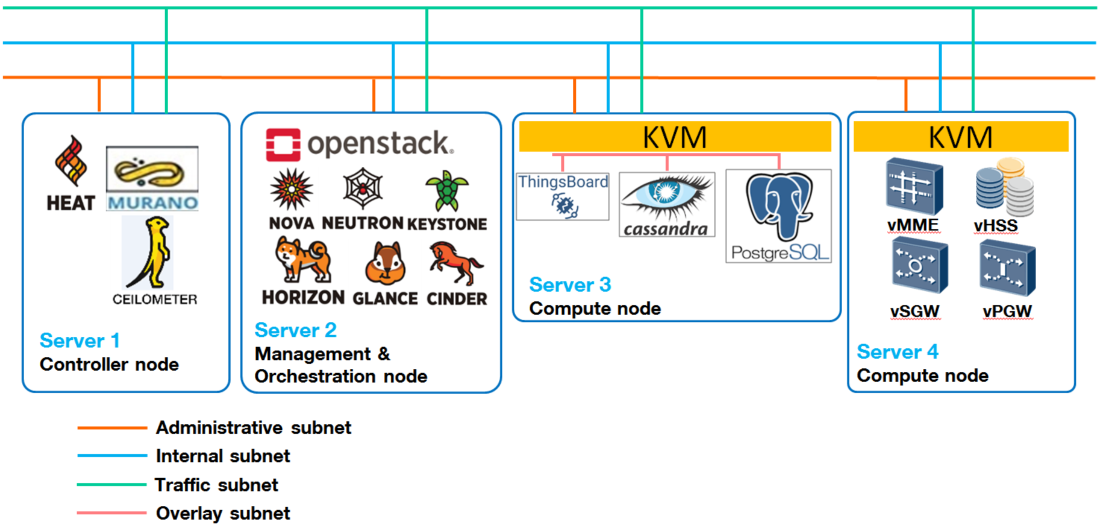

Figure 9: Smart City test bed connectivity layers.

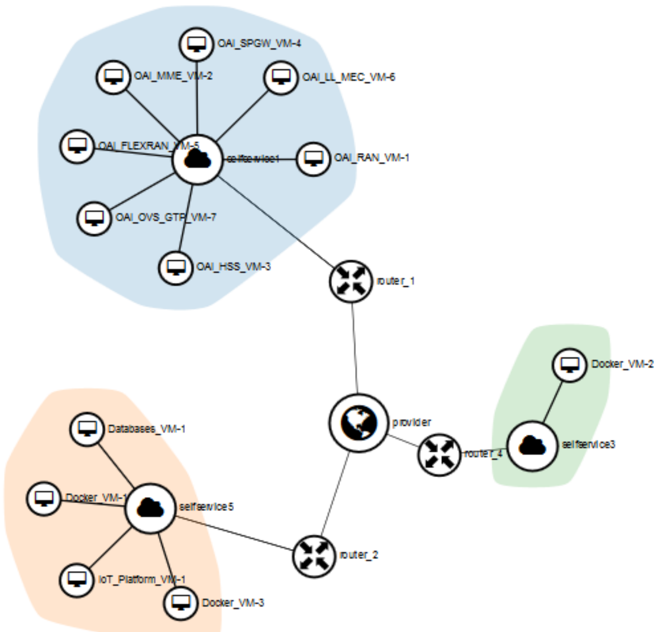

Figure 10: Smart City high level network topology.

4.2 UNIVBRIS Test Bed

University of Bristol's 5G test bed comprises several networking and computing technologies, interconnecting a significant area that expands across Millennium Square (MS), We The Curious (WTC) and the Smart Internet Lab (HPN) in the city of Bristol (United Kingdom). While most of the radio access network is available in MS and WTC, computing resources are located in HPN and WTC. This test bed combines existing technologies and innovations in development to allow the exploration and validation of the 5G ecosystem.

4.2.1 Test Bed Architecture

Figure 11 shows a geographical representation of the multiple access technologies deployed within MS and WTC. Connectivity terminates via fibre optic at distinct points in the square with onward links through the use of fixed wireless access mmWave radios. To allow for future expansion, termination locations have been over engineered with enough installed fibre and power capacity to allow the next generation of 5G connectivity. Space and power for mobile edge computing (MEC) has been provisioned at key locations in MS, HPN and VIRTUS to allow the deployment of virtual network functions (VNF) and low latency real-time application processing close to the end user.

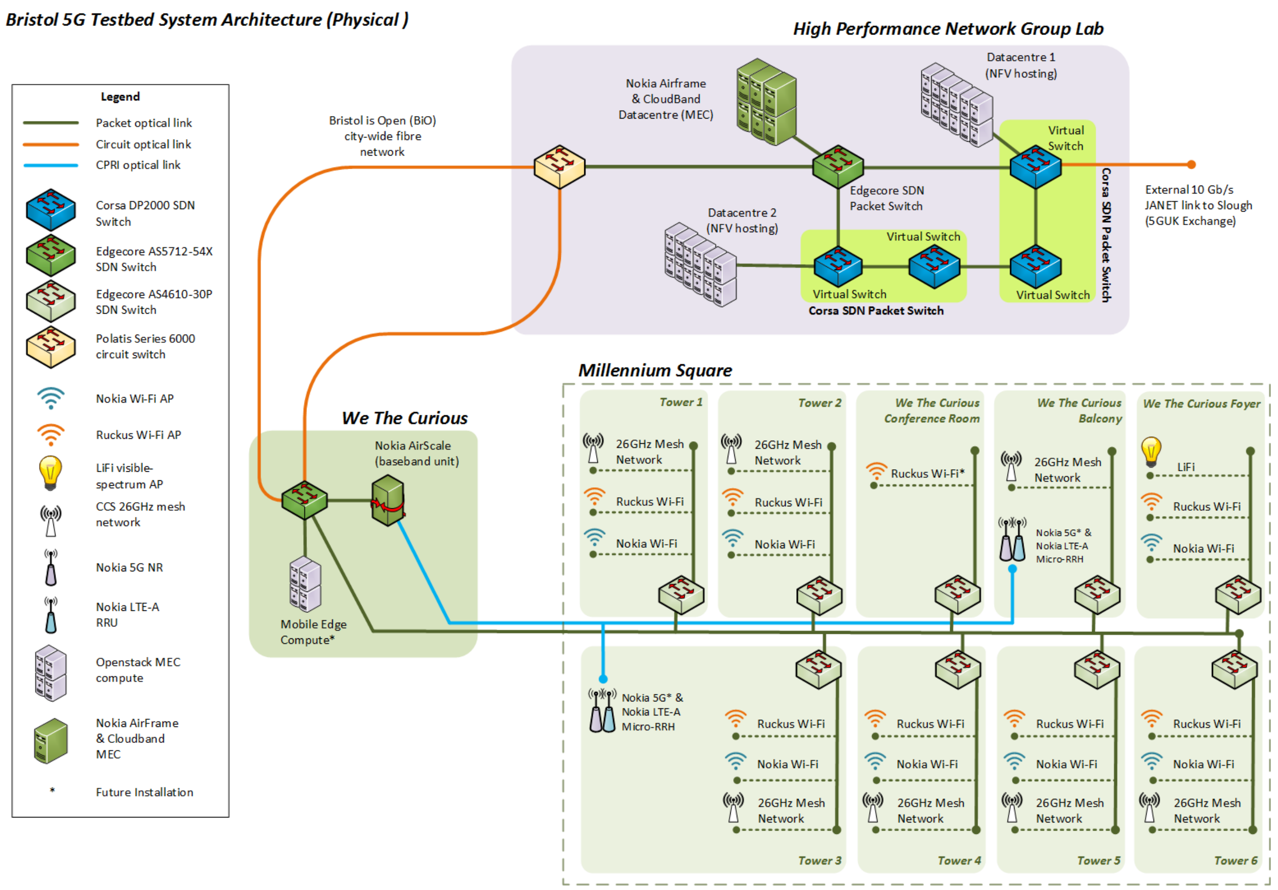

Figure 11: Top level architecture of equipment deployed in the test bed infrastructure.

4.2.2 Test bed HW infrastructure

The University of Bristol test bed infrastructure is composed of the following elements:

- Computing Nodes
  - 3x 24 Core x 2.4GHz, 64GB RAM, 400SSD 1000HDD, 2 x GbE, 8x PCIe
- Control Plane Switches
  - 10x Netgear M4300-24X24F
  - 3x Netgear GS110TP
- Data Plane Switches
  - 13x Edgecore AS4610-30P
  - 3x Edgecore AS5712-54X
- mmWave SDN mesh
  - 8x CCS Metnet 26GHz mmWave mesh node
- Radio Access
  - 1x Nokia Airscale LTE Basestation
    - 2x Nokia RRH
  - 1x Nokia LTE Pico BTS
  - 8x Nokia AC400 outdoor Wi-Fi AP
  - 7x Ruckus T710 outdoor Wi-Fi AP
  - 3x Ruckus R720 Indoor AP.

4.2.3 Test Bed Virtualized Infrastructure

To accommodate varying requirements, the test bed can provide different levels of virtualized infrastructure.

At the most basic level, a project can be given access to a quota of resources and functionality on a shared OpenStack datacentre [OpenStack]. This level of access will allow an experimenter to access a shared management network with limited scope to deploy personal network elements inside, and either an OpenStack web UI or CLI interface from which they can deploy VMs. Each user's OpenStack quota is private, virtual infrastructure is isolated from other researchers, and can enjoy access to their own private data-path network slice.

The current deployment at this level, as shown in Figure 12, consists of the following:

- two different NFV orchestration and management solutions:
  - Open Source MANO release THREE (open source) [OSM-THREE]
  - NOKIA CloudBand (proprietary based on a version of OSM and OpenStack, providing network slicing and virtualisation in rapid service creation) Available December 2018 [CloudBand]

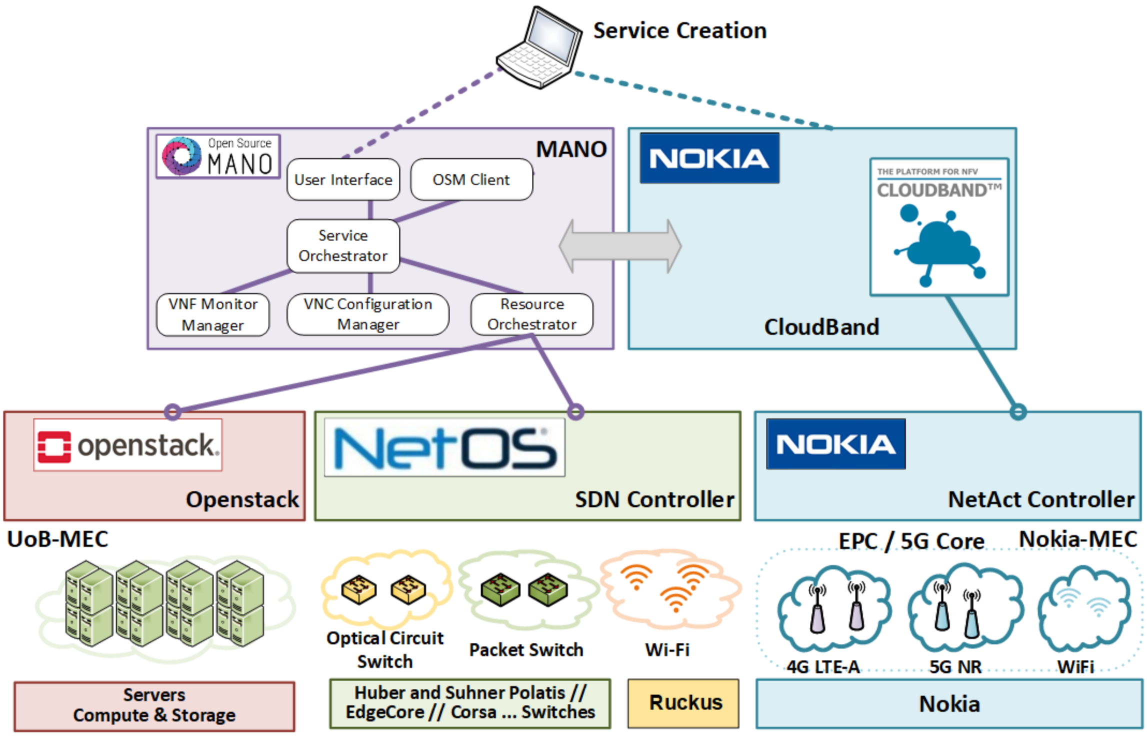

Figure 12: Software used for management and orchestration of the test bed resources.

- two cloud/edge computing solutions:
  - Openstack Queens (opensource) [OpenStack-QUEENS]
  - Nokia MEC (proprietary) [NokiaMEC]
- one SDN controller responsible for providing connectivity:
  - NetOS [NetOS] (proprietary, based on the Open Daylight [ODL] opensource).

When a project's requirements are more advanced, a hosting arrangement can be made where the test bed deploys layers of cloud infrastructure according to the requirements of the project. This can start as a bare-metal stand-alone deployment of an OpenStack environment, or an Open Source MANO orchestrator driving an OpenStack VIM. This mode of deployment allows projects to have full control over their cloud environment, including the use of GPUs, NICs and FPGA hardware, software versions and configuration and management network access.

4.2.4 Test Bed Network Topology

University of Bristol test bed infrastructure High speed, low latency connectivity is provided between HPN and WTC through the use of dedicated fibre and local compute resource. Multiple locations in MS are directly connected with HPN providing access to radio resources for use-case delivery. Figure 13and Figure 14 show the physical locations in MS connected to the 5GUK test bed.

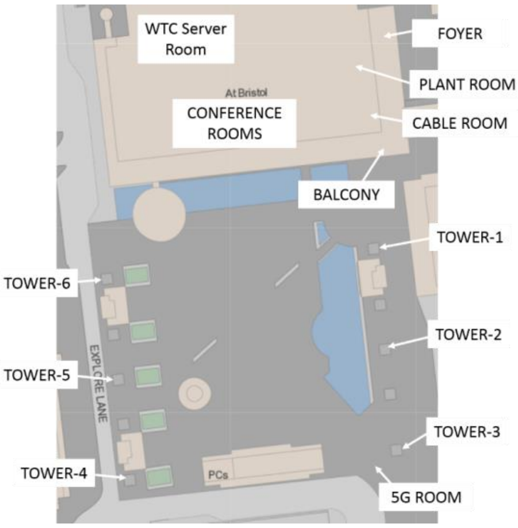

Figure 13: Map of locations at Millennium Square and We The Curious.

