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

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

DELIVERABLE D1.1

MATILDA FRAMEWORK AND REFERENCE ARCHITECTURE

Due Date of Delivery:	M6 (30/11/2017)
Actual Date of Delivery:	12/12/2017
Work package:	**WP1 – MATILDA Reference Architecture, Conceptualization and Use Cases**
Type of the Deliverable:	R
Dissemination level:	PU
Editors:	CNIT, ATOS, ERICSSON, INTRA, COSM, ORO, UBITECH, ININ, INC, SUITE5, NCSRD, UNIVBRIS, AALTO, UPRC, ITL, BIBA, EXXPERT
Version:	1.0

Call:
H2020-ICT-2016-2

Type of Action:
IA

Project Acronym:
MATILDA

Project ID:
761898

Duration:
30 months

Start Date:
01/06/2017

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 the Authors	
CNIT	Consorzio Nazionale Interuniversitario per le Telecomunicazioni
Franco Davoli, Roberto Bruschi	
ATOS	ATOS Spain SA
Aurora Ramos, Javier Melian	
ERICSSON	ERICSSON
Orazio Toscano	
INTRA	INTRASOFT INTERNATIONAL SA
Kostas Thivaios, Marios Logothetis	
COSM	COSMOTE KINITES TILEPIKOINONIES A.E.
George Lyberopoulos, Helen Theodoropoulou, Ioanna Mesogiti, Konstantinos Filis	
ORO	ORANGE Romania
Jean Ghenta, Marius Iordache, Cristian Patachia, Bogdan Rusti, Horia Stefanescu	
UBITECH	GIOUMPITEK Meleti Schediasmos Ylopoiisi kai Polisi Ergon Pliroforikis EPE
Panagiotis Gouvas , Anastasios Zafeiropoulos	
ININ	INTERNET INSTITUTE Ltd.
Luka Koršič, Mojca Volk, Janez Sterle	
INC	Incelligent
Panagiotis Demestichas, Kostas Tsagkaris, Athina Ropodi, Nikos Stasinopoulos, Stavroula Vassaki, Marinos Galiatsatos, Aristotelis Margaris, Dimitris Cardaris	
SUITE5	SUITE5 Data Intelligence Solutions
Fenareti Lampathaki, Minas Pertselakis, George Sideratos	
NCSRD	NATIONAL CENTER FOR SCIENTIFIC RESEARCH “DEMOKRITOS”
Eleni Trouva	
UNIVBRIS	UNIVERSITY OF BRISTOL
Anderson Bravalheri, Dimitrios Gkounis, Reza Nejabati, Dimitra Simeonidou	
AALTO	AALTO-KORKEAKOULUSÄÄTIÖ
Tarik Taleb, Afolabi Ibrahim, Bagaa Miloud	
UPRC	UNIVERSITY OF PIRAEUS RESEARCH CENTER
Dimosthenis Kyriazis, Chrysostomos Symvoulidis, Ilias Tsoumas	
ITL	ITALTEL
Pietro Paglierani	
BIBA	BIBA – BREMER INSTITUT FÜR PRODUKTION UND LOGISTIK GMBH
Kay Burow, Karl Hribernik, Zied Ghrairi	
EXXPERT	EXXPERTSYSTEMS GMBH
Earon Beckmann, Jörn Albrecht, Kishore Duganapalli	

Disclaimer

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

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

Copyright

The MATILDA Consortium consists of:

CONSORZIO NAZIONALE INTERUNIVERSITARIO PER LE TELECOMUNICAZIONI

ATOS SPAIN SA (ATOS)

ERICSSON TELECOMUNICAZIONI (ERICSSON)

INTRASOFT INTERNATIONAL SA (INTRA)

COSMOTE KINITES TILEPIKOINONIES AE (COSM)

ORANGE ROMANIA SA (ORO)

EXXPERTSYSTEMS GMBH (EXXPERT)

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

INTERNET INSTITUTE, COMMUNICATIONS SOLUTIONS AND CONSULTING LTD (ININ)

INCELLIGENT IDIOTIKI KEFALAIOUCHIKI ETAIREIA (INC)

SUITE5 DATA INTELLIGENCE SOLUTIONS LIMITED (SUIT5)

NATIONAL CENTER FOR SCIENTIFIC RESEARCH "DEMOKRITOS" (NCSRD)

UNIVERSITY OF BRISTOL (UNIVBRIS)

AALTO-KORKEAKOULUSAATIO (AALTO)

UNIVERSITY OF PIRAEUS RESEARCH CENTER (UPRC)

ITALTEL SPA (ITL)

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

Table of Contents

1 Executive Summary

The vision of MATILDA is to design and implement a novel holistic 5G end-to-end services operational framework tackling the overall lifecycle of design, development and orchestration of 5G-ready applications and 5G network services over programmable infrastructure, following a unified programmability model and a set of control abstractions.

This document elicits several requirements for those tasks by investigating various use cases that span across various verticals, such as media on demand, emergency communications, smart cities, manufacturing and automotive industry. These use cases are presented according to the standpoint of several stakeholders, such as providers for both computing and telecommunication infrastructure, application developers, network functions developers, service providers and final consumers.

Additionally, this document presents a meticulous review of the state of the art regarding not only multi-site virtual infrastructure and intelligent application orchestration mechanisms, but also application and virtual network function development toolkits and modern marketplaces. Gaps in the state of the art are identified, a set of requirements taking into account existing technologies and trends are extracted and proposals on how to fill them are discussed.

As a result, a reference architecture is derived, considering the most recent trends for the development of cloud-native applications with current standardization efforts from the telecommunications industry related to infrastructure virtualization and networks. The proposed architecture is designed to provide end-to-end flexibility for 5G-ready applications, focusing on scalability, resilience and performance optimization, and therefore, to support the development of new business models that can deliver high quality services to the end-user.

This deliverable provides the basis for the design and implementation work that will be performed in the work packages 2, 3 and 4.

2 Introduction

5G is seen as a key enabler to new business models that can promote economic and social growth [IHS-2017]. This new technology will not only further enhance radio performance in the next mobile generation, pushing capacity, transmission rates and connectivity even in challenging situations, but also make extensive use of softwarisation and programmability in order to provide a highly integrated infrastructure with end-to-end flexibility. In less than a decade from now, telecommunication networks and computing resources will be integrated in order to create a high capacity infrastructure that makes use of convergent technologies for both mobile and fixed access, bringing ubiquity and new unique service opportunities [5GPPP-2015].

Envisioned native capabilities of 5G networks, especially network slicing, edge computing and multi-tenancy, will support the introduction of digital technologies in multiple sectors, empowering different verticals, such as high-quality media on demand, emergency communications, smart cities, manufacturing and automotive industry. Therefore, appropriate abstractions are desired to lower the barriers for creating 5G-ready applications that are able to satisfy business and user necessities. The purpose of the MATILDA project is to provide such abstractions by stablishing a holistic framework that unifies the development, deployment and operation for this new kind of applications. Moreover, MATILDA will also provide intelligent mechanisms to automate most of those processes, focusing on scalability, resilience and performance optimization.

Accordingly, this deliverable aims to present the overall architecture of the MATILDA framework, including its main components and artefacts, as a result of a requirement analysis. To achieve this, key concepts are initially proposed in Section 3 as guidelines for the architectural approach. The terminology followed by this document is defined and the high-level architectural approach is presented to set the baselines for the rest of the sections. Furthermore, a number of actors and stakeholders for the MATILDA framework are identified and their roles are elaborated. Key expectations from some of those stakeholders are then extracted by means of a survey realized by partners from the Romanian market. This allows a clear definition of the objectives anticipated for 5G.

Based on these objectives, a methodology for designing the framework is then stablished, as described in Figure 2.0.1. This methodology uses two different approaches to elicit requirements. The first approach is to create use cases with realistic economic targets that span across various verticals' domains, require high bandwidth, low latency and ultra-reliable communications, and are in alignment with the MATILDA high level architectural view. In Section 4, those use cases are carefully inspected in order to picture the diversity of the 5G ecosystem and facilitate the definition of a generic architecture, flexible enough to demonstrate all the programmable capabilities expected in the 5G network. As a result, a set of application-inspired requirements is derived.

The second approach is to investigate the technology domains that the MATILDA platform focuses on. In addition to the use case derived requirements, 5G-related technology requirements are extracted in Section 5. To accomplish this, a meticulous review of the state of the art technologies is performed, regarding the main components of the high-level view of the architectural framework; specifically, the development toolkit, the marketplace, multi-site virtual infrastructure mechanisms (VIM) and intelligent application orchestration mechanisms. Within each technology category, the current capabilities of the existing technologies are reviewed, including challenges. Proposals on how the MATILDA framework will extend the state of the art are discussed. Each technology category contributes to technology requirements to facilitate defining a generalized architecture.

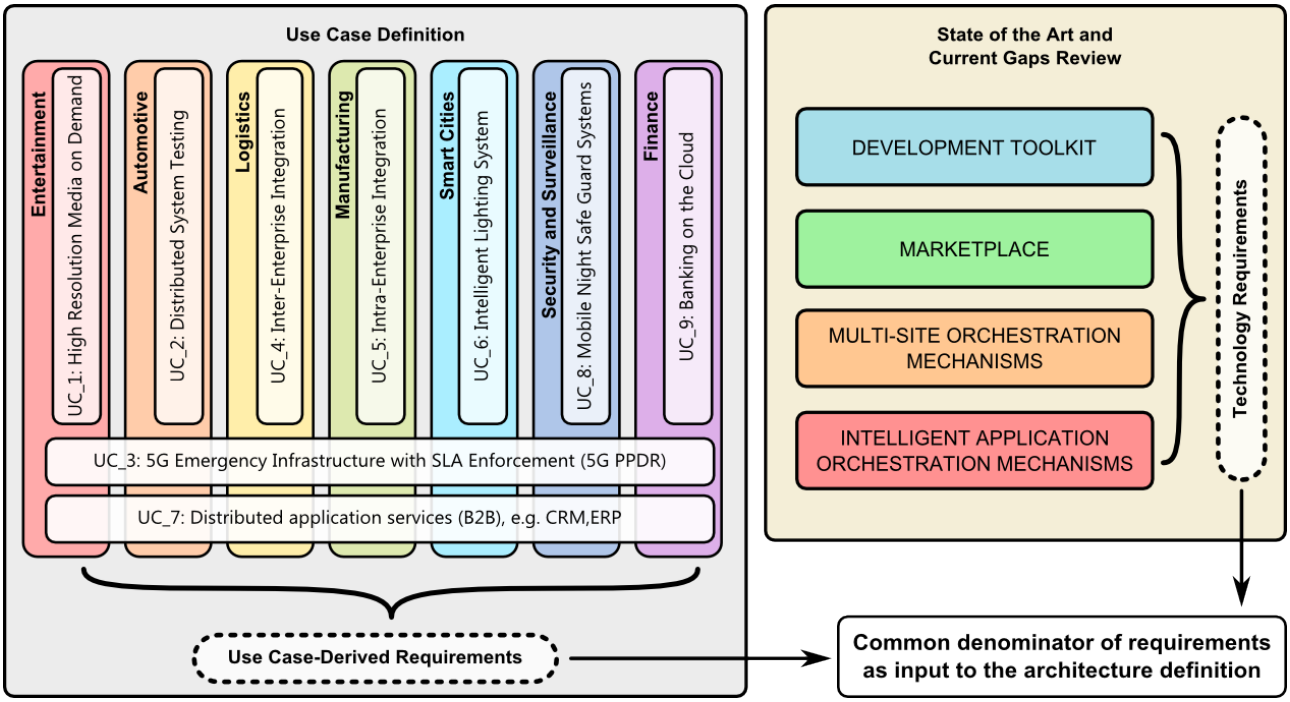


Figure 2.0.1: Methodology for designing the MATILDA framework reference architecture. This methodology elicits requirements by investigating example use cases and the state of the art. [UC: use cases, PPDR: Public Protection and Disaster Relief, CRM: Customer Relationship Management, ERP: Enterprise Resource Planning]

In Section 6, the MATILDA requirements are combined and generalized, providing input for the design of the reference architecture, which is then presented. The reference architecture details its components, combining modern approaches, such as the Service Mesh design (an evolution of the cloud-native applications paradigm) [Calçado-2017], with existing standardization efforts, such as the European Telecommunications Standards Institute (ETSI) Network Functions Virtualisation (NFV) Management and Orchestration (MANO) [ETSINFV-2014a]. A consistency check of the proposed architecture against MATILDA requirements is then performed and the compliance of the proposed architecture to well-known standards is discussed.

Finally, Section 7 summarizes the contribution of the deliverable and provides an outlook of the future work, including a brief description of how the presented work will interact with upcoming work packages.

3 MATILDA Project: Key Aspects

One of the main objectives of the 5G technology specification certainly resides in the enablement and support of a new class of vertical applications with very heterogeneous but extremely challenging performance and operating requirements. To this end, the 5G community is embracing novel well-known, still evolving technological frameworks such as Network Functions Virtualization (NFV) and Multi-access Edge Computing (MEC). Both frameworks are based on the unrestrainable "softwarisation" process in the telecommunication field, which is expected to transform network operator infrastructures into distributed datacentres with advanced IT virtualization capabilities.

In this context, MEC and NFV frameworks will have clear and well-separated objectives. As stated by the ETSI MEC working group in [ETSIMEC-2016a], "***Mobile Edge Computing*** *uses a virtualisation platform* ***for running applications*** *at the mobile network edge.* ***Network Functions Virtualisation*** *provides a virtualisation platform to* ***network functions"****.*

The infrastructure that hosts their respective applications or network functions is quite similar. In order to allow operators to benefit as much as possible from their investment, it would be beneficial to reuse the infrastructure and infrastructure management of NFV to the largest extent possible, by hosting both VNFs (Virtual Network Functions) and mobile edge applications on the same or similar infrastructure [ChaoHu-2015]. Subject to gap analysis, this might require a number of enhancements (e.g. regarding the sharing of resources with NFV Management and Orchestration, etc.).

Thus, from a perspective of service/application lifecycle management, NFV and MEC are frameworks with a highly complementary nature, that might be independently applied over a "softwarised" telecommunication infrastructure. Even if operated by different stakeholders (i.e., MEC by vertical industries and NFV by telecom service industries), both frameworks provide orchestration modules able to acquire resources as-a-Service from the VIM layer and manage the lifecycle of virtual machines containing components of their applications or network functions over the aforementioned resources.

However, from a functional perspective, these two frameworks are closely related. A trivial remark in this respect is that applications need the networking layer to communicate with 5G users and the public Internet. This however becomes less trivial in the case of 5G vertical applications. In such case, the application is dynamically and cognitively managed and composed by its own (MEC) orchestrator, which might take dynamic decisions based on the capabilities offered by the network and the status of network resources. On the network side, the NFV Orchestrator (NFVO) composes and manages the lifecycle of network services as chains of VNF instances, depending on OSS/BSS requirements. In turn, the network OSS/BSS provides interconnectivity among vertical applications' components and connected things in terms of "network slices" (i.e., instances of NFV services managed by the NFVO) [Hattachi-2015, Zhou-2016, Samdanis-2016, 3GPP-2017].

In this scenario, the 5G ecosystem will be clearly designed according to a hierarchical orchestration paradigm, whose architectural details, especially the ones towards vertical industries, are still largely unexplored.

The MATILDA project aims to cope with these open issues using a top-down approach, and, specifically, to design a holistic set of tools, mechanisms and architectural components to enable state of the art cloud applications (e.g., microservices-based applications, where each microservice regards an independently orchestratable component, along with control mechanisms supported via service-to-service communication schemes as realised in a service mesh approach) to rely on and to benefit from the performance/operational advantages of 5G infrastructures and services. In order to achieve this, state of the art cloud applications metadata have to provide indications regarding their infrastructure-oriented needs that can be exploited by orchestration mechanisms to optimize the exploitation of 5G ecosystems' capabilities. Towards this direction, a blend of cloud orchestration mechanisms with the enablement of appropriate network functionalities at the infrastructure level has to be realised.

According to the MATILDA vision, the vertical application orchestrator will be enabled to acquire as-a-Service customized 5G network slices to directly connect mobile terminals with application components, which, in turn, will be hosted in local computing facilities. The MATILDA framework will be designed to cope with the aforementioned functionalities, while assuring the highest possible levels of scalability, autonomicity, and flexibility.

The remainder of this section is organized as follows. Section 3.1 introduces the main terminology that will be used in the project specifications. Section 3.2 provides a first description of the MATILDA architectural approach, and Section 3.3 introduces the main stakeholders involved in the MATILDA framework.

3.1 Terminology – Definitions

Table 3.1.1: The MATILDA main terminology.

Term	Description
Cloud-native Software	A piece of software is cloud-native if it is developed according to a methodology that fully exploits the advantages of the cloud computing delivery model and, therefore, is ready to be deployed to a cloud environment [Brown-2014]
5G-ready Application	A distributed application consisting of independently orchestratable cloud-native components able to take advantage of both the programmable network and computational infrastructure.
Application Component	A cloud-native software entity, which is part of a 5G-ready application, and provides a defined set of functionalities. Each component should comply with the requirements of the cloud-native components (e.g., use interfaces for binding with other components, be horizontally scalable by design, be agnostic to the infrastructure)
Application Graph	An Application Graph represents a template that defines a 5G-ready Application. This template is serializable and can be used as a basis for instantiation. An application graph can be ungrounded (i.e. non-instantiated) or grounded (i.e. instantiated).
Service Mesh	A service mesh is a dedicated overlay for handling component-to-component communication. It is responsible for the reliable delivery of requests through the complex topology of components that comprise an application graph. In practice, the service mesh is typically implemented as an array of lightweight component intelligent-proxies which are deployed alongside the component, without the latter needing to be aware.

Intelligent Proxy	A lightweight, programmable, protocol-specific proxy that is installed on top of the cloud-native component and is able to abstract and support several layer-7 functionalities. Communication among proxies is addressed as Service Mesh Data Plane. The behaviour of the proxy is performed by a logically centralized Control Plane, while the proxy itself can dynamically load several layer-7 virtual functions (L7VF).
L7VF (Layer 7 Virtual Function)	A layer-7 function that can be dynamically loaded by the Intelligent Proxy. Several L7VFs can be loaded simultaneously in one proxy using a "chaining" paradigm.
Application Orchestrator	A main module of the MATILDA architecture that manages the 5G-ready Application Graph lifecycle. To do so, it interacts with the programmable resources across various domains (to ensure optimized allocation of the necessary resources and connectivity). Moreover, the Application Orchestrator is responsible for enforcing a specific policy defined by the Service Provider.
Network Slice	A network slice is a logical infrastructure partitioning with appropriate isolation, allocated resources and optimized topology to serve a particular purpose of an application graph.
Network Function (NF)	"*Functional block within a network infrastructure that has well-defined external interfaces and well-defined functional behaviour*." Source: [ETSINFV-2014c]
Virtual Network Function (VNF)	"*Implementation of an NF that can be deployed on a Network Function Virtualisation Infrastructure (NFVI).*" Source: [ETSINFV-2014c]
Physical Network Function (PNF)	"*Implementation of an NF via a tightly coupled software and hardware system.*" Source: [ETSINFV-2014c]
Virtual Network Functions Forwarding Graphs (VNF-FG)	"*Graph of logical links connecting NF nodes for the purpose of describing traffic flow between these network functions.*" Source: [ETSINFV-2014c]
Virtual Network Function Orchestrator (NFVO)	"*Functional block that manages the Network Service (NS) lifecycle and coordinates the management of NS lifecycle, VNF lifecycle (supported by the VNF manager) and NFVI resources (supported by the VIM) to ensure an optimized allocation of the necessary resources and connectivity.*" Source: [ETSINFV-2014c].

Slice Manager	The module of the MATILDA architecture that is responsible to manage the entire lifecycle of a Network Slice, i.e. planning, provision and deprovision.
Virtual Infrastructure Manager (VIM)	"*Functional block that is responsible for controlling and managing the compute, storage and network resources, usually within one operator's Infrastructure Domain.*" Source: [ETSINFV-2014c]
Wide Infrastructure Manager (WIM)	The architectural component that applies SDN concepts to the wide area network (WAN), to create a centrally-controlled overlay network that intelligently uses a distributed SDN-enabled infrastructure.
Operational Support System (OSS)	The term Operational Support System (OSS) generally refers to the systems that perform management, inventory, engineering, planning, and repair functions for communications service providers and their networks.
Business Support System (BSS)	The BSS typically refers to the systems used by telecommunication service providers in order to deal with customers, such as taking orders, managing customer data, processing bills and collecting payments.
Network Service (NS)	"*A network service is the composition of Network Functions and defined by its functional and behavioural specification. The Network Service contributes to the behaviour of the higher layer service, which is characterized by at least performance, dependability, and security specifications. The end-to-end network service behaviour is the result of the combination of the individual network function behaviours as well as the behaviours of the network infrastructure composition mechanism.*" Source: [ETSINFV-2014c]
5G Marketplace	A storefront including a set of repositories for reusable application components, application graphs and NFs (mainly VNFs and L7VFs).
Deployment Policy	A set of objectives and constraints that have to be taken into account during the placement of a 5G-Ready Application Graph over a network slice. The satisfaction of the requirements results in the definition of a "slice intent", which has to be offered through the instantiation of the appropriate slice.
Runtime Policy	A set of rules related to the runtime adaptation of the application graph.
Integrated Development Environment (IDE)	A software application that provides comprehensive facilities for software development.

Multi-site Management	Management of virtualized resources and software instances across multiple sites/points of presence.
Multi-tenancy	A platform/framework, running as a single logical instance, able to support multiple services from different client organizations in a fully isolated fashion.
Metamodel	"*Modeling is the process of converting our perceived view of the reality into a representation of it. Metamodeling is the process of specifying the requirements to be met by the modelling process or establishing the specifications which the modelling process must fulfil. […] The metamodel embodies the properties that are abstracted from all models*" Source: [Gigch-1991]

3.2 High Level View of MATILDA Architectural Approach

The vision of MATILDA is to design and implement a novel holistic 5G end-to-end services operational framework tackling the overall lifecycle of design, development and orchestration of 5G-ready applications and 5G network services over programmable infrastructure, following a unified programmability model and a set of control abstractions. MATILDA aims to devise and realize a radical shift in the development of software for 5G-ready applications, as well as virtual and physical network functions and network services, through the adoption of a unified programmability model, the definition of proper abstractions and the creation of an open development environment that may be used by application as well as network functions developers. Intelligent and unified orchestration mechanisms are going to be applied for the automated placement of the 5G-ready applications and the creation and maintenance of the required network slices. Deployment and runtime policy enforcement is provided through a set of optimisation mechanisms stablishing deployment plans based on high-level objectives and runtime adaptation of the application components and/or network functions, based on rules defined on behalf of a services provider. Multi-site management of the cloud/edge computing and IoT resources is supported by a multi-site VIM, while the lifecycle management of the supported Virtual Network Functions Forwarding Graphs (VNF-FGs), as well as a set of network management activities, are provided by a multi-site NFV Orchestrator (NFVO). Network and application-oriented analytics and profiling mechanisms are supported based both on real-time and a posteriori processing of the collected data from a set of monitoring streams. The developed 5G-ready application components, applications, virtual network functions and application-aware network services are made available for open-source or commercial purposes, reuse and extension through a 5G marketplace.

As already mentioned, to the best of our knowledge, no holistic framework exists that supports the tight interconnection among the development of 5G-ready applications, the creation of on-demand networking and computational infrastructure in the form of an application-aware network slice and the activation of the appropriate networking mechanisms for supporting industry vertical applications. MATILDA follows a layered approach with the consolidation of some functionalities into specific layers. The MATILDA layers, along with the main artefacts and key technological concepts comprising the MATILDA framework per layer, are depicted in Figure 3.2.1.

The Applications Layer corresponds to the Business Service and Business Function layer and takes into account the design and development of 5G-ready applications per industry vertical, along with the specification of the associated networking requirements. The associated networking requirements per vertical industry are tightly bound together with their respective 5G-ready applications' graph, which defines the business functions, as well as the service qualities of the individual application. At the application layer, the application graph together with the corresponding networking requirements are

coupled together to deliver a 5G-ready application. The Orchestration Layer is strategically positioned below the Application layer, in order to support the dynamic on-the-fly deployment and adaptation of the 5G-ready applications to its service requirements, by using a set of optimisation schemes and intelligent algorithms to provision needed resources across the available multi-site programmable infrastructure.

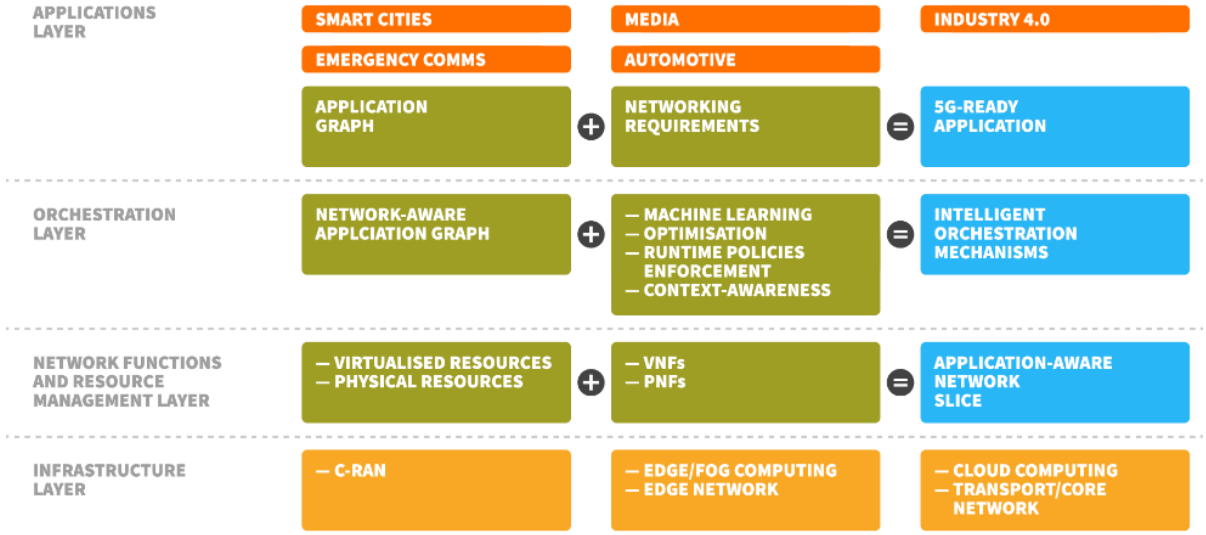


Figure 3.2.1: MATILDA High Level Architectural Approach.

The Orchestration Layer refers to both the application components and the attached virtual network functions, as well as the set of networking resources needed to chain the network functions together to provision a complete network service package. The orchestration layer also includes a set of intelligent mechanisms for optimal deployment, strategic placement, runtime policies enforcement, data mining and analysis and context awareness support for holistic 5G end-to-end network services. This layer utilises a set of high-level, high-performing and high-class software modules, which integrate network awareness with application graph functionalities, problem-solving mechanisms and sophisticated functions such as machine learning algorithms, optimization schemes, runtime policy enforcement agents and context-awareness functionalities, to build a smart and intelligent orchestration service system. The MATILDA orchestration layer was cleverly aligned directly above the network functions and resource management layer, in order to have an almost native impact on the virtual and the physical resources that will be used by the orchestrator to provision the 5G end-to-end network slices.

The Network Functions and Resource Management Layer sits between the Orchestration and the Infrastructure Layers. This positioning is deliberately made so that network functions and resources are as close as possible to the Orchestration Layer and the layer housing the physical infrastructure.

The Network Functions and Resource Management Layer refers to the implementation of the resource management functionalities over the available programmable infrastructure and to the lifecycle management of the activated virtual network functions. This layer uses both physical and abstracted virtualised resources, along with virtual network functions and physical network functions, to deliver application-aware network slices.

The Infrastructure Layer consists of the data communication network spanning across a set of cloud computing and storage resources such as C-RAN, Edge/Fog computing resources, Edge network, Transport/Core network, etc.

The key technological concepts and artefacts comprising the proposed MATILDA framework and constituting its unique selling points are listed hereinafter:

- a conceptual architecture for supporting the provision of 5G end-to-end services tackling the overall lifecycle of design, development and orchestration of 5G-ready applications and 5G network services over a programmable infrastructure.
- a set of metamodels representing the vertical industry applications' components and graphs, the virtual network functions and forwarding graphs.
- an innovative collaborative development environment supporting the design and development of 5G-ready applications and VNF-FGs, including a web-based IDE, verification and graphs composition mechanisms.
- an orchestrator that undertakes the responsibility of optimal deployment and orchestration of the developed applications over the available programmable infrastructure -taking into account a set of objectives and constraints, as well as the defined policies-, along with the instantiation of the required network functions for the support of the infrastructural-oriented functionalities. Policies enforcement is going to be supported by a context awareness engine, able to infer knowledge based on a set of data monitoring, analytics and profiling production streams.
- a multi-site wide infrastructure manager supporting the multi-site management of the allocated resources per network slice, along with a multi-site NFVO supporting the lifecycle management of the network functions embedded in the deployed application's graph as well as supporting a set of network monitoring and management mechanisms.
- a set of interfaces towards OSS/BSS systems of service providers, targeting at managing the lifecycle of network functions over the network slices, based on requests provided by the MATILDA orchestrator.
- a novel analytics and unified profiling framework consisting of a set of machine learning mechanisms, as well as design time profiling and runtime profiling towards the production of advanced analytics, and software runtime profiling.
- a marketplace including an applications' and virtual network functions' repository and a set of mechanisms for supporting the diverse 5G stakeholders.

The approach followed, as detailed in section 6.1, is going to be based on the specification of open interfaces and APIs as well as the adoption and extension of existing open-source frameworks.

3.3 Actors/Stakeholders, Roles and Expectations

The MATILDA value chain consists of a number of stakeholders currently existing in the Information and Communication Technology (ICT) market. However, the stakeholders' role is expanded with more responsibilities depending on their level of involvement in the MATILDA framework development and operation. More specifically, the main MATILDA stakeholders/stakeholders' roles are the following:

- **Infrastructure Providers**, with the main role to provide infrastructure resources (network resources, storage space, compute resources) to third parties dynamically, either by handling the MATILDA application graphs of the 5G applications or by exposing programmable interfaces directly to third parties (possibly *Service Providers*) for the support of the MATILDA framework. Depending on the nature of the required infrastructural resources, and the assets of the stakeholders, this role can be split to many smaller ones, performed by one or more stakeholders, namely:
  - **Telecommunication Infrastructure Providers**- undertaken by Mobile (and possibly also Fixed) network infrastructure providers- operating a programmable (5G) network infrastructure spanning from the radio and/or fixed access to the edge, transport and core network. In the MATILDA value chain, the main *Telecommunication Infrastructure Providers*' role is to provide network resources for 5G services/applications. Particularly, either they perform the programmable dynamic provisioning of network resources by handling VNF graphs and network services, or they directly expose programmable network interfaces to third parties (possibly *Service Providers*) for the support of the MATILDA framework.

  - **Cloud Infrastructure Providers**, operating centralized (locally) or distributed (in more than one location) cloud/edge deployments and offering compute and storage resources in a programmable way. In the MATILDA value chain, the role of *Cloud Infrastructure Providers* is to provide cloud resources, by either handling directly MATILDA application graphs and performing the programmable dynamic provisioning of cloud resources, or by exposing programmable interfaces of the cloud deployment directly to third parties (*Service Providers*) to be used for the support of the MATILDA framework.

- **Application Developers** i.e. software engineers/programmers who develop 5G applications. These 5G applications must be modular and "chainable-by-design", i.e. consisting of a number of chainable application components, abstracted from physical and networking resources as well as "reactive-by-design". Each chainable application component shall adhere to a specific metamodel, while the combination of several chainable application components will formulate the application graph. Depending on the application, the role of the *Application Developers* can include the creation of the 5G application graphs. In the MATILDA value chain, *Application Developers* announce (to *Service Providers* or to *Infrastructure Providers* depending on the business model), in a formal way, the resource (networking, storage, compute) requirements that should be satisfied when the application graph is instantiated, so that the 5G- application is deployed over MATILDA.
- **VNF/PNF Developers** creating virtual network functions or programmable physical equipment, thus practically implementing the MATILDA programmable network layer functions, and delivering these components/functions to the *Service Providers* and *Infrastructure Providers*.
- **Service Providers,** responsible for creating the application graph, thus making it 5G-ready, and possibly –depending on the agreements between *Telecommunication and Cloud Infrastructure Providers* and *Service Providers*- instantiating a 5G application/service on top of reserved programmable infrastructural resources (based on the 5G application graph), and providing it to the end-users/service customers. In other words, the *Service Provider* comprises the mediation entity between the *Application Developers*, the *Infrastructure Providers*, and the *Service Customers*.
- **Application Stores/Marketplaces**, through which the MATILDA applications (applications' updates, upgrades, etc.) are available to be downloaded.
- **Service Consumers**, which are the individual or corporate users to finally consume the 5G applications/services while being static and/or on the move. *Service Customers* practically correspond to the set of vertical industry clients that will benefit from MATILDA.

In the case of a commercial MATILDA deployment, one stakeholder may undertake more than one role, or a stakeholder's role may be split to more than one stakeholder depending on the nature of the 5G application/service and the resources that it requires. For instance:

- the *Telecommunications Infrastructure Provider* may undertake also the role of the *Cloud Infrastructure Provider* depending on their infrastructure assets, or in some cases even the role of the *Service Provider*,
- the roles of the *Application Developers* and the *Service Provider* could be played by the same stakeholder,
- the role of the *Service Provider* can be played by a vertical (if it has the know-how), or by a software house providing the 5G-ready application as a service to a vertical, while
- as *Service Consumers* many stakeholders can be considered depending on the nature of the applications, e.g. they can be either customers of the verticals or the verticals themselves, etc.

In order to identify key expectations for these stakeholders, and the 5G ecosystem in general, a survey was conducted with key partners in the Romanian market, as presented in Annex 1. The results of this survey, along with a potential extension in the future are going to be used towards the preparation of the MATILDA exploitation and business plans, based on the work that is going to be realised in WP7.

4 5G Network-Aware Application Use Cases

By virtue of the high degree of innovation and complexity related to the development of the 5G infrastructure, the aims of the MATILDA project, as described in Section 3, are significantly ambitious. In order to design and develop an architecture that enables the achievement of these aims, a meticulous survey and analysis about 5G Application use cases is indispensable. Use cases allow the project not only to highlight its full reach but also to facilitate the identification of functional and non-function requirements. This section enumerates a few important use cases that can be carried out with the help of the MATILDA framework. They were chosen to highlight different verticals such as media & entertainment, emergency infrastructure, manufacturing, smart cities and automotive industries. For each use case, corresponding requirements for the framework are derived.

4.1 Use Case 1: High Resolution Media on Demand

4.1.1 Scenario Description

In recent years, online video content has shaped the development of the Internet, and has become one of the most successful business in the entertainment industry. Under the constant pressure of dwindling revenues, due to increasing investments in infrastructure and intense competition, Telecommunication Service Providers indeed see in this market segment a big opportunity to increase their margins. This kind of service is not only more profitable than simply providing connectivity, but also aligned with the softwarisation of the 5G network infrastructure and the related emergence of mobile edge cloud computing [5GPPP-2016]. Particularly, a Service Provider (usually but not exclusively a telecom company) can make use of a 5G framework to create an application that provides an innovative high-resolution media on-demand service to its mobile and fixed users (the Service Consumers).

The 5G media sector is now witnessing the emergence of many novel services based on High Definition Content delivery. In particular, the provision of immersive video services during crowded events is one of the most attractive use cases under development by many important Telecom Equipment Vendors. During a crowded event, a high number of Service Consumers are concentrated in a small geographical area for a relatively short time, typically ranging from few hours to a week. Well-known examples of crowded events are sport events or concerts in stadiums, exhibitions hosted by dedicated venues and international events spread over a university campus or even an entire city. Immersive video services enable the possibility of sharing High Definition video contents, anywhere, at any time and via any device, with the opportunity of (perceived) real-time interaction with the system and among Service Consumers. In addition to traditional content delivery systems, in which Service Consumers only play the role of content consumers, the most innovative immersive video services offer the possibility to create and share video contents in real-time - for example within a pre-defined group of peers.

In this type of services, the media content (video) must be sent to a number of diverse user terminals such as smartphone, PC, TV, each of them supporting different screen size, resolution, and codecs. A high value service, for the Service Provider, is related to flash events, such as football matches, that must be rapidly delivered to the Service Consumers**,** to be consumed in a very restricted time-frame. The main challenge is to effectively provide the wanted content in the shortest possible time, adapted to the end-user device, while optimizing bandwidth utilization and power consumption. This is important especially when the contents are sent to mobile devices such as smartphones or tablets.

In this challenge Telecom Equipment Vendors may see a business opportunity to create specialized VNFs that can be integrated by Application Developers to compose the final 5G application related to the service. Since this kind of VNFs are based on caching the media content in advance as close as possible to the Service Consumer and performing on-demand adaptations for the specific user device,

network and edge computing resources (provided by third-party Infrastructure Providers or the Service Provider itself) are required.

This use case is based on the Enhanced Video Services (i-EVS) framework provided by ITL. In addition to traditional video content delivery services, i-EVS also offers the participants of a crowded event the capability to locally create and share video contents in real-time. The i-EVS framework, in fact, leveraging multi-access edge computing and network function virtualization principles, enables the use of high performance computing resources at the network edge, offering the possibility – during the crowded event - of sharing high definition video contents, anywhere, at any time and via any device, with the opportunity of (perceived) real-time interaction with the system and among users.