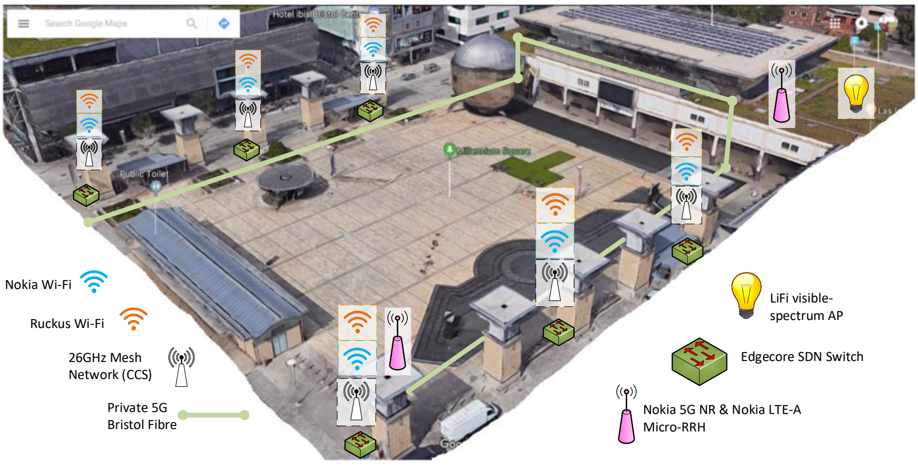

Figure 14: Aerial view of Millennium Square and We The Curious, depicting network infrastructure.

4.2.5 Test bed monitoring capabilities

University of Bristol test bed current deployment is monitored using the respective Controller or Network Management System of each subsystem individually. The tools available for the network administrator to monitor the network performance and status of its operation are:

- Ruckus Wireless Controller
- CCS EMS mmWave
- Nokia EPC NMS, Nokia AirScale NMS and Nokia Flexizone NMS.
    - Nokia NetAct is in the progress of integration to provide NMS for the complete monitoring of Nokia's subsystems.
- PureLiFi EMS
- NetOs.

4.3 CNIT Test Bed

The CNIT test bed is the integration of the CNIT, UBITECH and COSM infrastructures. The purpose of the test bed is to provide a managed infrastructure for the deployment and execution of industrial, vertical 5G-ready applications.

The test bed is composed of a set of high-performance equipment and several latest generation programmable radio, computing and networking technologies.

4.3.1 Test Bed Architecture

The CNIT test bed architecture is composed of two main parts: (*i*) the radio access part and (*ii*) the virtualized infrastructure part.

- The radio access part is composed of:
    - 3 Amarisoft eNodeBs (Intel NUC i7-32 GB + NI USRP B210 board) with support for NB-IoT.
    - 4/5G User Equipment (UEs) including tablets, handled devices, residential gateway boxes (FRITZBOX 6820 LTE), and SODAQ NB-IoT sensors.
- The virtualized infrastructure is composed of a set of high-performance pieces of equipment supporting OpenFlow, SDN [ONF] and the OpenStack project [OpenStack] as depicted in Figure 15.

4.3.2 Test Bed HW Infrastructure

The CNIT test bed infrastructure is composed of:

- A set of servers for the compute nodes
    - 6x blade servers Intel KP-S2600KPR (3x Xeon E5-2630v4 Q1'16/3x E5-2660v4 Q1'16, 2.2/2.0Ghz, 20/28 cores, 64/128GB RAM).
    - 1x Supermicro server (2x Xeon E5-2630v2 Q3'13, 2.6GHz, 12 cores, 32GB RAM).
- A set of servers for the controller nodes
    - 3x server HPE DL360 Gen9 (1x Xeon E5-2620 v4 Q1'16, 2.1GHz, 8 cores, 64GB RAM). (These servers will be probably upgraded to double the number of cores and the RAM). One of these servers hosts the controller nodes, network nodes and block storage of the OpenStack instances.
- High performance Ethernet switches with OpenFlow and SDN support
    - 1x Pica8 Pronto 3920 48 Port 10 GE with 4 Port 40 GE.
    - 2x Pica8 Pronto 3290 48GE with 4 Port 10 GE.
    - 1x Mellanox SX6036 36 Port 40 GE, with 10GbE adapters. (This switch connects the 3 datacenters).
    - 1x HP 2920-48G.
- Network adapters for server connectivity
    - 6/12x Intel X710, dual port, 10 GbE.
    - 3x Intel x520, dual port, 10 GbE .
    - 2x Mellanox ConnectX-2, dual port, 10 GbE.
    - 2x Mellanox ConnectX-3, dual port, 40 GbE.

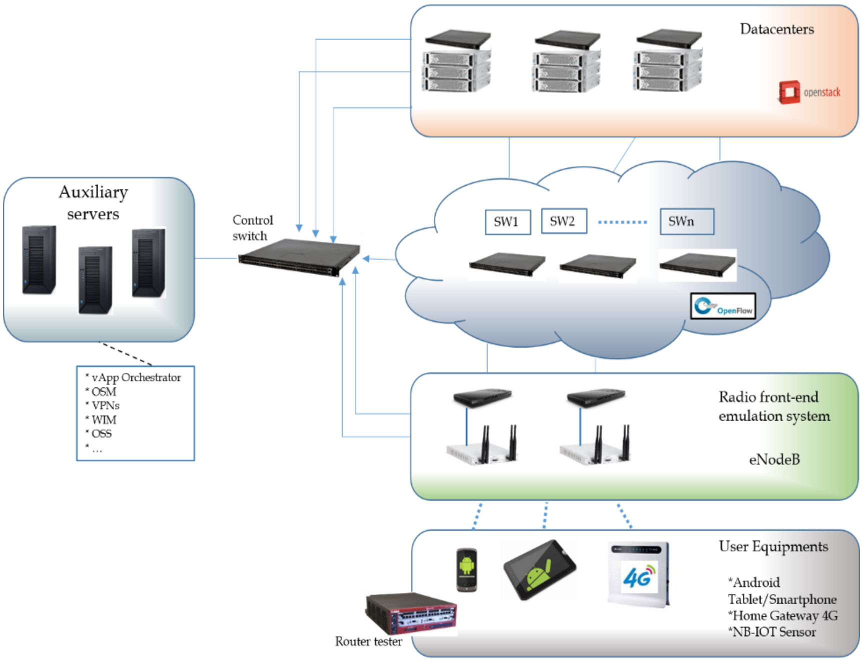

Figure 15: CNIT test bed physical architecture.

4.3.3 Test Bed Virtualized Infrastructure

The virtualized infrastructure is composed of three OpenStack instances. Each OS instance contains a controller node, a network node, a block storage and at least two compute nodes.

The version of the OpenStack instances is Queen, and the modules installed on the different components of the instances are listed below:

Controller Node

- Keystone
- Glance
- Nova
  - nova-api, nova-conductor, nova-consoleauth, nova-novncproxy, nova-scheduler, nova-placement-api
- Neutron
  - neutron-server
- Horizon
- Cinder
  - cinder-api, cinder-scheduler
- MariaDB
  - mariadb-server, python-mysql

Network node

- Neutron
  - neutron-openvswitch-agent, neutron-dhcp-agent, neutron-l3-agent, neutron-metadata-agent

Compute node

- Nova
  - nova-compute
- Neutron
  - neutron-openvswitch-agent, neutron-sriov-nic-agent

Block Storage node

- Cinder
  - cinder-scheduler

4.3.4 Test Bed Network Topology

The test bed network topology is divided into three sections as shown in Figure 16. The main feature of the test bed is its ability to adapt to the needs of the network administrator and, in particular, the requirements of the industry vertical use cases to be implemented in the project.

The OpenStack section can host a variable number of OpenStack instances (for the moment, the test bed counts with three OS instances). Each OS instance is composed of three VMs for the control nodes (as described in Section 4.3.3), and they can support both self-service networks (VxLAN, VLAN and SR-IoV tpes) and provider networks (VLAN and Flat types).

The OpenStack instances are connected to the OpenFlow switches that emulate the Wide Area Network (WAN). The physical switches can be virtualized into a variable number of virtual switches in order to create a meaningful network for the project (the precise topology

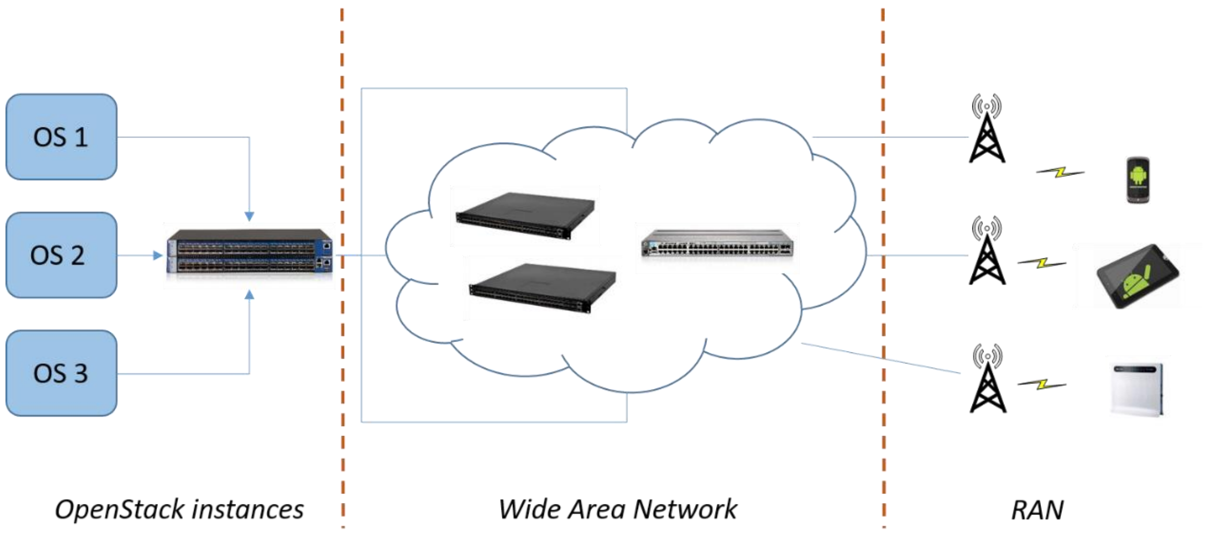

Figure 16: CNIT test bed network topology.

for the MATILDA test bed is still under study/development at the time of writing). The eNodeB, representing the Radio Access Network (RAN) section, will be connected to this network.

4.3.1 Test bed monitoring capabilities

The test bed is able to expose the SNMP (Simple Network Management Protocol) MIBs of the network devices that support it.

4.4 Demonstrator 1: 5GPACE - High Resolution Media on Demand & Banking on the Cloud

5GPACE is a 5G Application that provides high value, innovative services to end users during crowded events. The system combines the i-EVS framework by Italtel, for immersive video services, and the Incelligent system, which can provide personalized recommendations to end users in real time [JMM09], from the analysis of historical and mobility data using machine learning techniques.

The scenario in play involves a user moving around a crowded venue, such as a shopping mall or an open-air market area, sharing high quality video with her peers while making purchases around the brick and mortar shops that are part of this venue, as shown in Figure 17.

The user, through her mobile device, can discover personalized recommendations and offers, based on her-his exact current location, and her-his preference to consume certain types of products or services (possibly in sequences or bundles). Such data-driven retail recommendations come in the form of a "**Move to next shop that offers an A% discount just for you**" visual aid on the user's mobile screen, and are created by applying advanced machine learning methods on the user's purchase history and mobility data, in user-perceived real time.

In 5GPACE, the Italtel i-EVS system provides User and Group Database Management, High Resolution Media Processing, Content Storage and Geo-localization functionalities.

The Incelligent system for personalized recommendations includes a series of advanced Machine Learning methods, and incorporates various aspects of the Customer Relationship Management modelling. These intelligent methodologies involve data pre-processing,

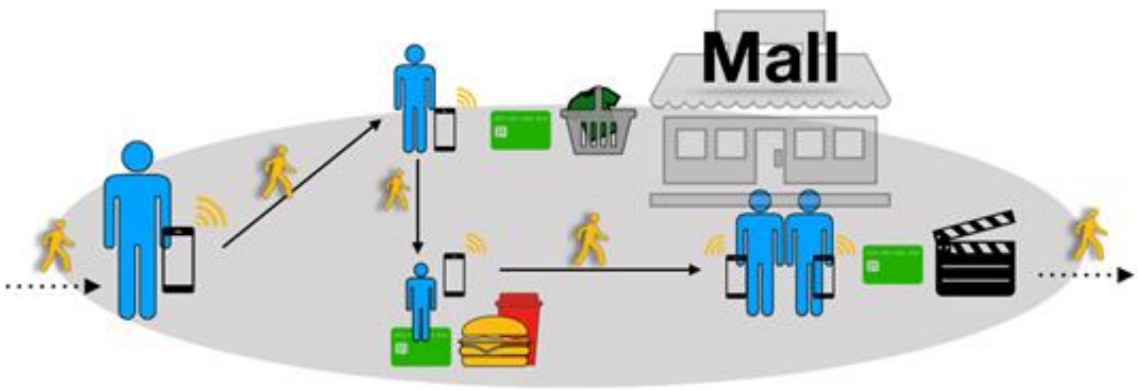

Figure 17: A user journey inside a shopping mall area. The user receives high quality media content and personalized recommendations as she moves around the shops.

dimensionality reduction, feature selection, advanced customer segmentation and profiling, prediction modelling for customer propensities to buy and near real-time recommendations at the network edge.

4.4.1 Tested Scenarios

Table 5 summarizes a possible list of the main functional tests related to 5GPACE. Such tests are briefly presented and discussed as an example of the approach that should be adopted for 5GPACE, and do not represent the exhaustive description of the testing procedure adopted to fully validate the framework.

Table 5: 5GPACE Functional Tests.