4.1.2 Objectives

From this point on, the possibility to create, share or receive low-latency, high definition video contents, anywhere and with any device in a defined area, with real-time interaction with the system, will be referred as immersive video services. Immersive video services are attracting a lot of interest, but they still impose very stringent requirements, due to the huge needs of compute, storage and networking resources that High Definition video brings about.

The main objective of this use case is offering immersive video services, as described above, during a crowded event or in places of particular relevance (for instance, touristic), exploiting edge computing capabilities, and leveraging the overall framework and architectural view developed by the MATILDA project. The content delivery system will be implemented by the ITL i-EVS framework, consisting of a specific VNF for HD video processing and an ad-hoc App that can be downloaded to the smart phone or tablet of the end users. The final goal is deploying and validating the use case summarized above in the Bristol Is Open infrastructure, made available by the University of Bristol.

Challenges and Innovation

The 5G media sector is witnessing the emergence of many innovative services based on High Definition video processing. In this context, the provision of immersive video services is one of the most attractive use cases, and many industrial players are presently working to implement it. In particular, in crowded events a high number of end users are concentrated in a small geographical area for a relatively short time interval. The i-EVS framework innovation in this field consists of enabling the possibility of sharing HD video contents, anywhere, at any time and via any device, with the opportunity of (perceived) real-time interaction with the system and among users. This use case will also make use of the advanced capabilities, developed by the MATILDA project, to orchestrate the virtualized high-performance computing resources needed by the i-EVS system and located at the network edge.

Intelligent orchestration mechanisms will also be essential, in particular related to the dynamic management of the network slice over which the i-EVS based 5G-ready application will run. As will be further elaborated in the next paragraphs, during crowded events the resource usage can be highly varying with time, but usually with an almost-deterministic behaviour. This will allow to dynamically adjust the amount of resources allocated to the slice, so as to efficiently cope both with peak-traffic and inactivity periods, thus optimizing resource usage. At the same time, the overall framework developed by the MATILDA project will also continuously monitor the resource usage, so as to promptly react to any unpredictable event, which could result in unexpected traffic peaks.

To support immersive video services during crowded events, the network infrastructure has to overcome several challenges, namely: the high density of user devices (more than tens of thousands per km^2), high data-rates to enable HD video streaming (at least 7 Mbps per user) and low latencies (in the range 10-50 ms) to guarantee the expected level of perceptual quality and user experience (QoS/QoE). Moreover, HD video services require large IT resources, both in terms of compute and

storage capabilities at the network's edge. For this reason, ad hoc IT resources must be used, including hardware accelerating devices for video processing, by the i-EVS VNF.

4.1.3 Scenario Workflow

To provide immersive video services to Service Customers, the Service Provider will at first need to accommodate all the required functional blocks onto the available infrastructure. At deployment time, the service provider will thus need to reserve programmable resources with cloud and telecom infrastructure providers. Based on those resources, it will then select the VNF-FGs that satisfy the set of network requirements of the desired application graph, taking into account also optimization aspects. At this point, the service can finally be instantiated.

In this specific use case, the service provider must reserve edge computing capabilities next to the final user, to perform cache and device targeting adjustments. It is also possible that the service provider would like to reserve core computing capabilities to pre-process the original content, or at least to create a central point of distribution. The MATILDA framework will enable the creation of a proper network slice that will include all the computing, storage and networking resources needed to provide the above-described immersive video services.

In particular, the 5G-ready application will make use of the i-EVS VNF that ITL, as a Telecom equipment provider will offer through the Marketplace.

The ITL i-EVS framework is mainly based on a VNF, which encompasses the following VNFC:

- dBM: VNFC which gathers profile information for each user who has subscribed to the service;
- sTORE: VNFC responsible to manage storage resources, which allows:
  - caching of uploaded videos;
  - download of videos previously uploaded by users belonging to the same group or a super-user (such as the event organizer);
- vTU: VNFC which processes the videos shared among the users; two different VNFCs can be configured:
  - vTU-GPU: VNFC for massive video processing (transcoding/trans-sizing/trans-rating) leveraging off-the-shelf Graphical Processing Units, available in the server at the edge; processing can be performed both on real-time video, and on already cached video contents;
  - vTU: VNFC for video processing (transcoding/trans-sizing/trans-rating) based on standard virtual compute resources; processing can be performed both on real-time video and on those already cached.

A high-level description of the i-EVS VNF is summarized in Figure 4.1.1, where also the ITL VNF Manager is shown.

For the three VNFCs, internal connectivity within the i-EVS VNF must be provided. Moreover, each VNFC must also have an external connection point, to be reached by Service Consumers when accessing the offered Immersive Video Services. In particular, the dBM VNFC manages the Service Consumer registration process, and all the related information. Conversely, the vTU instances process the highly demanding (in terms of needed bandwidth and low latency) video traffic. Finally, the sTORE VNFC will transfer the locally stored video contents to other storage systems, located elsewhere, to make them retrievable by the Service Consumers for a given period of time after the event occurred.

From the networking point of view, the traffic handled by the vTU and the sTORE VNFCs can be highly demanding, and presents specific peculiarities. The video content traffic typically presents a specific pattern, with peaks at predictable times, and periods of low or null activity. For instance, when the i-EVS is deployed at a venue where an exhibition takes place for a period of a few days, no activity by the Service Consumers is expected at night, while peaks are usually experienced at specific time periods. Thus, periods of low activity for the vTU can be used by the sTORE to transfer the video contents to a centralized storage system.

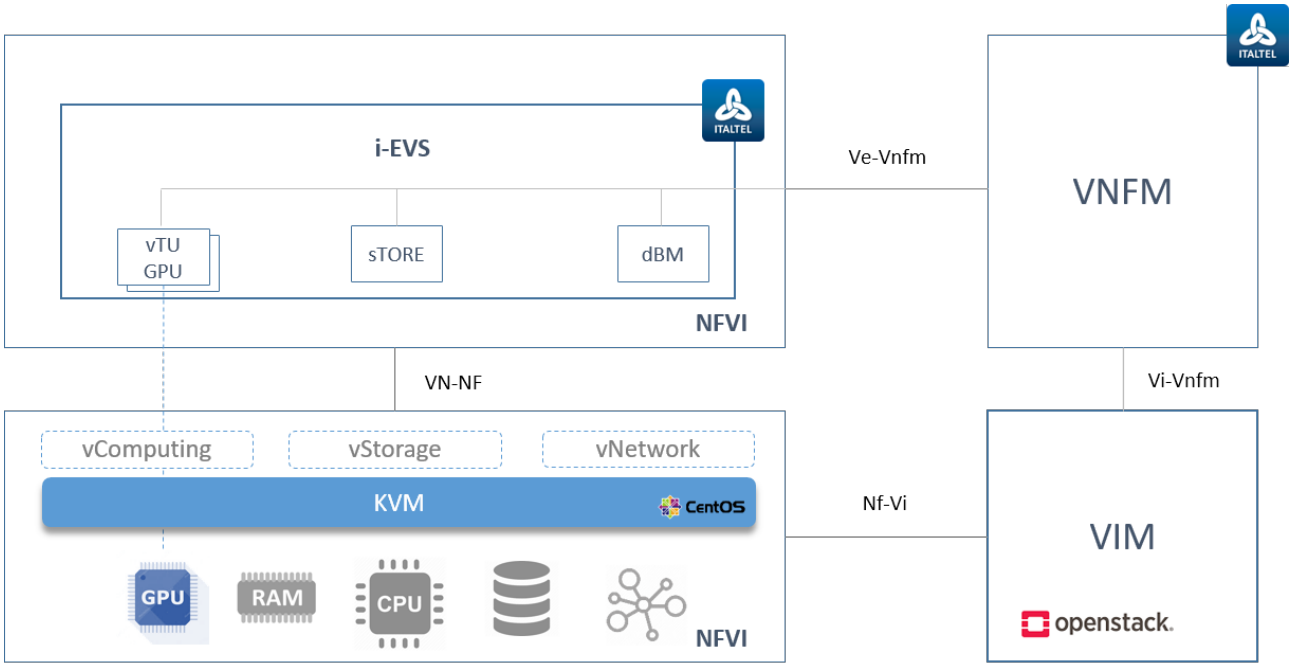


Figure 4.1.1: Block scheme of the ITL i-EVS VNF, ITL VNF Manager, Network Function Virtualized Infrastructure (NFVI) and its Manager (VIM).

To cope with these needs, network slicing and adjustable bandwidth capabilities should be provided by the MATILDA framework, as envisaged by the MATILDA high level architectural view. In particular, the capability to adjust resource allocation to effectively manage the infrastructure in an efficient way will be based on the context-aware policy enforcement developed within the MATILDA framework. Also, resource usage shall be continuously monitored, to promptly detect unexpected events, and react, for instance to face abrupt traffic peaks due to unpredictable events.

To this end, resources will need to be constantly monitored. In fact, at runtime, several events can induce anomalies in the metrics being monitored. These anomalies will be processed by the analytics and profiling framework (for example, by making use of machine learning techniques) and may trigger changes by the orchestrator (thanks to the context awareness engine) in the installed infrastructure to enforce the defined policies.

In this use case, Service Consumers start their interaction with the i-EVS framework by downloading the i-EVS App; this can happen whenever a Service Consumer enters the area covered by the i-EVS service. As an alternative, such services are also available through any browser, in the area where the event occurs.

4.1.4 Use Case-Derived Requirements

ID	UC1_1
Unique Name/Title	Network Slicing Capability.
Priority	High
Type	Performance
Brief Description	Network Slicing is needed by the vTU and the sTORE VNFCs of the i-EVS VNF to handle different types of traffic, with specific, time-varying features.
Rationale	The video content traffic has specific, strict requirements in terms of bandwidth and latency, and typically presents a time-varying pattern, with peak periods and periods of low activity. The traffic originated by the sTORE VNFC must exploit the low activity periods of the vTU to transfer elsewhere the stored video contents. To effectively handle these activities, network slicing is needed.

Validation method/Relevant KPI	Test of the correct allocation of network slices to the corresponding VNFC(s). KPI: throughput (in Mbps) of each network slice.

ID	UC1_2
Unique Name/Title	Adjustable Bandwidth allocation.
Priority	High
Type	Performance
Brief Description	Adjustable bandwidth allocation is needed to cope with the time varying features of the different types of traffic within network slices.
Rationale	The different network slices carry different types of traffic to the i-EVS VNFC, with time varying patterns. To effectively manage network connectivity, the bandwidth allocated to the different slices should vary to adjust to the current traffic conditions.
Validation method/Relevant KPI	Measuring of allocated bandwidth, and comparison with target values. KPI: throughput (in Mbps) of each link.

ID	UC1_3
Unique Name/Title	Low Delay/Latency Guarantees
Priority	High
Type	Performance
Brief Description	Low Delay is required for interactive video sharing.
Rationale	To offer real-time video services and real-time interaction with the system low delay connectivity is needed.
Validation method/Relevant KPI	KPI: measurement of end-to-end delay time. Maximum 100 ms end-to-end delay for real-time applications.

ID	UC1_4
Unique Name/Title	Resource Usage Monitoring
Priority	High
Type	Infrastructure
Brief Description	Monitoring of resource usage to enable VNF scalability
Rationale	Resource usage must be continuously monitored to allow VNF scaling
Validation method/Relevant KPI	Availability of monitoring data; KPI: availability of the monitoring capability.

ID	UC1_5
Unique Name/Title	Policy Enforcement
Priority	High
Type	Functional
Brief Description	The MATILDA framework must be able to dynamically enforce traffic policy rules.
Rationale	The traffic generated by Service Consumers can present specific trends; to optimize network resource usage, ad hoc policies must be enforced, for instance to cope with peak-traffic or inactivity periods.

Validation method/Relevant KPI	Testing performance after the enforcement of a specific policy. KPI: availability of policy enforcement.

ID	UC1_6
Unique Name/Title	VNF Scalability (in/out)
Priority	High
Type	Non-Functional
Brief Description	Scaling (in/out) of the VNFs.
Rationale	As expected in the use case, many users will be offered the same service at the same time, each having its own instance. Furthermore, for each instance multiple applications components may be required.
Validation method/Relevant KPI	Test of scalability of the i-EVS VNF. Scaling in/out of available VNFs

4.2 Use Case 2: Testing 4.0 - Distributed System Testing

4.2.1 Scenario Description

To a great extent, all major industries throughout Europe and the world require a form of distributed communication of Industrial Bus Signals between machines and/or software that fulfils the following necessities:

- High Quality of Service
- Real Time
- Guaranteed Data Delivery with High Data Volume Capabilities (Reliable)
- Interoperable (Plug & Play)
- Modular & Scalable

Presently, compliance with these communication necessities can only be assured in an isolated scenario, i.e. with a dedicated physically connected local network infrastructure, and not between geographically separated locations using Wireless Wide Area Networks (WWAN) communication technologies.

Furthermore, today's systems, especially in the Automotive Industry, are made up of complex integrated systems with several systems, sub-systems and components being developed at geographically distributed locations. Integration and functional testing of these highly coupled systems involves a great deal of logistics, high amount of personnel, and most importantly is time consuming and costly.

FastWAN is an experimental communication technology that was developed as a solution for the enablement of geographically separated real-time industrial test benches. FastWAN presently provides the following features:

- Supports and provides virtual extensions over the Internet for a variety of industry BUS Standards such as AFDX, ARINC 429, Digital, Analog, etc.;
- Ensures on time and on quality delivery for all signal data with efficient data acquisition and packing;

- Deterministic and reliable reconstruction of signals at destination by the use of accurate GPS signal time stamping, configurable fixed delays, and jitter compensation;
- Integrates fast standard data "container" transmission services such as Ethernet UDP;
- Ensures reliable data transmission integrity by way of failure detection (detection of lost or incorrectly ordered packets), as well as recovery mechanisms (packet re-ordering and re-transmission);
- Efficient bandwidth usage with customized proprietary packing methods based on signal priorities and destination requirements;
- Comprehensive system control and monitoring GUI to monitor the data logistics network and virtual connection statistics.

Within the Automotive Industry, FastWAN can enable interconnected integration and functional testing of these end product systems over WWAN infrastructure. The use of the FastWAN technology has proven to lead to significant reduction in system development life cycles times and costs in an environment where system complexity, competition and certification requirements are dramatically increasing. This is achieved by way of enabling earlier system integration testing through the reduction of time and cost impacts with respect to logistic efforts.

Figure 4.2.1 shows an example of the geographically separated interconnection of the emission control system to enable early integration & functional testing for Automobile Industry.

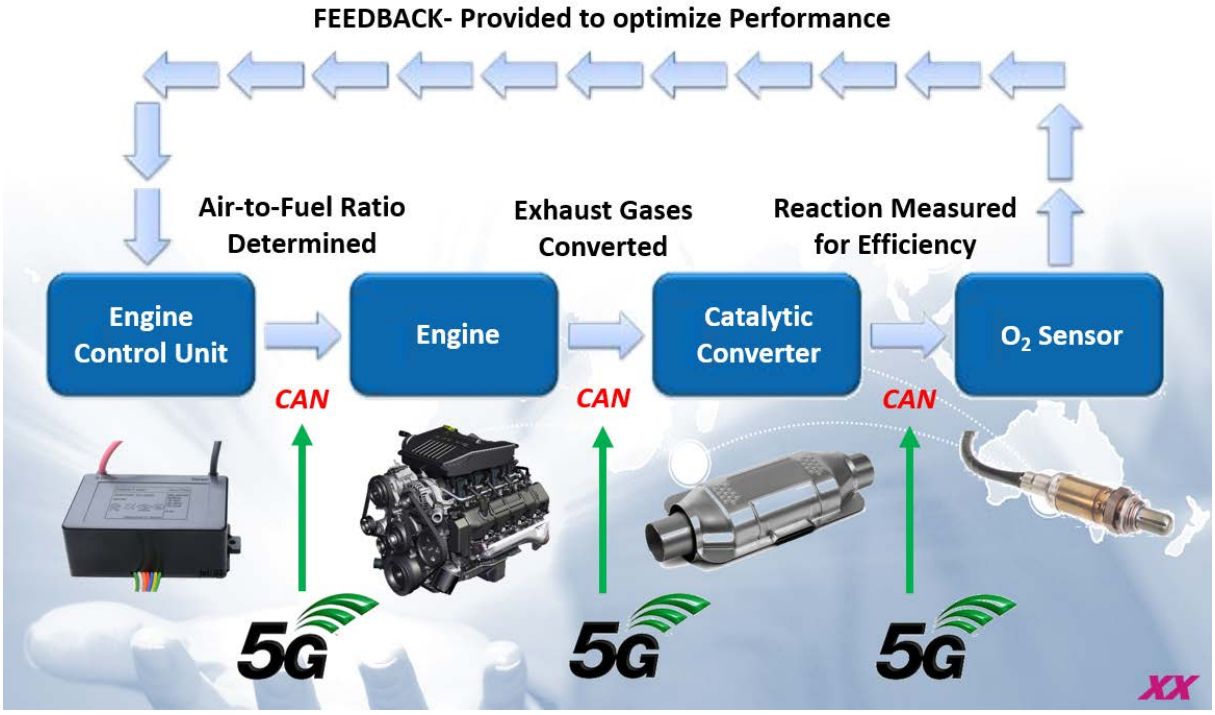


Figure 4.2.1: Example of the geographically separated interconnection of the emission control system to enable early integration & functional testing.

The main actors / stakeholders for this use case are as follows:

- **VNF/PNF Developer:** ExxpertSystems will adopt the role of telecommunication equipment vendor (converting industrial Bus Protocol Signals into Internet Packets and vice versa) by providing a Physical Network Function (PNF); namely, the ExxpertSystems FastWAN solution.

- **Infrastructure Provider:** The data will be sent over a common Internet Infrastructure provided by a Telecommunications Network Provider.
- **5G Application Developer:** Engineers responsible for the creation of the testing system.
- **Service Consumers:** Automobile component, sub-system, system suppliers or original equipment manufacturers (OEMs).
- **Service Providers:** Automobile component, sub-system, system suppliers, OEMs, or third-party source that supplies a high-level automobile testing solution or framework.

Usually the development of test systems for the Automobile Industry is done internally by the same company that want to use it. So, it is common that the roles of 5G Application Developer, Service Consumer and Service Provider are played by the same stakeholder: the final consumer itself.

4.2.2 Objectives

As shown in the Figure 4.2.1 above, an interconnected integration and functional testing system represents a chain of components from a closed loop system that are developed at different sites and/or by different suppliers. Today, it is necessary to bring all these components 'physically' together before Integration and functional tests can be performed. This not only requires all hardware to be fully developed and sent to a common location, but also requires all experts and associated testing hardware and software to be present at one common site before integration and functional testing can be performed. The main objective of this use case is to exploit 5G capabilities to enable the flexible (and potentially remote) interconnection of mobile systems, e.g. automobile systems, sub systems and components.

Challenges and Innovation

The innovation in this use case consists of improving by an order of magnitude the performance, or more specifically, higher capacity, lower latency, enhanced mobility, better accuracy of terminal location, increased reliability and availability for systems and equipment under test. The network infrastructure has to overcome several challenges, namely: high data-rates to enable HD video streaming and large Industrial Bus Signal traffic, as well as low latencies (optimally in the range of 1 ms) to guarantee the expected level of quality (QoS).

4.2.3 Scenario Workflow

The MATILDA architecture will be used to realise the use case defined in this section. In this Use Case, an Automobile Component, Sub-System, System Suppliers or OEM will incorporate the ExxpertSystems FastWAN solution into their test set-up in order to geographically test systems, sub-systems or equipment as if it were a single, locally connected end system. The Automobile Component, Sub-System, System Suppliers or OEM, in the role of 5G Application Developer, will compile an application graph that will be used to compose and define which application components will be used. The graph composer and the marketplace will be used to choose between all the possible components, and the toolkit will be used to edit these components. The programmable (layer 2 to layer 4) infrastructure (that may span from the radio-access to the edge, transport and core network) will be defined. The required Virtual & Physical Network Functions (VNFs & PNFs) required for the application will then be defined and incorporated, ensuring that the VNF/PNF metamodel meets the requirement of the application graph. The 5G Application will be modular and chainable by-design, abstracted from physical and networking resources and reactive-by-design, thus guaranteeing that the application meets the requirements for the MATILDA environment. The Automobile Component, Sub-System, System Suppliers or OEM will play the role of the service provider by integrating this service into their own test benches, or into test benches within the supply chain

4.2.4 Use Case-Derived Requirements

ID	UC2_1
Unique Name/Title	Flexible Bandwidth Allocation
Priority	High
Type	Performance
Brief Description	Flexible bandwidth allocation is needed between geographically distributed systems/sub-systems under test to ensure the integrity and required performance of distributed functional and integration testing.
Rationale	It would be optimal to have a flexible network that supports data rates of up to 10 Mbits/s (Mbps) per Node (FastWAN Unit). At present, a dynamic bandwidth policy is built-in to the FastWAN product; however, the ability to offload this responsibility, to a certain degree, to the 5G network would be beneficial.
Validation method/Relevant KPI	Measuring of bandwidth fluctuations between interconnected test systems, and comparison with target values. KPI: throughput (in Mbps) of each link.

ID	UC2_2
Unique Name/Title	Low Delay/Latency
Priority	High
Type	Performance
Brief Description	Low Delay is required between geographically distributed systems / sub-systems under test to ensure the integrity and required performance of functional and integration testing.
Rationale	Low Delay/Latency is required between geographically distributed systems / sub-systems under test to ensure the integrity and required performance of functional and integration testing. Therefore, strict maximum delay requirements are posed for the links between the different components, sub-systems and systems under test. The optimal Maximum Delay / Latency requirements are as follows: • Inside Germany - Approximately 50 ms latency between nodes • Inside Europe - Approximately 100 ms latency between nodes • Worldwide - Approximately 200 ms latency between nodes
Validation method/Relevant KPI	Measuring of delay fluctuations between interconnected systems / sub-systems under test, and comparison with target values. KPI: delay time (in ms – depending on the test case) of each link.

ID	UC2_3
Unique Name/Title	High Availability
Priority	High
Type	Performance
Brief Description	The test systems interconnection infrastructure/services shall be always available.
Rationale	The testing of distributed test systems poses strict availability requirements to ensure the integrity and success of test campaigns.
Validation method/Relevant KPI	The availability level shall reach 99.99% of operational time, and will be measured after the completion of the MATILDA development stage. Relevant KPIs are: (time the service is available) / (total time from service deployment up to the time of measurement)

ID	UC2_4
Unique Name/Title	Interoperability with Various Access Networks (WAN, LTE, 5G, etc.)
Priority	High
Type	Non-Functional
Brief Description	The infrastructure/services for deploying FastWAN Test Systems shall be supported seamlessly over various Access Networks.
Rationale	Mobility is a key feature for deploying FastWAN Test Systems, implying that they can be served by different access networks depending on their location and the availability of each technology present at each location. Therefore, the FastWAN Test Systems shall be supported seamlessly over various Access Networks; meaning that the underlying MATILDA framework shall be interoperable with various access networks.
Validation method/Relevant KPI	Testing of services' operation when end users are served by different access networks (WAN, LTE, 5G, etc.).

ID	UC2_5
Unique Name/Title	Security & Privacy
Priority	High
Type	Non-Functional
Brief Description	The interconnection of test systems (e.g. test system data, user information/authorisation, etc.) shall be secure, in order to preserve system integrity.
Rationale	Since the interconnection of test systems involves highly sensitive data transfer, all operations must be highly secured and subject to specific access rules. This needs to be provided and ensured by the network infrastructure.
Validation method/Relevant KPI	Testing of access to data / authorization levels for all MATILDA procedures/operations.

ID	UC2_6
Unique Name/Title	Dynamic QoS Provisioning
Priority	High
Type	Functional
Brief Description	Support and enforcement of dynamic QoS provisioning (especially for Jitter Compensation and Packet Loss)
Rationale	Dynamic QoS provisioning will allow for resource management optimisation and meeting performance demands of different FastWAN Test Nodes at all times, while avoiding overprovisioning. A high QoS will guarantee test campaign success and seamless interconnection in cases of several services that are provided simultaneously.
Validation method/Relevant KPI	Testing performance with various FastWAN Nodes and measuring per-case QoS metrics.

ID	UC2_7
Unique Name/Title	Network Programmability
Priority	High
Type	Functional
Brief Description	Network Programmability is required in order to achieve optimal allocation in

	terms of network resources for the different links between the FastWAN components, processing power in FastWAN nodes, etc.
Rationale	Given the fact that the resources required from various test benches vary in time, network programmability will allow for optimisation of resources management based on actual needs per FastWAN Node.
Validation method/Relevant KPI	This requirement can be validated by different FastWAN Node demands.

ID	UC2_8
Unique Name/Title	Network Monitoring
Priority	High
Type	Non-Functional
Brief Description	Network monitoring is necessary in order to deploy, detect issues / problems, reconfigure and reallocate resources.
Rationale	In order to offer undisturbed operations, it is important to monitor whether the deployment of resources took place or to detect any issues that might occur, so that reallocation of resources can take place.
Validation method/Relevant KPI	This requirement can be validated at the design phase.

4.3 Use Case 3: 5G Emergency Infrastructure with SLA Enforcement (5G PPDR)

4.3.1 Scenario Description

5G Emergency Infrastructure and Services Orchestration with Service Level Agreement (SLA) Enforcement Use Case is based on the implementation of a 5G-enabled emergency response pilot provided with the iMON product suite for real-time intervention monitoring and critical infrastructure protection, extended with performance monitoring engines and advanced Operation, Administration and Management (OAM) capabilities of the qMON solution for supporting SLA.

The SLA would mean a contract between the subscriber (MATILDA service provider) and the operator (MATILDA infrastructure provider), specifying the agreed service level requirements and commitments. It typically includes a Service Level Specification (SLS), which basically represents the technical part of the SLA (e.g. service objectives, metrics definitions, measurement of metrics, bandwidth profile details, etc.).

The 5G emergency scenario targets support of a suite of reliable and survivable services and applications on top of a 5G telecom infrastructure providers for emergency response teams both in day-to-day operations and during extreme situations requiring large on-site interventions. 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 (Figure 4.3.1).

The applications and services used in the emergency scenarios are commonly addressed as Broadband Public Protection and Disaster Relief (BB-PPDR) applications and services.

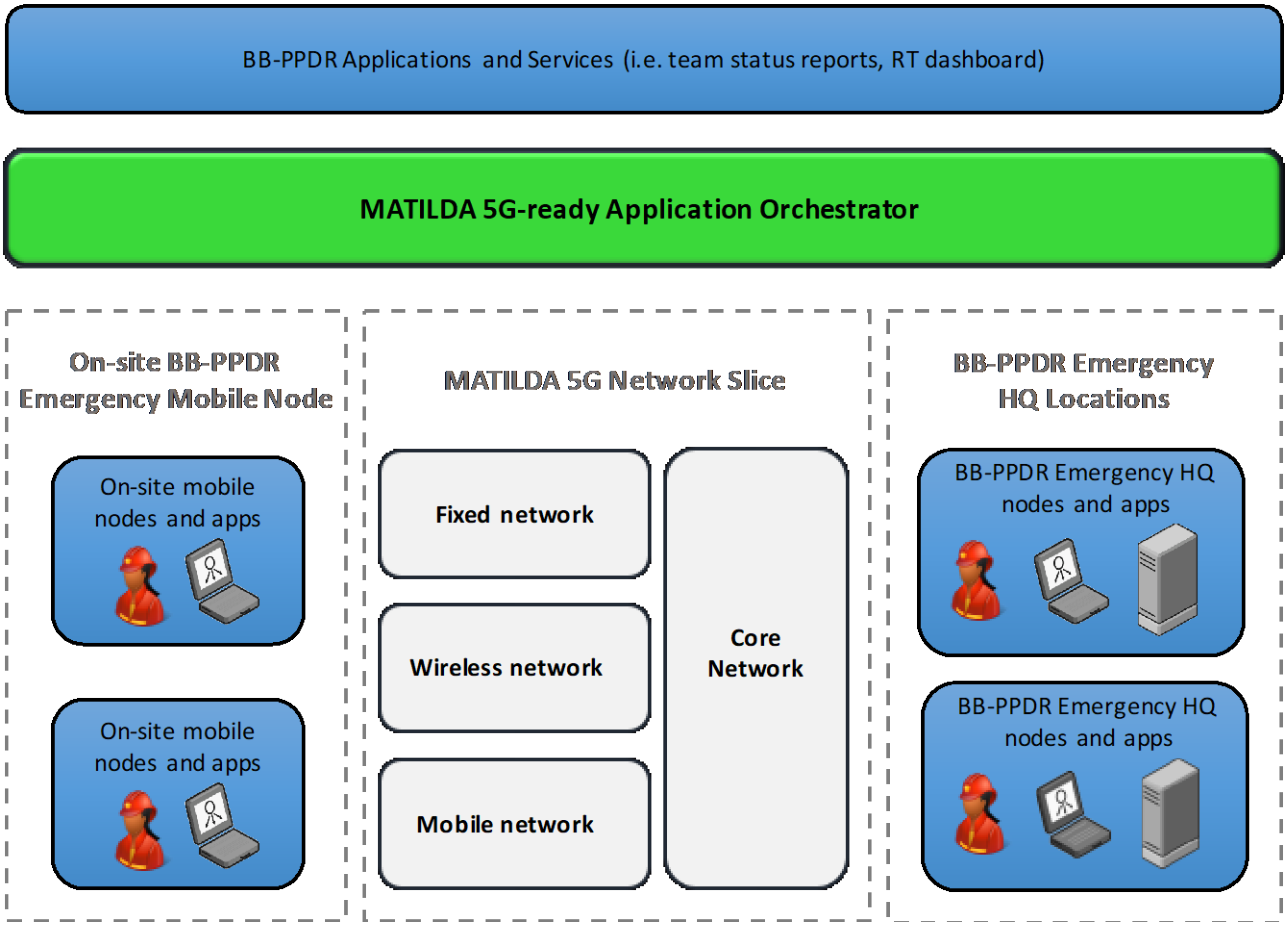


Figure 4.3.1: 5G Emergency building blocks.

For such BB-PPDR applications and services to operate reliably and with high survivability and availability characteristics, support of high-performance communications with added service provisioning intelligence is required in both normal day-to-day conditions and in extreme situations, since the provided emergency applications are executed in a distributed system with distributed service intelligence. The complex requirements for service provisioning on top of a 5G infrastructure and powerful SLS/SLA monitoring are realised in this use case using the iMON solution for real-time emergency intervention monitoring and critical infrastructure protection, extended with qMON performance monitoring engines and advanced OAM capabilities (Figure 4.3.2).

The following actors and stakeholder are part of the MATILDA based 5G emergency ecosystem:

- ***BB-PPDR network operator*** provides and operates ***MATILDA telecom and cloud infrastructure***.
- Each emergency response organization (ERO) has a dedicated **BB-PPDR service provider** functioning as ***MATILDA service provider***.
- ***Emergency response teams and end users*** (policemen, firefighters, rescuers) are in the role of ***MATILDA service consumers***.

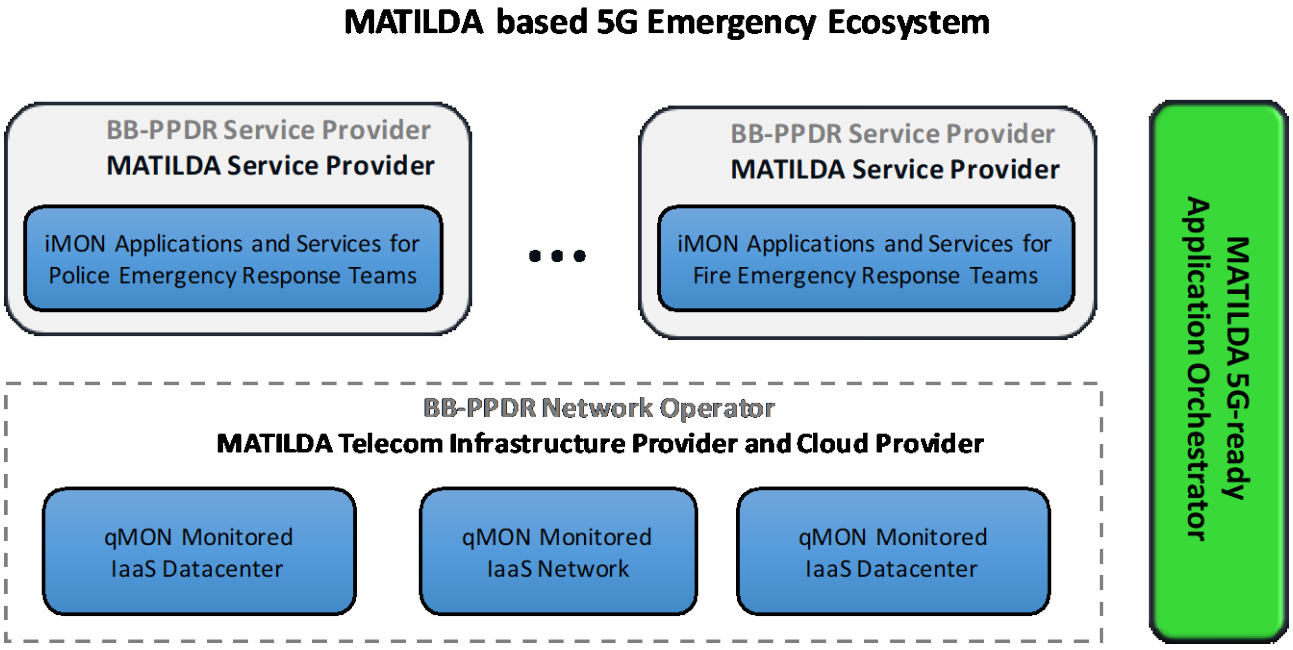


Figure 4.3.2: MATILDA based 5G Emergency Ecosystem.

The pilot environment for real-time emergency intervention monitoring and critical infrastructure protection in a 5G environment will be built based on the iMON product suite. iMON is a product suite designed for use by first responders and public safety agencies and 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. It comprises three main components:

- iMON Core – Disaster Communications Node with smart routing/tunnelling for survivable backhaul connectivity from intervention sites. It includes algorithms for automatic fall back at network failure, pre-defined network selection priority; support for mobile, wireless and fixed networks (5G, 4G, satellite, fixed, WiFi); support for telco network integration (dedicated VPNs with roaming support and QoS management, network authentication mechanisms); Implementation options (portable ruggedized or non-ruggedized, fixed in datacentre, built-in in tactical command vehicle).
- iMON Dashboard – Tactical Dashboard exposing real-time PPDR services for common operational picture, situational awareness and intervention management. It features a rich client HTML5/PHP web application, supporting: real-time common operational picture, integrated video feeds from the field, real-time assets tracking and backlog, data analytics and visualisations, intervention reports and logs; Backend: real-time notifications, exposed APIs, automated hierarchical (group) user and device management.
- iMON Mobile App – Android Mobile application for triage and tracking from the field. Native mobile app with field sensing, time and distance-based location tracking, automated triage reporting (official procedural reporting formats, image attachments, automatically retrieved location data); automatic sync with COP; use of Commercial off-the-shelf (COTS) mobile devices with IP67 International Protection marking (e.g. resistance to dust and water) [ResourceSupply-2008].

The iMON implementation will be realised on top of a 5G telecom and cloud infrastructure with specific MATILDA technology support for added intelligence in service provisioning and policing of distributed applications and network components that have to operate reliably and with high availability and survivability features under extreme emergency conditions.

In addition, the qMON product suite will be used to extend the pilot environment with capabilities for advanced application-level operation, administration and maintenance with real-time performance and quality measurement capabilities of mobile and fixed networks and applications based on distributed agent-based probing and active user emulation. A heterogeneous 5G-enabled environment will be targeted, and specific extensions will be designed and implemented to support quality monitoring and application OAM capabilities for 5G. qMON allows for configurable measurement scenario definitions with automated remote configuration updates on distributed agents, data and measured KPIs collection and central storage, as well as data mining and visualizations using professional Business Intelligence (BI) analytics tools. The solution is adaptable for specific architectures and extendable for KPI, SLA and performance metrics, which allows for flexibility in applying real-time performance and quality validation also in a 5G setting. It enables advanced application-level operation, administration and maintenance with real-time performance and quality measurement capabilities collecting more than 500 relevant KPIs.

The qMON system comprises 4 main parts based either on cloud VMs or dedicated consumer/industrial HW):

- qMON NetworkSensor – Autonomous distributed agent for emulating user (mobile/fixed) activities and active services/applications. Linux-based code for various form factors: VMs, industrial/consumer HW; support for LTE, HSPA, EDGE and other IP/Ethernet based networks;
- qMON Manage – cloud-based back-end agent management and data collection infrastructure;
- qMON Monitor – cloud-based real-time dashboard for KPIs visualization and monitoring;
- qMON Insight – Advanced KPI analysis and visualizations

4.3.2 Objectives

Use of end-to-end SDN and NFV capabilities to build heterogeneous 5G emergency communications infrastructure and services for the PPDR environment.

Challenges and Innovation

The main targeted innovation in this project that will be added in the iMON solution and the qMON quality monitoring and OAM engines is based on the implemented MATILDA technology, particularly the use of MATILDA 5G-ready Application Controller implementation in conjunction with SDN and NFV technologies for virtual channel establishment allowing for transparent communication via one or several available backhaul networks represented as a single virtual channel with SLS/SLA monitoring (Figure 4.3.3). Application context-driven routing with dynamic QoS support and dynamic cross-network admission control and service pre-emption will be supported, while based on the provided implementation, very high availability will be supported with reroute capabilities allowing for (almost) instantaneous fall-back scenarios in case of extreme conditions causing failure of individual networks information driven servicing (Figure 4.3.4). In addition, information-driven user, service and network prioritization will be provided based on network and service contexts.