System Under test	Operation Under Test	Role of Testing Operator
MATILDA Orchestrator	On-boarding Instantiation Termination	Application Developer Service Provider Service Provider
5GPACE	User creation User Registration Group Creation Content Uploading/Downloading Real Time Video Streaming Real Time Video Transcoding and Streaming Geo-localization Retail recommendation transmission	End User

In Table 5, the first column highlights the main functional part of the system under test. The second column indicates the specific tested operation, while in the third column we report the role played by the tester in the considered use case.

The first three tests in Table 5 are mainly related to the behaviour of the MATILDA orchestrator, and in particular to the management functions offered to Application Developers and Service Providers. They analyse the capability to on-board a 5G-ready App by a Developer, and the possibility to instantiate and terminate it by a Service Provider.

The functional tests performed on 5GPACE aim at verifying the complete availability of the service. Such functions, in fact, can be provided to end users only after a successful on-boarding and instantiation carried out by the MATILDA orchestrator.

4.4.2 First Demonstrator Release

Features

In the first demonstrator release, the following features will be presented:

Table 6: 5GPACE demonstrator features (first release).

Feature name	Feature details
Application Provisioning and Deployment	Provisioning and deployment of the 5GPACE application through the MATILDA orchestrator (whole application graph).
Service Orchestration	Context and policy-based application orchestration.
Service Scaling	Horizontal scaling of the 5GPACE application components.
5G Slice Provisioning	Slice intent request/reply mechanism with 5G network slice provisioning.
5GPACE Application	5GPACE is deployed as an application component providing basic video sharing capabilities and user/group management.

Test Bed Requirements

For the first demonstrator release, the test bed should satisfy the following requirements:

Table 7: Test bed requirements for 5GPACE demonstrator (first release).

Requirement name	Requirement details
Cloud Hosting Environment	OpenStack-based cloud hosting services.
Network Slice Provisioning	Programmable 4G/5G mobile network that provides dedicated mobile network slice.
Service Orchestration	MATILDA Orchestrator must be connected to the OpenStack cloud, which acts as an IaaS provider.

Acceptance Criteria

The following table provides basic target values for the first demonstrator release:

Table 8: Acceptance criteria for 5GPACE Demonstrator (first release).

Metric	Target value	Baseline value
Time needed to deploy an individual component of an application graph.	< 2min	3 min
Time needed to deploy the entire application graph.	< 5 minutes	15 minutes
Time required to adapt to increased or decreased load.	< 1 min	3 min
Time needed to provision the 5G network slice.	< 3 min	Not possible at this time.

4.4.3 Second Demonstrator Release

Features

In the second demonstrator release, the following features will be presented:

Table 9: 5GPACE demonstrator features (second release).

Feature name	Feature details
High Availability at the application level	Scaling and redundancy mechanisms at the application component level.
High Availability at the network level	Network-based failover mechanisms.
5G Slice Orchestration	Network context (i.e., slice intent) and policy-based application orchestration.
VNF Orchestration	VNF provisioning and orchestration (e.g. router, VPN, FW…).
SLA Monitoring	Performance parameters are acquired from 5GPACE and the VIM, to verify SLA

Test Bed Requirements

For the second demonstrator release, the test bed should satisfy the following requirements:

Table 10: Test bed requirements for 5GPACE demonstrator (second release).

Requirement name	Requirement details
Cloud Hosting Environment	OpenStack-based cloud hosting services.
Network Slice Provisioning	Programmable 4G/5G mobile network that provides dedicated mobile network slice. Availability of GPU resources in the virtualization infrastructure.
Service Orchestration	MATILDA Orchestrator must be connected to the OpenStack cloud which acts as an IaaS provider.
Network Orchestration	Programmable 4G/5G mobile network that provides dedicated mobile network slice based on the context and policies set by the MATILDA orchestrator.
VNF Orchestration	Open Source MANO connected to the MATILDA Orchestrator and provides VNF-based component orchestration.

Acceptance Criteria

The following table provides basic target values for the second demonstrator release:

Table 11: Acceptance criteria for 5GPACE Demonstrator (second release).

Metric	Target value	Baseline value
Time needed to deploy an individual component of an application graph	< 1min	3 min
Time needed to deploy the entire application graph.	< 2 minutes	15 minutes
Time required to adapt to increased or decreased load.	< 30 sec	3 min
Time needed to provision the 5G network slice.	< 1 min	Not possible at this time.
Time needed to reconfigure the 5G network slice.	< 1 min	Not possible at this time.
Time needed to deploy the VNF-based components (i.e. forwarding graph).	< 2 minutes	15 minutes
Time needed to deploy and provision SLA monitoring.	< 3 minutes	30 minutes

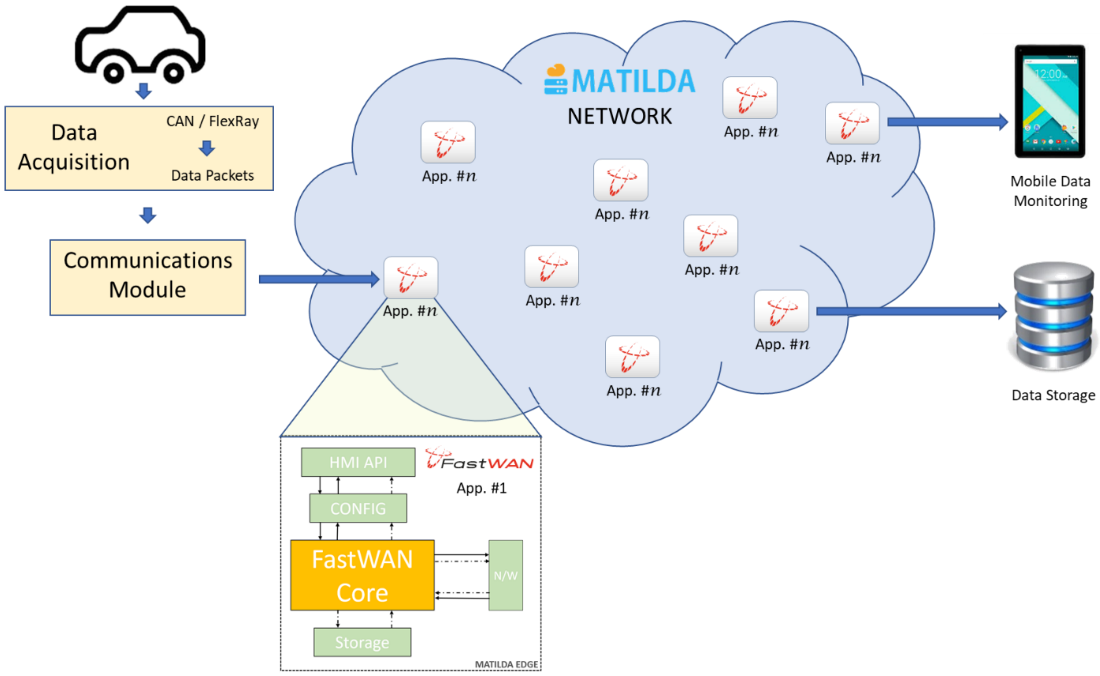

Figure 18: Testing 4.0 - Distributed System Testing.

4.5 Demonstrator 2: Testing 4.0 - Distributed System Testing

Testing 4.0 – Distributed System Testing demonstrates the data acquisition and remote testing of Mobile System Under Test (SUT) units using the FastWAN solution by ExxpertSystems over 5G network. The FastWAN solution consists of the following components:

- High speed data acquisition and abstraction module
- Virtual signal extension and data distribution module
- Efficient data storage system
- Configurable UI for data (e.g. Statistics) visualization of all transmitted data.

4.5.1 Tested Scenarios

The components of the ExxpertSystems FastWAN [FastWAN] product are combined in a scenario where tests performed on a mobile SUT, such as an automobile, will be controlled and monitored from a remote location (Command and Control Center). In addition, the raw data acquired from the test unit will be transferred to the remote location. The scenario is shown in Figure 18.

In this test scenario, raw signal data (e.g. FlexRAY [FlexRay] and/or CAN bus based data) is acquired from the automobile control systems and converted to packetized format in the FastWAN system. This step involves abstracting the data facilitation compression as well as reduced bandwidth usage.

The FastWAN communications module establishes connection with the network, communicates the intent and configuration information for the test and transfers the data packets onto the network. In case of connection failure, the communication module caches the data locally and transfers the cached data on connection re-establishment.

The FastWAN core system performs configuration distribution, network slice negotiation, graph management, Quality of Service (QoS) management and data transfer. Based on the bandwidth and QoS requirements from the test configuration and communications module, network slice will be re-negotiated.

Table 12 lists the possible test cases that will be performed to evaluate the feasibility of distributed system testing on the MATILDA/5G framework.

Table 12: Testing 4.0 Demonstrator functional tests.

System Under test	Operation Under Test	Role of Testing Operator
MATILDA Orchestrator	On-boarding of the application components (FastWAN Core network functions) Application deployment and instantiation High availability at the application level	Application Developer 5G Application SW Developer Service Provider
Testing 4.0 – Data Acquisition and Communication Module	SUT test configuration upload Data Acquisition and Abstraction Data Caching Network connection management	End User Application Developer Application Developer Application Developer
FastWAN Core/5G Network Slice	Slice Intent Request/Reply Dynamic intent provisioning	Application Developer Service provider
Data Storage & Mobile Application (HMI)	End to end data delivery and storage HMI bandwidth negotiation – Change of visualizations options and checking if end to end communication operates without glitches or losses	Application Develope 5G Application SW Developer / Application Developer

In Table 12, the first column highlights the main functional part of the system under test. The second column indicates the specific tested operation, while in the third column we report the Role played by the tester in the considered use case.

The first three tests in Table 12 are mainly related to the behaviour of the MATILDA orchestrator, and in particular to the management functions offered to Application Developers and Service Providers. They analyse the capability to on-board a 5G-ready App by a Developer, and the possibility to instantiate and terminate it by a Service Provider.

4.5.2 First Demonstrator Release

Features

For the first demonstrator release the following features will be presented:

Table 13: Testing 4.0 demonstrator features (first release).

Feature name	Feature details
DATA ACQUISITION	Acquiring the data from the I/O devices for, e.g., FlexRay and/or CAN etc...
COMMUNICATION MODULE – Connection MGMT (Monitoring and Resending)	Establishing and ensuring end to end connection between the source and the destination nodes. If the packets are lost on the network, resending the lost packet based on the acknowledgement.
DATA STORAGE – Implementation Level 1	A simple client on the edge basically representing a distributed storage solution
HMI API / GUI (mobile data monitoring) – Implementation Level 1	A simple client on the edge representing use case for data visualization. Only raw data in real time is displayed without any analytics
Configuration Management-Implementation Level 1	Configuration information including the participating clients and the test acquisition data information to be provided to Network application by the communications module.
Network Management-Implementation Level 1	Network monitoring of activities to support the requirements of the service, identification of the end systems for data transmission and reception.
Data Transmission-Implementation Level 1	Transmission of the data acquired from the I/O devices on to the network.
Protocol Definitions-Implementation Level 1	Basic FastWAN data transfer protocol implementation without any QoS policies

Test Bed Requirements

The demonstrator implementation is based on the Test bed setup provided by UNIVBRIS. This implementation will utilize the following features and infrastructure provided by the test bed setup. These requirements will remain the same for all the phases of implementation.

Table 14: Test bed requirements for Testing 4.0 demonstrator (all releases).

Requirement name	Requirement details
Cloud Hosting Environment	OpenStack-based cloud hosting services.
Network Slice Provisioning	Programmable 4G/5G mobile network that provides dedicated mobile network slice.
Service Orchestration	MATILDA Orchestrator must be connected to the OpenStack cloud which acts as an IaaS provider.

Acceptance Criteria

Narrow acceptance criterion scope is presented here due to the use case of the Testing 4.0. These are not exhaustive but are mandatory and shall remain identical for all phases of implementation.

Table 15: Acceptance criteria for 5GPACE Demonstrator (first release).

Metric	Target value	Baseline value
Time needed to deploy an individual component of an application graph.	< 2min	3 min
Time needed to deploy the entire application graph.	< 5 minutes	15 minutes
Time required to adapt to increased or decreased load.	< 1 min	3 min
Time needed to provision the 5G network slice.	< 3 min	Not possible at this time.

4.5.3 Second Demonstrator Release

Features

In the second release the following advancements will be provided:

Table 16: Testing 4.0 demonstrator features (second release).

Feature name	Feature details
Data Abstraction	Development of data abstraction algorithms to reduce the data traffic based on intelligent data pattern recognition.
COMMUNICATION MODULE – Lookup Table for sent IDs	Data communication update based on the Data Abstraction algorithms implementation.
DATA STORAGE- Implementation Level 2	Compressed data storage. Data retrieval mechanisms for serving requests from clients.
HMI API / GUI (mobile data monitoring)- Implementation Level 2	Decompressing the compressed and stored and/or real-time data stored and visualizing the data .
Configuration Management – Implementation Level 2	Adding basic QoS configuration information to the existing configuration in Phase I.
Network Management- Implementation Level 2	Adding network level QoSs and basic network slice brokering.
Data Transmission - Implementation Level 2	Transmission of Data from the I/O will now be done based on the IDs stored in the Lookup tables.
Protocol Definitions - Implementation Level 2	FastWAN data transfer protocol implementation with basic QoS policies such as jitter compensation.
QoSs- Implementations Level 1	Adding QoSs both at the network level and application level.

Test Bed Requirements

The requirements for the test bed are identical for all releases and can be found in Table 14.

Acceptance Criteria

The acceptance criteria is identical for all releases and can be found in Table 15.

4.5.4 Third Demonstration Release

Features

In the final release the following advanced capabilities are provided:

Table 17: Testing 4.0 demonstrator features (third release).

Feature name	Feature details
HMI API/GUI (mobile data monitoring)- Implementation Level 3	Fluid interactive plots for data acquired. The data can also be retrieved from data storage clients on the network.
Configuration Management- Implementation Level 3	Advanced QoSs usage for specific channels of different signal types.
Network Management- Implementation Level 3	Advanced QoSs implementation.
Cache Management	Low level temporary caching of data on the network function application for faster recovery from losses and quickening the responses to the nearest edge system retransmission requests.
QoSs- Implementation Level2	Advanced QoSs implementation and integration.
Protocol Definitions- Implementation Level3	FastWAN data transfer protocol implementation with advanced QoS policies such as loss compensation and bandwidth provisioning.

Test Bed Requirements

The requirements for the test bed are identical for all releases and can be found in Table 14.

Acceptance Criteria

The acceptance criteria is identical for all releases and can be found in Table 15.

4.6 Demonstrator 3: 5G Emergency Infrastructure with SLA Enforcement

5G Emergency Infrastructure with SLA Enforcement is a use case based on the implementation of a 5G-enabled emergency response capabilities provided with the iMON product suite [iMON] (iMON Cloud based dashboard application and iMON mobile application) for real time intervention monitoring, extended with continuous performance monitoring engines of the qMON solution for supporting active and passive SLA monitoring and SLS enforcement. iMON is a product suite designed for use by the first responders and public safety agencies during planned and un-planned interventions. In combination with the iMON mobile application, used by on-site first responder teams, solution provides modular emergency communications capabilities, Common Operational Picture (COP) in real time and a suite of IoT-supported intervention management tools with on-site sensing and tracking capabilities.

5G Emergency Infrastructure with SLA Enforcement scenario targets support of a suite of reliable and survivable services and applications on top of a 5G infrastructure providers for public safety teams both in day-to-day operations and during planned and un-planed extreme situations requiring large on-site intervention teams. In particular, applications and services for on-site intervention monitoring are targeted as well as a series of mobility and location tracking characteristics that can be used in the field during various types and sizes of emergency interventions.

Due to the nature of the public safety operations automation of the iMON system deployment, high availability, resilience and system flexibility are key KPIs that needs to be achieved for the public safety operational use. Therefore, test scenarios need to verify the folowing high-level system operational requirements

- Automated iMON deployment with 5G network slice management;
- Vertical and horizontal scalability of iMON services and application components;
- Strict network QoS assurance of the provided network and compute slice;
- High availability based on multi IaaS services deployment;
- Resilience of the Services and the PPDR system based on MEC deployment scenarios;
- PPDR SLA/SLS enforcement and dynamic resources allocation based on continuous active and passive monitoring.

4.6.1 Tested Scenarios

Table 18 summarizes a high-level list of key functional test scenarios related to the "Emergency Infrastructure with SLA enforcement" demonstration.

Table 18: 5G Emergency Infrastructure with SLA Enforcement Demonstrator functional tests.

System Under test	Operation Under Test	Role of Testing Operator
MATILDA Orchestrator	On-boarding of the application components Application deployment and instantiation (whole application graph) Context and policy-based application orchestration Horizontal/Vertical scaling High Availability at the application level	Infrastructure provider 5G Application SW developer Service provider
iMON Dashboard	User creation Device creation Group creation Report uploading Alarms reporting User tracking	Service provider Service consumer Operator at the command centre End user in the field (e.g. fireman)
5G Network Slice	Slice Intent Request/Reply Dynamic intent provisioning VNF Provisioning (e.g. router, FW, VPN…) VNF Operation (e.g. router, FW, VPN…)	Infrastructure provider Service provider
SLA Monitoring	Deployment of qMON components Slice intent KPI measurements Continuous SLA Monitoring	Infrastructure provider Service provider

The first column represents different testing areas while the basic functional test scenarios are stated in the second column.

At first, deployment and orchestration of the iMON dashboard has to be tested. This mainly applies to the roles of infrastructure provider (e.g. data centre owner/administrator) and service provider that intends to offer the iMON Dashboard service. When the iMON application is successfully deployed and can be managed through the orchestrator, functional tests regarding iMON Dashboard can follow.

Functional testing of iMON Dashboard is primarily aimed at the roles of service provider and service consumer that can be either operator at the command centre (i.e. operates the iMON Dashboard) or end user in the field (i.e. fireman at the accident site using iMON smart phone application). Basic operation testing should include administrative tasks such as creating users and tasks involving the end users, for example creating triage report from the smart phone application.

Another test area is 5G network slice which enables provisioning of network path and network elements as VNFs along the path. Testing involves the roles of infrastructure (telecom) provider and service provider. The main test target is to verify the operation of slice request/reply mechanism and provisioning of network slice which provides network connectivity to the iMON application. Moreover, the onboarding and operation of VNFs providing advanced network functions (e.g. firewall) should be tested in this phase.

The last step is to test the SLA monitoring mechanism which will be provided by the qMON agent realized as a component. Included MATILDA roles are infrastructure (telecom) provider and service provider that uses SLA monitoring mechanism to continuously monitor the network KPIs of a provisioned 5G network slice. The scenarios should include provisioning, administration and basic operation of qMON agents and verify that monitoring data is available to the service provider operating the MATILDA orchestrator.

4.6.2 First Demonstrator Release

Features

In the first demonstrator release, the following features will be presented:

Table 19: 5G Emergency Infrastructure with SLA Enforcement Demonstrator features (first release).

Feature name	Feature details
Application Provisioning and Deployment	Provisioning and deployment of the iMON Dashboard application through the MATILDA orchestrator (whole application graph).
Service Orchestration	Context and policy-based application orchestration.
Service Scaling	Horizontal and vertical scaling of iMON Dashboard application components.
5G Slice Provisioning	Slice intent request/reply mechanism with 5G network slice provisioning.
Application QoS Monitoring	qMON Agent is deployed as an application component providing basic network and application QoS metrics.

Test Bed Requirements

For the first demonstrator release, the test bed should satisfy the following requirements:

Table 20: Test bed requirements for 5G Emergency Infrastructure with SLA Enforcement Demonstrator (first release).

Requirement name	Requirement details
Cloud Hosting Environment	OpenStack-based cloud hosting services.
Network Slice Provisioning	Programmable 4G/5G mobile network that provides dedicated mobile network slice.
Service Orchestration	MATILDA Orchestrator must be connected to the OpenStack cloud which acts as an IaaS provider.

Acceptance Criteria

The following table provides basic target values for the first demonstrator release:

Table 21: Acceptance criteria for 5G Emergency Infrastructure with SLA Enforcement Demonstrator (first release).

Metric	Target value	Baseline value
Time needed to deploy an individual component of an application graph.	< 2min	3 min
Time needed to deploy the entire application graph.	< 5 minutes	15 minutes
Time required to adapt to increased or decreased load.	< 1 min	3 min
Time needed to provision the 5G network slice.	< 3 min	Not possible at this time.

4.6.3 Second Demonstrator Release

Features

In the second demonstrator release, the following features will be presented:

Table 22: 5G Emergency Infrastructure with SLA Enforcement Demonstrator features (second release).

Feature name	Feature details
High Availability at the application level	Scaling and redundancy mechanisms at the application component level.
High Availability at the network level	Network-based failover mechanisms.
5G Slice Orchestration	Network context (ie. slice intent) and policy-based application orchestration.
VNF Orchestration	VNF provisioning and orchestration (e.g. router, VPN, FW…).
SLA Monitoring	qMON Agent is deployed as a component providing advanced QoS metrics including 5G network slice KPIs.

Test Bed Requirements

For the second demonstrator release, the test bed should satisfy the following requirements:

Table 23: Test bed requirements for 5G Emergency Infrastructure with SLA Enforcement Demonstrator (second release).

Requirement name	Requirement details
Cloud Hosting Environment	OpenStack-based cloud hosting services.
Network Slice Provisioning	Programmable 4G/5G mobile network that provides dedicated mobile network slice.
Service Orchestration	MATILDA Orchestrator must be connected to the OpenStack cloud which acts as an IaaS provider.
Network Orchestration	Programmable 4G/5G mobile network that provides dedicated mobile network slice based on the context and policies set by the MATILDA orchestrator.
VNF Orchestration	Open Source MANO connected to the MATILDA Orchestrator and provides VNF-based component orchestration.

Acceptance Criteria

The following table provides basic target values for the second demonstrator release:

Table 24: Acceptance criteria for 5G Emergency Infrastructure with SLA Enforcement Demonstrator (second release).

Metric	Target value	Baseline value
Time needed to deploy an individual component of an application graph	< 1min	3 min
Time needed to deploy the entire application graph.	< 2 minutes	15 minutes
Time required to adapt to increased or decreased load.	< 30 sec	3 min
Time needed to provision the 5G network slice.	< 1 min	Not possible at this time.
Time needed to reconfigure the 5G network slice.	< 1 min	Not possible at this time.
Time needed to deploy the VNF-based components (i.e. forwarding graph).	< 2 minutes	15 minutes
Time needed to deploy and provision SLA monitoring.	< 3 minutes	30 minutes

4.7 Demonstrator 4: Industry 4.0 Smart Factory – Inter and Intra-Enterprise Integration

The Industry 4.0 Smart Factory – Inter and Intra-Enterprise Integration demonstrator is an application that provides two scenarios: the logistics scenario and the production scenario.

- Logistics scenario: 5G based Tracking of transport vehicles and monitoring of transported goods
- Production scenario: real-time pattern detection for quality assurance and distance calculation in human-robot collaborative (HRC) production environment for collision avoidance.

4.7.1 Tested Scenarios

In the logistics scenario, a transport shall be tracked, starting from a supplier and moving to a production facility. In addition, the goods loaded on a transporter shall be monitored in real-time, as in this scenario the goods are very fragile and need a sensitive handling. For this purpose, temperature, humidity and vibrations are transmitted in real-time to give the client a possibility to monitor his goods all over the time of transport. For tracking, the goods are scanned first at the supplier (start), second at the transportation and third at the final destination at the production facility (end). This will be done by tagging the goods with RFID labels.

For the application the first step is to scan the goods at the supplier which is on a fixed location to get the information that goods have to be transported. Second step is scanning the goods at the transportation. From here the transportation phase begins. Data like positioning via GPS and the housekeeping of the goods (humidity, temperature, vibration) shall be monitored and sent directly to a client. Next to the monitoring, the data shall be collected for further analysis to optimize the transportation phase. The final step is scanning the goods at the production facility. With doing, so the scenario ends as the goods reach its final destination. In Figure 19 an overview of components and modules (hard and software) for this use case scenario is shown.

The production scenario, presented in Figure 20, consists of pattern detection for quality assurance and real-time distance calculation in human-robot collaborative (HRC) production environment. The assembly process of mass customized automotive parts (e.g. steering wheel car, car door, rear light, etc.), is still requiring human eyes to check the quality and standard of the finished product. Within MATILDA, BIBA is trying to evaluate the integration of 5G with a production assembly system, in which the quality (correctness) of assembled parts will be monitored by the use of image processing. The efficiency in this scenario is in term of time of response and pattern detection. Once some defect or mistakes (wrong colour of rear light) is found in the actual stage of the product assembly, an alarm, triggered by the process control system in the production facility, is raised for the particular assembly station.

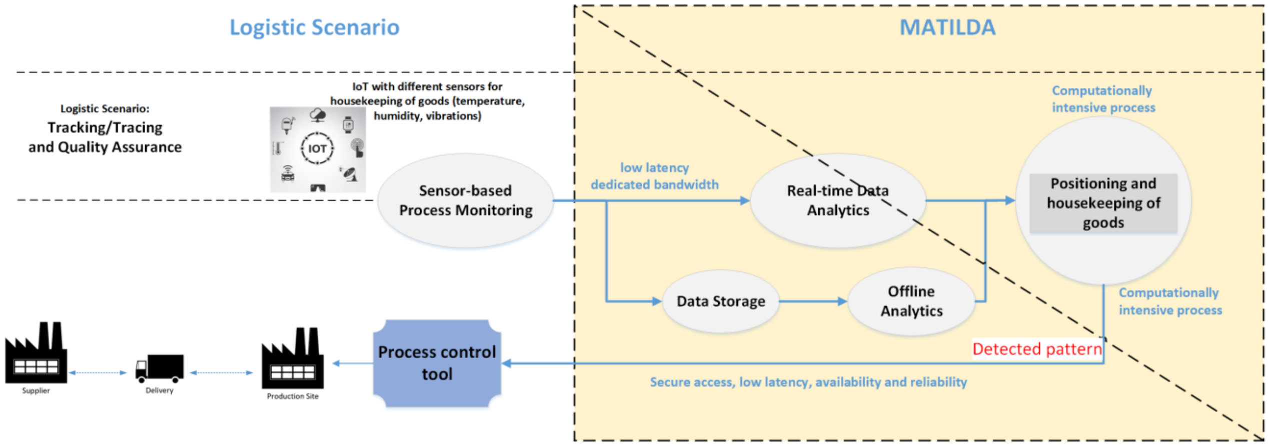

Figure 19: Logistics scenario – modules and components.

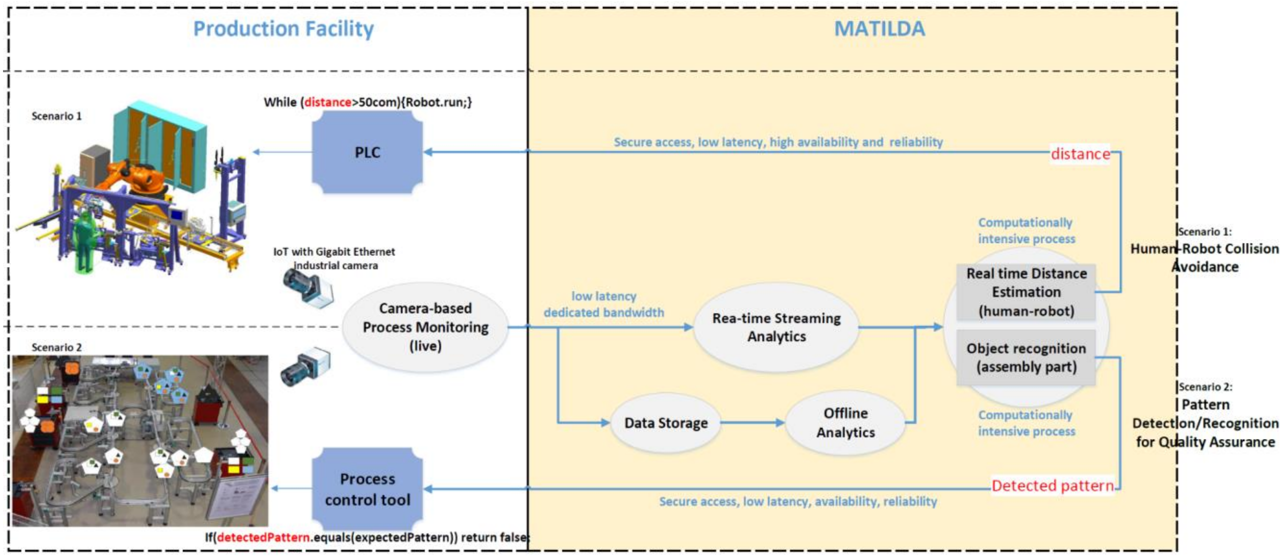

Figure 20: Production scenario – modules and components.