The examples of an application graph and network-aware application graph are presented in Figure 4.3.3 and Figure 4.3.4, respectively. The terminology used in those figures is the following:

- UA - User Agent (can be a PC running a web application in a browser or mobile app; not in the scope of MATILDA),
- Network components:
  - FW – FireWall,
  - SEC – Security (e.g. VPN concentration/termination),
  - R – Router,

- Q – Quality of Service (QoS) enforcement,
- Con – Connection (i.e. Network path over the 5G Network slice);

- Application components:
  - PHP BL – PHP Business Logic,
  - DB- Database,
  - BLOB – Binary Large Object.

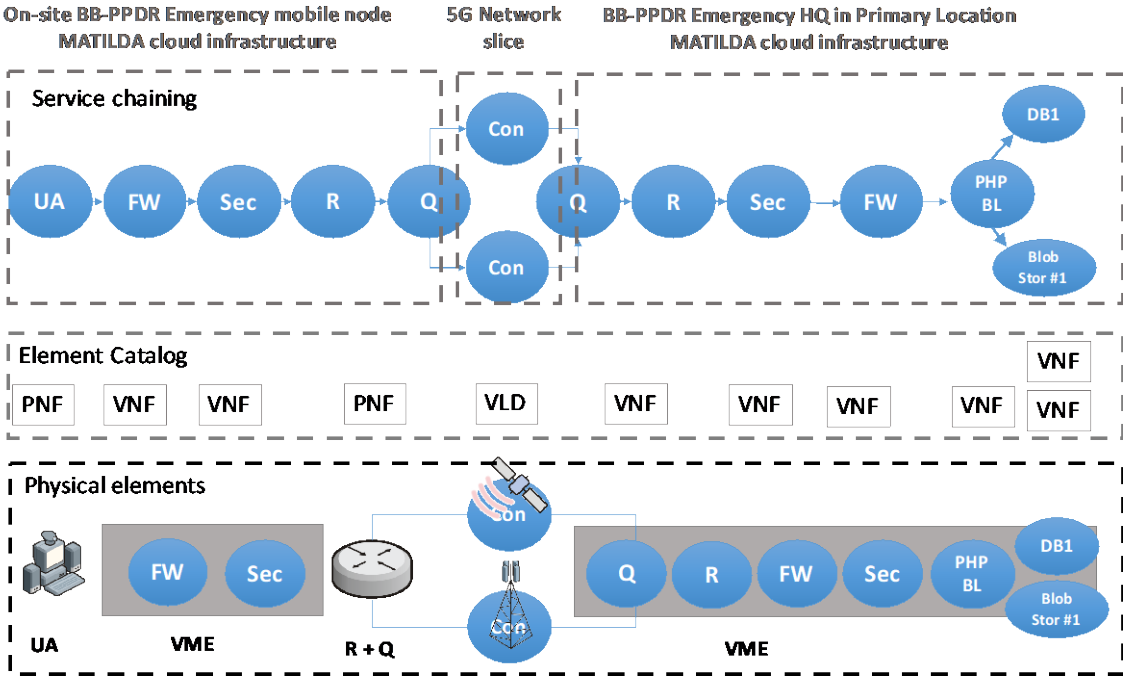


Figure 4.3.3: Sample of an application graph for the emergency applications connecting virtual (VNF) and physical (PNF) elements implemented over the heterogeneous 5G telecom and cloud infrastructure and controlled by the MATILDA 5G-ready Application Orchestrator.

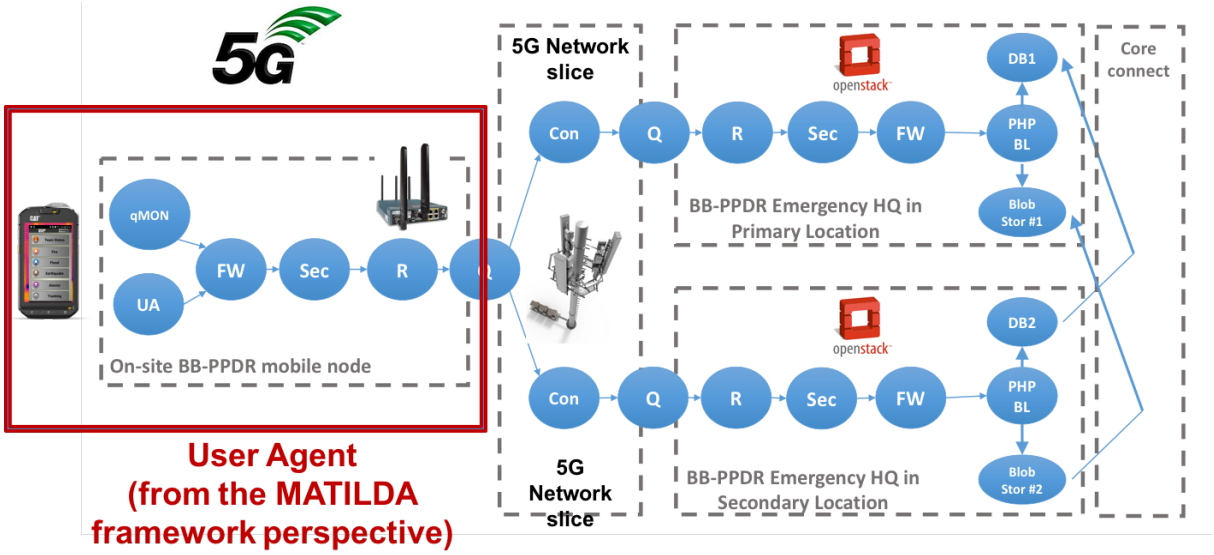


Figure 4.3.4: Sample of a network-aware application graph for support of high performance and survivable emergency services in 5G

Some network components can be deployed in standalone mode or are provisioned in one VME (Virtual Machine Environment), such as FW and SEC functions. Furthermore, some components can be

realized as a PNF, for example a "mobile" router PNF which would provide routing and QoS enforcement functions.

4.3.3 Scenario Workflow

Based on the request by the BB-PPDR service provider (e.g. Police or Fire ERO department), the MATILDA environment will provide end-to-end service (iMON) and network components (e.g. 5G network slice, router and FW) provisioning and orchestration. Following the context of the deployed applications it will also enforce SLA and monitor SLS based on qMON solution elements.

The iMON Dashboard, which represents the backend of the iMON solution, can be established in geographically separated datacentres. As shown in Figure 4.3.4, each instance of the iMON Dashboard is itself a distributed application comprised of the following application components:

- Database - The purpose of this component is to maintain the data of the registered users, alarms, events and reports along with several metadata that accompany them. It is used by two components of the application service graph.
- PHP BL - The purpose of this component is to enable the business logic in the shape of a tactical dashboard, displaying field operatives with location, various alarms and events, and also provide detail reports with multimedia metadata. As a component, it is dependent on the MySQL DB and on a BLOB storage component in order to achieve stateless behaviour regarding the PHP sessions and storage of multimedia files. The native 6inACTION application written in PHP will be used as a base component for wrapping.
- BLOB - The purpose of this component is to provide single shared storage for multimedia files and PHP related data to all PHP BL components in a service graph and is needed to achieve stateless operation of a PHP-based business logic. Furthermore, it also provides the file sync service between two BLOB components. The component is used by two other components of the service graph. The SMB [SAMBA] file share engine and Syncthing P2P application [Syncthing] will be used as the base components for wrapping.

The networking components, such as router (R) or firewall (FW) will be provisioned as VNFs available at the MATILDA 5G Marketplace.

Moreover, the qMON solution will be implemented in the MATILDA framework as NFV-based solution for QoS and SLS/SLA monitoring. The qMON Network Sensor will be implemented as a VNF and will be also available as PNF, if required (e.g. in case a mobile uplink and radio parameters should also be observed and analysed). The qMON Network Sensor VNF provisioning and basic configuration will be made through the MATILDA 5G-ready Application Orchestrator.

Basic network KPIs about various network connections (e.g. datacentre uplink, connection to remote datacentre, VPN monitoring) will be provided to the MATILDA 5G-ready Application Orchestrator to enable network context-based policies and rules. This will allow various distributed applications to monitor their network services and also dynamically reconfigure their networks based on the data provided by the qMON Network Sensor.

Furthermore, additional KPIs will be gathered with the goal to provide detailed post-analytics and data enrichment (not in the scope of the MATILDA project).

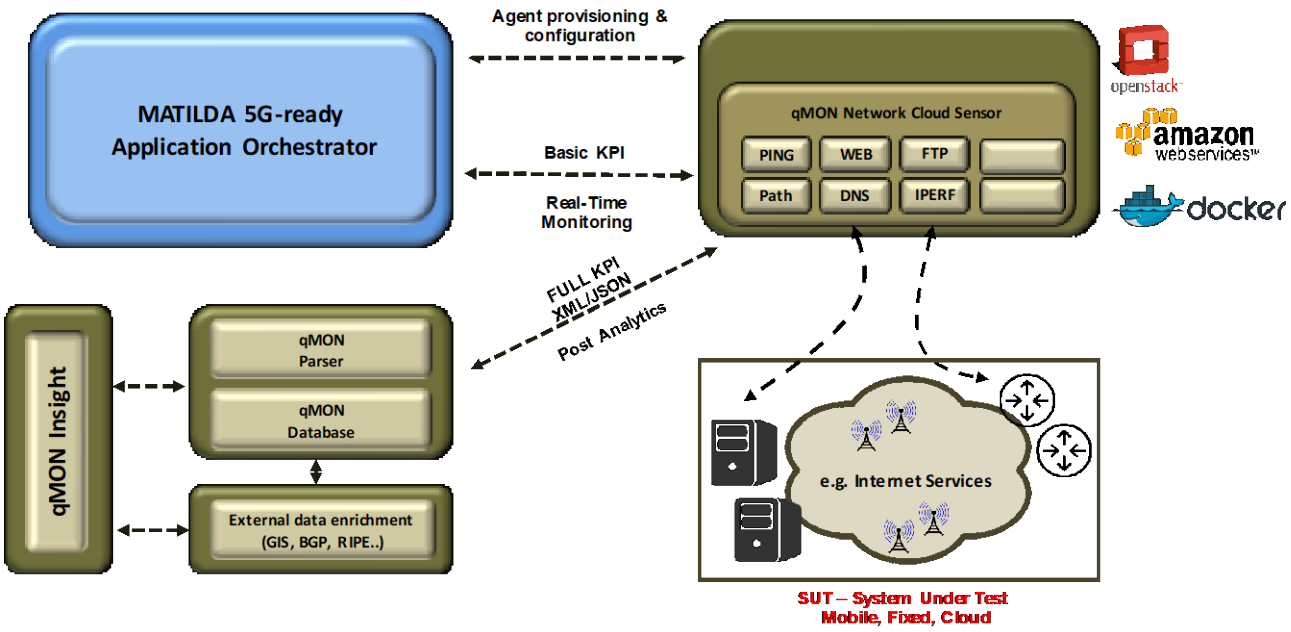


Figure 4.3.5: qMON solution integration with the MATILDA Framework

4.3.4 Use Case-Derived Requirements

ID	UC3_1
Unique Name/Title	Distributed application components
Priority	High
Type	Application/Non-Functional
Brief Description	The MATILDA framework must support deployment of distributed 5G-ready applications comprising multiple application components.
Rationale	The iMON Dashboard application will be implemented as a distributed 5G-ready emergency application comprising multiple application components. Distributed application paradigms must be used to allow different provisioning, scaling and fail-over mechanisms.
Validation method/Relevant KPI	This requirement can be validated at applications design phases.

ID	UC3_2
Unique Name/Title	Support vertical/horizontal scalability
Priority	High
Type	Application/Non-Functional
Brief Description	The MATILDA framework must support horizontal and vertical scaling of distributed application components.
Rationale	The “PHP BL” component (frontend) will be horizontally scalable, the “Database” and “BLOB” components (backend) will be vertically scalable. Horizontal scaling of “PHP BL” component enables the web dashboard service to react to a different number of users, while the vertical scaling allows the backend components to adjust their resources accordingly.
Validation method/Relevant KPI	This requirement can be validated at applications design phases.

ID	UC3_3
Unique Name/Title	Chainability of components
Priority	High
Type	Application/Non-Functional
Brief Description	All components comprising a distributed application according to the MATILDA framework should provide chainable endpoints/interfaces to allow connecting to the other components.
Rationale	The chainable components are used to create the application graph of the iMON Dashboard distributed application.
Validation method/Relevant KPI	This requirement can be validated at applications design phases.

ID	UC3_4
Unique Name/Title	Context-based application orchestration
Priority	High
Type	Application/Functional
Brief Description	The MATILDA framework must support context-based application orchestration.
Rationale	The iMON Dashboard application will be deployed and orchestrated according to the context parameters of the application. The context-based service orchestration allows provisioning and dynamic reconfiguration of distributed 5G-ready applications.
Validation method/Relevant KPI	This requirement can be validated after application deployment phase by checking if all application components are provisioned and are operating as defined with the application context.

ID	UC3_5
Unique Name/Title	Policy-based dynamic reconfiguration
Priority	High
Type	Application/Functional
Brief Description	The MATILDA framework must support the dynamic reconfiguration of a distributed application and components based on policy rules.
Rationale	Policy-based reconfiguration enables distributed applications to react on certain events or changed state. The iMON Dashboard application will allow dynamic reconfiguration based on the policy rules provided by the application administrator (e.g., if the PHP BL component's CPU usage is above 80%, deploy a new instance of PHP BL).
Validation method/Relevant KPI	This requirement can be validated after application deployment by simulating different usage scenarios and checking if the application components reacted as they should. For example, sending a large number of user requests should trigger horizontal scaling of "PHP BL" components.

ID	UC3_6
Unique Name/Title	Redundancy and resilience mechanisms
Priority	High
Type	Application/Functional
Brief Description	The MATILDA framework should provide some mechanisms for the distributed 5G-ready application to be operating at the highest possible level of availability.
Rationale	As the iMON Dashboard provides emergency services, it is very important that the

	application be able to minimize or eliminate the downtime of iMON Dashboard at critical events.
Validation method/Relevant KPI	This requirement can be validated after application deployment by simulating the disaster scenario, where one of the IaaSes hosting the iMON Dashboard application becomes unavailable and the instance located in the remote IaaS takes over all the load.

ID	UC3_7
Unique Name/Title	Support of legacy applications and services
Priority	High
Type	Application/Non-Functional
Brief Description	The MATILDA framework must provide support for the deployment of legacy applications and services.
Rationale	As the iMON Dashboard was not initially developed according to the 5G and MATILDA paradigm, the framework should allow the integration of such application through software wrappers.
Validation method/Relevant KPI	This requirement can be validated at application's design phases.

ID	UC3_8
Unique Name/Title	Service monitoring
Priority	High
Type	Application/Functional
Brief Description	The MATILDA framework must provide support for monitoring application and application components KPIs.
Rationale	The iMON Dashboard application components will provide various application KPIs, which can be used to specify some rules and assign the appropriate actions.
Validation method/Relevant KPI	This requirement can be validated after application deployment by measuring application KPIs, such as CPU usage, RAM usage, disk load, number of concurrent web requests, etc.

ID	UC3_9
Unique Name/Title	Support of VNF and PNF
Priority	High
Type	Network/Non-Functional
Brief Description	The MATILDA framework must provide support for VNF and PNF components.
Rationale	The iMON Dashboard network components will be deployed as VNF and optionally as PNF (e.g., if a mobile network connection is used). The network components along with the application components create the network-aware application graph.
Validation method/Relevant KPI	This requirement can be validated at application's design phases.

ID	UC3_10
Unique Name/Title	Context-based network orchestration
Priority	High
Type	Network/Functional

Brief Description	The MATILDA framework must support context-based network service orchestration.
Rationale	The iMON Dashboard 5G-ready application will support multiple network connections to the Internet and the MATILDA framework should allow usage and reconfiguration of different connections based on the current network context (i.e., round-trip time, available bandwidth, etc.).
Validation method/Relevant KPI	This requirement can be validated after application deployment phase by checking if all network components are provisioned and are operating as defined with the network context of the network-aware application.

ID	UC3_11
Unique Name/Title	Dynamic QoS and pre-emption provisioning, support and enforcement
Priority	High
Type	Network/Non-Functional
Brief Description	The MATILDA framework must support dynamic QoS provisioning and enforcement mechanisms.
Rationale	The iMON Dashboard 5G-ready application includes VNFs/PNFs that support QoS provisioning and enforcement mechanisms that can provide more efficient resource management at the network level.
Validation method/Relevant KPI	This requirement can be validated after application deployment by measuring different network parameters (e.g., kbps, RTT) and comparing them to the target values.

ID	UC3_12
Unique Name/Title	Network Monitoring
Priority	High
Type	Network/ Functional
Brief Description	The MATILDA framework must provide support for monitoring network conditions and provide network KPIs.
Rationale	For the iMON Dashboard to be able to always choose the best available network connection, the qMON solution will be integrated in the MATILDA framework as a VNF that is responsible for network monitoring and will provide network KPIs to the orchestrator.
Validation method/Relevant KPI	This requirement can be validated after application deployment by measuring network KPIs, such as round-trip time (RTT), bandwidth download/upload, packet loss, etc. The data is provided by the qMON VNF.

ID	UC3_13
Unique Name/Title	Networking across IaaS
Priority	High
Type	Network/ Functional
Brief Description	The MATILDA framework must provide support for network connections across and between multiple datacentres (IaaS).
Rationale	For the iMON Dashboard to achieve the highest possible level of availability, multiple iMON Dashboards will be deployed across multiple IaaS and should communicate with each other through the MATILDA-enabled network connections.
Validation method/Relevant	This requirement can be validated by deploying iMON Dashboard instances at multiple datacentres and checking if all data and services are properly synced

KPI	among them.

ID	UC3_14
Unique Name/Title	Privacy and isolation
Priority	High
Type	Network/Functional
Brief Description	The MATILDA framework should support some privacy and isolation mechanisms (e.g. VPN).
Rationale	If the iMON Dashboard is deployed on various sites, crossing various public networks, the VPN solution can be used to provide privacy for sensitive data exchanged between iMON Dashboard application components (e.g., sync between database components in different datacentres).
Validation method/Relevant KPI	This requirement can be validated by deploying iMON Dashboard instances at multiple datacentres and checking if data is securely sent over the communication channel (e.g., packet sniffer at the network link).

ID	UC3_15
Unique Name/Title	Security
Priority	High
Type	Network/Functional
Brief Description	The MATILDA framework should support basic network security mechanisms.
Rationale	The iMON Dashboard 5G-ready application components should reside behind the firewall implemented as a VNF from the MATILDA 5G Marketplace.
Validation method/Relevant KPI	This requirement can be validated at applications design phases.

ID	UC3_16
Unique Name/Title	SLA and Service Level Policy
Priority	High
Type	Network/Non-Functional
Brief Description	The MATILDA framework should support mechanisms for SLA/SLS monitoring.
Rationale	The qMON measurement tool will provide the basic KPIs of a network service to allow SLA/SLS monitoring of a network connection.
Validation method/Relevant KPI	This requirement can be validated after application deployment by measuring network KPIs provided by the qMON VNF and comparing them to the values specified in a SLA/SLS.

ID	UC3_17
Unique Name/Title	Programmability
Priority	High
Type	Network/Non-Functional
Brief Description	The MATILDA framework should support network programmability.
Rationale	If some level of network programmability is available within the MATILDA framework, the iMON Dashboard network components could be optimized in the sense of resource management and efficiency.
Validation method/Relevant KPI	This requirement can be validated at application's design phases.

ID	UC3_18
Unique Name/Title	Multiple network access types
Priority	High
Type	Network/Non-Functional
Brief Description	The MATILDA framework should support various network access types (e.g. fixed, wireless, mobile).
Rationale	The 5G-ready iMON Dashboard application can use various network access types to provide services to the users. During critical events, it can happen that certain communications infrastructure is not available and having the option to choose between access technologies will greatly enhance the resiliency and high-availability of the iMON Dashboard.
Validation method/Relevant KPI	This requirement can be validated at orchestrator design phases.

ID	UC3_19
Unique Name/Title	High-availability support
Priority	High
Type	Network/Non-Functional
Brief Description	The MATILDA framework should provide some network resilience and fail-over mechanisms to allow the distributed 5G-ready application to be operating at the highest possible level of availability.
Rationale	As the iMON Dashboard provides emergency services it is very important that the network-aware application is able to minimize or eliminate the downtime of the iMON Dashboard at critical events. At the network level, the MATILDA framework should allow setting basic high-availability features, such as setting redundant network paths, etc.
Validation method/Relevant KPI	This requirement can be validated after application deployment by simulating primary network connection outage with backup network connection taking over.

4.4 Use Case 4: Industry 4.0 Smart Factory Use Case – Inter-Enterprise Integration

4.4.1 Scenario Description

With the main focus of logistics, this use case addresses the scenario of connected manufacturing facilities, which are embedded in the lab infrastructure of BIBA. Main challenges are expected by the diversity of multiple stakeholders, which are typically running different technologies and management solutions.

One aspect of this scenario is the customer having the possibility to get a deeper look inside the customer's order. After making the order, the customer can track the different steps from achieving the raw components, following the delivery, the production itself and finally the warehousing, which are shown in Figure 4.4.1. It shall be possible for the customer to have constant access to every detail about the whole delivery process, such as: "Where are the raw components?", "When does the delivery arrive?", "When is the expected time for finishing the order?". As it is already very complex, this use case mainly focuses on the part of delivery in the supply chain. In this specific use case, the BIBA truck,

Figure 4.4.2, can be used. It has to be equipped with tracking and communication devices such as a GPS transmitter.

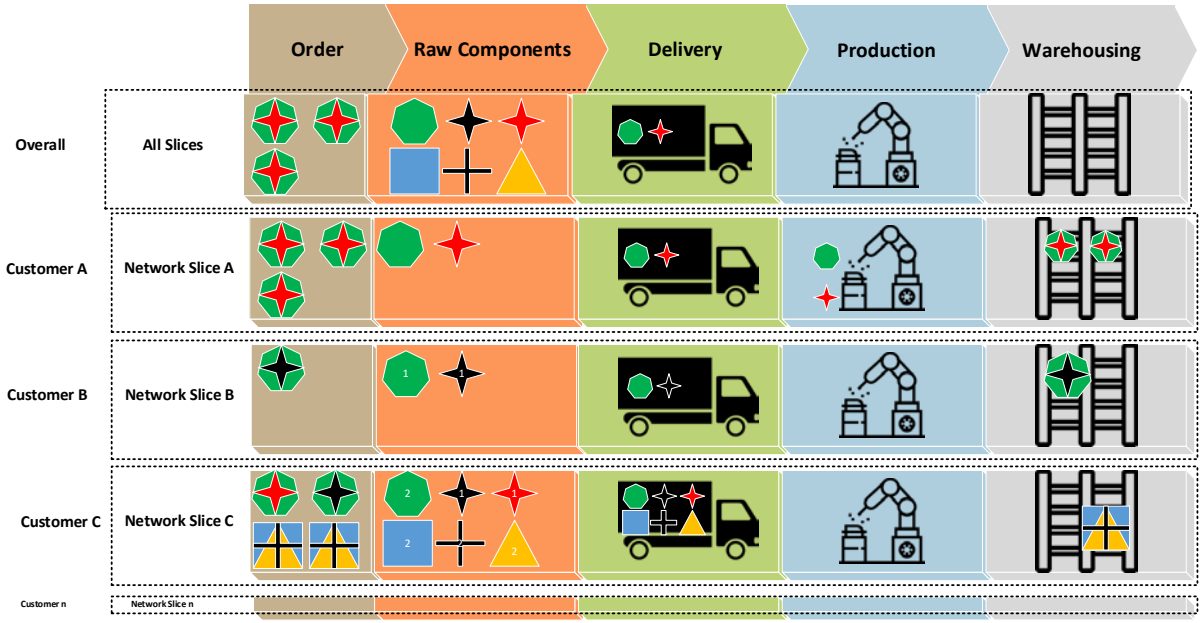


Figure 4.4.1: General concept for the logistic chain dashboard.

Optionally, the possibility of monitoring the freight is given by a collaboration of the German project SaSCh, which has its focus in supply chain transparency [SaSCh]. The basis of the project are sensor nodes, which are at the load carrier, in order to monitor the state of the charge and, if necessary, send alarm messages. Examples of sensitive goods in the automotive supply chain are batteries, which may only be transported at a certain temperature interval, and electronic components that are shock-sensitive.

Figure 4.4.2: BIBA truck.

Following this scenario, every customer can only access its own order. Also, it shall be possible to set a priority for an order (e.g., it has to be finished within a certain time). For the logistic chain, all orders are running in parallel. Therefore, a negotiation of the products and components is required to fulfil the orders in an optimal way.

For this use case, we have to link the facilities with a smart object, e.g. the BIBA truck, and require clear service provisioning and real-time capability of the overall communication network. This encompasses interfacing to existing legacy systems of OEMs, as well as interconnecting suppliers or logistics service providers. Network slices shall be used for a customer's order.

4.4.2 Objectives

- Use of end-to-end SDN and NFV capabilities to build heterogeneous 5G communication infrastructure and services for production/assembly, as well as from logistics service providers;
- To track and follow the change of the orders;
- Real time monitoring is required;
- The data shall be stored in a cloud service.

Challenges and Innovation

To achieve a reliable connection between the different stakeholders, the MATILDA network infrastructure has to manage different technologies and management solutions with low latencies, safety, and separate logistic chains; therefore, network slices shall be used. Furthermore, tracking of the components for traceability is required. At least, the delivery service needs a GPS transmitter.

One task is to provide the client with all information concerning the client's offer to see the ongoing process. The data shall be stored in a cloud environment where the client can trace, via a terminal or smart device, the client's own order. Besides, by using network slices, every client has only access to his/her own order. This also means that separated and flexible resources are required, depending on the orders.

Another challenge will be the scenario what will happen during interferences. In case of interference, much higher bandwidth is required as the communication between the products rises enormously.

In order to achieve the goals with this use-case and ensure the defined requirements in Section 4.4.4, many actors/stakeholders have to be involved during the implementation and testing of the expected prototype. Since the use case addresses the acquisition of huge production data, national cloud resources have to be offered for data processing and storage (Cloud Infrastructure Providers). For enabling a real-time interaction of the costumer with the production plant, Dashboards have to be implemented. The collaboration with 5G Application Software developers will ensure the implementation of generic robust and secure interfaces, which will be later available on Application stores/marketplaces. The integration of those type of services into industrial applications, such as the addressed scenario, could be the role of Service Providers.

4.4.3 Scenario Workflow

The main goal is, by using the idea of Industry 4.0 – Smart factory, to achieve a highly flexible automatic logistic production chain. In detail, the idea is to give three customers the possibility to track, change and prioritize their own orders. This includes information about when, where, and real-time monitoring. All the data shall be stored in a cloud service. For the logistic chain, the different facilities as well as the products and components have to communicate with each other. In addition, this shall be done by using a cloud service, as shown in Figure 4.4.3.

The scope for this use case shall begin with the order and achieving/delivering the raw components, followed by the assembly and finally warehousing the complete products. A dashboard for every client shall be provided, so that the clients can see the actual process of their own order. One solution could be similar to the sketch in Figure 4.4.1. The logistic chain consists of several facilities, which are available at BIBA. For example, for delivery the BIBA truck can be used, for the production/assembly a cyber-physical production system, which is described in Section 4.5, and for warehousing a high rack system is available.

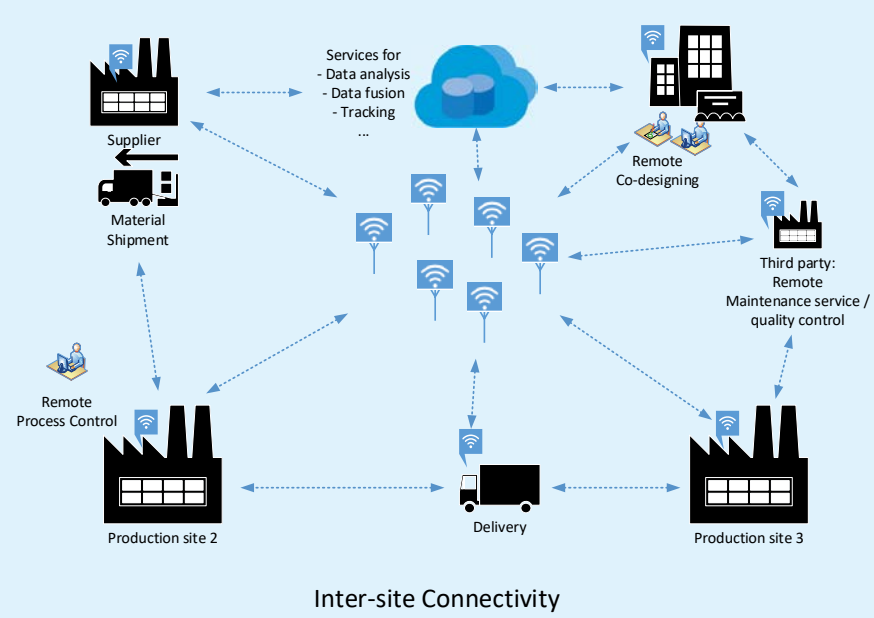


Figure 4.4.3: Automatic Logistic and Communication via a cloud service.

Furthermore, it shall be possible to change the order (e.g. amount, type of product, priority level). As each item belongs to a different customer, there shall only be access to that customer's certain order. This requires a very flexible system where all facilities and products – respectively, components – have to interact with each other. In normal mode, the communication between the products will be set with the order. Therefore, a bigger order (more products) requires a higher bandwidth. By using network slices every order shall receive enough bandwidth so that the logistic chain process can communicate and run smoothly. The main challenge relates to scenarios whereby interferences occur and how to react on this. In this case, the communication between facilities, products and components will increase. For example, for a small order the exchange of information requires much more bandwidth than during the normal mode.

4.4.4 Use Case-Derived Requirements

ID	UC4_1
Unique Name/Title	High Availability & Reliability
Priority	High
Type	Non-Functional
Brief Description	The test systems interconnection infrastructure/services shall always be available and reliable. Further, the whole infrastructure should guarantee a high level of availability & reliability, since service customers will consider the services as the interface with their products.
Rationale	Services need to be available at all times for service consumers (producers, customers). The services must also take place in a seamless manner and the user shall be able to trust that no disruptions will occur. The testing of distributed test systems poses strict availability requirements to ensure the integrity and success of test campaigns.
Validation method/Relevant KPI	The availability level shall reach 99.99% of operational time, and will be verified through extensive testing. KPIs: • (time the service is available) / (total time from service deployment up) • mean time between failure

ID	UC4_2
Unique Name/Title	Network Slicing Capability
Priority	High
Type	Performance
Brief Description	Network Slicing is needed to handle different sessions and types of traffic of service consumers (producer, customer), with specific, time-varying features.
Rationale	The traffic between production facilities and service consumers has specific, strict requirements in terms of bandwidth and latency, and typically presents a time-varying pattern, with peak periods and periods of low activity. The traffic must exploit the low activity periods. To handle these activities effectively, network slicing and context awareness are needed.
Validation method/Relevant KPI	Test of the correct allocation of network slices to the corresponding customer session. KPI: throughput (in Mbps) of each network slice.

ID	UC4_3
Unique Name/Title	Adjustable Bandwidth allocation
Priority	High
Type	Performance
Brief Description	Adjustable bandwidth allocation is needed to cope with the time varying features of the different types of traffic within network slices.
Rationale	The different network services carry different types of traffic to the Framework, with time varying patterns. To manage network connectivity effectively, the bandwidth allocated to the different services should vary to adjust to the current traffic conditions.
Validation method/Relevant KPI	Measurement of allocated bandwidth, and comparison with target values. KPI: throughput (in Mbps) of each link.

ID	UC4_4
Unique Name/Title	Security & Privacy
Priority	High
Type	Non-Functional
Brief Description	The interconnection of test systems (e.g. production data, user information/authorisation) shall be secure in order to preserve system integrity.
Rationale	Since the interconnection of test systems involves highly sensitive data transfer, all operations must be highly secured and subject to specific access rules. All actions/data are personal and available only to those with the appropriate authorization level. This needs to be provided and ensured by the network infrastructure.
Validation method/Relevant KPI	Testing of access to data / authorization levels for all MATILDA procedures/operations.

ID	UC4_5
Unique Name/Title	Scalability
Priority	High
Type	Application/Non-Functional
Brief Description	MATILDA components should be able to handle parallel data loads from multiple

	sources (e.g. huge amount of industrial IoT devices) in a scalable, high-throughput persistent approach.
Rationale	Since MATILDA will have to be able to support various users and applications, scalability is necessary for successful deployment of all the services.
Validation method/Relevant KPI	This requirement can be validated at applications design and testing (simulation) phases.

ID	UC4_6
Unique Name/Title	Interoperability with Various Access Networks
Priority	High
Type	Network/Non-Functional
Brief Description	The MATILDA framework should support various network access types (e.g. fixed, wireless, mobile).
Rationale	Mobility is a key feature for logistics, implying that the customers can be served by different access networks depending on their location and the availability of each technology present at each location. Therefore, logistics shall be supported seamlessly over various Access Networks; meaning that the underlying MATILDA framework shall be interoperable with various access networks.
Validation method/Relevant KPI	Through the design phase of the MATILDA framework itself and its extended testing phase.

ID	UC4_7
Unique Name/Title	Network Monitoring
Priority	Medium
Type	Network/ Functional
Brief Description	Network monitoring is necessary in order to deploy, detect issues / problems, reconfigure and reallocate resources.
Rationale	In order to offer undisturbed operations, it is important to monitor whether the deployment of resources took place or to detect any issues that might occur, so that reallocation of resources can take place.
Validation method/Relevant KPI	This requirement can be validated in the design phase.

ID	UC4_8
Unique Name/Title	Low Delay/Latency
Priority	Medium
Type	Performance
Brief Description	Low Delay is required for the real-time interaction with production facilities.
Rationale	To offer real-time monitoring services and real-time interaction with the system low delay connectivity is needed.
Validation method/Relevant KPI	Measurement of delay fluctuations between interconnected systems / sub-systems under test, and comparison with target values. KPI: measurement of end-to-end delay time.

4.5 Use Case 5: Industry 4.0 Smart Factory Use Case – Intra-Enterprise Integration

4.5.1 Scenario Description

In the new supply chain's competitive markets, enterprises must provide customers with a range of information services in real-time, tracking capabilities, as well as product customisation alternatives in product configuration. In the new environment, customers want to know at all times the status of their orders and the relative delivery date. They want assurances that their particular needs can be incorporated into product configuration.

Figure 4.5.1: Cyber-physical platform as prototype for automotive assembly.

The aim of the proposed scenario is the prototypically development of a cyber-physical production system, that deals with the new environment's requirements. For a real-life demonstration of the intended goals within this use case, an existing BIBA prototype, shown in Figure 4.5.1, will be used as a production environment for automotive assembly. The demonstrator considers a production/ assembly process of mass customized products, e.g. fabrication/ assembly of automotive parts (e.g. steering wheel car, car door, rear light, etc.). This use-case focuses on a production scenario in which mass customized automotive parts are ordered by a customer and produced by the different production facilities into the production plant. A generic ordering procedure is shown in Figure 4.5.2.

The internally distributed production facilities form a network by means of being connected to the intranet/Internet and the ability to communicate with each other using defined interfaces and protocols. For configuring and ordering the customized product, a cloud service is provided. With this service, the customers are able to develop customized products by selecting from a number of components and options, e.g. the colour/size of the automotive part and additional characteristics.

After the configuration of the expected product is completed, the customer's order is forwarded to a provider that automatically plans and deploys the necessary production steps to the production plant. For this step, a local production facility (e.g., production facility A for assembling, production facility B for painting, etc) is selected. During the production of the automotive part, the different facilities are monitored for diagnosis to enable further optimization of the production process and, if necessary, reasoning about possible reconfiguration measures in case of failures.

In order to implement the selected scenario, novel software paradigms and architectural approaches will be applied for the production facilities that include workers, machines, tooling, components of final products, products and for the customer interfaces.

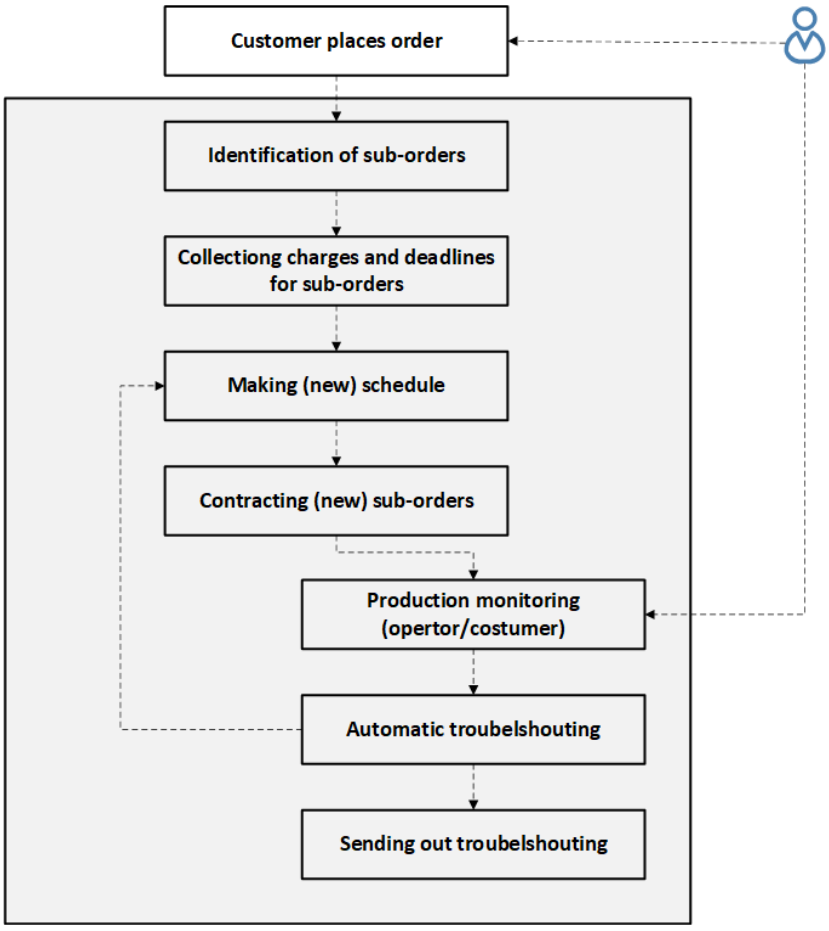


Figure 4.5.1: Generic ordering procedure.

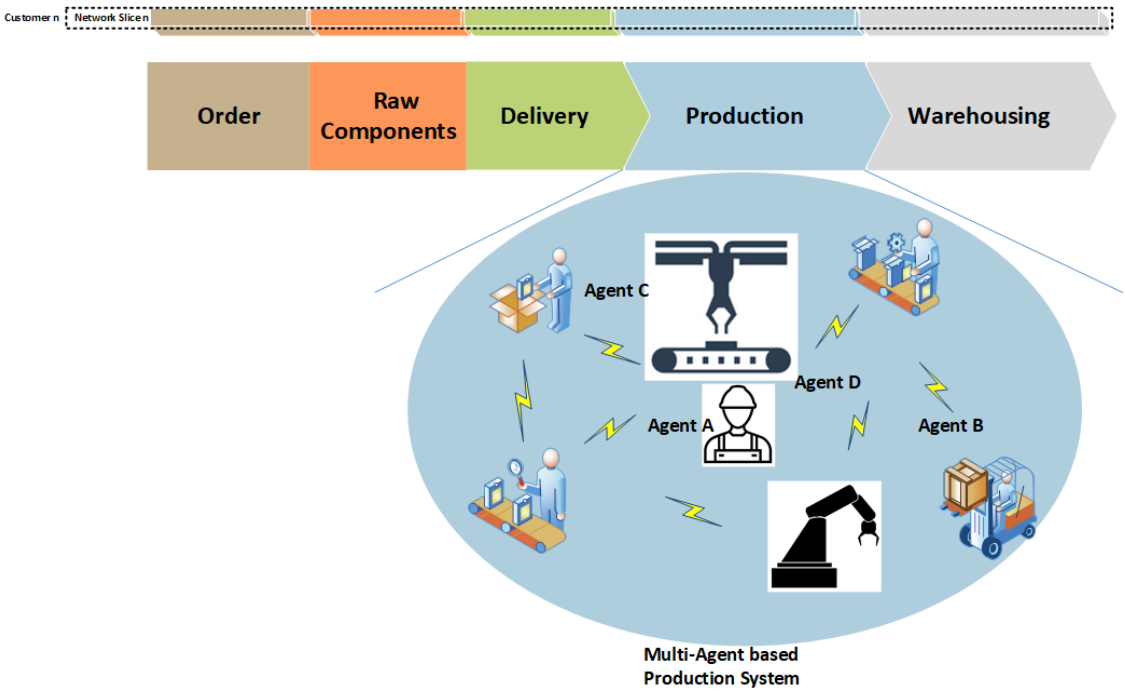


Figure 4.5.2: Multi-Agent based production system.

The adoption of systems based on a multi-agent architecture has proven its utility in several works. Multi-agent systems have been widely investigated for providing a support for modularization, decentralization, autonomy and reusability, and are increasingly adopted in order to realize different functionality required to enhance production automation and control. Most applications of software agent concepts for production systems address different reconfiguration issues of the production system to handle modules'/workstations' breakdowns or structural changes of the system, or to realize distributed production planning. Based on task descriptions to realize a certain production process, software agents that control production system's components and monitor the state of products will be implemented. Decision options for reconfigurations of the production system will be realized by, e.g., redundant control functions offered by different modules. Therefore, the users can reconfigure the automation system's behaviour of inner logistic systems online. The reconfiguration of a production automation system to compensate components' or machines'/workstations' breakdowns under varying throughput conditions will be developed. Each machine or workstation is equipped with an IoT module for edge computing to guarantee 5G requirements.

To complete the assembly of automotive parts, many processing steps have to be carried out and many product types are available. The diverse assembly parts are illustrated in a so called "variant tree" in Figure 4.5.3. By using the variant tree and considering the model-specific components, a product type corridor can be derived. This product type corridor describes the options a product has after each production step.

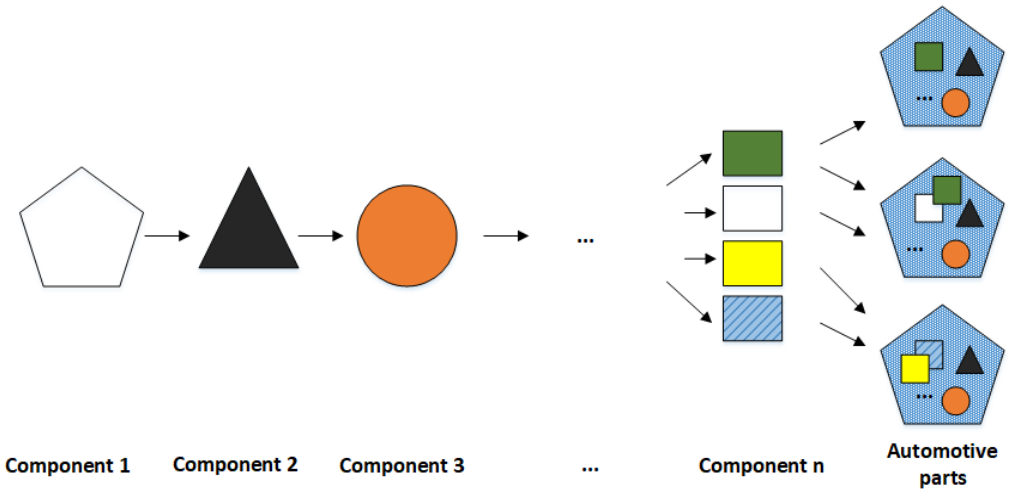


Figure 4.5.3: Example of a variant tree for assembly parts.

This scenario includes the flexibility to change the sequence of the assembly steps. The aspect of flexibility is taken into consideration by offering many workstations for assembling a specific component. For each assembly step many workstations can be assigned (see Figure 4.5.4). Thereby, a number of sequences are possible, and the products/automotive parts have the opportunity to choose between these alternative workstations. However, in some cases, some components have to be assembled before others.

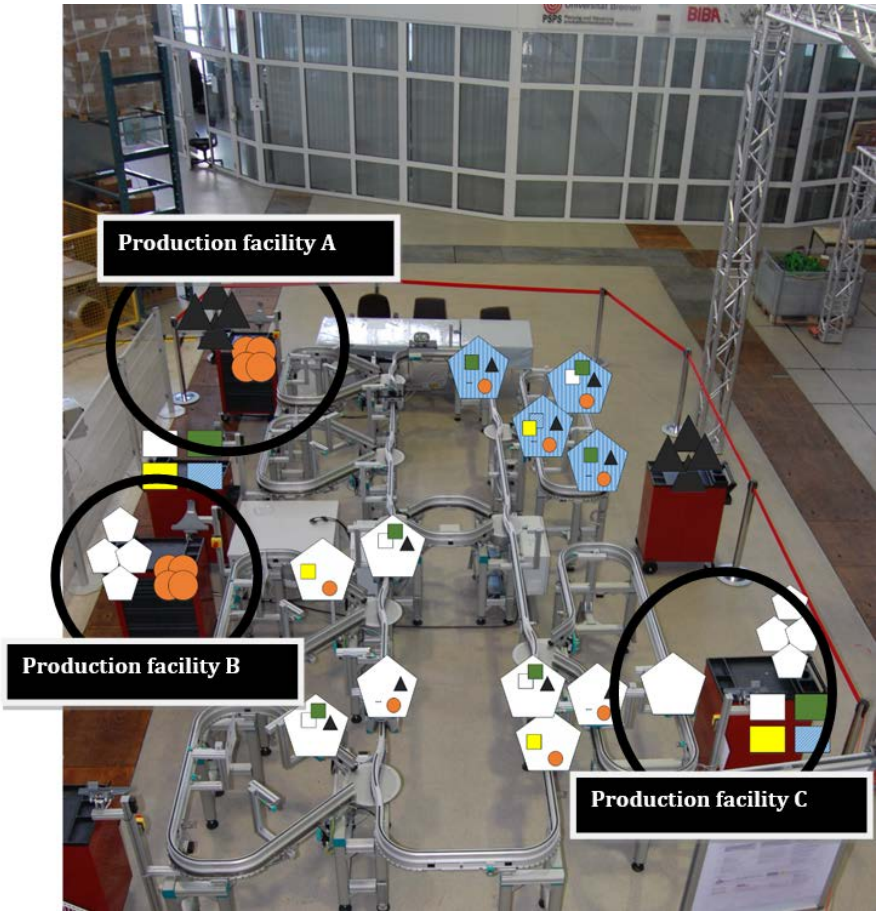


Figure 4.5.4: Infrastructure for assembly/production scenario.

The demonstration scenario will start with three customer orders for different automotive parts. Every customer will be able to monitor the status of the customer's own order and interact directly with the assembly process. During processing, orders can be reconfigured (e.g., change the colour of a component, choose another cable type) and the customer has the opportunity to cancel/create orders. Thus, some products change their target variant to prevent overproduction. In addition, the scenario will include the case of a workstation failure and provide Service Consumers with the new delivery time of their orders. The products should react independently on this event and try to avoid this workstation. Either they change the sequence of the process steps and come back later when the workstation is fixed, or they take an alternative workstation. In this way, the products reduce their processing time. In order to achieve the goals and at the same time ensure the defined requirements in Section 4.5.4, many actors/stakeholders have to be involved during the implementation and testing of the expected prototype. Since the use case is addressing the acquisition of huge production data, national cloud resources have to be offered for data processing and storage (Cloud Infrastructure Providers). To enable real-time interaction of the costumer with the production plant, Dashboards have to be implemented. The collaboration with 5G Application Software developers will ensure the implementation of generic robust and secure interfaces, which will be later available on Application stores/marketplaces. The integration of those types of services into industrial applications, such as the addressed scenario, could be the role of Service Providers.

4.5.2 Objectives

The main objective of this use case is offering customers multiple independent instances on one physical network, the production plant. Within the instances, the following features will be enabled:

- Real-time production monitoring: this includes capabilities that enable users/customers to see the current status of specific products/orders and to check production as it occurs. This provides instantaneous feedback on critical parameters such as total parts created, production time, downtime, scrap, rejects, parts remaining to be produced and cavitation changes. Users define which performance and measurement parameters to capture for each item or tooling configuration, thus providing flexibility to gather the unique, job-specific statistics they need to drive additional activities and processes.
- Product configuration/reconfiguration: the customer has the ability to adapt the configuration of the customer's own products/orders. Furthermore, the option of changing the sequence of the assembly steps can be provided (customer as process engineer). The products can autonomously react fast and flexibly to dynamic influences.

These services will be provided by the BIBA Framework, consisting of ad-hoc App/dashboards that can be downloaded to the smartphone or tablet of the end users. The final goal is to deploy and validate the use case described above in the infrastructure made available by BIBA.

Challenges and Innovation

In order to achieve the goals within the intended approach for production services, many challenges need to be faced, starting with the proper transfer of the traditional monitoring approaches from intranet to a cloud/edge deployment. For this reason, different requirements at multiple production compartments have to be explored. Thus, the overall network infrastructure has to overcome several challenges; namely: the high density of industrial IoT devices (sensors, actuators, etc.), high data-rates to enable real-time capabilities and low latencies. Expected services require large IT resources, both in terms of processing and storage capabilities at the network's edge. Due to the fact that customers' needs are different, the use of network slices will optimise the resource allocation, by considering context-awareness approaches.

4.5.3 Scenario Workflow

To demonstrate the Industry 4.0 Smart Factory use case "Intra-Enterprise Integration", the following have to be considered:

- Service customer: The service provider offers a 5G-enabled application for the addressed use-case. This application, offered in the Marketplace, will be used by the service customers for their benefit (see Section 4.5.2).
- API: an API will take the role of aggregating all relevant application data (user, profile, connectivity …). Afterwards, appropriate mechanisms for process data collection, processing and visualisation will be defined. Here it is important to mention that some application data have to be checked continuously, such as connectivity, in order to adapt the resource allocation to the present profile.
- Production environment, which consists of the network of connected agents/controllers/ machines that will be deployed within the production facilities. These are interconnected by means of communication technologies such as WLAN, LAN, ZigBee. The API layer should take into consideration the huge number of interconnected devices and the resulting amount of data that need to be processed in real-time.

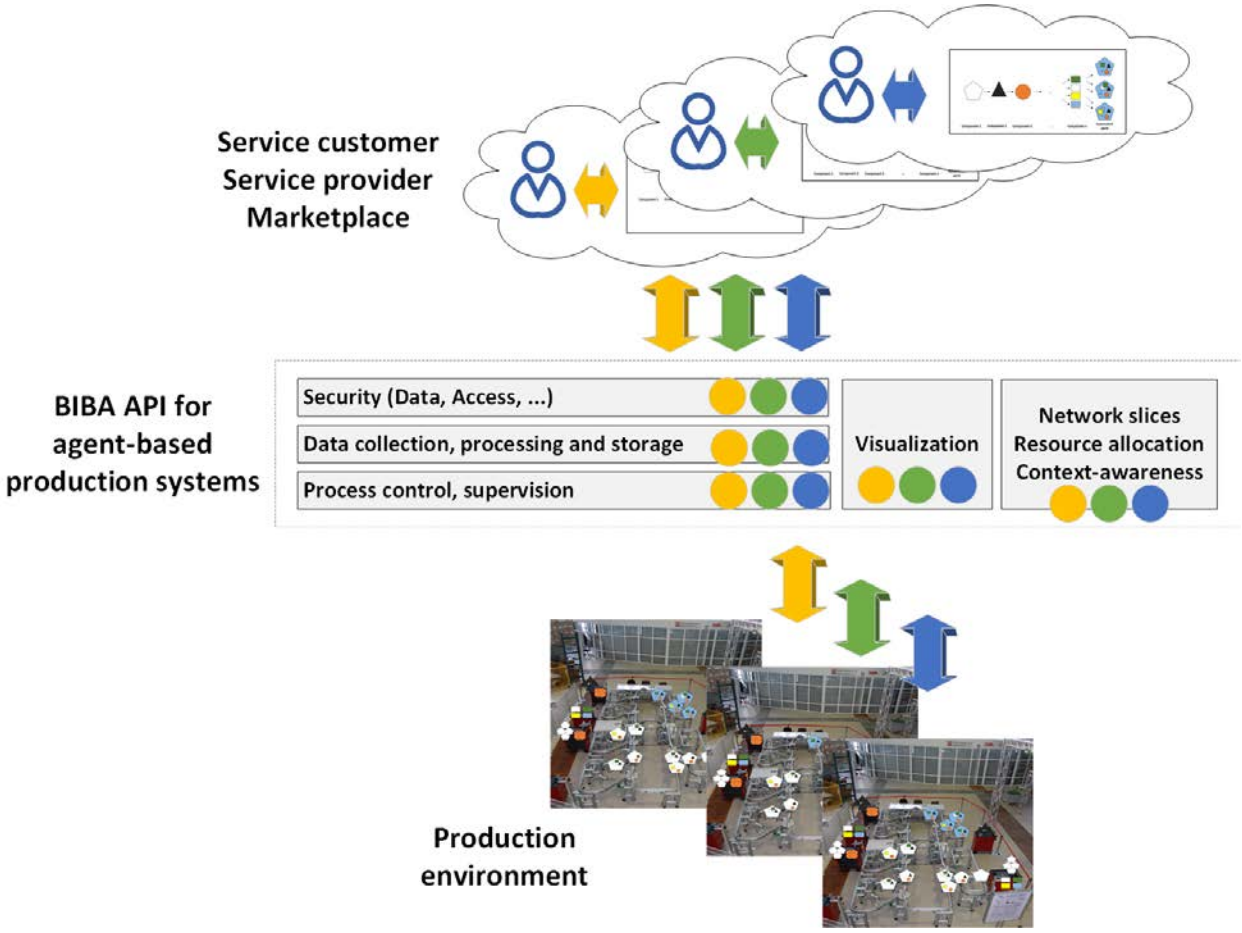


Figure 4.5.4 Workflow diagram.

For the demonstration of the scenario, the existing infrastructure has to be extended mainly in terms of hardware components similar to those offered by "Bristol is Open". The sharing of Bristol infrastructure could be an alternative in order to avoid the acquisition of similar platforms. This could be possible once the connection of the two entities (production facility and Bristol IT infrastructure) is realized.

4.5.4 Use Case-Derived Requirements

ID	UC5_1
Unique Name/Title	Network Slicing Capability
Priority	High
Type	Performance
Brief Description	Network Slicing is needed to handle different sessions and types of traffic of service costumers (producer, costumer), with specific, time-varying features.
Rationale	The traffic between production facilities and service customers has specific, strict requirements in terms of bandwidth and latency, and typically presents a time-varying pattern, with peak periods and periods of low activity. The traffic must exploit the low activity periods. To effectively handle these activities, network slicing and context awareness are needed.
Validation method/Relevant KPI	Test of the correct allocation of network slices to the corresponding customer session. KPI: throughput (in Mbps) of each network slice.

ID	UC5_2
Unique Name/Title	Adjustable Bandwidth allocation
Priority	High
Type	Performance
Brief Description	Adjustable bandwidth allocation is needed to cope with the time varying features of the different types of traffic within network slices.
Rationale	The different network slices carry different types of traffic to the Framework, with time varying patterns. To effectively manage network connectivity, the bandwidth allocated to the different slices should vary to adjust to the current traffic conditions.
Validation method/Relevant KPI	Measurement of allocated bandwidth, and comparison with target values. KPI: throughput (in Mbps) of each link.

ID	UC5_3
Unique Name/Title	Low Delay/Latency
Priority	High
Type	Performance
Brief Description	Low Delay is required for the real-time interaction with production facilities.
Rationale	To offer real-time monitoring services and real-time interaction with the system low delay connectivity is needed.
Validation method/Relevant KPI	Measurement of delay fluctuations between interconnected systems / sub-systems under test, and comparison with target values. KPI: measurement of end-to-end delay time.

ID	UC5_4
Unique Name/Title	Security & Privacy
Priority	High
Type	Non-Functional
Brief Description	The interconnection of test systems (e.g. production data, user information/authorisation etc.) shall be secure, in order to preserve system integrity.
Rationale	Since the interconnection of production systems involves highly sensitive data transfer, all operations must be highly secured and subject to specific access rules. This needs to be provided and ensured by the network infrastructure.
Validation method/Relevant KPI	Testing of access to data / authorization levels for all MATILDA procedures/operations.

ID	UC5_5
Unique Name/Title	High Availability & Reliability
Priority	High
Type	Non-Functional
Brief Description	The whole infrastructure should guarantee a high level of availability & reliability, since service customers will consider the services as the interface with their products.
Rationale	Services need to be available at all times for service consumers (producers, customers). The services must also take place in a seamless manner and the user be able to trust that no disruptions will occur that may result to economic damages.

Validation method/Relevant KPI	The availability level shall reach 99.99% of operational time, and will be verified through extensive testing. KPIs: • (time the service is available) / (total time since service deployment) • mean time between failures

ID	UC5_6
Unique Name/Title	Scalability
Priority	High
Type	Non-Functional
Brief Description	MATILDA components should be able to handle parallel data loads from multiple sources (e.g. huge amount of industrial IoT devices) in a scalable, high-throughput persistent approach.
Rationale	Since MATILDA will have to be able to support various users and applications, scalability is necessary for successful deployment of all the services.
Validation method/Relevant KPI	This requirement can be validated at applications design and testing (simulation) phases.

4.6 Use Case 6: Smart City Intelligent Lighting System

4.6.1 Scenario Description

Smart City initiatives are an important worldwide phenomenon, and in the EU only the annual smart city benefits from 5G is estimated to reach 8.1 billion Euros in 2025 [SelfNet].

Over the last decade, the evolution of information technologies and communications networks, sensors, actuators, cloud infrastructure, big data and products/services based on these enablers has changed the way people live in a city. Access to information, services and communication is now provided anywhere and anytime by smartphones, and modern people have adapted to this new way of living. Meanwhile, various service providers that create "smart city technologies" are trying to convince the governments and the public administrations that these technologies can help cities improve the efficiency, availability, quality and cost of provided city services. While governments are making the transition to online services, they must ensure that no one is left behind, not even those without access to this technology.

In this use case, we will observe how Alba Iulia, a small to middle size city in Romania with about 70k inhabitants, is moving forward as a smart city by adopting the latest Information and Communication technologies (ICT) including Long Range WAN (LoRaWAN – a Low Power Wide Area Network - LPWAN- specification defined by LoRa Alliance), LTE-M (Machine Type Communication introduced in 3GPP Release 13) and finally 5G enablers. For the smart city use cases, Orange proposes an open data strategy and open architecture that give access to further development of new applications, by monetizing datasets from the city itself. The high-level architecture is constructed on three layers: data collection and transport, open IoT middleware and application. The data collection and transport layer will provide LoRaWAN, LTE-M and 5G specific connectivity for all sensors, actuators and consequently raw datasets that will be generated from the smart city solutions. These datasets will be sent to the open middleware platform to be stored, processed and secured. The open middleware can work also with other datasets that are not accessible in real-time through sensors or actuators. For example, we can consider the 1k datasets available on the Romanian Government Open Data Portal [DATA.GOV.RO], the 11k datasets available on the EU Open Data Portal [EUODP] or the 197k datasets available on the US Government's data portal [US DATA.GOV]. There is the possibility that middleware

becomes available to trusted application developers using REST APIs, creating the opportunity of building a dedicated marketplace for the smart city ecosystem.

Alba Iulia has been selected by Orange to demonstrate the capabilities of the targeted smart city high-level architecture in dealing with critical smart lighting infrastructure. In the case of Alba Iulia, we plan to build a live testing infrastructure of at least 100 smart controllers (actuators) that will be deployed on the main roads of the city. This will help the authorities understand the aggregated benefits of the solution and compare them with the status quo.

The Intelligent Lighting use case will facilitate three key functionalities that are impossible or hard to be efficiently provided today over the legacy city lighting infrastructure and even modern lighting infrastructure, regardless of the development of modern Low Power Wide Area Networking (LoRaWAN) technology or cellular IoT technologies (LTE-M, NB-IoT):

- According to this use case, the entity in charge will be able to remotely control every single lighting pole in real-time and in a secure way from the target network, in order to adjust the lighting intensity and efficiently manage energy consumption. The system will offer the public lighting distribution company reports to the city manager the ability to automatize the control of the lights, including the on/off and dimming capability according to certain policies (e.g. daytime moment, natural light intensity, location, traffic). This system, combined with the adoption of more efficient LED based ballast lamps, is anticipated to generate a reduction of energy costs of up to 80%, and a return on investment in just four to five years. According to a report [Philips-2017], only about 10% of the 300 million street lights poles in the world are using energy-efficient LEDs, and just 2% are connected thanks to legacy communication technologies such as Programmable Logic Controllers (PLCs) and 2G/3G.
- Moreover, the system will allow real-time and history-based energy consumption measurements. The city of Los Angeles [Philips-2017] made energy savings of 63% in 2016 just by switching to 100% LED street lighting, generating cost savings of USD 9m and reducing its annual greenhouse gas emissions associated with public lighting by 47,000 metric tons. This is equivalent to the greenhouse gas emissions from almost 10,000 passenger vehicles driven for one year.
- The entity in charge of the street lighting infrastructure operation and maintenance will be able to proactively spot the malfunctions, energy loss or energy theft attempts on the public lighting network, as the system will generate intervention tickets in real-time per pole or branch of pole. This capability will highly improve the city lighting service availability and will decrease the operational costs with maintenance activities. There is an international standard [ISO 37120] that specifies a set of indicators meant to define and measure the performance of quality of life and city services. This is applicable to any city or municipality that targets to measure its performance in a comparable and verifiable manner, irrespective of size and location. Street lighting can consume between 15 – 50% of public electricity [ISO 37120]. Electricity consumption of public street lighting is calculated as the total electricity consumption of public street lighting (numerator) divided by the total distance of streets where street lights are present (denominator). The result shall be expressed as kWh per kilometre per year.

Therefore, increasing the street lighting's efficiency is one of the most relevant and cost-effective steps that a municipality can consider to improve energy efficiency. Increasing the efficiency and quality of public street lighting generates multiple co-benefits including improved citizen perception of public safety and reduced crime rates, reduced maintenance costs, improved street and traffic safety, enhanced city attractiveness and community identity, improved air quality, and increasing economic productivity by extending business hours in commercial areas. According to [Philips-2017] Los Angeles administration highlighted a 10.5% drop in crime rates regarding vehicle theft, burglary and vandalism in the first 2 years of its LED program.

In order to enable such a use case, the following key stakeholders should partner (the responsibilities of the stakeholders are linked with the high-level building blocks from Figure 4.6.1):

- Telecom Infrastructure Provider – in charge of providing the connectivity of the sensors/actuators over the LoRaWAN/LTE-M/5G access network.
- Cloud Infrastructure Providers – in charge of providing the cloud servers on which the Middleware and Lighting and Energy Management Application are deployed.
- Service Provider – in charge of providing the hardware and software components: actuators/sensors, the middleware and the Lighting and Energy Management Application, but also of deploying the application graph.
- 5G Application SW developer – in charge of building the actual application that is further commercialized by the Service Provider.
- Service Consumer – benefits from the service, but is also involved in defining the business requirements.

4.6.2 Objectives

The main goals of this use case are:

- Specification of business, functional and security requirements for Smart City IoT – Smart Lighting infrastructure with focus on energy consumption optimization and intelligent lighting to be supported by the 5G architecture
- Demonstration of the coexistence of selected Smart City IoT applications in the shared 5G infrastructure, without decreasing the value for KPIs achieved in the initial setup demonstration.
- Integration and testing of this vertical use case within the project's 5G communication framework.

A best practice will be to create a measurement framework that can monitor and evaluate city-level impacts of smart and connected lighting investments thanks to the Intelligent Lighting System. The adoption of 5G based smart city datasets will help to build the investment case for smart technology projects. These datasets can clearly define how investments can improve infrastructure QoS and Quality of Experience (QoE) across a city and deliver benefits to its citizens.

Challenges and Innovation

The solution proposed in this use case should be able to connect and manage 24/7 over 9000 poles that are provided with power only during the night. Besides the actual deployment of the solution, there are two main challenges identified: (1) dealing with the signalling form 9000 connected poles over the mobile access infrastructure and (2) during the day the sensors/actuators from the poles should work on batteries in order to provide 24/7 access to each of them. Within MATILDA, both challenges will be overcome by (1) enabling network slicing so that the signalling traffic from this IoT solution will be isolated and will not flood the network and (2) leveraging on low power communication mechanisms.

Therefore, within MATILDA it is important to assess different connectivity scenarios LoRaWAN/LTE-M/5G with virtualization mechanisms enabled, benchmarking the performance of the solution on predefined KPIs like availability, reliability and power efficiency.

4.6.3 Scenario Workflow

The Intelligent Lighting use case to be demonstrated in Alba Iulia pilot will consider the following high-level building blocks:

1. The network of connected actuators/controllers that will be deployed one per each public lighting pole and poles aggregation node from the selected area;
2. The competing connectivity networks that will include: LoRaWAN, LTE-M and 5G technologies, each with its own access layer, transport layer, security layer, management layer and core layer network components;
3. The open IoT middleware, which allows the main Service Provider to decide which other Service Providers could benefit from the collected data. This could further enable enhancement of the Smart Lighting 5G use case.
4. The street lighting and energy management layer that integrates the connected actuators/controllers with web-based management applications, including a remote street lighting pole and energy management tool for the city to measure, manage and monitor connected public street lights, by using a real-time, map-based view, and a street lighting pole asset management application, which helps maintenance planning and operations management.

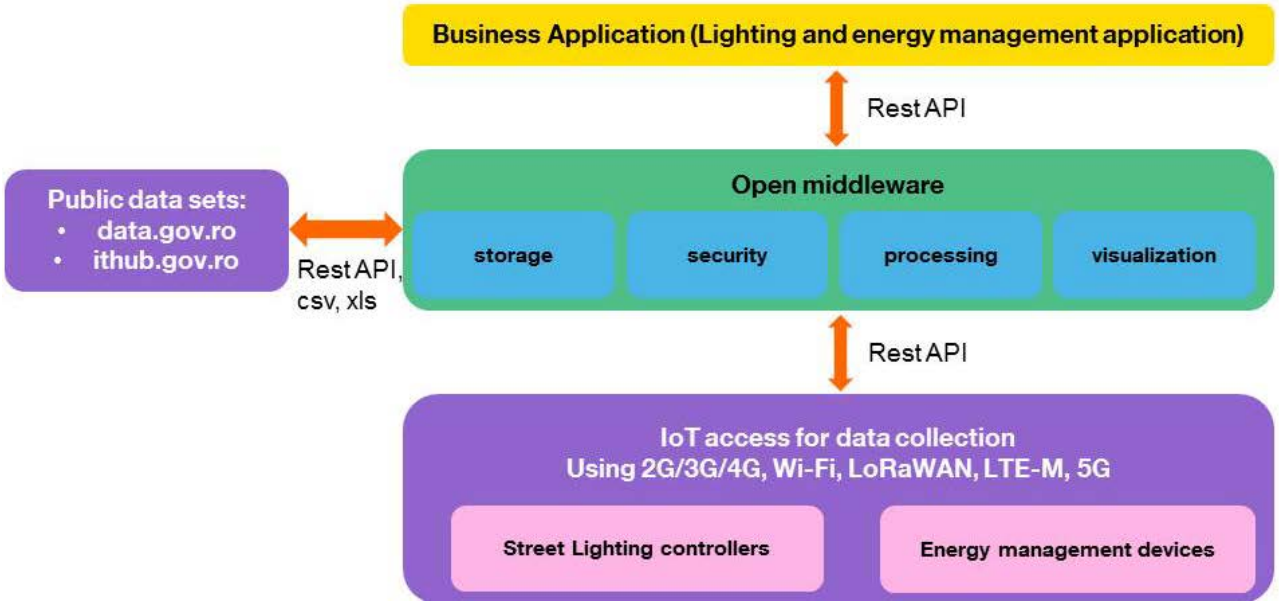


Figure 4.6.1: High level building block for Smart Lighting Use Case.

4.6.4 Use Case-Derived Requirements

ID	UC6_1
Unique Name/Title	Low power consumption
Priority	High
Type	Non- Functional
Brief Description	The system should assure low power consumption.
Rationale	The poles are powered only during the night, meaning that during the day the sensors should be able to work on accumulators that should be recharged during the night.
Validation method/Relevant KPI	Energy consumption

ID	UC6_2

Unique Name/Title	High density signalling
Priority	High
Type	Functional
Brief Description	The street lighting actuators/controllers shall be able to communicate with the gateway (aggregator).
Rationale	The solution will be deployed over 9000 public lighting poles, meaning that the signalling in the network will be significant and should be supported over the RAN part.
Validation method/Relevant KPI	% signalling messages successfully received.

ID	UC6_3
Unique Name/Title	Scalability
Priority	High
Type	Non-Functional
Brief Description	The solution should be able to scale easily.
Rationale	These solutions are deployed in steps and usually the opportunity to scale a solution with ease is key. The scalability of the system should be permitted both from the platform we propose, but also from the network perspective (RAN).
Validation method/Relevant KPI	Correlate number of street lightning actuators/controllers deployed per area (area to be defined in this context) with the requirements on the Intelligent Lighting Platform and network (RAN).

ID	UC6_4
Unique Name/Title	Security
Priority	High
Type	Functional
Brief Description	The solution should be able to assure security and privacy of users' data.
Rationale	The solution must ensure globally the protection of resources and encompassing several dimensions such as authentication, data confidentiality, data integrity and access control.
Validation method/Relevant KPI	Passing the penetration tests, Protection against cyber-attacks, Isolation.

ID	UC6_5
Unique Name/Title	Dynamic QoS provisioning, support and enforcement.
Priority	High
Type	Functional
Brief Description	Activation of slice specific QoS and QoE metrics.
Rationale	The solution must be able to assure on-demand provisioning of the use case according to specific slice metrics and real-time network conditions.
Validation method/Relevant KPI	The QoS and QoE metrics are maintained during the slice lifecycle.

ID	UC6_6
Unique Name/Title	Policy based dynamic reconfiguration

Priority	High
Type	Functional
Brief Description	Enforcement of traffic policy for dynamic network reconfiguration.
Rationale	According to resource layer load and use case requirements, control plane uses a policy based approach for dynamic reconfiguration, in order to ensure end-to-end performances of the solution.
Validation method/Relevant KPI	Traffic redirection according to enforced policy.

ID	UC6_7
Unique Name/Title	Redundancy and resilience mechanisms
Priority	Medium
Type	Functional
Brief Description	The solution must provide redundancy and resilience to avoid slice disruptions.
Rationale	Physical and logical redundancy strategies should be implemented to provide resilience according to criticality of the slice.
Validation method/Relevant KPI	No slice disruptions when different solution components are failing.

ID	UC6_8
Unique Name/Title	Service monitoring
Priority	Medium
Type	Functional
Brief Description	Performance monitoring of slice from multiple perspectives.
Rationale	The data sets obtained from the monitoring performed at the data plane are employed at the cognition-based management plane to the self-reconfiguration of the slice, to keep, for example, the expected QoE.
Validation method/Relevant KPI	Availability of network performance KPIs: packet loss ratio, latency, jitter.

ID	UC6_9
Unique Name/Title	Distributed application components
Priority	Low
Type	Functional
Brief Description	The solution must be able to make data available for different applications.
Rationale	Based on the three-layer approach, the data obtained from the network and available through middle layer can be accessed by any application using a REST API.
Validation method/Relevant KPI	Data availability for different applications.

ID	UC6_10
Unique Name/Title	Support of VNFs and PNFs
Priority	Medium
Type	Non-functional

Brief Description	The solution must be able to interwork with both PNFs and VNFs.
Rationale	Edge devices (lighting poles and Intelligent Lighting platform) should be unaware of the type of network functions (physical or virtual) the frame is transiting as long as slice requirements are maintained during its lifecycle.
Validation method/Relevant KPI	Seamless end-to-end solution functioning over both VNFs or PNFs.

ID	UC6_11
Unique Name/Title	Heterogeneous network access types (LoRaWAN/LTE-M/5G)
Priority	High
Type	Non-functional
Brief Description	Seamless connectivity to Intelligent Lighting platform.
Rationale	The sensors should be able to authenticate and communicate over LoRaWAN, LTE-M or 5G in order to transmit the lighting pole data to the Intelligent Lighting platform.
Validation method/Relevant KPI	Successful authentication procedure. Communication between lighting poles and Intelligent Lighting platform is unaltered.

4.7 Use Case 7: Provisioning of distributed application services (B2B) such as CRM/ERP services

4.7.1 Scenario Description

Nowadays, many professions require individuals working from a remote location as in the office, thus implying having their "office environment/desktop" whenever -at least during working hours- and wherever they are (at their customers' premises, at their home or at other remote locations), which is not often achieved. At the same time, it is forecasted that in the near future more and more business activities –besides the commercial/sales- will be performed remotely utilising the so called "Mobile/Virtual Office" approach. Examples of those activities are: journalists processing captured news from a number of remote locations, engineers running applications and simulations remotely, doctors updating patient records and consulting medical support systems, etc. This will necessitate the support of remote access of possibly highly demanding applications in terms of latency, storage capacity, responsiveness, etc., as offered on premises.

Additionally, it is common for businesses to outsource part of their operational activities to third parties. Such services could be part of the common business processes such as economic, human resources management, logistics, IT, Customer Relationship Management (CRM), Enterprise Resource Planning (ERP), or even part of specific business activities. In practice, this outsourcing could take the form of provisioning and operation of Software Applications-Services as a Service (SaaS), or even the provisioning and operation of Platform as a Service (PaaS). Usually the applications/services are installed on the business customer premises and are supported and sometimes even operated by the Service/Platform provider. This however requires a lot of resources from the Service Provider to maintain a high QoS level, as it may require their physical presence in a large number of locations where the business customers' premises are located. It also may require high investments in hardware/software systems by the business customer or by the Service Provider, depending on their agreement and the type of service. It is also possible for a CRM/ERP service to be deployed on the cloud and provided as a service, which, however, may introduce delays and may lead to low application responsiveness.

In this context, this use case focuses on a CRM/ERP service provider aiming to support multiple business customers (Business to Business) with their CRM/ERP services. The service provider is necessary (a) to provide multiple instances of its CRM/ERP application to different Business Customers –i.e. run a multitenant application-, and (b) depending on the individual organizational structure of each business customer, to provide and support instances of the various CRM/ERP application components to different business units –of the same business customer- residing in various geographical locations, while being able to perform remote management and orchestration of these services.

This service provisioning will run over cloud/edge deployments instead of on-premises deployments, in order for both the business customer and the CRM/ERP service provider to benefit from the large computing power/ capacity/ connectivity resources of cloud/edge computing and to avoid investment in HW/SW infrastructure. However, there may be strict requirements, which need to be satisfied regarding the application responsiveness, the application/service components/repositories/etc. synchronisation and the storage space of specific application/service components. For this purpose, orchestration of various instances, and resource management and allocation are necessary, so that the CRM/ERP service provider can maintain its SLAs with their customers. The various distributed application components of the CRM/ERP service/application and the interactions of various stakeholders are presented in Fig. 4.7.1.