Since within such safety critical collaborative application, where safety fences will be eliminated, any collision between robots and worker must be avoided, a predefined safety distance between worker and heavy payload robots has to be ensured in order to avoid dangerous hazards. The goal of this use case is to evaluate the benefits and limits of 5G within safety critical applications such as Human-Robot Collaboration. The contribution of MATILDA within this scenario consists of detecting the humans (workers) close to the operated robot and provide the shortest distance between workers and robot in real-time. Based on this information, the PLC (controller), will control the speed (stop/reduce/full speed) of the robot.

Following table summarizes a possible list of the main functional tests related to the two Industry 4.0 Smart Factory scenarios. Such tests are briefly presented and discussed as an example of the approach that should be adopted, and do not represent the exhaustive description of the testing procedure adopted to fully validate the framework.

Table 25: Industry 4.0 Functional Tests.

System Under test	Operation Under Test	Role of Testing Operator
MATILDA Orchestrator	On-boarding of the application components Application deployment and instantiation High availability at the application level	Application Developer 5G Application SW Developer Service Provider
Industry 4.0 – Data Acquisition and Communication Module Services	Data Acquisition and Abstraction Communication between Application and cloud services	End User Application Developer
Industry 4.0 – Data Processing and Analysis Services	Data Processing Data Storage Real time Data Analysis (Logistics, Collision avoidance, pattern Detection) Offline Data Analysis (Logistics, Collision avoidance, pattern Detection)	End User Application Developer
Industry 4.0 – Data Visualisation Service	Dashboards	End User Application Developer
Industry 4.0 – Process control Service	Machine control from cloud-based services	End User Application Developer

In Table 25, the first column highlights the main functional part of the system under test. The second column indicates the specific tested operation, while in the third column we report the role played by the tester in the considered use case.

4.7.2 First Demonstrator Release

Features

In the first demonstrator release, the following features will be presented:

Table 26: Industry 4.0 Smart Factory demonstrator features (first release).

Feature name	Feature details
Data Acquisition	Real time acquisition of process data from connected devices (sensors, cameras, …) Real time monitoring of sensors from the logistics scenario
Application components development	The development of the software tools for the sensor data processing (localization, pattern detection, collision avoidance, …)
Application Provisioning and Deployment	Provisioning and deployment of the Industry 4.0 application through the MATILDA orchestrator (whole application graph).
Industry 4.0 Application	Industry 4.0 is deployed as an application component providing the addressed functionalities

Test Bed Requirements

The demonstrator implementation is based on the test bed setup provided by UNIVBRIS. This implementation will utilize the following features and infrastructure provided by the test bed setup. The test bed should satisfy the following requirements:

Table 27: Test bed requirements for Industry 4.0 Smart Factory demonstrator (first release).

Requirement name	Requirement details
Cloud Hosting Environment	OpenStack-based cloud hosting services.
Network Slice Provisioning	Programmable 4G/5G mobile network that provides dedicated mobile network slice.
Connectivity	VPN connection between BIBA IT Infrastructure and UNIVBRIS
Availability	The test bed IT infrastructure should ensure a high-level connectivity with the application in Bremen.
Security	Process data (e.g. live video streams) and decisions (responses of the cloud-based services) should be transferred in a secure way between end-points

Acceptance Criteria

The following table provides basic target values for the first demonstrator release:

Table 28: Acceptance criteria for Industry 4.0 Smart Factory Demonstrator (first release).

Metric	Target value
Demonstrator test bed deployment time	< 5min
Demonstrator component availability/reliability performance	99.9%
Resource verification, reservation, allocation	< 5 min
Device sensor authentication /authorization /encryption/provisioning	99.9%
Delay/ Latency	Collision avoidance: 100ms Pattern matching: 250ms Logistics: 60 seconds

4.7.3 Second Demonstrator Release

Features

In the second demonstrator release, the following features will be presented:

Table 29: Industry 4.0 Smart Factory demonstrator features (second release).

Feature name	Feature details
Process control	5G application should be able to control industrial processes (safety critical, non-safety critical)
Process monitoring	5G application should be able to monitor the given logistics scenario

Test Bed Requirements

For the second demonstrator release, the test bed should satisfy the following requirements:

Table 30: Test bed requirements for Industry 4.0 Smart Factory demonstrator (second release).

Requirement name	Requirement details
Security	Process data (e.g. live video streams) and decisions (responses of the cloud-based services) should be transferred in a secure way between end-points
Latency	Response time between application & services / services and application should satisfy application needs in term of real-time

Acceptance Criteria

The following table provides basic target values for the second demonstrator release:

Table 31: Acceptance criteria for Industry 4.0 Smart Factory Demonstrator (second release).

Metric	Target value
Demonstrator test bed deployment time	< 3min
Demonstrator component availability/reliability performance	99.9%
Resource verification, reservation, allocation	< 3 min
Device sensor authentication /authorization /encryption/provisioning	99.9%
Delay/ Latency	Collision avoidance: 100ms Pattern matching: 250ms Logistics: 30 seconds
Process monitoring capabilities	99.9%
Machine Control capabilities	99.9%

4.8 Demonstrator 5: Smart City Intelligent Lighting System

The Smart City Intelligent Lighting System is a use case based on the implementation of the Smart Lighting solution, designed and optimized for LoRaWAN and LTE-M networks, to perform autonomous operation based on predefined schedules and light level sensors. It is designed to provide an efficient bandwidth with minimal communication requirements for under/overpower smart monitoring, voltage monitoring and daytime/nighttime consumption mismatches that will be reported in real-time to a central server. The solution provides map visualization capabilities of managed devices (maps, grid, power supply cabinet), ON/OFF lights switching and diming functionalities, real time lighting sensors monitoring. The system architecture is composed of three main application functionalities:

1. IoT Aggregator component, a fixed component used for data collection using LoRAWAN, LTE-M, LTE street lighting assuring a secured connection with the lighting sensors
2. IoT Middleware component with the role of provisioning, processing, visualization and lighting sensor management
3. Dashboard, Monitoring and Billing components, which give the administrators the possibility to remote control and access lighting sensors, to supervise and keep track of service required intervention, to perform traffic evaluation and monetization activities.

High availability, redundancy and resilience, scalability, dynamic QoS provisioning are key characteristics for Smart City public deployment vertical. The evaluation scenarios should cover the following requirements:

- automated end to end instantiation of the solution as a vertical slice
- high availability, redundancy and resilience, no slice disruptions in the case of individual components failure
- fast scalability, the deployment of the solution will be realized in steps and the scalability of the system is mandatory
- dynamic QoS provisioning, support and enforcement to assure on-demand provisioning of the use case according to specific slice metrics and real-time network conditions. The QoS metrics are maintained during the slice lifecycle.

The Smart City use case demonstrator is composed of the following application component descriptors:

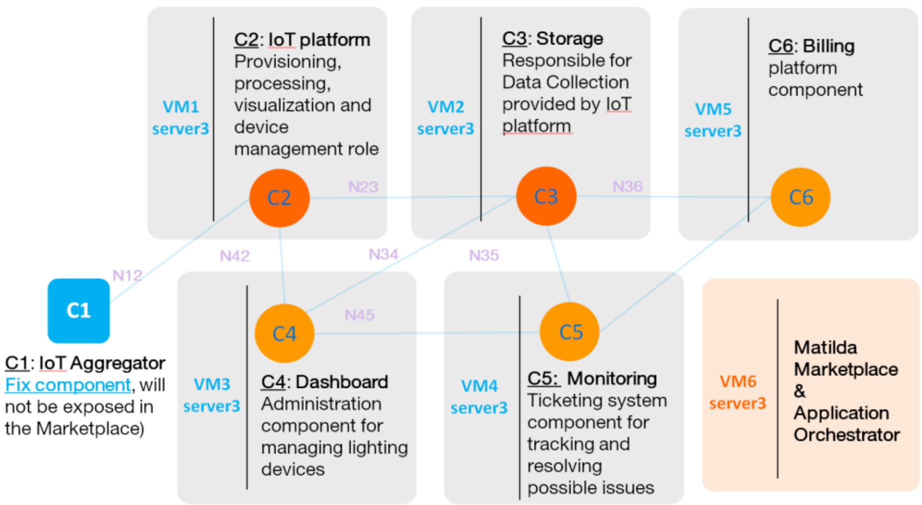

Figure 21: Smart City components - test bed infrastructure mapping.

Table 32: Test bed requirements for 5G Emergency Infrastructure with SLA Enforcement Demonstrator (second release).

Demonstrator components	Release version
IoT platform component - Thingsboard.io	v2.1
Database PostgreSQL	v9.5.13
Database Cassandra NoSQL - hosting no relational database	v2.1.0
Dashboard	alpha (under deployment)
Monitoring	alpha (under deployment)
Billing	alpha (under deployment)
One stop API	alpha (under deployment)
Docker	v 18.03.1, 0.14.0, 1.21.2

- **IoT platform (Thingsboard.io)** is the main component of Smart City demonstrator. It is an open-source IoT component that facilitates the development, management and scaling of device lighting sensors. Thingsboard.io is horizontally scalable, a single server node being capable of handling hundreds of thousands of devices depending on the use case. It provides a large pallet of features and mechanisms assuring the entire workflow of an IoT cloud ecosystem with roles of provisioning, control of lighting sensor devices, different use case specific enabler through customizable rule chains, enabling multiple workflows based on device life-cycle event, REST API event.
- **Storage - SQL database (PostgreSQL)** is a database demonstrator component responsible for data collection provided by the IoT platform. It is an open source object-relational database system that safely stores and scales complex data workloads. PostgreSQL comes with many features aimed to protect data integrity and build fault-tolerant environments.
- **Storage - SQL database (Cassandra NoSQL)** is the second database component to store Smart City device sensor data to further processed by the Dashboard, Monitoring and Billing components. NoSQL database is a non-relational and largely distributed database system that enables rapid, ad-hoc organization and analysis of extremely high-volume, disparate data types.
- **Dashboard** is a demonstrator component with the administrative role of the lighting sensor devices
- **Monitoring** provides the ticketing capabilities responsible for device identification, health status for servers and database demonstrator components
- **Billing** is in charge of traffic evaluation and monetization functions.

4.8.1 Tested Scenarios

A set of test scenarios related to the "Smart City Lighting System" demonstrator are proposed in Table 33:

Table 33: Smart City Intelligent Lighting Demonstrator functional tests.

MATILDA-Slicenet system components	Evaluation steps	Roles
Smart City IoT Enterprise / MATILDA Marketplace	Application components deployment and application graph creation Application components hosting	5G Application SW developer Infrastructure provider Service provider
Smart City IoT Enterprise/MATILDA Orchestrator	Smart City Service composition evaluation (VNF-FG composition) Slice Intent creation Context and policy-based application orchestration	Infrastructure provider Service provider
SliceNet Programable Infrastrcture	Smart City slice Intent deployment based on pre-defined metamodel through open API Resource and Service Orchestration	Infrastructure provider Service provider
Smart City IoT platform/Service Dashboard	Smart lighting service end to end functionality (device authentication, device control (ON/OFF switching, dimming, etc) Automatic provisioning (configuration, network resource allocations)	Service provider Service consumer
Smart City IoT Enterprise/ Service Monitoring	Slice monitoring - real time monitoring of sliced network activity and health information delivery Traffic monitoring - user data traffic inside the slice (ex: resources consumed in OAI-RAN smart lighting slice) Topology monitoring – evaluate if monitoring information of the slice topology is provided VNF & PNF monitoring (keep alive mechanism functionality evaluation) QoS metrics and Resources metrics delivery	Infrastructure provider Service provider

The envisaged Smart Lighting use case testing scenario is composed of four main steps:

1. Smart City service graph composition. The Service Provider will access the MATILDA Marketplace hosted in the ORO test bed virtual infrastructure and download individually predefined application components or an aggregated solution in the form of an application graph already created based on a predefined metamodel.
2. Smart City network slice instantiation. Through the MATILDA Application Orchestrator hosted on a dedicated VM, the Smart Lighting application graph will be

instantiated through an open API in the form of an end to end network slice over the whole test bed infrastructure. The composition of the service is defined based on a network functions workflow in the form of a VNF forwarding graph (VNF-FG) and specifies the types of VNFs to be connected and their order and requirements in terms of connectivity to provide the Smart City service.

3. Smart City service delivery. Once the network slice is set and the Smart City Lighting service is up and running, the Smart Lighting enterprise applications framework will go into a dynamic operation mode and the Service Consumers will benefit from all the capabilities enabled by MATILDA Smart City applications.
4. Smart City Lighting service performance monitoring. Once the solution is live, a set of test and monitoring scenarios shall run automatically and shall provide a set of test reports and data models in order to assure the correct functionality as per the acceptance criteria, including cognition (based on sensors capabilities and metrics), prediction, scalability and performance.

4.8.2 First Demonstrator Release

Features

The first Smart City demonstrator release will have the following features:

Table 34: Smart City Intelligent Lighting Demonstrator features (first release).

Feature name	Feature details
Deployment and provisioning	Local deployment of Marketplace & App orchestrator (e.g. on a laptop)
Network Slice Provisioning	Slice intent (MATILDA format) interoperable with One Stop API SLICENET – definition and development phase
Management and scaling	Dashboard administration component for managing lighting devices - first release (Dockerized)
Service monitoring and service supervision	Monitoring component - ticketing system -1st release (Dockerized)
Service orchestration	Context and policy-based application orchestration.

Test Bed Requirements

For the first demonstrator release, the test bed should satisfy the following requirements:

Table 35: Test bed requirements for Smart City Intelligent Lighting Demonstrator (first release).