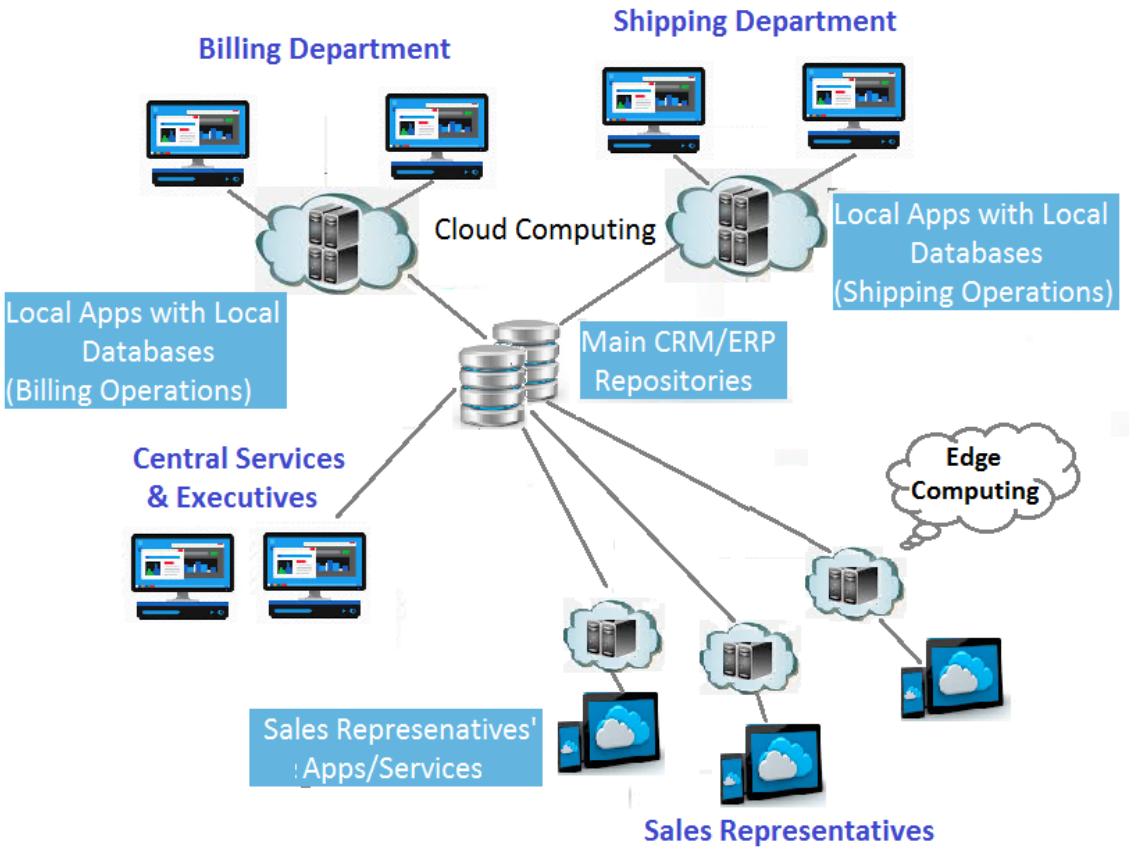


Figure 4.7.1: Distributed CRM/ERP services.

The way in which different stakeholders interact with the CRM/ERP service/application deployed over the MATILDA framework is presented in Fig. 4.7.2.

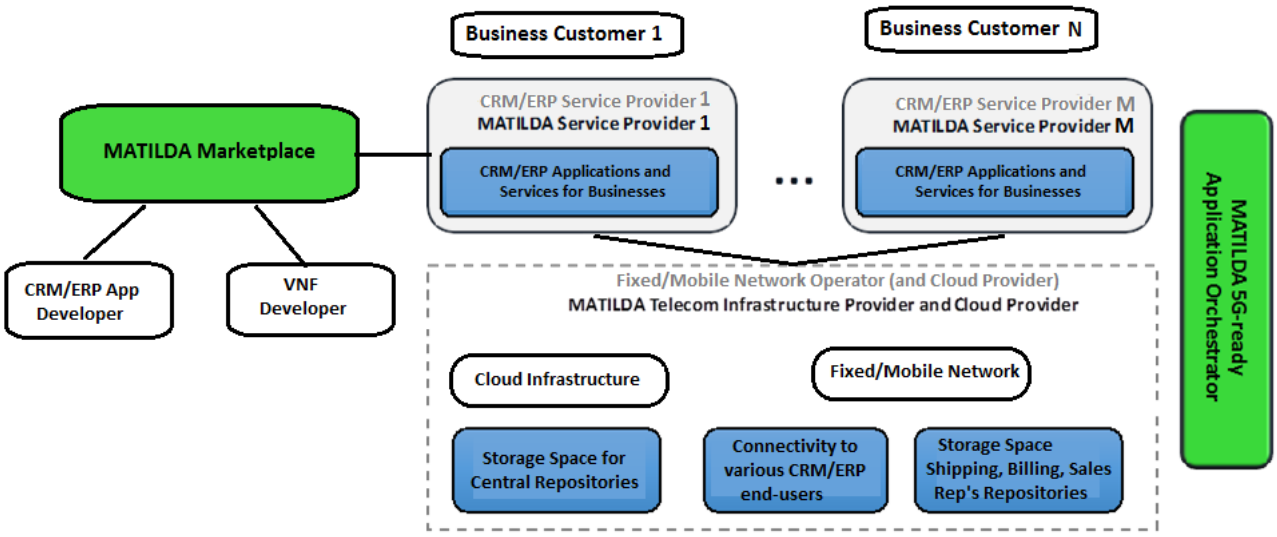


Figure 4.7.2: MATILDA based CRM/ERP services.

4.7.2 Objectives

The main objective of this use case is to provide CRM/ERP services over a cloud/edge deployment with resources at geographically distributed locations instead of on-premises, exploiting the edge computing and SDN/NFV capabilities. The ultimate goal is to optimise resource allocation per service/application instance (per service customer-tenant) and per service/application instance component over time, while pertaining to the SLAs between the CRM/ERP service provider and its business customers.

Challenges and Innovation

To support the envisioned distributed CRM/ERP services many challenges need to be overcome, starting with the proper transfer of a CRM/ERP application SW from on-premises to a cloud/edge deployment, which implies splitting the application SW into multiple components, each with different resource requirements, residing at different locations. Additionally, towards maintaining the SLAs between the CRM/ERP service provider and their business customers, another great challenge is the orchestration of resources to serve multiple tenants, possibly with distributed service/application components, each one having different resource/performance requirements that change with time.

For this purpose, contrary to existing centralised deployments (on-premises or on the cloud) and static network slicing approaches, MATILDA innovates by bringing the application components to the edge –thus reducing latency/delay-and by introducing Application-Aware Network slicing and 5G-Ready Application orchestration.

4.7.3 Scenario Workflow

Usually CRM/ERP services consist of various components, such as global and local repositories, data entry components, data retrieval/reporting components, data processing components, etc., for various operations. CRM and ERP systems resemble a lot in terms of software architecture and QoS requirements per application components, and for this reason we consider both under this use case.

More specifically, by CRM we refer to a system aiming at the rapid sharing (recording, storing and reporting) of information related to customer interactions. In the context of this use case, we consider a distributed CRM system, which consists of: (1) local storage components used by sales representatives to store and record information about their clients and thus maintain contact with them –deployed close to the sales representatives' area of operation-; (2) central repositories and processing components used by executives to obtain complete reports on business customers and

sales – deployed close to the executives' central offices-; (3) local components used by shipping clerks to verify addresses – deployed close to the shipping clerks' area of operation -; (4) central components to be used by the billing department to create invoices –deployed close to the billing departments' offices-; and other components depending on the additional operations/functionalities of the system.

Similarly, by ERP we refer to a system focusing on the digitization of business processes allowing for rapid sharing of business information throughout various departments. In the context of this use case, we consider a distributed ERP system, which consists of: (1) product planning components –deployed close to their respective product planning department-, (2) product development components – deployed close to their respective department-, (3) human resources management components – either collecting information from employees – thus, deployed close to employees' locations, or supporting reporting functionalities – thus, deployed close to the human resources management departments-; (4) financial processes related components - deployed close to their respective financial department-; and other components depending on the additional operations/functionalities of the system.

In both cases the users of the CRM/ERP systems may be on the move/work out of office, which implies the need for easy access to the CRM/ERP systems, while the business competitiveness sometimes requires access out of working hours/when the employee is not at the business premises. More specifically, in the scenario workflow of this use case, a sales representative operating in a specific area requires access to his/her client files anywhere and at any time. Given the fact that most times the sales representative is at his/her clients' premises, and he/she requires to have access to this content as from his/her office environment, the local repository and the CRM component handling this information is deployed at the "edge", these components should be able to scale, according to storage and bandwidth requests.

Upon insertion of a new order by the sales representative (or by a client) into the CRM local storage component, the central repository is also updated/synchronised with the new data, and also this information is forwarded to the local components of the billing and then the shipping departments, in order to handle the relevant payment and shipping processes. The respective clerks/payment - shipping departments generate the data accompanying these processes, accordingly. During this process, information is infused –at different stages- to the central repository so that the "global" reports on business customers and sales are updated.

It is obvious that in the case of large enterprises, with intense commercial activities, performed by a large number of sales representatives scattered in various geographical locations, massive connectivity is essential. In such cases low latency and network traffic minimisation can be achieved by distributing smaller instances of the CRM/ERP application close to those employees (utilising edge-computing). The main challenge is to ensure having the available resources during the various operational states of the service (that is; during different time windows/periods, which are characterised by the execution of different sets of operations), while avoiding over-provisioning and therefore minimise costs. Actually, resources need to be guaranteed (a) for a large number of commercial transactions performed around a large geographical area almost exclusively during working hours, (b) for business-running operations performed also almost exclusively during working hours involving interactions with central business systems, as well as (c) for repositories' synchronisation during low traffic periods (e.g. during nights, weekends, etc.). Therefore, MATILDA's auto-scaling capabilities and optimization mechanisms are necessary for a successful deployment.

The MATILDA architecture will facilitate deployment and operation of the overall network-aware application over the 5G access network fulfilling the strict performance and delay requirements. In case of a CRM/ERP service provider offering services to more than one customer, different network slicing per tenant/customer and possibly per application may also be offered. An indicative service graph is presented in for the various service/application interactions and can be extended for more than one tenant.

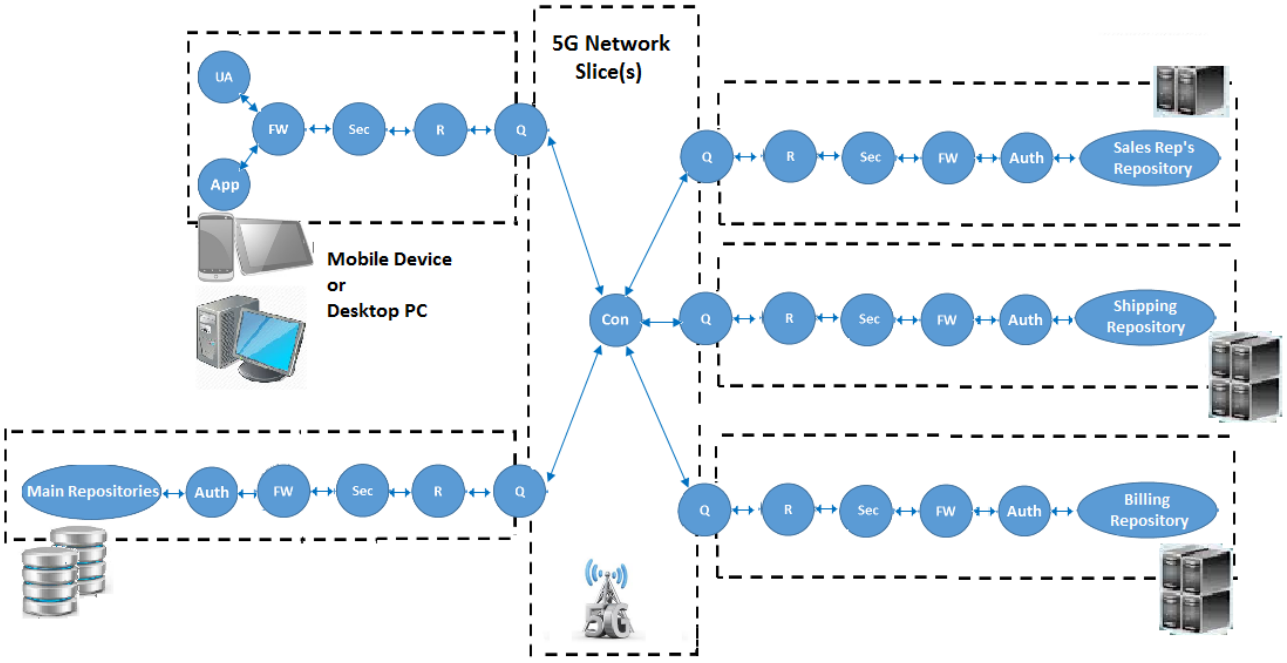


Figure 4.7.3: Indicative service graph for CRM/ERP services [UA: user access, FW: firewall, Sec: Security, R: Routing, Q: Load balancing, Con: Controller/Orchestrator, Auth: Authorization].

4.7.4 Use Case-Derived Requirement

ID	UC7_1
Unique Name/Title	Flexible Bandwidth allocation.
Priority	High
Type	Performance
Brief Description	Flexible bandwidth allocation is needed in different links of the application graph, depending on the CRM/ERP application and the state in which it is found.
Rationale	The links between the users and the local repositories both in CRM and ERP apps, are not expected to pose significant bandwidth requirements as the amount of data transferred is usually low. However, high bandwidth is occasionally necessary for the synchronisation of the central repository with the local repositories, which of course depends on the amount of data that are stored/used/recorded by the CRM/ERP application. Therefore, static bandwidth allocation per tenant is not the optimal way to handle bandwidth resources; flexible bandwidth allocation shall be considered instead.
Validation method/Relevant KPI	Measurement of bandwidth fluctuations of the application graph links, and comparison with target values. KPI: throughput (in Mbps) of each link.

ID	UC7_2
Unique Name/Title	Low Delay/Latency
Priority	High
Type	Performance
Brief Description	Low Delay is required in different links of the application graph, depending on the CRM/ERP application, the connected components and the application state.
Rationale	High interactivity and responsiveness are required, especially for the application components used by the sales representatives, the billing/shipping departments

	and the executives. Therefore, strict delay requirements are posed for the links between the users and the local repositories both in CRM and ERP apps.
Validation method/Relevant KPI	Measurement of delay fluctuations of the application graph links, and comparison with target values. KPI: delay time (in ms/s (depending on the application components' state)) of each link.

ID	UC7_3
Unique Name/Title	High Availability
Priority	High
Type	Performance
Brief Description	The CRM/ERP services shall be always available.
Rationale	Corporate/business users pose strict availability requirements to the CRM/ERP services, since potential unavailability may disrupt businesses' operations and may potentially incur severe economic damage.
Validation method/Relevant KPI	The availability level shall reach 99.99% of operational time, and will be measured after the completion of the MATILDA development stage. Relevant KPIs are: (time the service is available) / (total time from service deployment up to the time of measurement)

ID	UC7_4
Unique Name/Title	Interoperability with various Access Networks (WAN, LTE, 5G, etc.)
Priority	High
Type	Non-Functional
Brief Description	The CRM/ERP services shall be supported seamlessly over various Access Networks.
Rationale	Mobility is a key feature of the users of CRM/ERP systems, implying that they can be served by different access networks depending on their location and the availability of each technology there. Therefore, the CRM/ERP services shall be supported seamlessly over various Access Networks; meaning that the underlying MATILDA framework shall be interoperable with various access networks.
Validation method/Relevant KPI	Testing of services' operation when end users are served by different access networks (WAN, LTE, 5G, etc.).

ID	UC7_5
Unique Name/Title	Dynamic QoS provisioning
Priority	High
Type	Functional
Brief Description	The MATILDA Framework shall support dynamic QoS provisioning.
Rationale	Given the fact that QoS requirements vary in time especially for corporate apps, dynamic QoS provisioning will allow for resource management optimisation for different applications and/or tenants so that high QoS is guaranteed when needed, while resources are released during inactivity times.
Validation method/Relevant KPI	Performance of tests with various tenants/applications, measuring QoS metrics and comparing the results against target values defined by the services' graphs.

ID	UC7_6

Unique Name/Title	Network Programmability
Priority	High
Type	Functional
Brief Description	Network Programmability is required in order to achieve optimal allocation in terms of storage space in edge nodes, network resources for the different links between the application components, processing power in edge nodes, etc.
Rationale	Given the fact that the resources required from various application components/tenants/etc. vary in time, network programmability will allow for optimisation of resource management based on actual needs per component/tenant/etc.
Validation method/Relevant KPI	This requirement can be validated by design.

ID	UC7_7
Unique Name/Title	Network Slicing
Priority	High
Type	Functional
Brief Description	Network slicing per tenant and per application/service shall be supported.
Rationale	Network slicing is required, in order to guarantee end-to-end QoS for a specific application (instance)/tenant, over a deployment consisting of multiple, interconnected (application) components residing in different network edges.
Validation method/Relevant KPI	This requirement can be validated through monitoring the MATILDA Orchestrator functions related to network slicing, e.g., the identification/creation of network slices, the resource allocation to specific slices, etc.

ID	UC7_8
Unique Name/Title	Distributed application components
Priority	High
Type	Non-Functional
Brief Description	The CRM/ERP applications shall be modular, consisting of a number of application components, which can be deployed at different (remote) locations.
Rationale	In order to benefit from the MATILDA framework and edge computing, applications shall consist of a number of components that can be deployed in separate locations, while operating as a single application deployed on-premises.
Validation method/Relevant KPI	This requirement can be validated at applications' design phases.

ID	UC7_9
Unique Name/Title	Scalability
Priority	High
Type	Application/Non-Functional
Brief Description	The CRM/ERP application shall be able to scale both in terms of supporting a large number of application instances and in terms of supporting a number of components per instance.
Rationale	Practically, scalability is required so that the CRM/ERP service provider is capable to support many businesses/users, each having their own instance. At the same time, scalability is required since one instance may need to include multiple repositories/local application components/etc., for example, in order to support a

	large number of sales representatives, etc.
Validation method/Relevant KPI	This requirement can be validated at applications design phases.

4.8 Use Case 8: Mobile Night Safeguard Systems

4.8.1 Scenario Description

In the near future, Ultra-High-Definition (UHD) video services are expected to contribute a lot to the growth of data traffic. However, this type of services requires a lot of network resources to maintain a high QoS level, including low latency and high bandwidth capacity. This use case focuses on services involving Video Streaming/Recording and real-time video transmission over a network. The camera can be either moving or static. Indicative scenarios of this use case are the following:

- surveillance services including recording performed by a drone equipped with a camera,
- security/surveillance (or even events/TV broadcasting) services including recording from a moving camera (i.e., inside a vehicle) and transmitting video in real-time,
- a set of static cameras operating over a wide area and recording and transmitting videos in real-time when triggered by specific events, etc.

In such cases, video capturing takes place at specific locations (e.g., assets/people/borders' locations), usually distributed around a large geographical area, while video processing/analysis/storage/etc. is performed at specific remote locations with high compute and storage capabilities, thus requiring the transmission of the UHD video over an access network. At the same time, the remote control of the video capturing devices (which can be many) requires low latency network connectivity.

In this context, we can consider the use case of a Private Security Service Provider. More specifically, a Private Security Service Provider wants to introduce an IoT automated service to cut its personnel costs down, to avoid risking their safety in particularly dangerous scenarios and to provide its customers a set of extended features (e.g., video acquisitions from ground and aerial view, real-time video elaboration and alarming systems and so on). For this purpose, it wants to put in service a number of driverless cars and drones able to follow configured paths in the city, providing the subscribed customers periodical monitoring of their assets, by capturing relative high definition videos from different views. In public security implementations, a real-time video processing service will allow face recognition leveraging on criminal databases and fast and highly available alarming systems.

According to usual surveillance criteria the acquired data will not be stored on board, but immediately transmitted to a remote storage centre able to provide a safe and highly available redundant storage system. Moreover, to avoid transmission of huge amount of data to a remote Cloud, the first stage of storage is foreseen in the local cloud (referred as Fog) for a preconfigured (timeout) amount of days. After the timeout expiration, any crime-related data will be transferred to central storage facilities - private or public Cloud datacentres (DC)- with a proper service able to guarantee the required confidentiality, integrity, and availability requirements, while the remaining data will be destroyed.

The camera acquisitions will be enriched by a configurable number of advanced sensors to provide improved intrusion detection capabilities well behind the human guards' capabilities (e.g., Infrared thermographic cameras and so on).

Moreover, the IoT devices will maintain a constant contact with the Security Service Provider Headquarters to allow central supervisors to keep control at any moment (manual directions in emergency cases, direct video analysis on demand, direct video camera directions control or IoT device control on demand, and so on).

4.8.2 Objectives

The main objective of this use case is to show the MATILDA ability to coordinate a complex multi-site edge and cloud scenario to provide to a vertical customer an evolved service leveraging on a dedicated proper dynamic network slice. A practical deployment and test of the driverless car or of the drone case would be an attractive way to demonstrate MATILDA framework's capabilities.

Challenges and Innovation

The main challenges are related to the desirable high density of user devices to control a big town or a region, the huge amount of high data-rates HD videos, the low-latency and availability requirements on the network slice used for the alarm system and the data transfers in both ways (from criminal faces databases, to video storage centres in local cloud or remote cloud) and – central innovative point – the dynamic coordination of mobile IoT services connected to the local/remote clouds and network slices with optimized allocation and use of virtual functions and network infrastructures.

4.8.3 Scenario Workflow

The various distributed components of the Mobile Night Safeguard service/application and the interactions of stakeholders are presented in Fig. 4.8.1.

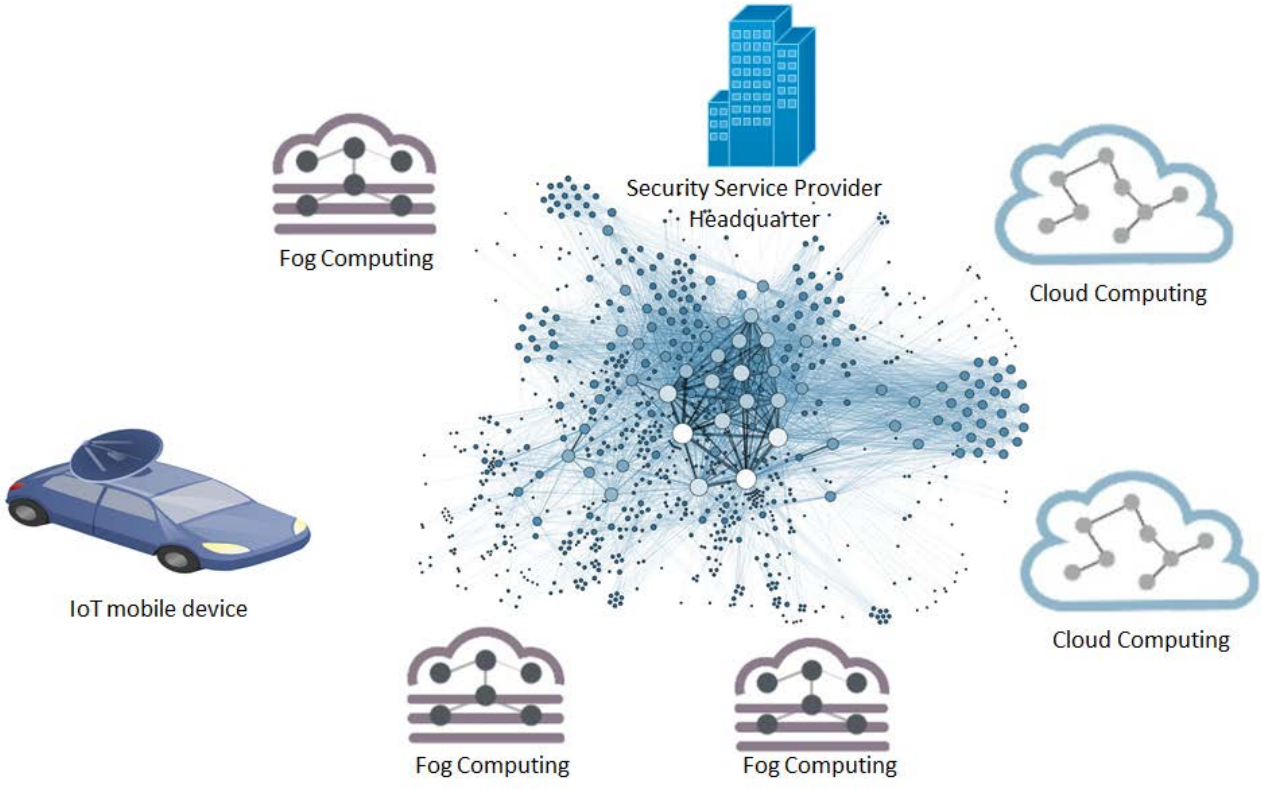


Figure 4.8.1: Mobile Night Safeguard services using Fog and Cloud Computing

From a high-level point of view the main blocks will be constituted by:

- Advanced IoT mobile devices (driverless cars and drones) equipped with a rich set of cameras/sensors and all the proper communication functions;
- Local Cloud infrastructures (Fog for short term storage) able to satisfy the required storage space, high availability and security requirements;

- Remote Cloud infrastructures for long term storage of selected videos also with high availability and security constraints;
- A reliable network (slice) infrastructure to connect the IoT devices with the central Headquarters and the Cloud infrastructures with additional low latency requirements;
- A holistic orchestration to dynamically optimize the communication and cloud infrastructure resources according to dynamic IoT positions and all the relevant criteria (e.g. network congestions, availabilities, and so on).

The MATILDA architecture will facilitate deployment and operation of the overall network-aware application over the 5G access network fulfilling the strict performance and delay requirements. In case of Mobile Night Safeguard services provisioning to more than one customers/agencies/services, different network slicing per tenant/customer and possibly per application may also be offered. An indicative service graph for the various service/application interactions is presented in Fig. 4.8.2 and can be extended for more than one tenant.

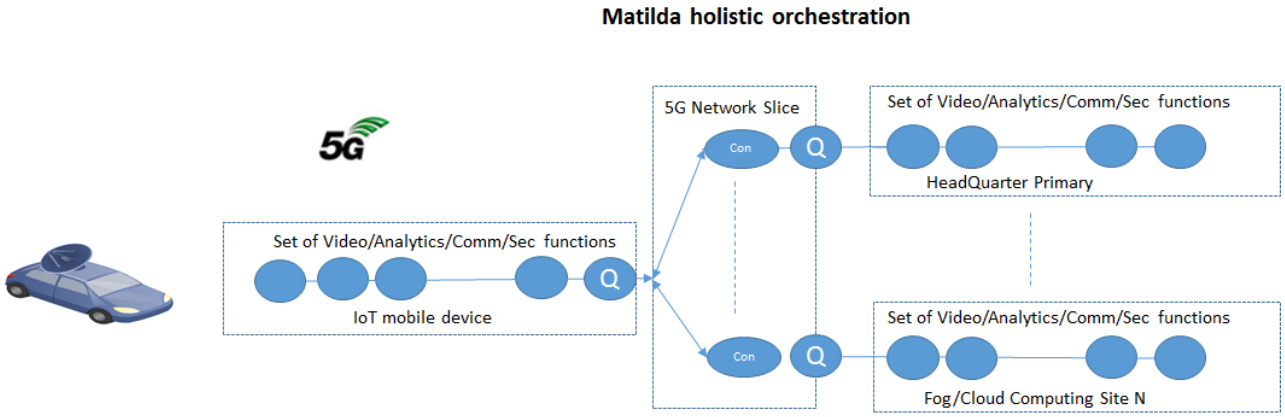


Figure 4.8.2: Indicative service graph for Mobile Night Safeguard services [Q: Load balancing, Con: Controller/Orchestrator].

4.8.4 Use Case-Derived Requirements

ID	UC8_1
Unique Name/Title	Flexible Bandwidth allocation.
Priority	High
Type	Performance
Brief Description	Flexible bandwidth allocation is needed for various links of the application graph.
Rationale	Bandwidth requirements for the various connection links can vary a lot depending on the time of the day, and on the occurrence of an event. For scheduled surveillance activities, high bandwidth connections between the IoT devices and local cloud servers will be required during the surveillance times, as well as between the local cloud servers and the central servers during the scheduled synchronisation times (i.e. after the preconfigured timeouts). Upon occurrence and detection of an incident however, high bandwidth may be requested outside the scheduled times for the link between a local cloud and the on-the-fly centralized human control server(s). In this case, the system shall be able to provision this connectivity. Due to the high bandwidth demands, static bandwidth allocation per tenant is not the optimal way to handle bandwidth resources; flexible bandwidth allocation shall be considered instead.

Validation method/Relevant KPI	Measurement of bandwidth fluctuations of the application graph links, and comparison with target values. KPI: throughput (in Mbps) of each link.

ID	UC8_2
Unique Name/Title	Low Latency
Priority	High
Type	Performance
Brief Description	Low Latency is required in different links of the application graph.
Rationale	Low latency is required for the links that are used for transferring alarms and for the support of alarm signalling actions. At the same time high interactivity and responsiveness are required especially for controlling the various surveillance IoT devices such as drones, vehicles, alarm systems, etc.
Validation method/Relevant KPI	Measurement of latency fluctuations of the application graph links, and comparison with target values. KPI: delay time (in ms/s (depending on the application components' state)) of each link.

ID	UC8_3
Unique Name/Title	High Availability
Priority	High
Type	Performance
Brief Description	The Surveillance/Security services shall be always available.
Rationale	Given the criticality of the security services provisioning and the fact that the time an incident occurs is unknown, high availability is a critical requirement.
Validation method/Relevant KPI	The availability level shall reach 99.99% of operational time, and will be measured after the completion of the MATILDA development stage. Relevant KPIs are: (time the service is available) / (total time from service deployment up to the time of measurement).

ID	UC8_4
Unique Name/Title	Interoperability with various Access Networks (WAN, LTE, 5G, etc.)
Priority	High
Type	Non-Functional
Brief Description	The surveillance/security services shall be supported seamlessly over various Access Networks.
Rationale	Different surveillance/security services and end-devices may utilise various access network technologies depending on the availability of such technologies in the covered area. Therefore, the security services shall be supported seamlessly over various Access Networks; meaning that the underlying MATILDA framework shall be interoperable with various access networks.
Validation method/Relevant KPI	Testing of services' operation when end users are served by different access networks (WAN, LTE, 5G, etc.).

ID	UC8_5
Unique Name/Title	Network Programmability
Priority	High

Type	Functional
Brief Description	Network Programmability is required in order to achieve optimal allocation in terms of storage space in fog nodes, network resources for the different links between the application components, processing power in fog nodes, etc. Besides, the local cloud storage location should dynamically change based on the position of the IoT device, especially in the wide area cases, in order to achieve low latency and optimise performance.
Rationale	Given the fact that the resources required from various application components /tenants/etc. vary in time (depending on the occurrence of an event) and location (depending on the location of the end IoT device), network programmability will allow for optimisation of resource management based on actual needs per component/tenant/etc.
Validation method/Relevant KPI	This requirement can be validated by design.

ID	UC8_6
Unique Name/Title	Network Slicing
Priority	High
Type	Functional
Brief Description	Network slicing per tenant and per service shall be supported.
Rationale	Network slicing is required, in order to guarantee end-to-end QoS for a specific security service and/or tenant, over a deployment consisting of multiple, interconnected (application) components residing in different network edges.
Validation method/Relevant KPI	This requirement can be validated through monitoring the MATILDA Orchestrator functions related to network slicing, e.g. the identification/creation of network slices, the resource allocation to specific slices, etc.

ID	UC8_7
Unique Name/Title	Optimised Storage Location in Distributed Storage Facilities
Priority	High
Type	Functional
Brief Description	The transferring of interesting data from a local cloud to different target DCs -part of the central storage facilities (central DC)- should satisfy a number of optimisation criteria.
Rationale	In case the central storage facilities (DC) consists of multiple DCs, the interesting data from local cloud should be placed to the most appropriate target DC taking into account several criteria, such as: availability of DC, availability of resources (i.e. storage space, compute resources, network connectivity), proximity (to minimise latency), and so on.
Validation method/Relevant KPI	This requirement can be validated at applications' design phases.

4.9 Use Case 9: Banking on the Cloud

4.9.1 Scenario Description

The focus of this use case lies in the application of advanced network characteristics to the area of financial services. Similar to the majority of the other sectors, banks and financial service companies can also benefit from cloud computing, creating a more flexible, agile business model to meet the growing business needs in a dynamic and competitive landscape. In particular, cloud computing can offer a vast range of capabilities that banks need on a flexible basis and help them to transform their business processes and enhance their ability to grow in new sectors or regions without the time and cost burdens involved with establishing a physical presence. A significant benefit that should also be mentioned is the efficiency that a cloud-based infrastructure may bring into decision-making and policy implementation. Having remote access to the information regarding new implementations and internal changes, the cloud brings efficiency in providing access to all involved parties in a single format and a place to securely evaluate the matter. Finally, banks will have a much better ability to provide consistent service to customers across branches, geographies and integrate a plethora of disjoint customer information and analytics.

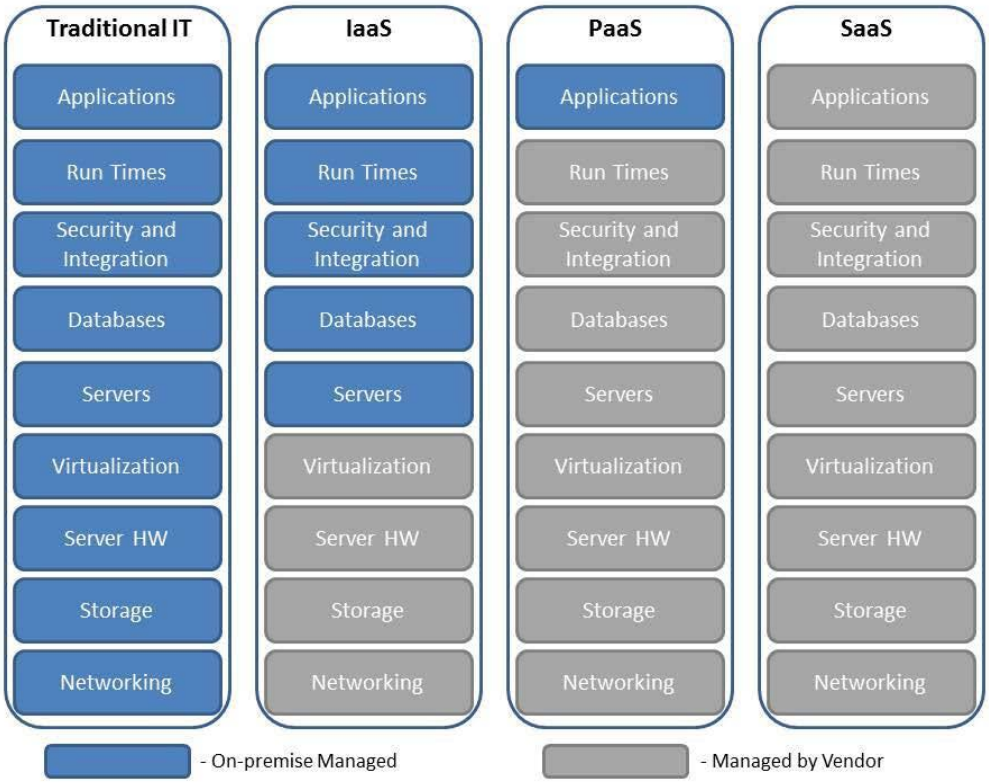


Figure 4.9.1: Cloud service models available for banking. Source: Wenk, "D. Porter's Five Forces Analysis of Cloud Computing", Jan 2015, SCIP Magazine, cited by [Blazheski-2016].

As depicted also in Figure 4.9.1, several cloud service models could be considered for the financial institutions. The Business Process-as-a-Service (BPaaS) model refers to the one where the cloud is used for standard business processes such as billing, payroll, or human resources. Software-as-a-Service (SaaS) model is a model in which a cloud service provider houses the business software and related data, and users access the software and data via their web browser, without being able to manage or control the underlying cloud infrastructure. In this case, the types of software that can be delivered include accounting, customer relationship management, enterprise resource planning, invoicing, human resource management, content management and service desk management. In the Platform-as-a-Service (PaaS) model, the cloud service provider offers a complete platform for

application, interface, database development, storage and testing, allowing businesses to streamline the development, maintenance and support of custom applications, lowering IT costs and minimizing the need for hardware, software, and hosting environments. In this case, the users have more control as they are allowed to deploy their own applications onto the cloud by using the adequate libraries, services and tools. Finally, the Infrastructure-as-a-Service (IaaS) model allows businesses to buy servers, software, datacentre space or network equipment as a fully outsourced service, offering even more control to the users.

Focusing on providing banking services though applications, i.e. on-line and mobile payments, different stakeholder interactions will take place. The service consumers are either the individual or corporate clients of the financial institution, while the bank, along with collaborating partners, depending on the service offered will be the service provider, also in charge of the development of the applications. The telecom infrastructure provider will offer network resources spanning from 5G to fixed access network capabilities among multiple sites. For security and regulatory purposes, the cloud infrastructure may be provided by the bank itself.

4.9.2 Objectives

Successful deployment of both core banking services, as well as branch transactions and Internet/ mobile services in the cloud, banks face a number of challenges. To narrow down the use case objectives, the use case focuses on the following features:

- Automated creation of promotional offers for credit card or mobile money products taking into account nearby collaborating merchants,
- Deployment of an Augmented Reality (AR) service in central markets and malls, where -based on high-efficiency positioning technologies- for each collaborating shop recorded on camera the benefits (e.g., % discount) are displayed on the mobile phone/smart device.
- Successful biometric authentication based on image and voice recognition coupled with traditional PIN code implementation.