Requirement name	Requirement details
Hosting Environment	OpenStack-based hosting services.
IoT aggregation platform	Raw data information aggregation functionality form device sensors to the IoT Enterprise application (IoT platform, dashboard, command and control).
Heterogeneous network access types (LoRaWAN/LTE-M/5G)	Seamless connectivity to Intelligent Lighting platform.
Network Slice Provisioning	Mobile network programmable infrastructure to sustain network slice deployment.
IoT Middleware	IoT middleware communication biddings with MATILDA Application Orchestrator are needed for end-2-end slice instantiation

Acceptance Criteria

The following table provides basic target values for the first demonstrator release:

Table 36: Acceptance criteria for Smart City Intelligent Lighting Demonstrator (first release).

Metric	Target value
Demonstrator test bed deployment time	< 5min
Demonstrator component availability/reliability performance	99.9%
Resource verification, reservation, allocation	< 5 min
Device sensor authentication/authorization/encryption/provisioning	99.9%

4.8.3 Second Demonstrator Release

Features

In the second demonstrator release, the following features will be presented:

Table 37: Smart City Intelligent Lighting Demonstrator features (second release).

Feature name	Feature details
QoS performance	Capabilities to adapt to different range of scenarios during Smart city use instantiation/functionality (e.g. a data traffic burst scenario)
Scalability (Resource Control/Management)	Resource Control and Management based on the service capacity requirements. Smart isolation over the network assuring the resource control and management.
Plug & Play capabilities	Automate provisioning facilities for end to end network deployment (configuration, network resource allocations)
High Availability at the application level	Scaling and redundancy mechanisms at the application component level.
Deployment and provisioning	Deployment of Marketplace & App orchestrator in the test bed.
Network Slice Provisioning	Slice intent (MATILDA format) interoperable with One Stop API SLICENET - partial integration

Test Bed Requirements

For the second demonstrator release, the test bed should satisfy the following requirements:

Table 38: Test bed requirements for Smart City Intelligent Lighting Demonstrator (second release).

Requirement name	Requirement details
Cognition requirements	Intelligence adaptation based on the network situational context scenario providing rapid responses to exceptional cases (failure, congestion)
Communication service	Unicast
FCAPS requirements	Fault-management, configuration, accounting, performance, and security.
Multi-tenancy	Physical capabilities resources sharing among multiple tenants (slices)

Acceptance Criteria

The following table provides basic target values for the second demonstrator release:

Table 39: Acceptance criteria for Smart City Intelligent Lighting Demonstrator (second release).

Metric	Target value
Demonstrator test bed deployment time	< 3min
Demonstrator component availability/reliability performance	99.9%
Resource verification, reservation, allocation	< 3 min
Device sensor authentication /authorization /encryption/provisioning	99.9%
Fault detection	<5min

4.8.4 Third Demonstrator Release

Features

In the third demonstrator release, the following features will be presented:

Table 40: Smart City Intelligent Lighting Demonstrator features (third release).

Feature name	Feature details
5G Network Slice Provisioning	Slice intent (MATILDA format) interoperable with One Stop API SLICENET - final integration
Service monitoring and service supervision, slice and topology monitoring, Qos and QoE capabilities	Monitoring component - ticketing system -2nd release (Dockerized).
Management and scaling	Dashboard administration component for managing lighting devices - 2nd release (Dockerized)

Test Bed Requirements

For the third demonstrator release, the test bed should satisfy the following requirements:

Table 41: Test bed requirements for Smart City Intelligent Lighting Demonstrator (third release).

Requirement name	Requirement details
5G Application and Slice Orchestration	Market Place, Application Orchestrator and Slice Orchestrator integration
QoS/QoE sensors and actuation	QoE sensors integrated and through ML/AI techniques actuation takes over the system, performance prediction and adaptation
Self-scaling network for service management requests	Plug & Play capabilities integration in the system for lighting devices management activities (software updates)

Acceptance Criteria

The following table provides basic target values for the third demonstrator release:

Table 42:Acceptance criteria for Smart City Intelligent Lighting Demonstrator (third release).

Metric	Target value
IoT platform device data collection accuracy performance	99.9%
Storage database (PostgreSQL/Cassandra NoSQL) query response time performance	<100msec
Database scalability	Yes
Dashboard processing/display time performance	<1 sec

5 Master Plan

The master plan contains the scheduling for:

- the components that will be developed within MATILDA
- the three testbeds that support the deployment of the demonstrators
- the five demonstrators planned to be developed within MATILDA running the use cases for the 5G verticals.

The complete master plan in presented in Annex 1. The test beds will be developed with full capabilities by the 7th project quarter. The demonstrators will be developed during the second half of the project and are organized per releases, each of them with a relevant set of capabilities to be demoed during international conferences or project review meetings. Each of the demonstrator releases is deployed over a certain test bed. For the sake of clarity, the correspondence among the demonstrator releases, test beds and demos is depicted in Table 43.

Table 43: Demonstrator releases, test beds and demos.

Demonstrator	Owner	Release	Test bed	Demo
5GPACE- High Resolution Media on Demand & Banking on the Cloud	ITL & INC	Release 1	UNIVBRIS	MWC 2019[1]
		Release 2	UNIVBRIS	EUCNC 2019[2]
		Release 2 – extra features	UNIVBRIS	MWC 2020
Testing 4.0 - Distributed System Testing	EXXPERT	Release 1	UNIVBRIS	N/A (internal testing)
		Release 2	UNIVBRIS	Review Meeting (June 2019)
		Release 3	UNIVBRIS	Feb 2020
5G Emergency Infrastructure with SLA Enforcement	ININ	Release 1 (phase 1)	CNIT	EUCNC 2018[3]
		Release 1 (phase 2)	CNIT	Review meeting (November 2018)
		Release 2 (phase 1)	CNIT	MWC 2019
		Release 2 (phase 2)	CNIT	Review Meeting (March 2020)
Industry 4.0 Smart Factory – Inter and Intra-Enterprise Integration	BIBA	Release 1	UNIVBRIS	Review meeting (June 2019)
		Release 2	UNIVBRIS	February 2020
Smart City Intelligent Lighting System	ORO	Release 1	ORO	MWC 2019
		Release 2	ORO	EUCNC 2019
		Release 3	ORO	Review meeting (February 2020)

All in all, the master plan provides a unique view of all the project phases, confirming the planning consistency among the components to be developed, test beds and demonstrators.

[1] Mobile World Congress, https://www.mobileworldcongress.com.
[2] European Conference on Networks and Communications, https://www.eucnc.eu.
[3] Past event

6 Conclusions

This deliverable has described the demonstrators of the MATILDA Project and the related scenarios and test cases, along with the evaluation framework and validation methodologies that have been selected to assess the architectural soundness of the MATILDA framework.

Testing procedures have been defined to characterize the applicability and potential of the innovations brought forth by the MATILDA Project. In more detail, the testing categories that have been identified so far focus on evaluating specific applications' QoS KPIs, the performance of the functionalities/capabilities to be delivered by a single or by multiple MATILDA component, and the solution as a whole for the development and definition of 5G-ready applications over a sliced network infrastructure.

The evaluation framework provided in Section 3 stems from both the definition of the set of performance evaluation metrics and acceptance criteria developed in the scope of WP1 and the 5G-PPP KPIs. A preliminary definition of the tests has been produced as well, and is reported in Annex 2, which provides an example of how tests will be performed and results will be presented in the upcoming deliverable reports (namely, D6.7 and D6.13). As previously mentioned, these definitions may be subject to changes according to the development in the testbeds and use case applications and the framework components throughout the Project.

It is worth noting that the goal of the evaluation is to exploit the collected data to provide an overall assessment of MATILDA following the framework evolution at the various phases of its development. The pursuit of this goal is reflected by the way the use cases development and demonstrations are planned. Their role of showcasing both the correct application of the MATILDA platform and its impact on the secure and trusted microservice engineering using the platform will be explicitly addressed in order to provide useful adoption guidelines for the efficient design, development and orchestration of vertical applications.

Accordingly, the description of the testbeds and demonstrators involved in the validation of the MATILDA framework follows this same line of work. Compared to the information provided in the D1.6 report, in this document the focus has been on how the scenarios will undergo testing and on the features that will be made available for the various releases of the prototypes.

Following this same path, the descriptions of the test beds put emphasis on the interconnectivity aspects and requirements and the related interfaces exposed to the outside. The Orange test bed, a joint effort of the MATILDA and SLICENET 5G-PPP EU projects, is a small-scale network following an ETSI MANO architectural framework for virtualization. The University of Bristol provides a smart city research and development network platform composed of several computing nodes, control plane/data plane switches and radio access components interconnecting different areas, with the capability of providing different levels of virtualization. Finally, CNIT, UBITECH and COSM are designing an integrated test bed infrastructure providing support for NB-IoT and 4G/5G user equipment devices, residential gateway boxes and NB-IoT sensors and the virtualized infrastructure part composed of a set of high-performance equipment supporting OpenFlow, SDN and the OpenStack Project.

As for the use cases, the five demonstrators are meant to represent different examples of 5G verticals. Their descriptions, as provided in this deliverable report, put a spotlight on the functional testing and the features, requirements and acceptance criteria of the various releases of the demonstrators. In fact, each demonstrator will evolve in two or three releases, with the first release focusing on the interoperability of the different components of the framework and the further functionalities related to 5G application graph deployment, slice adaptation, etc., made available for the following releases of the demonstrators. Detailed documentation on the individual use cases implementation will be tackled in the upcoming D6.2-D6.6 (for the first phase) and D6.8-D6.12 (for the second phase) deliverable reports.

Finally, a thorough planning has been provided to summarize the development steps, the partial releases, and the demonstrations that will take place until the end of the Project. This planning describes, for the different project phases, the development of the MATILDA platform components, test beds and use case applications by highlighting for each one of them the expected completion of each task and the achievement of the milestones with respect to the planned project review meetings and international conferences.

A simplified view of this plan is reported in Section 5, which provides a concise overview of the correspondence among the demonstrator releases, test beds and demos. A more complete version is available in Annex 1 and it contains the whole planning organized according to the components that will be developed within MATILDA, the three testbeds that support the deployment of the demonstrators, and the five demonstrators planned to be developed within MATILDA running the use cases for the 5G verticals.

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
[ASR 9K]	Docs available online at: https://www.cisco.com/c/en/us/products/routers/asr-9000-series-aggregation-services-routers/index.html
[Big-IP F5]	Managing the F5 BIG-IP Load Balancer, available online at: https://www.cisco.com/c/en/us/td/docs/unified_computing/ucs/ucs-director/f5-bigip-loadbalancer-mgmt-guide/5-5/b_Cisco_UCS_Director_F5_LoadBalancer_ManagementGuide_55/b_Cisco_UCS_Director_F5_LoadBalancer_ManagementGuide_55_chapter_010.pdf
[CEILOMETER]	OpenStack project for data collection, docs available online at: https://docs.openstack.org/ceilometer/queens/
[CINDER]	OpenStack Block Storage Service, docs available online at: https://docs.openstack.org/cinder/latest/
[CloudBand]	Nokia, CloudBand Product Page. URL: https://networks.nokia.com/products/cloudband
[COAP]	RFC 7252 Constrained Application Protocol, web transfer protocol for use with constrained nodes and constrained networks in the Internet of Things, available online at: http://coap.technology/
[FastWAN]	http://www.exxpertsystems.de/#fastwanLink
[FLASHNET]	Intelligent energy management system, available online at: https://www.flashnet.ro
[FlexRay]	https://svn.ipd.kit.edu/nlrp/public/FlexRay/FlexRay%E2%84%A2%20Protocol%20Specification%20Version%203.0.1.pdf
[Fortinet Appliance]	Network security solution, docs available online at: https://www.fortinet.com/products/next-generation-firewall.html
[GLANCE]	OpenStack project, provides a service where users can upload and discover data assets that are meant to be used with other services, docs available at: https://docs.openstack.org/glance/latest/
[HEAT]	Open Stack project, developed to orchestrate composite cloud applications, description available online at: https://docs.openstack.org/heat/pike
[HORIZON]	OpenStack Dashboard Project, docs available online at: https://docs.openstack.org/horizon/latest/
[HP DL360]	Docs available online at: https://www.hpe.com/us/en/product-catalog/servers/proliant-servers/pip.hpe-proliant-dl360-gen10-server.1010007891.html
[HP DL380]	Docs available online at: https://www.hpe.com/us/en/product-catalog/servers/proliant-servers/pip.specifications.hpe-proliant-dl380-gen9-server.7271241.html

[IARIA17]	A. Albanese, P. S. Crosta, C. Meani, & P. Paglierani, "Immersive Video Services at the Edge: an Energy-Aware Approach", IARIA International Journal on Advances in Telecommunications, vol 10, n. 3&4, January 2017
[iMON]	www.imon.si
[JMM09]	C. Bailey, P.R. Baines, H. Wilson & M. Clark, "Segmentation and customer insight in contemporary services marketing practice: why grouping customers is no longer enough". Journal of Marketing Management, vol. 25 (3-4), pp.2 27–252, 2009
[KEYSTONE]	OpenStack service that provides API client authentication, service discovery, and distributed multi-tenant authorization, docs available online at: https://docs.openstack.org/keystone/pike/
[KVM]	Kernel Virtual Machine, docs available online at: https://www.linux-kvm.org/page/Main_Page
[LORA]	LoRa Alliance, LoRaWAN - Long Range Low Power Wide Area networking protocol, available online at: https://lora-alliance.org/about-lorawan
[MATILDA-D.1.1]	MATILDA project deliverable, D1.1 - MATILDA Framework and Reference Architecture, available online at: http://www.MATILDA-5g.eu/index.php/outcomes
[MPLSoUDP]	RFC 7510 Encapsulating MPLS in UDP, docs available online at: https://tools.ietf.org/html/rfc7510
[MQTT]	Message Queuing Telemetry Transport, messaging protocol, designed for constrained devices and low-bandwidth, high-latency or unreliable networks, available online at: http://mqtt.org
[MURANO]	OpenStack Application Orchestration and Catalog project, docs available online at: https://www.mirantis.com/software/application-catalog-murano/
[NetOS]	Zeetta Networks, NetOS. URL: https://zeetta.com/technology/netos/
[NEUTRON]	OpenStack project, provides "network connectivity as a service", docs available online at: https://docs.openstack.org/neutron/pike/
[NEXUS 9K]	Docs available online at: https://www.cisco.com/c/en/us/support/switches/nexus-9000v-switch/model.html
[NokiaMEC]	Nokia, Multi-access Edge Computing. URL: https://networks.nokia.com/solutions/multi-access-edge-computing
[NOVA]	OpenStack project that provide a way to provision compute instances, docs available online at: https://docs.openstack.org/nova/latest/
[OAI SW stack]	Open Air Interface software stack, available online at: https://www.semanticscholar.org/paper/OpenAirInterface%3A-A-Flexible-Platform-for-5G-Nikaein-Marina/8f078271d8bd6b9ff21818de0dc3b4294e5fac12/figure/0
[ODL]	The OpenDaylight Project. URL: https://www.opendaylight.org/
[ONF]	Open Networking Foundation. URL: https://www.opennetworking.org/