Challenges and Innovation

The most inhibiting factor for deploying a bank service on the cloud concerns security and regulatory issues. Indeed, banks and financial institutions in general are governed by regulations for mission critical applications and customer data on-premises, and “data sovereignty” may be unclear in the cases where the chosen cloud storage providers are located in a different jurisdiction. This is why the majority of financial institutions prefer deployment in private cloud infrastructures.

For security purposes, a MATILDA implementation will have to offer separate network slices for the authentication with the bank service, which will guarantee very good network performance with low latency characteristics. As an external cloud will bring an increased need to balance the demands for speed, agility, and autonomy with security requirements, MATILDA will have to focus on encryption and obfuscation.

Other challenges that leave room for innovation involve the delivery of enhanced customer experience, higher availability of critical systems (mainly core banking) and the potential reduction of running costs. This could be achieved in the MATILDA framework with the use of dedicated network slices, monitoring of QoS metrics, applying policies based on SLA and QoS constraints and the development of intelligent application orchestration schemes, which will offer agile automated assignment of resources based on machine learning approaches for workload prediction, profiling and proactive resource allocation, ensuring high-quality of offered services.

To support a banking on the cloud scenario, the network must be able to provide low service latency, since offers must be made in user-perceived real-time. So, the round trip of getting user mobility

characteristics and identifying purchases to the analytics module that comes up with a next-stop suggestion must be short. Even more, since such places are crowded the slice must be prepared to other this service to client-dense vicinity while maintaining secure network access. The addition of an Augmented Reality presentation layer ups the requirements in terms of data-rate required.

4.9.3 Scenario Workflow

The main goal is to achieve a highly agile and reliable system that will involve all banking services, i.e. core banking, branch/ATM/credit card transactions, on-line and mobile payments, transaction with other financial institutions, etc., where the different services/components will communicate in real-time (see Figure 4.9.2).

In detail, customers and the bank itself will be able to follow in real-time the changes in the customer's account balances based on a transaction taking place, e.g., in a branch or an ATM. Customers will be able to perform different transactions and follow their progress. Additionally, the bank will be able to real-time monitor all transactions per customer, per employee or as a whole, in order to monitor specific regulatory issues. It will also be able to calculate and monitor advanced performance metrics corresponding to operational regulations (e.g., bank's debt exposure at any given time), as well as internal high-level objectives placed by the management. All the data will be stored in a cloud service. Clients will be able to access their services through their mobile device, tablet or computer. As expected, they will only have access to their data. Furthermore, certain transactions and clients will be considered of higher importance and therefore have priority, placing stricter QoS constraints for specific cases.

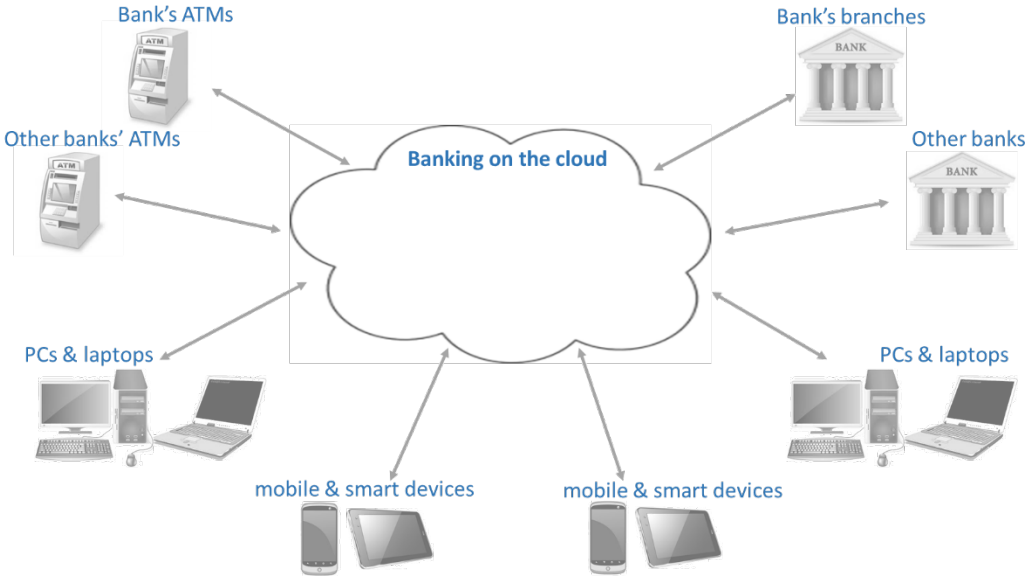


Figure 4.9.2: The "Banking on the cloud" vision.

Focusing on the above-mentioned advanced promotional offers case, this use case involves promoting specific offers through the bank's strategic partnerships with merchants (e.g. providing special offers/ benefits & store discounts, points for bank's loyalty schemes, or even return of percentage of money spent) mostly associated with credit card and/or mobile wallet products. In this scenario, a customer goes to a previously mapped large mall or central shopping area/ market using the credit card and/or mobile money app in his/her mobile device. As soon as the customer enters the general area a message/offer is received based on his/her location and known preferences (propensity models for shopping preferences based on historical data), exploiting the MATILDA analytics capabilities. Since 5G technology can offer high-efficiency positioning, this application can be further extended by an AR service where, based on the customer position, for each collaborating shop recorded in real-time on camera the related offers are displayed on the mobile phone/smart device. While AR can be provided

as a service without prior authentication, in the case of money transactions or personalized offers an authentication process must also be employed, either by using the traditional PIN/password approach or enhanced by image and voice recognition modelling.

It is obvious that, in the case of large malls and densely populated market areas the positioning data, the AR data and the image & voice data constitute a case where capacity enhancement and massive connectivity is of essence. Different network slicing per tenant and per application may also be offered. An indicative service graph is presented in Figure 4.9.3 and can be extended for more than one service per tenant and for various locations. The main challenge will consist of always having the available resources, while avoiding over-provisioning and therefore increase costs, especially in cases where a large number of transactions is taking place at the same time. Therefore, MATILDA's auto-scaling capabilities and optimization mechanisms are necessary for a successful deployment.

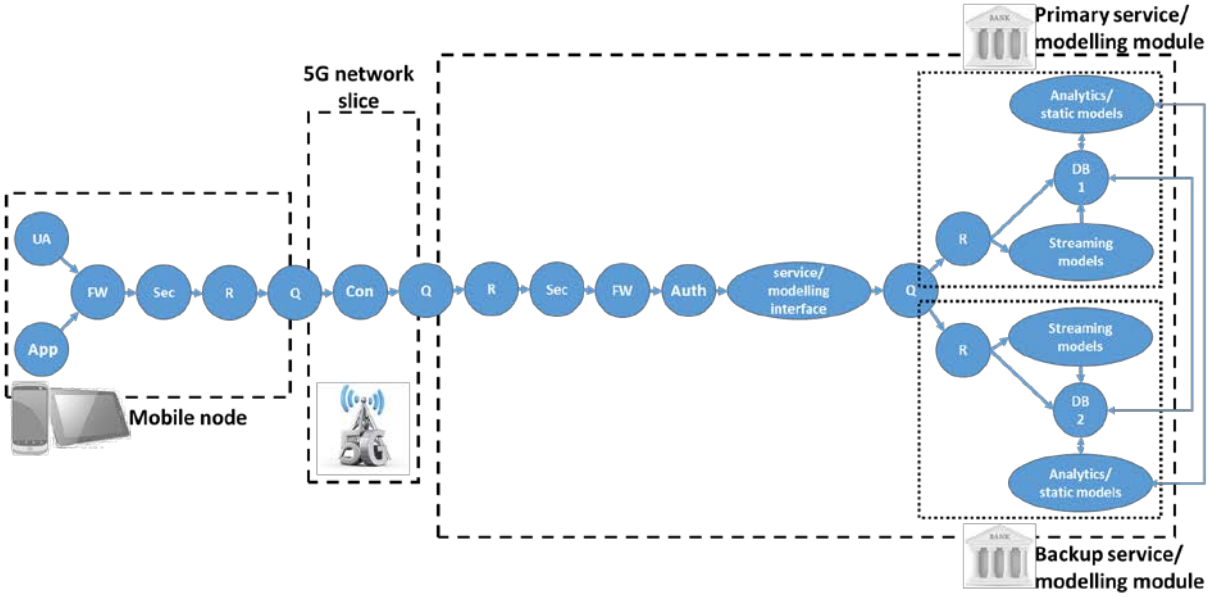


Figure 4.9.3: Indicative service graph for personalized banking services [UA: user access, FW: firewall, Sec: Security, R: Routing, Q: Load balancing, Con: Controller/Orchestrator, Auth: Authorization, DB: database(s)].

4.9.4 Use Case-Derived Requirements

ID	UC9_1
Unique Name/Title	High Availability & Reliability
Priority	High
Type	Non-Functional
Brief Description	The core banking services and specifically all transactions shall be available at all times.
Rationale	Banking services need to be available at all times for both corporate/business users and private users. The services must also take place in a seamless manner and the user be able to trust that no disruptions will occur that may result to economic damages.
Validation method/Relevant KPI	The availability level shall reach 99.99% of operational time, and will be verified through extensive testing. KPIs: • (time the service is available) / (total time since service deployment) • mean time between failure

ID	UC9_2
Unique Name/Title	Interoperability with various Access Networks (WAN, LTE, 5G, etc.) & High Coverage
Priority	High
Type	Non-Functional
Brief Description	Services must be provided over various Access Networks at various locations.
Rationale	MATILDA will offer 5G-ready applications and network services. In addition, banking services may be also accessed through various networks (WAN, LTE, 5G, etc.), at various locations/areas. For this reason, interoperability and a smooth connection between different access networks should be supported.
Validation method/Relevant KPI	Through the design phase of the MATILDA framework itself and its extended testing phase.

ID	UC9_3
Unique Name/Title	Security & Privacy
Priority	High
Type	Non-Functional
Brief Description	Transactions by different users and different levels of authorization shall be secure, in order to preserve system integrity.
Rationale	Since the services offered depend on the users, their roles and their authorization level, all operations must be secured and subject to specific access rules. Each client has access to specific data and services, while users can perform some -and not necessarily all- actions in the system. Lastly, all actions/data are personal and available only to those with the appropriate authorization level.
Validation method/Relevant KPI	Testing of various authorization levels for all MATILDA components that implement authorisation and access control.

ID	UC9_4
Unique Name/Title	Low Service Latency
Priority	High
Type	Non-Functional
Brief Description	Low Delay is required. The targeted value may be different per application/ different links of the application graph, and the connected components.
Rationale	All application components will have to offer real-time services displaying high responsiveness, e.g. updates on transactions and progress of specified operations taking place at different locations, local repositories synchronizing with central repositories, etc. For this reason, low delay for the links between the users and components is of great importance.
Validation method/Relevant KPI	KPI: delay time in ms for each link, at the testing phase.

ID	UC9_5
Unique Name/Title	Dynamic QoS provisioning
Priority	High
Type	Functional
Brief Description	Support and enforcement of dynamic QoS provisioning.

Rationale	Dynamic QoS provisioning will allow for resource management optimisation of different applications and/or app tenants at all times while avoiding overprovisioning. A high QoS will guarantee customer satisfaction and seamless transactions in cases of several services that are provided simultaneously. Additionally, in an effort to minimize costs, resources are released during times of low activity.
Validation method/Relevant KPI	Testing performance with various tenants/ applications and measuring per case QoS metrics.

ID	UC9_6
Unique Name/Title	Network Slicing
Priority	High
Type	Functional
Brief Description	MATILDA will offer network slicing per tenant and per application.
Rationale	Network slicing is important, in order to guarantee QoS for specific instances of services and users, while at the same time these services/ applications may be deployed at the same or different locations/ network edges.
Validation method/Relevant KPI	Testing MATILDA Orchestrator functions for creation/ resource allocation to specific slices, etc.

ID	UC9_7
Unique Name/Title	Scalability
Priority	High
Type	Non-Functional
Brief Description	It is important for the services/applications offered to support multiple instances of the application, as well as scaling instances with many components.
Rationale	Many users will be offered the same banking service at the same time, each having its own instance. Furthermore, for each instance multiple applications components may be required.
Validation method/Relevant KPI	This requirement can be validated at applications design phases.

ID	UC9_8
Unique Name/Title	Redundancy & Resilience mechanisms
Priority	High
Type	Non-Functional
Brief Description	Redundancy & resilience mechanisms are important, owing to the relevance of banking transactions and the critical nature of the services provided.
Rationale	The critical nature of all banking operations requires the provision of data storage mechanisms, as well as provision of services that will ensure the undisrupted continuation of operations even after a serious system failure, e.g. loss of data on a specific server.
Validation method/Relevant KPI	This requirement can be validated at applications design phases.

ID	UC9_9

Unique Name/Title	Network Monitoring
Priority	High
Type	Non-Functional
Brief Description	Network monitoring is necessary in order to deploy, detect issues/ problems, reconfigure and reallocate resources.
Rationale	In order to offer undisturbed operations, it is important to monitor whether the deployment of resources took place or to detect any issues that might occur, so that reallocation of resources can take place. In addition, monitoring will assist in QoS assurance and compliance with SLAs and service level policies, through dynamic or rule-based network reconfiguration.
Validation method/Relevant KPI	This requirement can be validated at the design phase.

4.10 Generic Use Case Requirements

A set of general requirements for the entire MATILDA framework are also defined, taking into account the individual use case requirements and generalizing part of them. Such requirements are presented in this section.

ID	GEN_1
Unique Name/Title	Modularity
Priority	High
Type	Non-Functional
Brief Description	Delivery of software/code (implementing the Toolkits' functionalities/features) that is modular and well-documented. Existence of supporting material per toolkit/component.
Rationale	Both the 5G-ready applications' and the MATILDA software must be designed according to the most recent design patterns and architectures. The different functionalities/features must be assembled in one or more logical blocks, thus implementing a modular architecture (in terms of Software) that will help improving software development and maintainability. Each block must interoperate with the others by means of one or more well-documented interfaces, thus enhancing their conceptual separation.
Validation method/Relevant KPI	This requirement can be validated at MATILDA system design and development phases. KPI: Function diagrams are used to indicate design patterns and improve code readability and understandability.

ID	GEN_2
Unique Name/Title	Extensibility/Upgradability
Priority	High
Type	Non-Functional
Brief Description	The MATILDA framework shall constitute a future-proof solution being easily extensible and upgradable.
Rationale	The MATILDA framework shall constitute a future-proof solution by continually keeping pace with state of the art developments and innovations. Therefore, the various MATILDA components shall be extensible/upgradable in terms of: • supporting software/hardware enhancements, • advanced features/functionality

	• new services/applications.
Validation method/Relevant KPI	This requirement can be validated at MATILDA system design and development phases. KPI: The delivered SW/code is well structured and well-documented, so that it is easy to extend and upgrade it to include future MATILDA components and services/applications.

ID	GEN_3
Unique Name/Title	Maintainability
Priority	High
Type	Non-Functional
Brief Description	The MATILDA components shall be maintainable throughout their lifecycle, from initial development, and during commercial operation, as all enterprise system solutions.
Rationale	The MATILDA components shall be maintainable, so that development and deployment of changes shall be performed with minimal risk of regression, and no changes to the system would negatively affect currently working functionalities.
Validation method/Relevant KPI	This requirement can be validated at MATILDA system design and development phases. KPI: Usage of versioning techniques and suitable SW tools to minimize maintenance effort.

ID	GEN_4
Unique Name/Title	Openness
Priority	High
Type	Non- Functional
Brief Description	The MATILDA Framework should be based on open technologies and APIs.
Rationale	Usage of open software and APIs is required in order to provide a platform that can be extensible, interoperable and reusable by a wide community.
Validation method/Relevant KPI	This requirement can be validated at MATILDA system design and development phases. KPI: Documentation of set of APIs and software tools.

ID	GEN_5
Unique Name/Title	User friendliness
Type	Non-Functional
Brief Description	The MATILDA Framework should provide user-friendly view to developers, service providers and service consumers.
Rationale	The usage of the various components of the MATILDA framework has to be user friendly, facilitating its adoption by a wide community.
Validation method/Relevant KPI	This requirement can be validated at MATILDA system design and development phases. KPI: Evaluation feedback collected by end users of the MATILDA platform.

5 5G Ecosystem Technologies Requirements

The MATILDA project will offer all the necessary components for the design, development and orchestration of 5G-ready applications and network services. Hence, it is essential to not only investigate all the related 5G Ecosystem Technologies that currently exist and their respective capabilities, but also to explore the challenges connected with the aforementioned technologies, as well as to propose new approaches beyond the state of the art. Based on this approach, a critical set of requirements can be produced that will be the cornerstone of the MATILDA framework implementation. In the following subsections, a detailed description on State of the Art technologies and requirements connected with the MATILDA's main components is offered; namely, as regards the following issues:

- The **5G-ready applications development approach and toolkit** targeted at the development of 5G-ready applications, incorporating the integrated development environment (IDE) and a set of application graph composers and other relevant tools (such as the policy editor) for the design and development of chainable application components.
- The **Marketplace** interfacing between the system and the external users, making available the (already) developed 5G-ready application components, applications, L7VFs and virtual network functions.
- The **Multi-site resource management and orchestration mechanisms**, managing the complex work of network slices creation and management and the support of the set of Layer2-3 network functionalities.
- Lastly, the **Intelligent Application Orchestration Mechanisms** including deployment and runtime policies enforcement, data monitoring, fusion and analysis and a context awareness engine for inference of knowledge based on the collected information.

5.1 5G-ready Applications Design and Development Approach

MATILDA aims to provide a **next-generation design, development and operational environment** for **5G-ready applications**. To this end, MATILDA will deliver a stack that will support the entire **lifecycle** of these applications spanning from their **deployment** and **runtime reconfiguration** to **destruction**, but before delving into the technical details, it is necessary to discuss the overall architectural choices considered towards the design of the MATILDA architecture, as it is detailed in Section 6 of the document.

In the subsection, two sets of trends are detailed. First, trends in the design and development of cloud-native applications are followed towards the development of 5G-ready applications. Finally, the trends in the design, development and usage of metamodels for the appropriate representation of the application components and their characteristics are explored. Such metamodels will ensure the coherence of the developed applications through the development environments.

5.1.1 Existing Technologies and Progress Beyond

Design and development of 5G-ready applications

5G-ready applications consist of several **cloud-native components** – i.e., components that have to collaborate in order to fulfil their operational scope. Collaboration implies that these components form a **logical graph based on their dependencies**. The term cloud-native refers to specific properties that these components should have to be ported to the cloud. However, there is no globally accepted definition of the term "cloud-native." On top of that, the emergence of programmable infrastructure added additional parameters that should be taken under consideration during a "strict" definition of a cloud-native application. Programmable infrastructure allows the dynamic reconfiguration of

provisioned resources (VCPUs, memory, storage, bandwidth, security groups, etc.), which are capabilities that are rarely taken under consideration during the development of cloud-native apps.

In the frame of MATILDA, a cloud-native software component has the following characteristics:

A. it exposes its **configuration parameters** along with their metadata (e.g., which are the acceptable values? can these parameters change during the execution?);
B. it exposes **chainable interfaces** that will be used by other cloud-native components in order to create a service graph;
C. it exposes **required interfaces** that will be also used during the creation of a service graph;
D. it exposes **quantitative metrics** regarding the QoS level required by the cloud-native component;
E. it encapsulates a **lifecycle-management programmability** layer to be used during the placement of one service graph in the infrastructural resources;
F. it is **stateless** in order to be **horizontally scalable by design;**
G. it is **reactive to runtime modification of offered resources** in order to be **vertically scalable by design;**
H. it is **agnostic to physical storage, network and general-purpose resources.**

Regarding **(A)**, it could be argued that, if a cloud-native component entails a specific configuration layer, it is extremely crucial to be reconfigurable-by-design (i.e., to adapt to the new configuration without interrupting its main thread of execution). Imagine a scenario of a video transcoder, exposed as a cloud-native component. If during a transcoding session the resolution of the video receiver is downgraded, the transcoding service should (upon request) adapt to the new "mode" without the termination of the session. Regarding **(B)** and **(C),** it is clear that dynamic coupling of services is highly valuable only when an actual binding can be fully automated during runtime. This level of automation raises severe prerequisites for the developed cloud-native components. The "profile" of the chaining should be clearly abstracted. Such profile includes the offered/required datatype, the ability to accept more than one chaining, etc. These metadata are often stored in highly efficient key-value stores (such as Consul [Consul]) in order to be queried by requesting cloud-native components. In the aforementioned transcoding example, such dynamic cloud-native component lookup could be performed by the transcoding cloud-native component in order to identify an object storage cloud-native component that has some constraints (e.g., a security constraint – to support seamless symmetric encryption, or a location constraint, e.g., the location of the storage service to be near the location of the transcoding service so as to minimize the storage delay).

Regarding **(D),** it should be noted that, while cloud-native component-agnostic metrics are easily measured, the quantification of business-logic-specific metrics cannot be performed if a developer does not implement specific application-level probes. Indicatively, following the transcoding example, the cloud-native component-agnostic metrics such as VCPU, memory utilization and network throughput can be extracted through the instrumentation of the underlying IaaS by making use of its programmability layer. However, metrics such as the sessions that are actively handled, the average delay per each transcoding session, etc. cannot be quantified if the cloud-native component developer does not provide a thread-safe interface which reports the measurement of these metrics.

Regarding **(E)**, the recent developments in the virtualization compendium provided novel management capabilities. For instance, the live migration from one hypervisor to another one has been now integrated as a core built-in feature of KVM [KVM] for more than a year. Hence, a cloud-native component that is running on a specific infrastructure may be literally "transported" to another one without any down-time. Yet the cloud-native component dependencies may be affected by this choice. Imagine the storage service that the transcoding application relies on, to be seamlessly migrated from Ireland to the USA within the same IaaS. This could violate some chaining constraint

(e.g. delay, legal aspects). As a result, both cloud-native components should expose a basic programmability layer, which handles the high-level cloud-native component lifecycle (e.g. remove chained dependency).

Regarding **(F)**, any service that is stateless can scale easily with the usage of some "facilitating" services such as network balancers or web balancers. Historically, these services were statically configured by systems/network administrators or by DevOps engineers. The emergence of the infrastructure programmability model will progressively "offload" this task to Virtualized Functions (VFs) that are controlled by a cloud orchestrator. Ensuring the stateless behaviour of a service graph is a challenging task, since the entire business logic should entail stateless behaviour in order to be horizontally scalable by design.

Regarding **(G)**, taking under consideration the developments in hypervisor technologies and OS kernels in the last two years, it could be argued that the barriers of dynamic provision and de-provision of resources on an operating system have been raised. However, the dynamic provisioning of resources to a virtualized system does not imply that these resources are automatically bound to the hosted cloud-native component. On the contrary, in most of the times the cloud-native component has to be (gracefully) restarted in order to make use of the new resources.

Finally, regarding **(H)**, it should be noted that not every valid cloud-native component is capable to be ported to a modern infrastructure. For example, imagine a Java developer that uses a file-system for persistency in the frame of the development of one cloud-native component. This is considered extremely anti-pattern since the cloud-native component cannot be hosted in any Platform-as-a-Service Provider.

The working group [12Factor] provides an in-depth analysis of these characteristics.

The obvious question that is posed now is "**are 5G-ready applications just a specialization of cloud-native applications? And if so, why are existing orchestrators not considered appropriate for managing 5G-ready applications?**".

The answer is **definitely no**. The 5G ecosystem introduces several challenges that are not addressed by cloud-native applications and their orchestration means. Indicatively, **zero-delay tolerance** and **excessive mobility** are indicative characteristics of 5G applications that do not fall under the definition of cloud-native applications. In fact, both of these characteristics refer to the **network layer** of the cloud-native applications.

So, the next logical question is "**is there a framework/pattern that tries to abstract the network requirements of cloud native components?**". The answer to this question is **yes**. Such efforts have been initiated during the last three years, and have resulted in the "**Service Mesh**" concept.

Setting the scene of 5G-enabled apps it is meaningful to clarify the concept of Service Mesh. According to Phil Calçado, a "*service mesh is a **dedicated infrastructure layer** for handling service-to-service communication. It is responsible for the reliable delivery of requests through the complex topology of services that comprise a modern, cloud-native application. In practice, the service mesh is typically implemented as an array of lightweight network proxies that are deployed alongside application code, without the application needing to be aware*" [Calçado-2017]. The need for such an infrastructure layer is not obvious at first sight. As Phil Calçado notices, the first generation of distributed apps suffered from some "**fallacies**" based on the assumptions that they were making for the underlying network. These fallacies include the assumption that a) the **network is reliable**, b) **latency is zero**, c) **bandwidth is infinite**, d) the **network is secure**, e) **topology does not change**, f) **there is one administrator**, g) **transport cost is zero** and h) the **network is homogeneous**. Relaxing these assumptions generates many hard requirements that have to be addressed.

Moreover, the **emergence of cloud computing/programmable infrastructure-as-a-code** paradigm, along with the dominance of microservice-driven architecture, generated additional requirements.

These include a) **rapid provisioning** of compute resources, b) **basic monitoring**, c) **rapid deployment**, d) easy to **provision** storage, e) easy **access** to the edge, f) **authentication/authorisation**, **g)** standardised interfaces (e.g., RPC, HTTP).

The requirements of 5G-ready applications coincide with the aforementioned requirements. Yet, they are **much more intensive**, since provisioning of infrastructure should be "instantaneous", topology is continuously changing, delay tolerance is minimum, etc. The concept of this dedicated infrastructure layer is depicted in Figure 5.1.1.

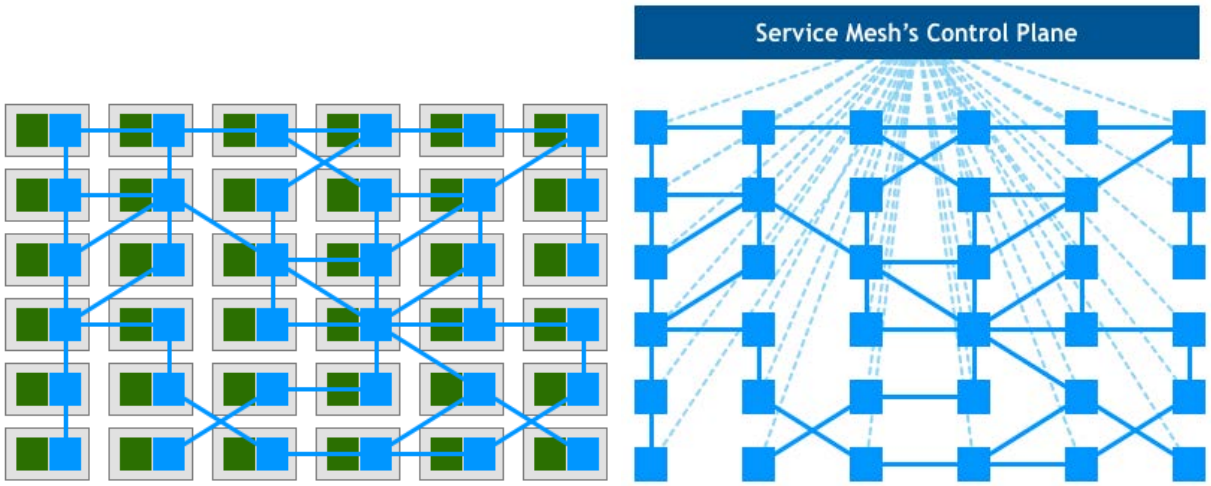


Figure 5.1.1: The concept of Service Mesh for cloud-native applications.

In conclusion, in the scope of MATILDA, a "5G-enabled application is a distributed application consisting of cloud-native components that rely on a service mesh infrastructure as a means of network abstraction. The service mesh per se has to operate on top of a programmable 5G environment". Towards these lines, the MATILDA architecture relies on a solid interplay between various logical layers such as the actual data plane, the service mesh control plane, and the configured virtualized resources that are offered by the telco provider as a proper slice.

Integrated development environment

MATILDA is going to provide a development toolkit targeted at the development of 5G-ready applications. The development toolkit is going to incorporate the integrated development environment (IDE) and a set of application graph composers and other relevant tools (such as the policy editor), aiming at supporting the design and development of chainable application components, VNFs, L7VFs and application graphs. The IDE is going to support the development of chainable application components, L7VFs and virtual network functions, respecting the metamodels that are going to be defined, namely (i) the Metamodel of Chainable Application Components, (ii) the 5G-ready Application Graphs and (iii) the VNF/PNF Metamodel. The graph composers are going to support the composition of application graphs taking into account the interfaces defined per application component and VNF/PNF and supporting the creation of appropriate bindings among applications components. Validation of the developed software and graphs with regards to adherence to the MATILDA metamodels and design principles is going to be realised.

The MATILDA IDE is going to rely on a web-based IDE facilitating collaboration and easy access to software developers. The most prominent tool that is considered for usage in MATILDA is Eclipse Che [EclipseChe], which is an IDE and developer workspace server that allows anyone to contribute to a project without having to install software. Given that Eclipse Che is a general-purpose development environment, a MATILDA plug-in has to be developed for checking the validity of the developed application components and VNFs/PNFs.

Concerning the graph composers, there are several existing open-source solutions for composition and visual representation of services that we are considering for use for the implementation of the MATILDA graph composers. The first is the SONATA service and function descriptor editor, which is an outcome of the 5G-PPP SONATA project [SONATA]. It is a JavaScript, HTML and CSS web application that has been released under the Apache 2.0 licence and provides a drag-and-drop interface for service composition. Another available application for consideration is a dashboard that has been developed in the frame of the VITAL EU-funded project [VITAL]. The dashboard has been implemented using SVG, JavaScript (D3.js library), CSS and HTML, and allows service composition through a drag-and-drop GUI. An additional option is the ARCADIA project GUI [ARCADIA] that supports the composition of application graphs consisting of chainable application components. A more advanced solution that could be used and modified for the development of the MATILDA graph composers is Juju-GUI [Juju], a web-based GUI for Juju that has been implemented using HTML and JavaScript.

Policy editor

At the higher level of the network-aware application graph, policies will be used to describe and define the boundaries and operational states of applications. Those will be based on rules, which will be designed with the support of a **Policy Editor** (e.g. based on the ARCADIA Policy Editor [ARCADIA-D.3.1]), based on the concepts to be included in the MATILDA metamodels (coming out of T1.3, T1.4 and T1.5) that will form the Policy Metamodel of MATILDA, and the context of each application.

Each policy is going to consist of a set of rules that are linked to attributes of the aforementioned models, and each rule consists of a "passive" part that includes expressions, denoting the conditions to be met and an "active" part, which denotes the actions to be executed upon the fulfilment of the conditions. Expressions may regard metrics that come out of different levels of the overall MATILDA stack, such as metrics relevant to the service graph, to the different chainable application components, or network resources usage metrics. The expressions will be designed using a user-friendly query-by-example paradigm i.e., the left part of a rule may comprise of several conditions or group of conditions. Complex expressions can be built in order to define the triggering condition of one rule. The right part of a rule can be a list of actions that will be performed in case a condition is met.

Policies are transformed into rules (using an engine such as Drools [Drools]), which are provided to the Optimisation & Context Awareness Engine for performing policy enforcement based on the current set of data and the active rules. By active rules we refer to the rules associated with the deployed service graphs at each point of time. Multiple policies could be assigned to a specific service graph; however, the application developer should explicitly state which single policy should be active.

As such, policies can be defined at two different levels:

- **Deployment Policies**, that deal with the deployment stage of a service graph and will be handled by the Deployment Manager.
- **Runtime Policies**, which will be handled by the Policy Manager. When attaching a specific runtime policy to a grounded service graph, the specified set of policy rules are deployed to the engine's production memory, while the working memory agent is constantly feeding the working memory with new facts.

Component & Application Graph Metamodel Management

An important functionality that has to be considered within the MATILDA development toolkit regards the definition of a set of network-oriented requirements on behalf of the software developer and their support during runtime through the exploitation of a service mesh architecture. Such requirements may regard Quality of Service (QoS) aspects, scalability aspects or profiling aspects. Each chainable application component or VNF/PNF may expose quantitative metrics regarding its QoS. These metrics are not related with the execution-ware metrics (i.e. CPU utilization, memory consumption, etc.), since the latter refer only to the quantification of the component's business logic metrics. Each chainable

application component or VNF/PNF may also be stateless, in order to be horizontally scalable by design. Each chainable application component should be reactive to runtime modification of reserved resources in order to be vertically scalable by design. Each chainable application component or VNF/PNF may also handle the served load by consuming provided resources based on a profile.

The scope of this sub-section is to provide a generic view of some of the current trends regarding each component/service and the entire applications/services graph modelling schemas that are going to be considered towards the definition of the MATILDA metamodels.

- ***NodeRED***

  It is a project of the JS Foundation. NodeRED [nodeRED] is a tool for wiring together hardware devices, APIs and online services. It tackles the problem of multi-connections in IoT. It provides a browser-based editor that makes it easy to connect together flows using the wide range of nodes in the palette that can be deployed to its runtime in a single-click. The flows created in Node-RED are stored using JSON, which can be easily shared with others. A built-in library allows one to save useful nodes and flows for reuse and an online flow library allows to share one's work with others. Focusing mostly on modelling aspects of NodeRED, it follows a very simple and light node and flow model schema. Especially, the service-model is addressed as "node" and it is described by two sets of properties:

  - **core-properties**: used by the runtime/editor for the basic node functionality;
  - **type-properties:** created by node-developer, where each node type can contain properties that are specific to it. In addition, a custom node could declare its own property to capture that information, and its runtime implementation knows what to do with it.

  To sum up, the application-graph model is addressed as "flow", and is represented by a Javascript array of objects. Each object is a node, with a set of core properties, and a set of type-specific properties. Furthermore, NodeRED checks the proper deployment of the flow, but it does not actually deal with nodes' orchestration, deployment, scaling, and management. Finally, it does not allow for the specification of (network) parameters regarding links between different components.

- ***Juju***

  Juju [Juju], developed by Canonical, is a framework that can be used to model, manage and scale services in the cloud. It contains some interaction tools such as a command line and a graphical user interface and is a solid solution that can reduce the workload for deployment and configuration. Thus, through these tools a DevOps user can easily embed a service or a web of services on top of multiple IaaS providers (e.g. OpenStack).
  The service metamodel of Juju is addressed as "charm" and contains a set of elements that are required in order, for a specific service to be composable and orchestratable. The service graph metamodel is addressed as "bundle" and it is a web of charms. Anybody can deploy a predefined charm or a bundle and use them. Both of them are described by some YAML files, and someone can moderate them with some commands called "hooks". In JUJU documentation, a strict list of commands per charm is provided, so that anybody can use them to configure it. However, the Juju platform was not built to address more specific network quality of service requirements and constraints.

- ***ARCADIA context model***

  This model is a product of the ARCADIA [ARCADIA-D.2.2] EU-funded research program. It deals with the modelling of services regarding highly distributed applications. With a focus on component and service-graph models, the ARCADIA Component Model represents the most granular executable unit of an ARCADIA application. A set of interconnected components

produces a service graph. Highly Distributed Applications (HDAs) are practically instantiations of a complex service graph. Each component is described through several properties and all this information is encapsulated in a XML file that is generated by a specific XML Schema Definition (XSD).
As already explained, many ARCADIA Component Models can be combined in order to create one ARCADIA Service Graph Model, which is practically a directed graph. Finally, the service graph model encapsulates the description of each component, as well as a description for each virtual link, accompanied with information regarding monitoring metrics that refer to the whole service graph.

- ***DOCKER Compose***

  Developed by Docker, Docker Compose [Docker Compose] is a tool for defining and running complex, multi-container applications with Docker. With Compose, a multi-container application can be defined in a file and the application is deployed and executed. A Dockerfile describes units (each docker-container is considered a granular unit), in which the possible user (i.e., Devops) could define what the container needs.

- ***PUPPET***

  It is categorized in the middleware level and aims to model, install and deploy infrastructure's applications and services [Puppet]. It also tracks down the dependencies between them. Puppet uses a Domain Specific Language (DSL) for the modelling. This specific language adds more complexity in Puppet, but on the other hand it becomes more consistent and offers a deeper layer of valid configurability on the spot.

- ***TOSCA***

  It is a standard description language regarding orchestration procedures by OASIS [TOSCA]. OpenStack HEAT, Cloudify and Ubicity are orchestrating tools that have based their models on it. Recently, a version of TOSCA about NFV was presented by OASIS. TOSCA is written in XML, but there is also a TOSCA's YAML simple profile. The main concept is structured on two types of entities: nodes and relationships. A node could be an infrastructure component, like a network, a server, a cluster of servers, or it can be a software component, like a service or a runtime environment. A relationship describes how nodes are interconnected.

- ***YANG***

  YANG [YANG] is a data modelling language that focuses on network services and network devices' configuration. It manipulates NETCONF protocol data. It is developed by the IETF NETCONF Data Modeling Language Working Group (NETMOD) and it is defined in RFC 7950. YANG is a modular language that represents the information in a XML tree format.

The proposed approach in MATILDA aims at the overall management of the component and application graph metamodels, both in terms of the information that will be captured in the metamodels (and exploited by other components) and in terms of the mechanisms that will support the metamodels regarding several aspects (e.g., chainability of components, QoS analysis, scalability, etc.). Based on the analysis of the state of the art, the ARCADIA Context Model will be exploited and extended with novel and ground-breaking approaches that deal with the design of a non-monolithic model norm, regarding highly distributed applications and addressing their complete lifecycle. The ARCADIA component and service graph metamodels will be extended to address the requirements and challenges of 5G environments. Regarding the VNF metamodel, existing specifications in ETSI NFV WG are going to constitute the basis for the specification of the relevant model in MATILDA.