[OpenAirInterface]	5G Software Alliance, open source software and hardware development for the core network (EPC), access network and user equipment (EUTRAN) of 3GPP cellular networks, available online at: http://www.openairinterface.org/
[OpenStack]	OpenStack software controls large pools of compute, storage, and networking resources throughout a datacenter, managed through a dashboard or via the OpenStack API, docs available online at: https://www.openstack.org/
[OpenStack-QUEENS]	Openstack, Queens Release Notes. URL: https://www.openstack.org/software/queens/
[OSM]	Open Source MANO, available online at: https://osm.etsi.org
[OSM-THREE]	The Open Source Mano project, Release THREE Notes. URL: https://osm.etsi.org/wikipub/index.php/OSM_Release_THREE
[OST]	OpenStack Telemetry , OpenStack project documentation, available online at: https://docs.openstack.org/mitaka/install-guide-rdo/common/get_started_telemetry.html
[REST]	REpresentational State Transfer, architectural style for providing standards between computer systems on the web, available online at: https://www.codecademy.com/articles/what-is-rest
[SaaS]	Software as a Service definition, docs available online at: https://searchcloudcomputing.techtarget.com/definition/Software-as-a-Service
[SLICENET-D.2.2]	SliceNet project deliverable, D2.2 - Overall Architecture and Interface Definition. URL: https://slicenet.eu/deliverables/
[SLICENET-D.3.1]	SliceNet project deliverable, D3.1 - Design and Prototyping of SliceNet Virtualised Mobile Edge Computing Infrastructure. URL: https://slicenet.eu/deliverables/
[SLICENET-D.3.2]	SliceNet project deliverable, D3.2 - Design and Prototyping of SliceNet Virtualised 5G RAN-Core Infrastructure. URL: https://slicenet.eu/deliverables/
[SDN controller]	Software Designed Network controller definition, docs available online at: https://searchsdn.techtarget.com/definition/SDN-controller-software-defined-networking-controller
[SDR]	Software Defined Radio overview, available online at: https://www.researchgate.net/publication/3227747_Software-defined_radio_A_brief_overview
[TACKER]	OpenStack service for NFV Orchestration, docs available online at https://docs.openstack.org/tacker/pike/
[ThingsBoard]	Open Source IoT Platform, available online at: https://thingsboard.io/
[TOR]	Top-of-Rack switching, docs available online at: https://searchnetworking.techtarget.com/definition/top-of-rack-switching
[USRP X310]	Scalable Software Defined Radio platform, available online at: https://www.ettus.com/product/details/X310-KIT

Annex 1: Complete WP6 Master Plan

The following images represent an aggregation of the complete master plan presented in Section 5.

It is worth noting that they represent the planning as expected at the time of the submission of this document. For a more up-to-date planning, and for better readability, please refer to the complete Excel spreadsheet, which is available for consultation at: https://private.matilda-5g.eu/Documents/PublicDownload/326.

	planning for testbed or demonstrator
	sub-category of testbed or demonstrator planning
	task planning
	milestone when demo takes place or component is released
Note	the colours from column A are not releated with the coulour above, they are used to link demonstrator to testbed.

	Task	dic-17	gen-18	feb-18	mar-18	apr-18	mag-18	giu-18	lug-18	ago-18	set-18	ott-18	nov-18	dic-18	gen-19	feb-19	mar-19	apr-19	mag-19	giu-19	lug-19	ago-19	set-19	ott-19	nov-19	dic-19	gen-20	feb-20	mar-20	apr-20
Components development - Common for all test beds	Component Development/Wrapping Toolkit												First release						Final release											
	Application Graph Editor												First release						Final release											
	Policy Editor														First release				Final release											
	Component Repository												First release						Final release											
	Application Graph Repository												First release						Final release											
	Optimization Engine														First release				Final release											
	Stream Aggregator														First release				Final release											
	Analytics Module														First release				Final release											
	Context Awarness/Policy Engine									First release									Final release prototype											
	NFVO												First release						Final release											
	WIM												First release						Final release											
	Computing Slice Broker												First release						Final release											
	OSS/BSS												First release						Final release											
ORO testbed	ORO testbed																													
	Access part																													
	portable smart lamp with LoRA/LTE-M connectivity																													
	4 smart lamps LoRA/LTE-M installed in Politehnica campus																													
	IOT aggregator (dedicated server)																													
	LoRA network server (dedicated server)																													
	Virtualized infrastructure																													
	deploy 4 servers in one PoP DC																													
	Server 1 controller node with VIM & OSS/BSS functionalities																													
	Server 2 Mng & Orch. node with VNFO, VNFM & Tacker Module																													
	Server 3 Computing Node using KVM																													
	Server 4 hosting vEPC																													
	Support for Demonstrators/Demonstrations																													
	Installation of MATILDA platform (incl. SLICENET integration) in the testbed																													
	Access administration																													
	General support																													
	Equipment/Network administration/maintenance																													
	General Software Installation/Update																													
ORO demonstrator- Smart City (Intelligent Lighting)	ORO demonstrator- Smart City (Intelligent Lighting)																													
	Set-up - phase 1																													
	Slice intent (MATILDA format) interoperable with One Stop API SLICENET -definition&dev																													
	IOT middleware platform																													
	Dashboard administration component for managing lighting devices -1st release (dockerized)																													
	Monitoring component - ticketing system -1st release (dockerized)																													
	Local deployment of MarketPlace & App orchestrator (e.g. on a laptop)																													
	1st demo preparation for MWC 2019																													
	demo at MWC 2019 takes place																													
	Phase 2																													
	Billing platform component																													
	deployment of Marketplace & App orchestrator in testbed																													
	Slice intent (MATILDA format) interoperable with One Stop API SLICENET -partial integration																													
	2nd demo preparation for EUCNC 2019																													
	demo at EUCNC 2019 takes place																													
	Phase 3																													
	Slice intent (MATILDA format) interoperable with One Stop API SLICENET -final integration																													
	Monitoring component - ticketing system -2nd release																													
	Dashboard administration component for managing lighting devices - 2nd release																													
	Final demo for EC in Bucharest (Politehnica campus)																													
	Final demo in Politehnica campus takes place																													

	Task	dic-17	gen-18	feb-18	mar-18	apr-18	mag-18	giu-18	lug-18	ago-18	set-18	ott-18	nov-18	dic-18	gen-19	feb-19	mar-19	apr-19	mag-19	giu-19	lug-19	ago-19	set-19	ott-19	nov-19	dic-19	gen-20	feb-20	mar-20	apr-20
CNIT testbed	CNIT testbed																													
	Access part																													
	2x Amarisoft eNodeBs (+1 in the UBITECH part)	available																												
	Small cells "Flexi Zone Multiband Indoor Pico BTS" (FZ MBI) provided by NOKIA (UBITECH)	available																												
	2x SODAQ NB-IoT sensors	available																												
	2x SAMSUNG Android Devices	available																												
	2x FRITZBOX 6820 LTE	available																												
	2x Android emulators (as VM) + LTE dongles	available																												
	Virtualized infrastructure																													
	* each instance with 1 controller, 1 network node, 1 storage node and >=2 compute nodes																													
	1x Openstack Pike for compute and storage resources (UBITECH)	available																												
	Server with: * VM for vApp Orchestrator * VM for OSM (NFVO) * VM for WIM * VM for OSS	available, for the components see the																												
	Ethernet switches with OpenFlow and SDN support	available																												
	available VNFs: * Amarisoft monolithic EPC (support for NB-IoT, dedicated bearers,etc.) * Amarisoft IMS * OpenEPC MME * OpenEPC S-GW * OpenEPC P-GW * OpenEPC HSS * CNIT S1-bypass for MEC * VNFs for routing, switching, tunneling, etc.																													
ININ demonstrator - 5G Emergency Infrastructure with SLA Enforcement	Demonstration planning - 5G Emergency Infrastructure with SLA Enforcement																													
	Set-up - phase 1																													
	Development of the application graph and slice intent																													
	Preparation for the first review in Brussels																													
	Presentation of the demonstrator at the first review																													
	Preparation of the demonstration for the EUCNC 2018																													
	First demo at the EUCNC 2018																													
	CNIT Testbed integration - phase 2																													
	Testing of the mobile terminals and smart phone application																													
	Attachment of the mobile terminals to the CNIT testbed mobile network																													
	Deployment of the application components at CNIT data centre																													
	qMON-based active monitoring (qMON engine at application level)																													
	Network slice provisioning (not integrated with the MATILDA orchestrator)																													
	Demonstration at the review meeting in Genoa (2nd demo)																													
	CNIT Testbed integration - phase 3																													
	Horizontal/vertical scaling of application components																													
	Dynamic network slice (i.e. through the MATILDA orchestrator)																													
	Integration of the VNFs																													
	CNIT Testbed integration - phase 4																													
	qMON Integration as a VNF																													
	SLA monitoring - application level																													
	SLA monitoring - network level																													
	Final demo																													

	Task	dic-17	gen-18	feb-18	mar-18	apr-18	mag-18	giu-18	lug-18	ago-18	set-18	ott-18	nov-18	dic-18	gen-19	feb-19	mar-19	apr-19	mag-19	giu-19	lug-19	ago-19	set-19	ott-19	nov-19	dic-19	gen-20	feb-20	mar-20	apr-20
UNIVBRIS testbed	UNIVBRIS testbed																													
	Access part																													
	7 x SDN enabled Ruckus Wi-Fi AP (T710 and R720) + 1 available for installation									available																				
	7 x Nokia Wifi AP (AC400)									available																				
	2 x 4G EPC & LTE-A (Dual FDD licensed bands for 1800MHz and 2600MHz; with 15MHz of T&D licence in 2600MHz band)									available - needs schedule for using													MATILDA OSS/BSS proxy would need to be integrated to nokia equipment							
	2 x 5G Core & 5G NR Massive MIMO (TDD band 42 at 3.5GHz; with 20MHz T&D licence) – expected availability after November 2018.															Risk: Depends on vendor ability to deliver/install on time							MATILDA OSS/BSS proxy would need to be integrated to nokia equipment							
	6 nodes x Self-organising multipoint-to-multipoint wireless mesh network CCS MetNet a 26GHz with 112MHz T&D licence providing 1.2Gbps throughput									available																				
	LiFi Access point - pureLiFi LiFi access points supporting 43Mbps									available																				
	2 x Corsa switch (Corsa DP2100)									available																				
	11 x Edgecore switch (Edgecore AS4610 series & AS5712-54X)									available																				
	1 x Polatis Series 6000 Optical Circuit Switch									available																				
	Virtualized Infrastructure																													
	1 instance of OpenStack running + variable number of compute nodes									available		If changes are required, needs planning																		
	1 instance of SDN controller (NetOS)									available		If changes are required, needs palnning																		
	1 instance of OSM									available		If changes are required, needs palnning																		
	1 instance of o Nokia MEC (proprietary)									available																				
	1 instance of o NOKIA CloudBand (proprietary, providing network slicing and virtualisation)									available																				
	11 x Dell PowerEdge T630 compute servers 700+ vCPU cores, 1TB+ RAM and 100TB of HDD storage									available - shared with other projects, better to specify the amount of resources needed for each demonstrator and schedule																				
	1 x NVIDIA V100														Risk: can be tricky to integrate to OpenStack					Needs scheduling for using										
	VPN Access to the control plane									available																				
	8 Samsung Android handsets									available - needs schedule for using					Not 5G compatible															
	Support for Demonstrators/Demonstrations																													
	Installation of MATILDA platform in the testbed													Starts after review			Risk: educated guess			Needs guidance/help from the partners developing the components, integration task leaders										
	Access administration																													
	General support																													
	Equipment/Network administration/maintenance																													
	General Software Installation/Update																													
ITL_INC- High Resolution Media on Demand & Banking on the Cloud	High Resolution Media on Demand & Banking on the Cloud																													
	Phase I: Integration @ CNIT																													
	Integration specification, Docker Compose integration																													
	Docker Compose on Matilda at CNIT, testing at CNIT												Initial release for Genoa Review Meeting																	
	Phase II: Integration @UNIVBRIS Testbed																													
	Deeper functionality integration @remote sites (re:application logic)																													
	Deployment work @Bristol Testbed																	Deployment @Bristol Testbed												
	Phase III: Final Integration, Performance testing																													
	Final Integration, Deployment at UNIVBRIS testbed																													
	Multiple user testing, edge/backend testing, asynchronous model updating @remote/CNIT/Bristol																													
	Expansion with extra features, Further experimentation, exploitation, etc. (open-ended)																													
	Possible Target Demo: MWC 19																													
	Possible Target Demo: EUCNC 19																													
	Possible Target Demo: MWC20																													