5.1.2 Technology Requirements

ID	Dev_Tool_1
Unique Name/Title	Applications denoted in the form of a Service Graph
Priority	High
Type	Functional
Brief Description	MATILDA applications must be easily deployable over a 5G ecosystem, consisting mainly of programmable infrastructure. Each application has to be developed in such a way that it can be broken down into a set of interconnected components/services (based on the definition of dependencies). This requires a modular architecture, where software components can be easily assembled and linked together.
Rationale	Technology review and Use cases.
Validation method/Relevant KPI	Validation based on the 5G-Ready Application Graph Policy Metamodel / Number of developed 5G-Ready application graphs.

ID	Dev_Tool_2
Unique Name/Title	Adherence to the MATILDA Metamodels
Priority	High
Type	Functional
Brief Description	Development of chainable application components, VNFs, application graphs and L7VFs have to be compatible with the set of applicable MATILDA metamodels. Validation functionalities have to be provided by the IDE and the graph composer tools. In this way, the developed software is going to be conformant with the characteristics of cloud-native applications.
Rationale	Technology review and Use cases.
Validation method/Relevant KPI	Validation through the IDE based on the metamodels' definition.

ID	Dev_Tool_3
Unique Name/Title	Web based and Collaborative development environment
Priority	Medium
Type	Functional
Brief Description	To facilitate the development of application graphs, consisting of set of application components, web-based tools facilitating the usage and the straightforward integration of available software should be supported. This applies to both the IDE and the graph composers.
Rationale	Technology review and Use cases.
Validation method/Relevant KPI	Release of the MATILDA web-based IDE.

ID	Dev_Tool_4
Unique Name/Title	Repositories for Sharing of Developed Software
Priority	High
Type	Functional
Brief Description	A software repository must be available to share code among developers and projects. The repository is used both to access available

	components/VNFs/L7VFs and application graphs, as well as to upload new ones.
Rationale	Technology review and Use cases.
Validation method/Relevant KPI	Release of MATILDA Marketplace.

ID	Dev_Tool_5
Unique Name/Title	Application Components and VNFs Configurability
Priority	High
Type	Functional
Brief Description	Each chainable component/VNF should expose its configuration parameters along with their metadata. This exposure is crucial, since during runtime policy enforcement the MATILDA Orchestrator may alter the configuration of a chainable component/VNF to satisfy a specific policy.
Rationale	Technology review and Use cases.
Validation method/Relevant KPI	Specification of MATILDA metamodels and development of relevant descriptors.

ID	Dev_Tool_6
Unique Name/Title	Application Components' and VNFs' Chainability
Priority	High
Type	Functional
Brief Description	Each chainable application component/VNF should expose chainable interfaces, which will be used by other chainable components/VNFs to create a 5G-ready application graph. Furthermore, it should expose required interfaces, which will be also used during the graph creation.
Rationale	Technology review and Use cases.
Validation method/Relevant KPI	Specification of MATILDA metamodels and development of relevant descriptors.

ID	Dev_Tool_7
Unique Name/Title	Application Components and VNFs' QoS Awareness
Priority	High
Type	Functional
Brief Description	Each chainable application component/VNF should expose quantitative metrics regarding its QoS. These metrics are not related to the execution-ware metrics (i.e., CPU utilization, memory consumption, etc.) since they refer only to the QoS aspects.
Rationale	Technology review and Use cases.
Validation method/Relevant KPI	Specification of MATILDA metamodels and development of relevant descriptors.

ID	Dev_Tool_8
Unique Name/Title	Application Components and VNFs Scalability
Priority	High
Type	Functional

Brief Description	Each chainable application component/VNF should declare whether it is stateless in order to be considered as horizontally scalable by design or not. It should also be reactive to runtime modification of reserved resources in order to be vertically scalable by design.
Rationale	Technology review and Use cases.
Validation method/Relevant KPI	Specification of MATILDA metamodels and development of relevant descriptors.

ID	Dev_Tool_9
Unique Name/Title	Infrastructure Agnostic Software Development
Priority	High
Type	Functional
Brief Description	Each chainable application component should be agnostic to physical storage, network and general-purpose resources. This will make the components de-facto portable to virtualized resources. Networking requirements are going to be declared leading to the configuration of the appropriate network slice.
Rationale	Technology review and Use cases.
Validation method/Relevant KPI	Specification of MATILDA metamodels and development of relevant descriptors. Appropriate description of slice intent for preparation of the network slice.

ID	Dev_Tool_10
Unique Name/Title	Application Components and VNFs Performance Profile
Priority	Medium
Type	Functional
Brief Description	Each chainable application component/VNF should handle the served load by consuming provided resources based on a profile. Profiling aspects are going to accompany the application components/VNF description through metadata.
Rationale	Technology review and Use cases.
Validation method/Relevant KPI	Support of profiling mechanisms in the data mining and analysis components. Profiling information available in the descriptors.

ID	Dev_Tool_11
Unique Name/Title	5G-ready Applications Composition through the Graph Composer
Priority	High
Type	Functional
Brief Description	The graph composer should provide the means to visually represent and graphically compose new 5G-ready applications.
Rationale	Technology review and Use cases.
Validation method/Relevant KPI	Appropriate composition and validation of 5G-ready applications through the graph composer.

ID	Dev_Tool_12
Unique Name/Title	Formal Language Expressing Networking Requirements
Priority	High

Type	Functional
Brief Description	A formal language should be defined to describe networking requirements.
Rationale	Technology review and Use cases.
Validation method/Relevant KPI	Specification of MATILDA metamodels and development of relevant descriptors.

ID	Policies_Editor_1
Unique Name/Title	Policies Assigned to 5G-ready Application Graphs
Priority	High
Type	Functional
Brief Description	For each application graph, one or multiple policies need to be designed, while a specific policy should be selected for enforcement during runtime. Each policy consists of a set of rules that are linked to attributes of the aforementioned models, and each rule consists of a "passive" part that includes expressions, denoting the conditions to be met and an "active" part, which denotes the actions to be executed upon the fulfilment of the conditions. It should be noted that based on a service mesh approach, policies rules are going to trigger the activation of L7VFs.
Rationale	Technology review and Use cases.
Validation method/Relevant KPI	Specification of MATILDA policy metamodel and development of relevant descriptor. Implementation of the Policy Editor.

ID	Policies_Editor_2
Unique Name/Title	Assigning Predefined Policies to 5G-ready Application Graphs
Priority	High
Type	Functional
Brief Description	To each application graph, one or multiple policies can be selected and assigned, out of the set of defined policies per application graph.
Rationale	Technology review and Use cases.
Validation method/Relevant KPI	Support of functionality through the Policies Editor.

ID	Policies_Editor_3
Unique Name/Title	Policy Levels for 5G-ready Application Graphs
Priority	High
Type	Functional
Brief Description	For each service graph, two policy categories may be defined, one dealing with the deployment environment, and one that deals with run-time execution conditions. These will allow applications to be deployed over the MATILDA infrastructure in the most efficient manner based on the requirements set, while during run-time they will safeguard the proper execution of the application, by enforcing operational conditions that have to be respected.
Rationale	Technology review and Use cases.
Validation method/Relevant KPI	Support of two different policy levels in the policies editor.

ID	Policies_Editor_4
Unique Name/Title	Pushing Application Graph Deployment Policies to the Deployment Manager
Priority	High
Type	Functional
Brief Description	Each deployment policy that is set as active in an application graph should be communicated to the Deployment Manager in a compatible format, in order for the latter to be able to enforce policies during runtime.
Rationale	Technology review and Use cases.
Validation method/Relevant KPI	• Interface to the Deployment Manager • One active policy for each application graph deployed/running

ID	Policies_Editor_5
Unique Name/Title	Pushing Service Graph Runtime Policies to the Policy Manager
Priority	High
Type	Functional
Brief Description	Each policy that is set as active in a service graph should be communicated to the Policy Manager in a compatible format, in order for the latter to be able to enforce policies during runtime.
Rationale	Technology review and Use cases.
Validation method/Relevant KPI	• Interface to the Policy Manager • 1 active policy for each service graph deployed/running

5.2 Marketplace

The MATILDA marketplace is the interface between the system and the MATILDA users, making available the (already) developed 5G-ready application components, applications, virtual network functions and L7VFs for open-source or commercial purposes, reuse and extension. That is, a graphical user interface (GUI) connecting a set of repositories and mechanisms for supporting the diverse 5G stakeholders. Through this GUI and depending on the stakeholders involved, the marketplace provides the means to accomplish several trading and commercial procedures related to the offered virtual network functions (VNFs), application components and 5G-ready applications such as procurement, cataloguing, offering and definition.

5.2.1 Existing Technologies and Progress Beyond

Research/innovation projects

- ***T-NOVA (7FP) marketplace***

  T-NOVA marketplace [T-NOVA] considers three different stakeholders: several VNF developers, a Service Provider and Customers, which are represented in Figure 5.2.1. Within the marketplace, two different catalogues are implemented: a NVF catalogue called NF Store (NSF) and a NS catalogue called Business Service Catalogue (BSC). These catalogues are independent repositories of VNFs (VNF descriptor + set of images) and Network Services (NS descriptor).

  The diverse stakeholders interact with the marketplace and the catalogues in different ways depending on the role they play in the system: VNF developers define and upload VNFs onto the NF store by means of a wizard that guides the user in the VNF definition process, creating a

VNF descriptor (VNFD, compliant with ETSI VNFD definition [ETSINFV-2014c]) as a result. A Service Provider creates Network Services using a similar graphical wizard to help in the generation of the NS descriptor (NSD, compliant with ETSI NSD definition [ETSINFV-2014c]). These NSs are based on the VNFs that are available in the NF store, which are combined to create complex NSs and are finally stored in the BSC to be exposed to the customers after a validation process.

Every relationship between stakeholders involves a commercial transaction, as every item has associated an SLA contract and billing information.

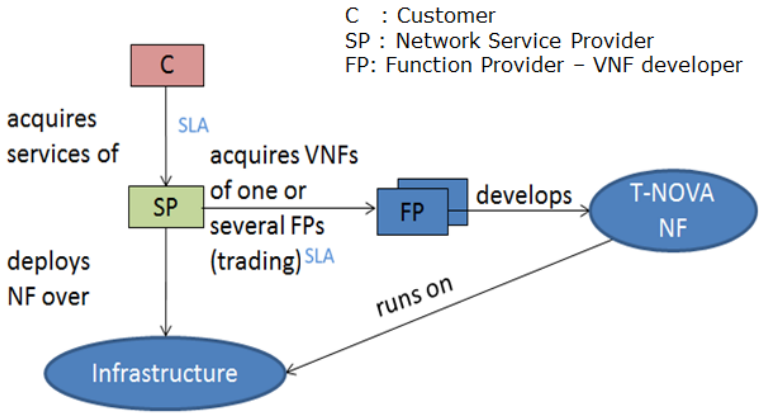


Figure 5.2.1: Relationships between the -NOVA stakeholders.

- ***5GEx (5G ph1)***

  The 5Gex project does not implement a marketplace itself, but a dashboard similar to the T-NOVA one, although the implementation of the catalogues in 5GEx is slightly more complex than in the T-NOVA project due to the multi-service provider/domain support.

  It contains also a local VNF repository and a local NS repository that store, respectively, the VNFs and NSs created by the local Service Provider, but both reunite in a common catalogue that references the repositories. This list exposes VNFs and NSs, allowing the local Service Provider to use VNFs and/or already existing NSs in the creation of (even more) complex Services. Additionally, the catalogue can communicate with peers in neighbour domains to exchange VNFs and NSs. The exchange of VNFs is done based on sharing policies: each administrative domain can configure which items are visible by the neighbour domains.

  The decision of adding an external Service to the local catalogue is taken by the local Service Provider and it needs to be adapted and validated before it becomes part of the local offer.

  Every catalogue item exposes relevant information that helps the potential user to choose between the offerings: domain owner, name, description, SLA and pricing information, etc. This information is used also to evaluate and ensure the quality of the services once they are running.

Standards and Opensource

- ***TMForum***

  There is an ongoing TMForum Catalyst project titled "Enabling the Digital Services Marketplace with Onboarding Automation" [TMF]. This Catalyst project identifies design patterns, from both business and technology aspects, with special attention to the VNF onboarding processes. The project aims to demonstrate a standards framework with model

driven approach to enable a dynamic marketplace that supports portability & interoperability of multi-vendor and cross-domain solutions, with the main objective of standardizing packaging and onboarding automation that enables a NFV marketplace. The Proof of Concept (PoC) planned consists of a well-enabled package within a smart city scenario with automated onboarding and lifecycle management to support the NFV marketplace.

- ***OSM***

  OSM [OSM] does not include a Marketplace, but introduces rift.io, which is the solution in northbounding with the orchestration layer as part of the OSM stack; it enables the mechanisms for VNF onboarding and service creation.

Commercial solutions

Among the commercial solutions, there are two of them, which support the customer interaction, by means of a GUI that allows also service design, although not including a proper marketplace. These are the solutions by AVDA [ADVA], and Virtuora [Virtuora].

MATILDA Marketplace

The MATILDA marketplace has a two-fold objective. On one hand, it acts as a centralised repository providing access to the set of developed application components, 5G-ready applications, L7VFs and VNFs/PNFs, either developed within the MATILDA development and verification environment or brought up by third parties. Open-source release of software and continuous update processes are going to be promoted in order to achieve a critical mass of popular 5G-ready applications, chainable application components, L7VFs and VNFs/PNFs that may be used by application developers and service providers/telecom operators.

On the other hand, the MATILDA marketplace will act as a mean to support service providers, network operators and software houses to commercialize new virtualized products and network-aware applications. Establishment of collaboration among specific stakeholders may be initiated through the marketplace, leading to customised software covering a service provider' needs. Several gaps which the MATILDA marketplace should cover have been identified in the marketplaces studied in the state of the art above:

- Support of new stakeholders that may require additional functionalities, e.g. vertical industries.
- Support of network-aware apps, L7VFs and vertical applications, which will imply new metamodels.
- Interfacing with verification, emulation and profiling environments. This includes interfaces with the MATILDA development and verification environment, the policy editor and the profiler. With regards to the verification, other solutions, such as the T-NOVA marketplace, provide only syntax validation of the produced descriptors. Previous solutions of emulation environments are included in some other research projects such as SONATA, but not as part of a marketplace.
- Interfacing with automatic service composer tools that allow service design and composition. This feature refers to the interface with the MATILDA 5G-ready Application Graph Composer.

A high-level diagram depicting the relationship between the marketplace components and the stakeholders that interact with it is shown in Figure 5.2.2:. It includes the repositories for the Applications and the Network Functions that will be integrated by the service provider into a 5G-ready application, which is also stored in a Marketplace repository. The developers access these repositories by means of a GUI, allowing them to easily define/upload the content, or directly through the development toolkits provided using the APIs exposed by the repositories.

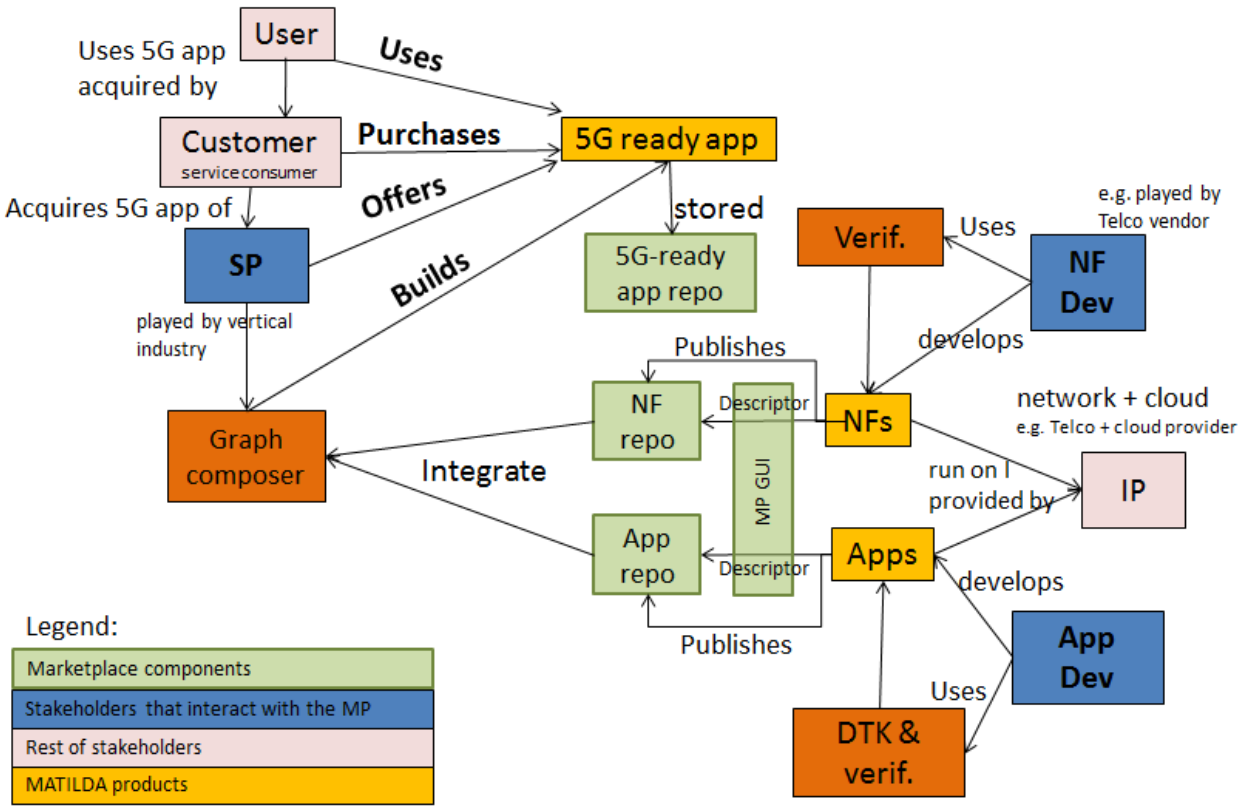


Figure 5.2.2: MATILDA Marketplace relationships.

5.2.2 Technology Requirements

ID	MP_1
Unique Name/Title	Graphical User Interface for Stakeholders
Priority	High
Type	Functional
Brief Description	The marketplace shall provide a graphical user interface for each one of the MATILDA stakeholders to perform their operations (each GUI shall include the functions allowed or required by each type of user/stakeholder) and shall be user-friendly.
Rationale	Technology review and Use cases.
Validation method/Relevant KPI	Successful delivery of a user-friendly GUI, which is different for different types of users/stakeholders.

ID	MP_2
Unique Name/Title	Service Trading from Third Party Developers
Priority	Medium
Type	Functional
Brief Description	The marketplace shall allow external developers to offer their developed services.
Rationale	Technology review and Use cases.
Validation method/Relevant	Capability of Third Party developers to perform service trading.

KPI	

ID	MP_3
Unique Name/Title	Support of Various, Different Profiles/Functions for Different Users/ Stakeholders/Roles
Priority	Medium
Type	Functional- Security
Brief Description	The marketplace shall support (the creation of) a number of different profiles for various different users/stakeholders/roles, each performing different functions (e.g., purchase, view, sell, etc.) via the Marketplace.
Rationale	Technology review and Use cases.
Validation method/Relevant KPI	Successful creation of a new profile. Testing of various profiles and their allowed operations through the Dashboard.

ID	MP_4
Unique Name/Title	Authentication, Authorization and Access Control
Priority	High
Type	Functional
Brief Description	All system operations must be subject to specific access restriction policies. The marketplace shall provide a "login" page for the different stakeholders to be authenticated. It shall support different rigidly defined levels of authorization and access rights depending on the individual users/stakeholders' profile, and their corresponding allowed functions on the Marketplace items and components. Furthermore, administrative/ maintenance/ configuration/ etc. procedures shall only be performed by appropriately authorized users, in order to preserve system integrity.
Rationale	Technology review and Use cases.
Validation method/Relevant KPI	Testing of user authentication and the various authorization levels for all MATILDA users/procedures/operations. KPI: Implementation of various rigidly defined authorization levels for different users/roles.

ID	MP_5
Unique Name/Title	Web Access
Priority	High
Type	Functional
Brief Description	The marketplace shall be accessible to authorised users via the Internet, while web-encryption shall be incorporated. The GUI shall be interoperable with all existing/current web-browsers.
Rationale	Technology review and Use cases.
Validation method/Relevant KPI	Successful access of the MATILDA marketplace through different web-browsers, while incorporating web-encryption.
Architectural Component	Marketplace GUI.

ID	MP_6
Unique Name/Title	Parallel Access and Synchronisation

Priority	High
Type	Functional
Brief Description	The marketplace should provide multiple users login (and no less than 10), and operation (modification) on the same or on linked components/Graphs, etc. In this context, synchronisation of all components on repositories (or on a specific repository) can be performed upon submission of a change or upon user request.
Rationale	Technology review and Use cases.
Validation method/Relevant KPI	Verification of the smooth operation of multiple users over the same or on linked components/graphs, etc., through multiple instances of the Dashboard. Successful synchronisation of components/repositories upon change or upon user request. Verification - through observation of the repositories' items.

ID	MP_7
Unique Name/Title	Availability
Priority	High
Type	Non-Functional
Brief Description	The availability level of the market place shall reach 99.99% of operational time.
Rationale	Technology review and Use cases.
Validation method/Relevant KPI	Availability can be measured after the completion of the MATILDA development stage. Relevant KPIs are: (time the service is available) / (total time from service deployment up to the time of measurement).

ID	MP_8
Unique Name/Title	Interface GUI - Catalogues
Priority	High
Type	Functional
Brief Description	The marketplace should expose internal communication interfaces between the dashboard and the catalogues.
Rationale	Technology review and Use cases.
Validation method/Relevant KPI	Validation through the Marketplace design and development.

ID	MP_9
Unique Name/Title	Interface Marketplace – Orchestrator
Priority	High
Type	Functional
Brief Description	The marketplace should expose external communication interfaces with the orchestrator.
Rationale	Technology review and Use cases.
Validation method/Relevant KPI	Validation through the Marketplace design and development.

ID	MP_10
Unique Name/Title	Concurrency and User Isolation (for the Marketplace)

Priority	High
Type	Non-Functional
Brief Description	Different users must be able to access the marketplace at the same time from different locations. User isolation is required especially between service consumers.
Rationale	Technology review and Use cases.
Validation method/Relevant KPI	Verification of smooth performance of operations by multiple users, without the operations of one user affecting the operations of another.

ID	MP_11
Unique Name/Title	App Repository
Priority	High
Type	Functional
Brief Description	The marketplace shall include a repository for Apps. This repository shall include the chainable application components and the application graphs.
Rationale	Technology review and Use cases.
Validation method/Relevant KPI	Validation through the Marketplace design and development.

ID	MP_12
Unique Name/Title	VNF Repository
Priority	High
Type	Functional
Brief Description	The marketplace shall include a repository for VNFs/PNFs L2-L3-L4 functions.
Rationale	Technology review and Use cases.
Validation method/Relevant KPI	Validation through the Marketplace design and development.

ID	MP_14
Unique Name/Title	L7VFs
Priority	High
Type	Functional
Brief Description	The marketplace shall include a repository for L7VFs. This repository shall include the set of L7VFs that can support L7 functionalities in a service mesh architecture.
Rationale	Technology review and Use cases.
Validation method/Relevant KPI	Validation through the Marketplace design and development.

ID	MP_14
Unique Name/Title	Policies Description
Priority	High
Type	Functional
Brief Description	A policies' repository has to be made available for storing information

	regarding policies associated with application graphs.
Rationale	Technology review and Use cases.
Validation method/Relevant KPI	Validation through the Marketplace design and development.

ID	MP_15
Unique Name/Title	Repository Operations
Priority	High
Type	Functional
Brief Description	The repositories shall provide the means to perform operations on the stored items such as: insertion, retrieval, list and withdrawal depending on different user authorization levels.
Rationale	Technology review and Use cases.
Validation method/Relevant KPI	Successful operation of insertion, retrieval, list and withdrawal by users with different authorization levels.

ID	MP_16
Unique Name/Title	Concurrency and Synchronisation (of Operations on Marketplace Repositories)
Priority	High
Type	Functional
Brief Description	The repositories shall support simultaneous access by multiple users. User isolation is required especially among users' operations on non-linked items, while specific synchronisation rules shall be established to manage simultaneous operations by multiple users on the same item.
Rationale	Technology review and Use cases.
Validation method/Relevant KPI	Verification of smooth performance of operations by multiple users, without the operations of one user affecting the operations of another.

ID	MP_17
Unique Name/Title	Security/Integrity of Repositories' Data
Priority	High
Type	Non-Functional
Brief Description	Specific administrative/ maintenance/ configuration/ deletion/ etc. functions shall only be allowed to appropriately authorized users, while for each such function a second control/check level may be introduced in order to preserve system integrity. Specific functions shall be reversible (e.g., a * deletion shall be reversible by automatically saving an instance of the repository before performing it, etc.). Moreover, repositories' data integrity/security shall be ensured through storage space encryption mechanisms.
Rationale	Technology review and Use cases.
Validation method/Relevant KPI	Validation through design and development using storage space encryption mechanisms.
Architectural Component	Repositories.

5.3 Multi-site Resource Management and Orchestration Mechanisms

5.3.1 Existing Technologies and Progress Beyond

Modern cloud technologies and architectures are largely recognized as the foundations of the upcoming 5G ecosystem [Szabo-2015, Vilalta-2017]. These technologies are expected to not only provide the needed means for allowing the softwarisation revolution in telecommunication infrastructures (mainly through the NFV framework), but also act as key enablers for new (more pervasive and more network-integrated) computing paradigms, like, for instance, fog and mobile edge computing [Soldani-2015, Shanhe-2015, ETSIMEC-2016a, Fernando-2013].

One of the most significant added-value aspects, which contribute to this ever-increasing success, can be found on the same cloud architecture, since it allows a clear and effective splitting of roles among the main actors involved. This splitting along with the disruptive "*as-a-Service*" paradigm and the agility of modern computing/networking virtualization technologies led to make the cloud a mass-market and irremissible means for any vertical market.

Anatomy of today's Cloud Computing

The anatomy of today's cloud ecosystems can be organized along three main layers [Mell-2011], namely *Infrastructure-as-a-Service*, *Platform-as-a-Service*, and *Software-as-a-Service*, which clearly define the type and the boundaries of offered service per each actor.

Actors offering IaaS services are providing their computing and/or networking infrastructures to third-party platform or software providers, usually referred to as "*tenants*", through *Virtual Infrastructure Managers* (VIM), also referred to as *Cloud Management Software* (CMS) [Sotomayor-2009, Manvi-2014].

Through such VIM interfaces, tenants are allowed to monitor and to manage the entire lifecycle of their applications and services. In detail, the VIM manages the computing, storage, and network physical infrastructure in a datacentre, and it serves as a sort of conduit for control-path interaction between multiple virtualized (isolated) infrastructures, each one associated to a tenant, and the physical level. Broadly speaking, the VIM provides tenants with inventories, provisions and de-provisions operations, and the management of virtual compute, storage and networking, while also communicating with the underlying physical resources (e.g., hypervisors, network switches, etc.). The VIM is also responsible for operational aspects such as logs, metrics, alerts, etc.

Among open-source VIM implementations, the most relevant are Eucalyptus [Eucalyptus], OpenNebula [OpenNebula], CloudStack [ApacheCloudStack], and the well-known OpenStack [OpenStack] [Vogel-2016, Shahzadi-2017].

Given the rising complexity of such services and the challenging performance requirements associated with them, a large part of these operations is often delegated and automated by a "Service Orchestrator" [Weerasiri-2012], which constitutes along with VIMs the backbone of any advanced and modern cloud system. Furthermore, it can be noted that also the definition of the ETSI NFV Working Group is perfectly compliant with the Orchestration/VIM layering infrastructure. Any actors (PaaS or SaaS providers) playing on top of the VIM layer shall have their own Orchestrator and, in case of PaaS providers, multiple Orchestrator modules might act in cascade.

Service Orchestrators generally include different mechanisms and sophisticated algorithms to cope with the following main functionalities:

- Automatically instantiating services and their components (in terms of execution environments – e.g., virtual machines).

- Monitoring the service and any of its components.
- Providing automatic service upgrade procedures.
- Acquiring or releasing computing/network/storage resources from VIM(s).
- Scaling the service to meet service level agreements and incoming workloads.

Relevant examples of well-known service orchestration platforms and related tools include Cloudify [Cloudify], Canonical Juju [Juju], OpenStack Heat [Heat], etc.

Although the NFV framework can be considered as an application of standard cloud computing technologies, as demonstrated in detail by its same designers [ETSINFV-2015], it is a common opinion that the rising of 5G technologies will significantly affect the cloud evolution. In this respect, Fog and Mobile Edge Computing are two clear preliminary signs of this trend [ETSIMEC, Taleb-2017, Fernando-2013].

The main aspects expected to produce this sort of back-pressure can be found in the challenging performance and operational requirements of the telecom sector, as well as in the ones to support new classes of vertical applications with mass-scale and/or real-time needs (see the next section) [Taleb-2017, Mijumbi-2016a, Mijumbi-2016b]. These requirements are leading to two key technological trends:

- A shift towards more distributed and heterogeneous infrastructure environments (e.g., spanning from centralized datacentres to small/medium ones and cloudlets).
- A much deeper, more flexible and autonomic integration among multiple applications and telecom software services.

Moreover, since the programmable resources will be integral part of the same 5G softwarised infrastructure, datacentres supporting 5G functions and vertical applications are supposed to be owned and maintained by telecom infrastructure providers, and to offer "private" (and in some case "hybrid") services [ETSINFV-2015]. Thus, public cloud platforms (like the aforementioned AWS, GoGrid, Microsoft Azure, Google Compute Engine, IBM Smart Cloud, etc.) are expected to have a limited relevance to the 5G/NFV ecosystem, while, during the latest few years, the NFV community selected OpenStack as the reference VIM for 5G/NFV environments (see Figure 5.3.1).

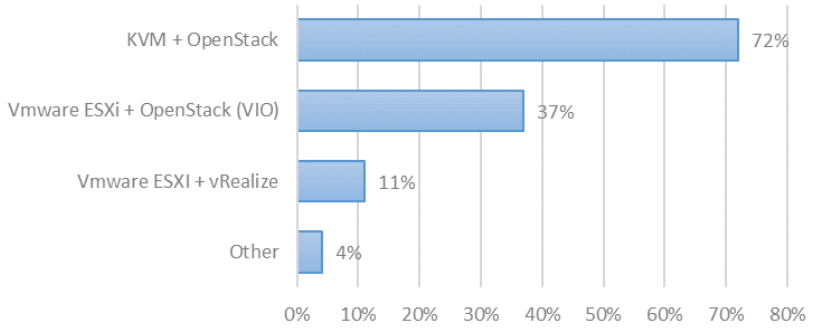


Figure 5.3.1: Current NFVI and VIM deployments. Survey (with multiple answers) conducted by SDxCentral. Source: [SDxCentral-2017a].

Starting from this scenario, a number of projects and platforms, often competing among themselves in a still complex and very fragmented landscape, have been arising during the latest years both to cope with NFV specifications and to enable the aforementioned opportunities [SDxCentral-2017a, SDxCentral-2017b]. However, many of these projects can be seen as added-value extensions/evolutions of state of the art cloud tools and means. In some cases (e.g., Cloudify), some cloud-native projects also released specific telecom editions to meet the challenging requirements of the 5G/NFV framework.

As noted in a Linux foundation whitepaper [LF-2017], the aforementioned complex landscape in telecommunications and networking has created the need for an umbrella architecture that harmonizes the multitude of standards and open source projects. Harmonization encompasses a number of aspects that affect both standardization and open source: *i*) Ease of integration through well-defined information models, APIs, and interfaces; *ii*) common development environment to ease the integration and testing of components in a highly automated manner; *iii*) close coordination among the activities, to align based on use cases, functional requirements, schedules, etc.

The Linux Foundation, maintaining numerous open-source projects relevant to 5G, has forged a unified Open Architecture for Networking and Orchestration OS-N&O, to position and harmonize the many OS-N&O projects and standards (see Figure 5.3.2 and Table 5.3.1). In detail, the unified OS-N&O relies on three layers, which correspond to the reference architectures defining high-level functionality. As introduced in more detail in the remainder of this sections, these layers (highly compliant to the MATILDA vision) roughly corresponds to the VIM, the NFV service and the vertical application orchestration, respectively.

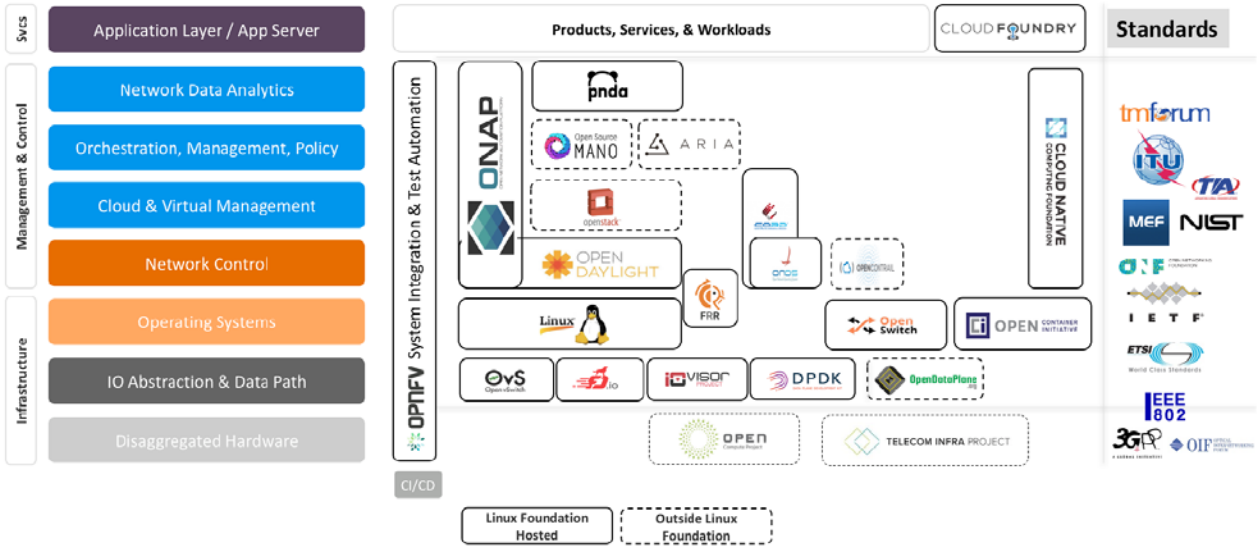


Figure 5.3.2: Linux Foundation perspective on the architectural landscape of Unified Open Networking & Orchestration. Source: [LF-2017].

Table 5.3.1: Unified Open Networking & Orchestration architecture description. Source: [LF-2017].

Layer	Description	Standard(s)	Open Source Project(s)
Orchestration & Service	Enable end-to-end composite services	MEF LSO TMForum Zoom ITU-T	ONAP (Open Orchestrator) PNDA (Network Analytics Platform) ARIA (TOSCA enablement)
Control & Management	Provide network control and management (NFV, SDN, and legacy networks)	NFV MANO IETF Routing IETF (many) ONF (OpenFlow)	OpenDaylight (SDN Controller); OpenSwitch (Whitebox NOS) JuJu (NFV G-VNFM) OpenStack (NFV VIM)
Infrastructure	Provide Network Data Plane and NFV Infrastructure	ETSI NFV-I IEEE 802 3GPP OIF	OpenvSwitch (virtual switch) FD.io (data plane acceleration); DPDK (fast packet processing) KVM (Hypervisor)

The following two sections summarize the main ongoing projects and platforms in the VIM and orchestration areas, respectively.

Moving Computing Towards the Network Edge

Multi-access Mobile Edge Computing (MEC) provides cloud computing hosting capabilities at the edge of the mobile network in close proximity to mobile users and connected devices, with the aim to offer zero-perceived latency and improved network awareness to critical layer-7 applications [ChaoHu-2015].

Mobile Edge Computing arouse as a natural evolution of mobile base stations, state of the art IT technologies, and networking means. According also to the 5G Infrastructure Public Private Partnership (5G-PPP), MEC is well-known to be one of the key emerging technologies for 5G networks (together with NFV and SDN). It allows exposing the edge mobile broadband network as hosting infrastructure to vertical applications, contributing in this way to satisfy the mission-critical requirements of 5G in terms of throughput, latency, scalability, awareness and automation. Moreover, MEC also provides, through specific APIs, real-time insight into radio network information and location awareness to hosted applications.

MEC is a technological paradigm complementary to NFV. While NVF provides network functions and services, the MEC framework enables layer-7 applications running at the edge of the network. Given the similarity of infrastructural dependencies between NFV and MEC, the MEC ETSI working group is actively working on defining guidelines on how making these two technological paradigms smoothly coexist in the same infrastructure. In detail, they envision MEC servers to be deployed at multiple locations, "*such as at the LTE macro base station (eNodeB) site, at the 3G Radio Network Controller (RNC) site, at a multi-Radio Access Technology (RAT) cell aggregation site, and at an aggregation point (which may also be at the edge of the core network). The multi-RAT cell aggregation site can be located indoors within an enterprise (e.g. hospital, large corporate HQ), or indoors/outdoors for a special public coverage scenario (e.g. stadium, shopping mall) to control a number of local multi-RAT access points providing radio coverage to the premises*" [ChaoHu-2015].

Figure 5.3.3 shows the reference architecture of the MEC framework as specified by ETSI [ETSIMEC]. It is composed of three main building blocks: *i*) a mobile edge host: a MEC server running the MEC platform and hosting the applications; *ii*) a mobile edge platform manager: a software entity providing control functionalities to the mobile edge host; and *iii*) a mobile edge orchestrator: a central network entity devised to manage and coordinate the operations of applications over multiple mobile edge hosts.