	Task	dic-17	gen-18	feb-18	mar-18	apr-18	mag-18	giu-18	lug-18	ago-18	set-18	ott-18	nov-18	dic-18	gen-19	feb-19	mar-19	apr-19	mag-19	giu-19	lug-19	ago-19	set-19	ott-19	nov-19	dic-19	gen-20	feb-20	mar-20	apr-20
EXXPERT - Testing 4.0 - Distributed System Testing	**Testing 4.0 - Distributed System Testing**																													
	Exxpertsystems Testbed Integration Phase 1:																													
	DATA IO - Data Acquisition - (e.g. CAN or FlexRay etc.)																													
	COMMUNICATION MODULE - Connection MGMT (Monitoring and Resending)																													
	DATA STORAGE- Implementation Level 1																													
	HMI API / GUI (mobile data monitoring)- Implementation Level 1																													
	Configuration Management- Implementation Level 1																													
	Network Management- Implementation Level 1																													
	Data Transmission- Implementation Level 1																													
	Protocol Definitions- Implementation Level 1																													
	Exxpertsystems Testbed Integration Phase 2:																													
	DATA IO - Data Abstraction - (e.g. CAN or FlexRay etc.)																													
	COMMUNICATION MODULE -Lookup Table for sent IDs																													
	DATA STORAGE- Implementation Level 2																													
	HMI API / GUI (mobile data monitoring)- Implementation Level 2																													
	Configuration Management- Implementation Level 2																													
	Network Management- Implementation Level 2																													
	QoSs- Implementation Level 1																													
	Data Transmission- Implementation Level 2																													
	Protocol Definitions- Implementation Level 2																													
	Demonstration at the review meeting																													
	Exxpertsystems Testbed Integration Phase 3:																													
	HMI API / GUI (Mobile Data Monitoring)- Implementation Level3																													
	Configuration Management- Implementation Level 3																													
	Network Management- Implementation Level 3																													
	- Cache Management																													
	QoSs- Implementation Level 2																													
	Protocol Definitions- Implementation Level 3																													
	Final demo																													
Industry 4.0 Smart Factory – Inter and Intra-Enterprise Integration	**Industry 4.0 Smart Factory – Inter and Intra-Enterprise Integration**																													
	Phase 1:Development and Testing at BIBA																													
	Data Acquisition																													
	Communication module																													
	Data Storage																													
	GUI																													
	Configuration Management																													
	Network Management- Implementation Level 1																													
	Phase 2: Preparation for Bristol Testbed																													
	Communication module																													
	Monitoring component																													
	Data Storage																													
	GUI (mobile data monitoring)																													
	Configuration Management																													
	Network Management																													
	QoSs																													
	Data Transmission																													
	Protocol Definitions																													
	Demonstration at the review meeting																													
	Phase 3: Testbed Deployment at Bristol																													
	Communication module																													
	Monitoring component																													
	Data Storage																													
	Network Management																													
	QoSs																													
	Protocol Definitions																													
	Final demo																													

Annex 2: Preliminary Definition of Tests

The following test definitions are preliminary and subject to change. Specific thresholds for KPIs will be reported in future deliverables.

End User Performance Testing

Table 1: Bandwidth Allocation Testing.

Number	1	**Type**	End User Performance
Title	Bandwidth Allocation		
Relevant UCs	All		
Validation method – Tests	Tests to be performed measuring the end-user experienced data rates against the provisioned ones for a number of data sessions (specific tools might be needed) and for various scenarios. Tests will focus on measuring data rates: • Of various provisioned slices (network level) • For the UCs applications (application level) • Under various scenarios requiring bandwidth allocation to be adjusted according to the predefined rules (being specified at application or network level) For bandwidth-intensive scenarios, towards verifying that it is possible to achieve the 5G-PPP bandwidth KPIs once supported by the radio access network.		
KPIs	**KPI:** Data Rates measured in Mbps. **Success Criteria:** It shall be verified that: • The achieved maximum and guaranteed data rates are in-line with the provisioned QoS class of the end-to-end slice that corresponds to each application/UC • The achieved maximum and guaranteed data rates are in-line with the provisioned QoS class of the sub-slices that correspond to specific application component interfaces.		
Components	End-to-End		
Testbed	All		

Table 2: Network Delay/Latency Testing.

Number	2	**Type**	End User Performance
Title	Network Delay/Latency Testing		
Relevant UCs	All esp. UC1, UC2, UC3		
Validation method – Tests	Tests to be performed measuring the delay/latency during a number of data sessions (specific tools might be needed). Tests will focus on measuring latency on: • User and Control Plane • For various provisioned slices (network level) • For the UCs applications (application level)		
KPIs	**KPI:** Latency measured in ms. **Success Criteria:** It shall be verified that: • The achieved latency is in-line with the provisioned QoS class of the slice • Where required latency restrictions are met		
Components	End-to-End		
Testbed	All		

Table 3: End-to-End QoS provisioning and maintenance Testing.

Number	3	**Type**	End User Performance
Title	End-to-End QoS provisioning and maintenance		
Relevant UCs	UC3		
Validation method – Tests	Tests to be performed measuring the end-user experienced QoS in terms of guaranteed, maximum/best achievable QoS, resources prioritisation over other traffic (including pre-emption policies enforcement): • Of various provisioned slices (network level) • For the UCs applications (application level) • For various applications/operational scenarios where pre-emption is required (following predefined policies, and adheres to SLAs).		
KPIs	**KPI:** Data Rates measured in Mbps. **Success Criteria:** It shall be verified that pre-emption follows predefined policies, and adheres to SLAs.		
Components	All		
Testbed	All		

Table 4: High Density Signalling Testing.

Number	4	**Type**	Functional
Title	High Density Signalling		
Relevant UCs	UC5		
Validation method – Tests	Tests to be performed will focus on ensuring the support of several public lighting poles, introducing a high volume of signalling traffic to be supported by the network.		
KPIs	**Success Criteria:** % of signalling messages successfully received.		
Components	End-to-End		
Testbed	All		

MATILDA Solution Components and Functionality Testing

Table 5: 5G-ready Applications Development Testing.

Number	5	**Type**	Functional
Title	5G-ready applications development using MATILDA Toolkit		
Relevant UCs	All		
Validation method – Tests	Tests to be performed are related to the 5G-ready applications' development and more specifically to: • Application/component development and wrapping so that it becomes cloud-native, including code/wrapping verification, and assessment of the MATILDA Development and Wrapping Toolkit • creation/edition of application service graphs on the basis of the MATILDA metamodels so that it becomes 5G-ready, including verification of understandability, completeness, assessment of metamodels' and MATILDA Application graph editor, • creation/edition of application/components' runtime policies at component level, through the MATILDA Policy Editor.		
KPIs	**Success Criteria:** Successful migration of an on-premises developed Application to a 5G-ready version of by using the MATILDA 5G-ready Application Development Toolkits.		
Components	Development and Wrapping Toolkit, Application graph editor, Policy Editor		
Testbed	UBITECH, CNIT		

Table 6: 5G-Ready Applications Lifecycle Management Testing.

<table>
<tr><td>Number</td><td>6</td><td>Type</td><td>Functional</td></tr>
<tr><td>Title</td><td colspan="3">5G-Ready Applications Lifecycle Management through MATILDA Marketplace</td></tr>
<tr><td>Relevant UCs</td><td colspan="3">All</td></tr>
<tr><td>Validation method – Tests</td><td colspan="3">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
instantiation
etc.
the handling of the user rights for various, different profiles/functions for different users/stakeholders/roles.</td></tr>
<tr><td>KPIs</td><td colspan="3">Success Criteria:
Successful performance of the functionalities that have been specified to be performed through the MATILDA marketplace interface on a per user/role/functionality basis.
Consistency maintained between the information shown through GUIs with the actual repository information, and the specified rules on a per user/role/functionality basis.</td></tr>
<tr><td>Components</td><td colspan="3">MATILDA Marketplace interface, Component Repository, Application Graph Repository</td></tr>
<tr><td>Testbed</td><td colspan="3">All</td></tr>
</table>

Table 7: 5G-ready Applications Orchestration Testing.

Number	7	**Type**	Functional
Title	5G-Ready Applications Orchestration		
Relevant UCs	All		
Validation method – Tests	Tests to be performed are related to the real-time deployment of a 5G-Ready Application through MATILDA, including: • delivery of Real-Time deployment planning of the 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 5G-ready application following a continuous match-resolve-act approach • support of re-active reconfigurations of application deployment based on monitoring data and analytics.		
KPIs	**Success Criteria:** Successful performance of the functionalities related to the optimized, real-time deployment of a 5G-ready Application. Optimised deployment 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 deployment of a 5G-ready Application		
Components	5G-Ready Applications Orchestration, Optimisation Engine, Policy Engine, Intelligent Proxy, Execution Manager,		
Testbed	All		

Table 8: 5G-ready Applications Deployment Monitoring Testing.

Number	8	**Type**	Functional
Title	5G-Ready Applications Deployment Monitoring		
Relevant UCs	All		
Validation method – Tests	Tests to be performed are related to the real-time and historical monitoring of a 5G-Ready Application, 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 regression analysis/predictive and prescriptive analytics/data mining etc. 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 and extraction of Analytics. The obtained results/information are valid, useful, versatile depending on the nature of application, effective towards undertaking corrective actions, and processing is sophisticated leading to advanced conclusions.		
Components	5G-Ready Applications Orchestration, Stream Aggregator, Analytics		
Testbed	All		

Table 9: Lifecycle Management of Application Instantiation Testing.

Number	9	**Type**	Functional
Title	Lifecycle Management of Application Instantiation		
Relevant UCs	All		

Validation method – Tests	Tests to be performed include:
	• the extraction of the slice intent from the service graphs definitions on the basis of the MATILDA metamodels
	• the lifecycle management of the application instantiation in terms of application components' and their chains' graph:
	  ○ allocation of resources (slice) according to the graph to instantiate the application on the infrastructure
	  ○ modification of the graph of application components (at runtime, based on events, etc.) to
	• termination of application instance operation.		

KPIs	**Success Criteria:** Successful performance of functionalities related to the lifecycle management of application instantiation on infrastructure. Consistency maintained between the VAO information and the actual application/application graph state.		
Components	Vertical Application Orchestrator (VAO)		
Testbed	All		

Table 10: OSS/BSS Operation Testing.

Number	10	**Type**	Functional
Title	OSS/BSS operation		
Relevant UCs	All		
Validation method – Tests	Tests to be performed include: • the interface between the Vertical Application Orchestrator 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 VNFs 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

Table 11: Lifecycle Management of Slices Testing.

Number	11	**Type**	Functional
Title	Lifecycle Management of Slices		
Relevant UCs	All		
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 modification of the slice based on application runtime policies, and/or infrastructure constraints/policies • 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. Consistency maintained between the Slice Manager information and the actual slice state.		
Components	Slice Manager		
Testbed	All		

Table 12: Management of Network Services Testing.

Number	12	**Type**	Functional
Title	Management of Network Services		
Relevant UCs	All		
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 VNFs • the instantiation of VNFs for a specific slice • the re-use and parameterization of VNFs from multiple tenants • the modification of VNFs based on runtime policies (possibly) • the termination of the VNF instance 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	Network Function Virtualisation Orchestrator (NFVO)		
Testbed	All		

Table 13: Management of Infrastructure Resources Testing.

Number	13	**Type**	Functional
Title	Management of Infrastructure Resources		
Relevant UCs	All		
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 NFVO and CSM/VAO • the instantiation of infrastructure resources upon request (after resolution of resources availability) • multi-tenancy • modification of resources allocation per tenant on the basis of runtime policies underpinning application/application components/slices/etc. operation • release of resources upon request.		
KPIs	**Success Criteria:** Successful performance of the functionalities that are related to the infrastructure resources management on a per PoP basis.		
Components	Virtual Infrastructure Manager (VIM)		
Testbed	All		

Table 14: Management of Wide-area Network Resources Testing.

Number	14	**Type**	Functional
Title	Management of Wide-area Network Resources		
Relevant UCs	All		
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	Wide-area Network Manager (WIM)		
Testbed	All		

Table 15: Multi-site Resource Management Testing.

Number	15	**Type**	Functional
Title	Multi-site Resource Management		
Relevant UCs	All		

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 Multi-Site Resource Manager and the PoPs regarding their availability of resources
	• deployment of applications/application components at Network Service Provider's edge facilities using:
	  ◦ end-user location and mobility information
	  ◦ 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, including migration to a different PoP, if needed.		

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	Multi-site Resource Manager, VIM, WIM
Testbed	CNIT

MATILDA General Solution Testing-Evaluation

Table 16: User Friendliness Evaluation.

Number	16	**Type**	Other
Title	User Friendliness		
Relevant UCs	All		
Validation method – Tests	User friendliness of the MATILDA solution interfaces to user/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		

Table 17: Speed of Application Deployment Evaluation.

Number	17	**Type**	Other
Title	Speed of Application Deployment		
Relevant UCs	All		
Validation method – Tests	Tests to be performed will include measuring 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

Table 18: Speed of Application Deployment Evaluation.

Number	18	**Type**	Non-Functional
Title	Interoperability with various Access Networks (WAN, LTE, 5G, LoRaWAN/LTE-M, etc.)		
Relevant UCs	UC1, UC2		
Validation method – Tests	Tests to be performed verifying the MATILDA framework operation over various Access Networks (WAN, LTE, 5G, etc.). In particular the following will be tested: • Definition of Access technology at MATILDA metamodels and delivery of an end-to-end network slice to the relevant access network nodes • Evaluation of the QoS definition for the various access network technologies and verification of delivery of a network slice/service with the defined performance/QoS. Tests will include network and application layers' performance measurements for the various UCs and Infrastructure. Especially the over LoRaWAN, LTE-M or 5G access network types will be tested in the context of UC6 at ORO testbed.		
KPIs	**Success Criteria:** Verification of MATILDA framework operation over various Access Networks (WAN, LTE, 5G, etc.)		
Components	Access network nodes, WIM, VIM, OSS/BSS		
Testbed	CNIT, ORO		

Table 19: Deployment across Multiple Infrastructures Testing.

Number	19	**Type**	Functional
Title	Deployment across multiple IaaS		
Relevant UCs	UC3		
Validation method – Tests	Tests to be performed verifying the iMON Dashboard instances are deployed at multiple datacentres and checking if all data and services are properly synced among them.		
KPIs	**Success Criteria:** Successful deployment of an instance of the iMON Dashboard components at multiple datacentres.		
Components	VIM, OSS/BSS		
Testbed	CNIT		

Table 20: Availability Evaluation.

Number	20	**Type**	Functional
Title	Availability		
Relevant UCs	All		
Validation method – Tests	Measuring infrastructure, and committed resources availability throughout a time period after the completion of the MATILDA development stage.		
KPIs	**Success Criteria:** The availability level shall reach a pre-specified threshold of operational time, and will be measured after the completion of the MATILDA development stage. **Relevant KPIs**: (time the service is available) / (total time from service deployment up to the time of measurement)		
Components	All		
Testbed	All		