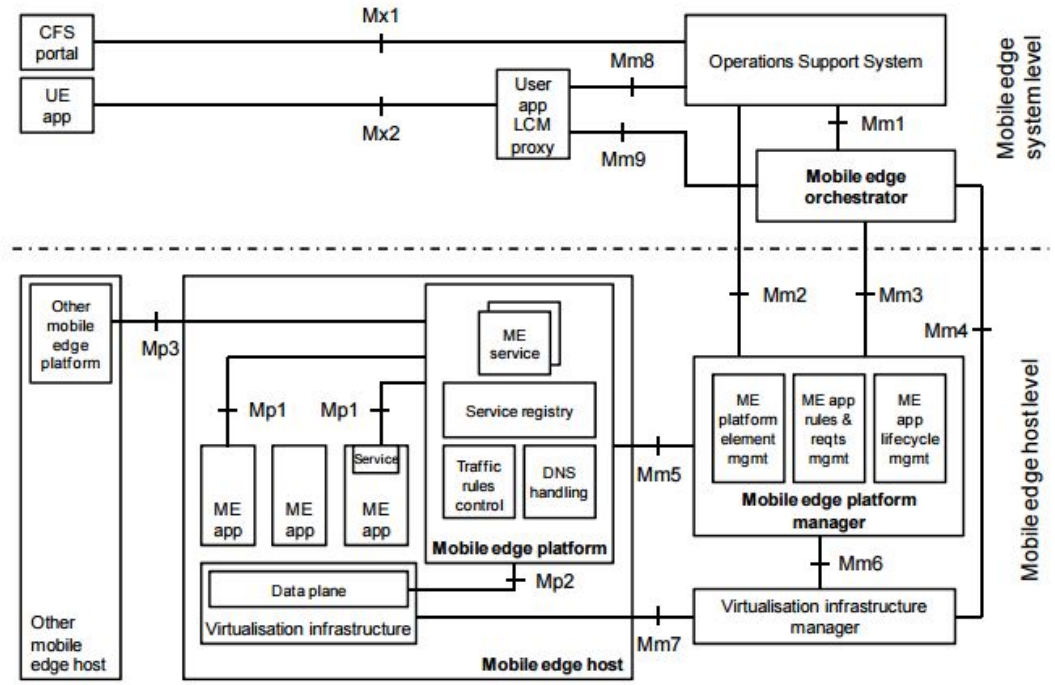


Figure 5.3.3: MEC reference architecture, building blocks, and reference points.

It can be noted how the MEC standard is still far from a complete technological maturity to cope with all the architectural and operational requirements of 5G and vertical applications. Despite its crucial role in the 5G landscape, current specifications only support monolithic applications, and the orchestration logic and tenant control interfaces are too simplistic to fully support many of the 5G vertical applications. For instance, current specifications do not allow external application orchestrators to manage the lifecycle of composite applications with components having heterogeneous requirements. Also, interconnectivity among applications is still under heavy investigation in order to provide easy and flexible interfaces towards NFV.

Virtual Infrastructure Managers (VIMs)

As previously sketched, the 5G network infrastructure is expected to include heterogeneous and variable-sized datacentre facilities, which will be interconnected through software-defined networking means. According to the ETSI NFV definitions, each of these facilities constitutes a NFV Infrastructure Point of Presence (NFVI-PoP), and exposes its virtualization capabilities through a VIM layer.

Various technological trends [Taleb-2017, Mijumbi-2016a, Mijumbi-2016b] suggest that telecom operators will:

- Deploy PoP facilities at various network aggregation levels, spanning from few large private datacentres at the network core, up to micro/pico cloudlets attached to mobile base stations (as suggested by the ETSI MEC Working Group [ETSIMEC]) or directly integrated in eNodeB units [NokiaLiquid].
- Exploit PoP facilities to host both NFV and vertical MEC/fog services.

As clearly analysed in various public reports [Chappell-2015] [Kavanagh-2015] [SDxCentral-2017a], even though **OpenStack** is becoming the *de-facto standard* VIM solution in the NFV ecosystem, it does not currently support many advanced capabilities required by the NFV specification. For instance, among other missing relevant features, OpenStack does not support live migrations of virtual machines among PoPs. To this purpose, the research community is actively discussing the dilemma if an OpenStack extension towards the needed additional capabilities would be more appropriate than a project fork.

The final decision on this complex dilemma will clearly and directly affect the software stack architecture needed for the joint lifecycle management and resource provisioning of NFV services and vertical applications.

In detail, telecom operators might opt for using "separated" or "integrated" VIMs to manage both service types. In the case of separated VIMs, the same PoP can provide an instance of NFV-compliant VIM, and a second one, general-purpose VIM, to host vertical applications. The main drawback with this option is that different VIM instances cannot share the same hardware infrastructure, which may lead telecom operators to rigidly divide the hardware resources in their PoPs between NFV services and vertical applications hosting.

In the integrated case, the PoP is abstracted by the same VIM instance able to expose general-purpose and NFV-compliant APIs. Obviously, the main and non-trivial drawback of an integrated VIM solution is its complexity.

The open-source **OpenVIM** project [OpenVIM], originally designed by Telefónica and now part of the ETSI Open Source MANO (OSM) project [OSM, OSMWP], is the reference implementation of an NFV VIM. OpenVIM is a lightweight piece of software specifically designed to support high and predictable performance.

As shown in Figure 5.3.4, it interfaces with the NFVI compute nodes and an OpenFlow controller to provide computing and networking capabilities and deploy virtual machines. It provides an OpenStack-like northbound interface (OpenVIM API), where enhanced cloud services are offered. This implementation follows ETSI's NFV-PER001 recommendations. From a general perspective, OpenVIM can be meant as a fork of OpenStack, but, in its latest release (release 2 announced in April 2017), it started to integrate components from the OpenStack project.

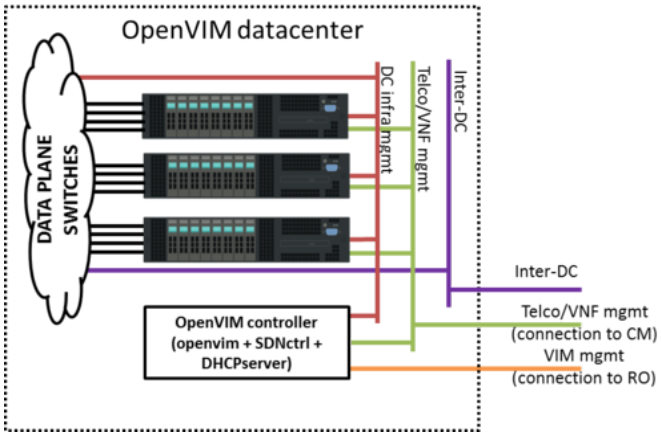


Figure 5.3.4: The OpenVIM architecture.

Table 5.3.2 reports a comparison among OpenVIM, OpenStack, Eucalyptus, Open Nebula, and CloudStack.

A different approach has been undertaken within the open-source **OpenVolcano** project [OpenVolcano, Bruschi-2016], designed from scratch in the context of the H2020 INPUT project [INPUT].

OpenVolcano can be seen, to the best of our knowledge, as the first open-source VIM prototype for personal applications in MEC/fog environments with support for NFV (but currently not completely supporting the NFV specification). Through a highly modular architecture, it provides a centralized OpenStack-like interface to tenants, allowing them to declare their modular services as "templates," which are then instantiated only upon end-user request on a per-user basis (i.e., each user has its own personal service instance).

OpenVolcano provides autonomic advanced functionalities for supporting and offloading MEC/fog services towards mobility. It includes specific VNFs for monitoring handovers for selected user mobile equipment, as well as for injecting/handling their traffic towards a user personal network.

The front-ends of MEC/fog services (referred to as Virtual Images) are attached to this virtual network, which has been realized through a sophisticated SDN slicing mechanism, called "SDN multi-centre overlay," specifically designed to scalably support bulk seamless migrations of attached virtual machines. Each application component is associated to a proximity class, which, in its turn, is mapped on a "centre" of the aforementioned SDN mechanism.

Upon handover events, OpenVolcano triggers internal policy reinforcement algorithms to evaluate if the position of application components meets the desired proximity and, if needed, it migrates the application components (centre per centre) closer to the user in a seamless fashion.

OpenVolcano also supports the interconnection of services with further components residing in the public cloud.

Table 5.3.2: Comparison of OpenStack, Eucalyptus, OpenNebula, CloudStack, and OpenVIM.

	OpenStack	Eucalyptus	Open Nebula	CloudStack	OpenVim
First release	21 October,2010	01 may, 2008	01 March 2008	06 November 2012	5 February, 2015
Latest release	Openstack Pike. 30th Aug. 2017	Eucalyptus 4.4.2 4th Aug. 2017	OpenNebula 5.4.1 19th Sept. 2017	Apache CloudStack 4.10.0 3rd July 2017	OSM release 2, April 2017
NFV compatibility	high	minimal	minimal	partial	full
Programming Language	Python	C and Java	C, C++, Java Script, Ruby	Java 1.7	Python
License	Apache V2.0	GPL V3.0	Apache V2.0	Apache V2.0	Apache V2.0
Installation difficulty	Difficult	Easy	Easy	Medium level	Easy
Cloud Type	Private, public and hybrid	Private and hybrid	Private, public, hybrid	Public, private, hybrid	Private
Hypervisor	KVM, LXC, QEMU, UML, VMWare VSphere update 1 and newer, Citrix Xenserver, Xen Cloud Platform and Baremetal service via pluggable sub drivers	Xen for CentOS 5 and RHEL 5, KVM for CentOS 6, RHEL 6 and Ubuntu, VMware's ESX	Xen, KVM, VMware	VMware, KVM, XenServer, Xen Cloud Platform(XCP) and Hyper-V.	KVM, VMWare VSphere
Architecture	Fragmented architecture-every OpenStack component is individual project	Monolithic	Monolithic	Monolithic	Monolithic
Public cloud compatibility	Amazon EC2 and Amazon S3	AWS	Microsoft Azure, AWS	AWS EC2 and S3	ETSI NFV-PER001
Database Support	RabbitMQ, MySQL, MongoDB, MariaDB	PostgreSQL	SQLite or MySQL	MySQL	-

Further commercial VIM platforms include Cisco NFV Infrastructure [CiscoNFVI], VMWare vCloud NFV Platform [vCloud], and Ericsson NFVi solution [EricssonNFVI].

Multi-VIM Cloud Orchestrator Platforms

Orchestrators are the component in the cloud ecosystem, where the rising of new projects and products, often with diverse visions and technological basis, largely concentrated. In the public cloud, almost every cloud providers offer proprietary service orchestration tools to their customers, while other platforms provide multi-cloud compatibility. Table 5.3.3 includes a short description of the most interesting and market-prominent orchestrators for cloud applications.

Table 5.3.3: Main State-of-the-Art Cloud Orchestration Platforms.

Name	Description
Amazon CloudFormation	It offers an easy way to create and manage a collection of related Amazon Web Services (AWS) resources, provisioning and updating them in an orderly and predictable fashion. It is based on declarative templates for describing the AWS resources, and any associated dependencies or runtime parameters, required to run the application.
AWS Elastic Beanstalk	Orchestration service offered by AWS for deploying an infrastructure, which orchestrates various AWS services, including EC2, S3, Simple Notification Service (SNS), CloudWatch, autoscaling, and Elastic Load Balancers.
IBM Cloud Orchestrator (ICO)	It is based on IBM's Business Process Manager foundation technology. With IBM's acquisition of Gravitant, IBM is rationalizing functionality across the Gravitant SaaS platform and the ICO on-premises platform. IBM intends to merge SmartCloud Cost Management into Gravitant (and remove it from ICO), with Gravitant becoming more of the aggregation platform for service catalogue, cost management and brokering over time, and ICO performing orchestration tasks.
VMWare VRealize	VMware has enhanced and integrated a set of products, starting with its DynamicOps acquisition in 2012. It is mostly an on-premises offering (some components are provided via SaaS). On-premises support includes vSphere, KVM and Hyper-V, while public cloud support includes AWS and Azure. vRealize Suite 7's strength is in its affinity for existing VMware tooling and the breadth of offerings within their cloud suite. A challenge for VMware is the continual need to lessen complexity around packaging (including pricing), deployment and implementation. vRealize Suite 7 is a step in the right direction.
HP Cloud orchestration	It automates the provisioning of infrastructure across hybrid environments. It also provides performance metrics and reporting on infrastructure and applications, so IT can monitor them to consistently meet SLAs and implement equitable chargeback processes. It automates operations tasks such as provisioning, patching, compliance auditing, monitoring, and remediation. It orchestrates complex cloud management processes and provides dashboards, reporting, and unified portal for self-service delivery of infrastructure and applications
Flexiant Cloud Orchestrator & Concerto	It enables service providers and enterprises to design, create and manage their own virtual public, private or hybrid cloud solutions. Flexiant Cloud Orchestrator can manage the entire cloud solution, from hardware, network and storage management to metering, billing & customer/end-user self-service. It can also be used to augment and orchestrate existing platforms such as VMware vSphere The Concerto extension extended the Flexiant solution towards service lifecycle management in multi-cloud systems.
RightScale Cloud Management Platform	Starting with cloud management of AWS in 2006, and then expanded to a multicloud CMP, it now supports Amazon, Azure, Google, Rackspace, IBM SoftLayer, VMware, and OpenStack-based public and private cloud infrastructures, as well as bare-metal servers. Its key strength is its relatively large size for a privately-held cloud management company, supporting both worldwide implementations, as well as providing managed and professional services to augment the breadth of its cloud management offering.
Gigaspace Cloudify	It focuses on taking complex, mission-critical legacy applications and giving them cloud attributes such as auto scaling, dynamic deployment, multi-cloud deployment and management. Cloudify is an open-source-based, on-premises, subscription-based software offering that targets large enterprises and telecommunications companies (for NFV).
Alien4Cloud	Open-source. It allows people in the enterprise to collaborate in order to provide self-service deployment of complex applications, taking in account the different experts through a role based portal. It leverages the following concepts: Location: Deployment target (cloud or set of physical machines); Components: Software components to deploy; Topologies (or blueprints): Description of multiple software components assembled together (to build an application); Applications: Actual applications to deploy with environments and versions, each of them being associated with a topology; TOSCA: An emerging standard to describe service components and their relationships.
Ubicity Central	It is a TOSCA-based service orchestrator. It includes: a repository for storing service templates, custom types, and customer-specific policies; service composition functionality to construct end-to-end services from primitive service templates; fulfilment functionality to resolve dangling requirements; policy enforcement; orchestration and lifecycle management of services.
Chef	It allows dynamically provision and de-provision one's infrastructure on demand, to keep up with peaks in usage and traffic. It enables new services and features to be deployed and updated more frequently, with little risk of downtime. With Chef, one can take advantage of all the flexibility and cost savings the cloud offers. Chef is integrated with all major cloud providers including Amazon AWS, Microsoft Azure, VMWare, IBM Smartcloud, Rackspace, OpenStack, HP Cloud, Google Compute Engine, and others.
Puppet Orchestrator	It models infrastructure's applications and services, as well as all the dependencies between them, using a proprietary configuration domain specific language (DSL). The Puppet Orchestrator uses the resulting model to intelligently and automatically determine the order of operations, how and where to securely discover and share information between services, and when to wait for a service to become available before continuing with an application deployment.
Apache Brooklyn	Open-source framework for modelling, deploying and managing distributed applications defined using declarative YAML blueprints. The design is influenced by Autonomic computing and promise theory and implements the OASIS CAMP (Cloud Application Management for Platforms) and TOSCA.
Avni	Founded in 2013 and officially launched as of 2015, Avni offers cloud management, as well as Layers 4–7 network functionality–enabling enterprises to reduce lock-in to any one cloud provider for network functions such as load balancing, caching, and application firewalls.
BMC	Launching into the market in 2010 with Cloud Lifecycle Management (CLM). CLM supports a variety of public and private cloud options. It also includes some automatic workload balancing functions.

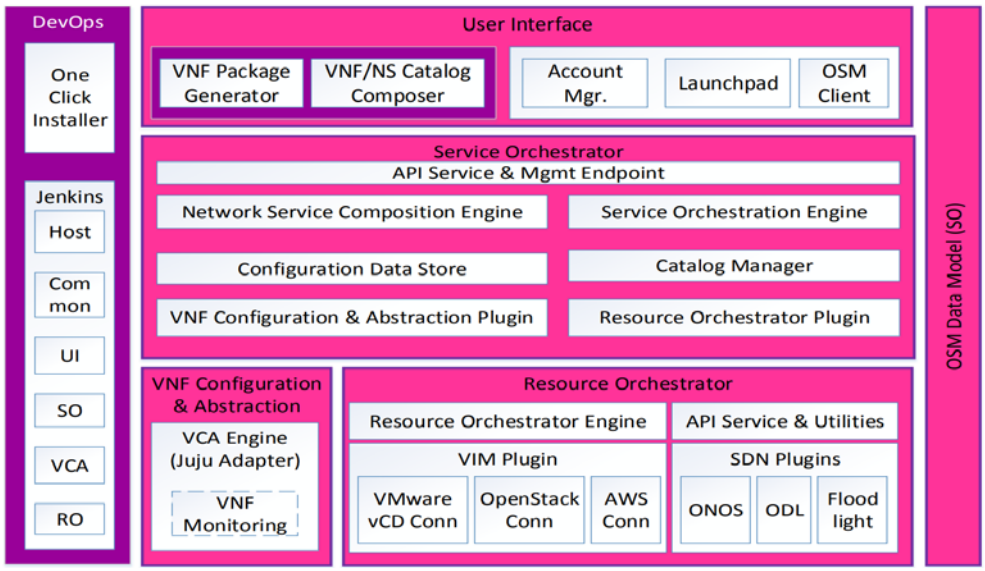
CloudBolt	Developed by an IT service provider that focused on the U.S. federal market, and entered the CMP market in 2012. It is a privately held company located in Campbell, CA, with approximately 35 customers. CloudBolt targets "brownfield" environments, where there is a need to orchestrate existing and diverse infrastructure components, including visibility and management for resources not provisioned by CloudBolt.
DivvyCloud	It offers on-premises cloud management operational software that allows enterprises to have visibility, management and active policy enforcement over multiple public and private IaaS cloud services. It does this from a single interface, by enabling near-real-time event processing and automated actions based on changes in state.
Embotics	Founded in 2006, and its CMP offering is vCommander, known for its ability to be easily deployed and its extensibility (that is, the ability to be fitted for an enterprise's unique requirements).
Bertram Capital Morpheus	Created in 2014. Designed specifically to be cloud agnostic, Morpheus provides one-click provisioning, seamless cloud to cloud migration, and takes less than a day to get fully up and running.
RedHat CloudForms	After acquiring ManageIQ in late 2012, Red Hat released CloudForms in 2013. CloudForms' key strengths are policy management through metadata tagging and management of existing brownfield environments via continuous auto-discovery.
Scalr	Open-source, launched in 2008. It offers three versions of its CMP, an on-premises enterprise edition, an open-source community edition, and a SaaS version. It is typically deployed as on-premises software. Scalr differentiates by focusing on the self-service cloud usage by application developers, desired-state infrastructure management, and transparent, automated policy enforcement across multi-cloud deployments.
ARCADIA H2020 Project Orchestrator	The ARCADIA Orchestrator and Smart Controller supports the dynamic setup and management of highly distributed applications over a programmable infrastructure. Any type of application/service can be denoted in the form of a service graph, including the parts related with the setup and support of network functionalities. A set of functionalities, like horizontal scaling capabilities, multi-IaaS network connectivity establishment, firewall setup and operation, are supported and activated –if requested- based on the requirements imposed on behalf of a services provider or even the application developer.

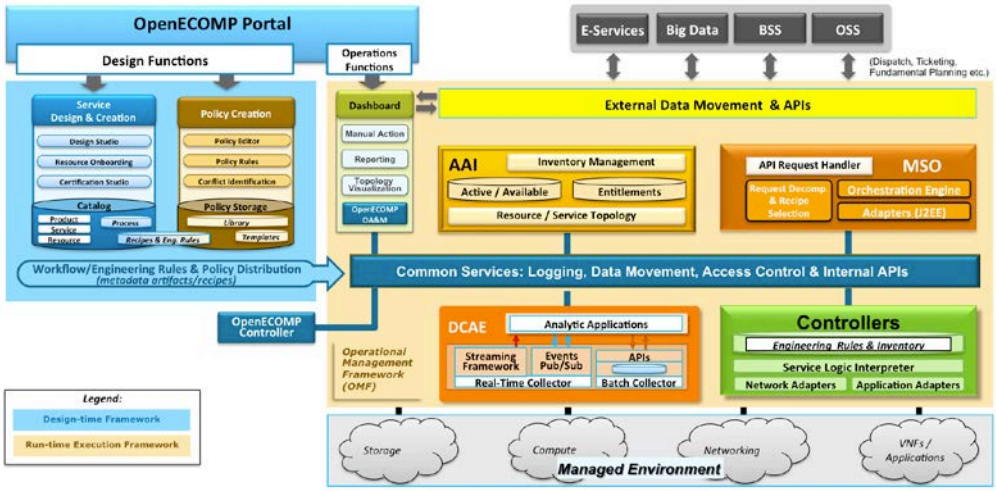
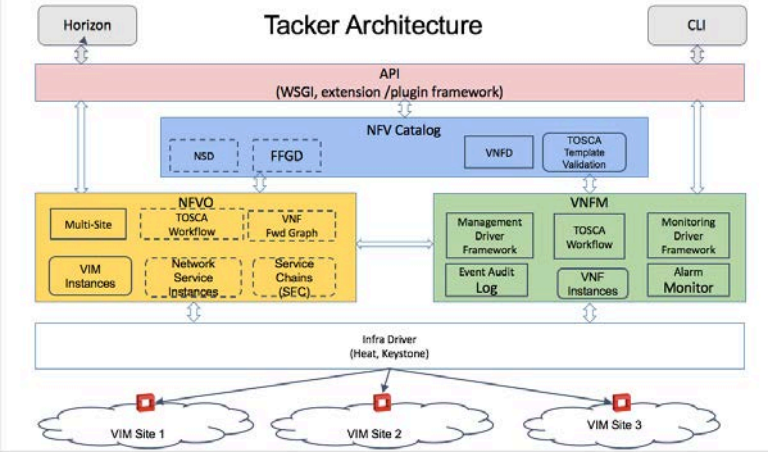
5G NFV Management and Orchestration

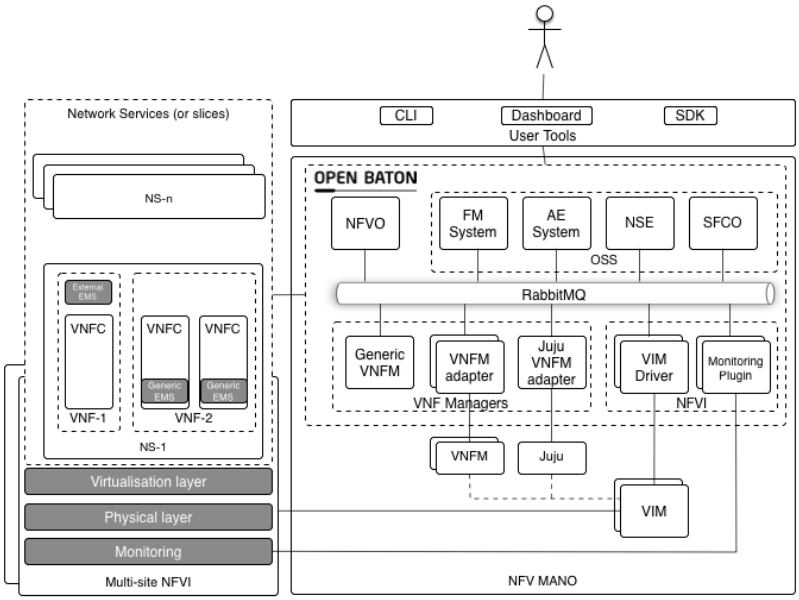
As previously sketched, the service orchestration landscape has become even more complex and fragmented with the rising of 5G/NFV technologies. Started with the “war” among SDN controllers (ONOS, OpenDayLight, FloodLight, etc.), the new generation of NFV-related open-source projects are competing under the umbrella of organizations/consortia like ETSI, the Linux foundation, and OpenStack. Although ETSI has made efforts to integrate these open-source projects into its community and promote a concerted effort on standards, open-source projects can clearly take on a life of their own.

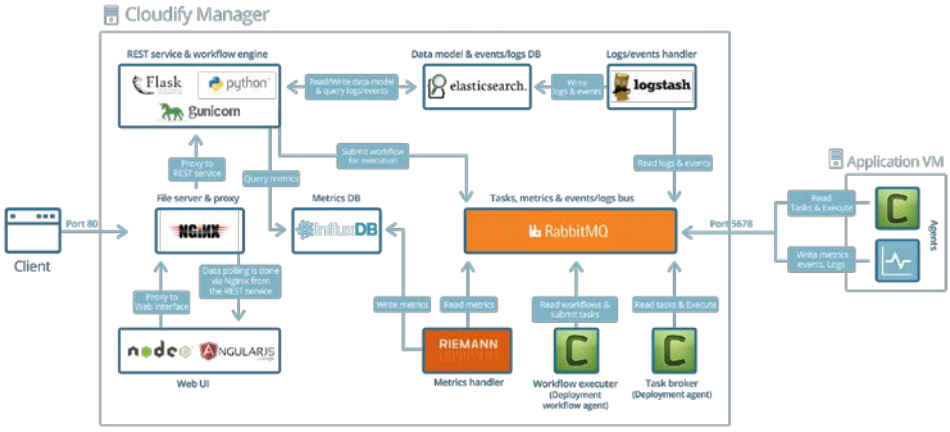
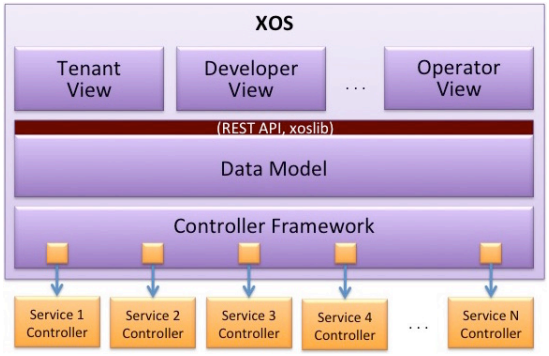
Table 5.3.4 summarizes the most interesting and well-known platforms for NFV orchestration. Further comparison among these platforms can be found in [Andrushko-2017].

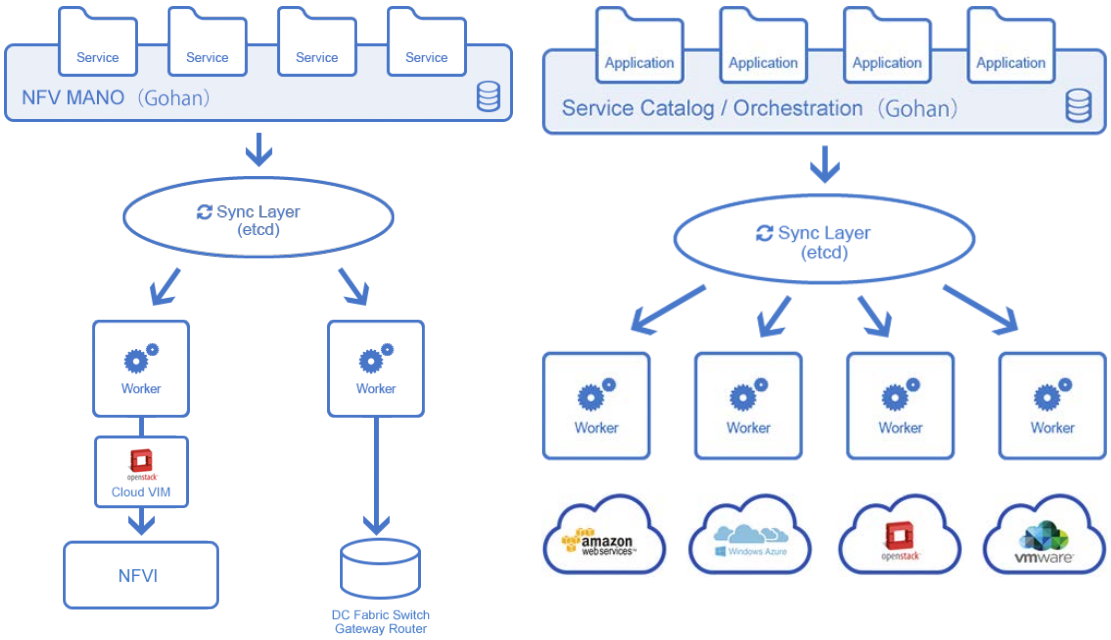
Table 5.3.4: Main NFV Orchestration Platforms.

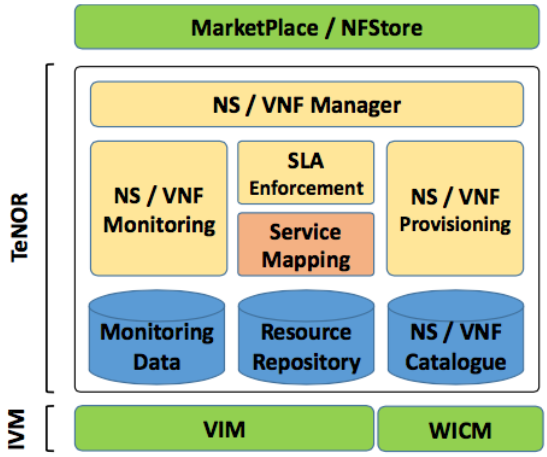
Name	Description
ETSI Open Source Mano (OSM) [OSM]	Open-source MANO (OSM) is an ETSI-hosted (https://osm.etsi.org/) project to develop a production quality Open-source NFV Management and Orchestration (MANO) software stack aligned with ETSI NFV and released under Apache 2 License. OSM was seeded by the OpenMANO efforts from Telefónica, and in combination with RIFT.io's orchestration and Canonical's Juju acting as a VNFM, provides for a more complete and standard-compliant MANO solution. OSM also includes OpenVIM (see the previous subsection). OSM has now garnered about 60-member organizations worldwide that are actively working on expanding the capabilities of the platform. As illustrated in Figure 5.3.5, the Service Orchestrator layer (based on RIFT.ware) is in charge of delivering end-to-end network services, while the Resource Orchestrator (based on OpenMANO) coordinates the allocation of network, computing and storage infrastructure. Finally, the VNF Configuration and Abstraction layer (based on JuJu) manages the network functions lifecycle, by allowing user-defined scripts to run according to the service status.  **Figure 5.3.5: OSM internal architecture and mapping with the ETSI NFV MANO standard. Source: [Hoban-2017].** The key aspects behind OSM are the usage of a strong data model based on the YANG language and the flexible architecture resulting from the usage of a plugin system. By leveraging the high-level API creation to automatic tools that parse the YANG models, the development of the OSM can be accelerated. Moreover, consistency can be assured in API calls using automatic validation. By using a plugin system, the resource orchestrator can be easily extended to support a wide range of commercially available SDNs and VIMs. Additionally, OSM is designed to support brown field developments and interoperability, easing the adoption of OSM in pre-existent infrastructure. Currently, the focus of the OSM development is providing a production-ready orchestration mechanism that supports service assurance, security (via authentication and rule based authorization), full scalability of VNFs and NSs, nested network services and service chaining.

Linux Foundation Open Network Automation Platform (ONAP) [ONAP]	Announced in early 2016, and born by the integration between Open-O and ECOMP. It is a comprehensive platform for real-time, policy-driven orchestration and automation of physical and virtual network functions that will enable software, network, IT and cloud providers and developers to rapidly create new services. ONAP consists of a number of software subsystems. These subsystems are part of two major architectural frameworks: *i*) a design-time environment to design, define and program the platform; *ii*) an execution-time environment to execute the logic programmed in the design phase. The design-time framework is an IDE with tools, techniques, and repositories for defining and describing deployable assets. It supports the development of new capabilities, augmentation of existing capabilities and continuous operational improvement throughout the life cycle of a service. The execution-time framework uses closed-loop, policy-driven automation to drive down operational costs. Built-in dynamic, policy-enforced functions are provided for component and workload shaping, placement, execution, and administration. Access to the design-time and execution-time frameworks are provided by the ONAP Portal, a role-based user interface.  **Figure 5.3.6: ONAP internal architecture. Source: [ONAPWIKI].**
OpenStack Tacker [Tacker]	Started in March 2015. Tacker is an official OpenStack project building a Generic VNF Manager (VNFM) and a NFV Orchestrator (NFVO) to deploy and operate Network Services and Virtual Network Functions (VNFs) on an NFV infrastructure platform like OpenStack. It is based on the ETSI MANO Architectural Framework and provides a functional stack to Orchestrate Network Services end-to-end using VNFs. Tacker uses TOSCA for VNF meta-data definition. Within TOSCA, Tacker used the NFV profile schema. It does not internally support Service Function Chaining, but it rather exposes north-bound APIs for this purpose.  **Figure 5.3.7: OpenStack Tacker internal architecture. Source: [Tacker].**

OpenBaton [Openbaton]	Designed by the Fraunhofer Institute of Open Communication Systems, Open Baton is an extensible and customizable framework capable of orchestrating network services across heterogeneous NFV Infrastructures. In its fourth release, Open Baton significantly increased the list of features provided. It can manage a diverse ecosystem of VNFs, through its Generic EMS and Generic VNFM, composing them runtime in any kind of network services. It integrates with existing VNFMs via a plug-and-play model, exposing AMQP and RESTful APIs, as well as SDKs in different programming languages (Java, Python, Go). Developing a VNFM adapter takes minutes using the SDKs and tutorials provided. It manages a multi-site NFVI supporting heterogeneous virtualization and cloud technologies. Although OpenStack is the major supported VIM, it provides a driver mechanism for supporting additional VIM types. It supports multi-tenancy, aka network slicing, at the infrastructure level, making use of SDN technologies for ensuring isolation between multiple network services sharing the same physical resources. It supports runtime operations fulfilling the needs of the FCAPS model integrating external Operational Support Services (OSS) components. For instance, it provides auto-scaling and fault management based on monitoring information coming from the monitoring system available at the NFVI level.  **Figure 5.3.8: OpenBaton internal architecture. Source: [Openbaton].**

Cloudify Telecom Edition [CloudifyTelco]	Cloudify Telecom Edition is an open source NFV orchestration framework. Performing the MANO functions in the NFV architecture: NFVO and Generic VNFM. Cloudify allows modelling Network Services and functions (VNFs) and automate their entire life cycle, including deployment on any NFVI, monitoring all aspects of the deployed resources, detecting issues and failures, manually or automatically remediating them and handling on-going maintenance tasks. In addition, Day 1 and Day 2 operations like scaling, healing, and upgrading the deployed services can be easily performed by Cloudify. NFV services and architecture in its entirety (NFVI, VNF, Application Code, Scripts, Tool Configuration, Metrics and Logs) can be described in a Blueprint. Written in a human readable YAML format, a blueprint allows for high granularity of configuration of VNFs. The complete lifecycle of each part of one's VNF and network service can be defined in a blueprint. Cloudify can deploy one's VNFs and manage them by utilizing the tools of one's choice, bringing DevOps best practices into management and orchestration for NFV.  **Figure 5.3.9: Cloudify internal architecture. Source: [Cloudify].**
Chord XOS [XOS]	XOS defines a collection of abstractions in support of services and service composition. It leverages existing datacentre cloud management systems (e.g., OpenStack) and SDN-based network controllers (e.g., ONOS), to provide explicit support for multi-tenant services. In doing so, XOS makes it possible to create, name, operationalize, manage and compose services as first-class operations. The XOS implementation is organized around three layers. At the core is a Data Model, which records the logically centralized state of the system. It is the Data Model that ties all of the services together, and enables them to interoperate reliably and efficiently. The logical centralization of this state is achieved through a clearcut separation between this authoritative state and the ongoing, fluctuating, and sometimes erroneous state of the remainder of the system: the so-called operational state. The ability to distinguish between the overall state of the system at these two levels (authoritative Data Model and operational backend) is a distinguishing property of XOS.  **Figure 5.3.10: XOS internal architecture. Source: [XOS].**

Gohan [Gohan]	Maintained by NTT, Gohan is a general-purpose API Gateway Server that enables the creation of RESTful services by orchestrating microservices. Due to its flexible architecture, shown in Figure 5.3.11, Gohan can be applied to several use cases, including Service Catalogue and Orchestration Layer on top of Cloud services and NFV MANO, which manages both VIM and legacy network devices. Similarly, to Opensource MANO, modelling plays an important role in Gohan, with the information model being defined in rich YAML schemas. Gohan is a particularly new initiative and has progressively gained momentum, mainly because of its applicability in the web industry.  **Figure 5.3.11: Gohan architecture when applied to different use cases.**
Tata Telco Cloud [TCS]	A TCS (Tata Consultancy Services) open-source initiative that provides an Open VNF Manager to enable NFV service orchestration on the OpenStack platform.

T-NOVA Orchestrator (TeNOR)	The T-NOVA project has designed and implemented a management/orchestration platform named TeNOR for the automated provision, configuration, monitoring and optimization of Network Functions -as-a-Service (NFaaS) over virtualised Network/IT infrastructures. In other words, T-NOVA combines IT/cloud virtualisation and Network-as-a-Service concepts to offer a complete end-to-end Cloud Network service. Figure 5.3.12 presents TeNOR 's high level architecture. The functional blocks represented as yellow, blue, and red are TeNOR specific functions, while the green blocks represent T-NOVA north and southbound system components.  **Figure 5.3.12: TeNOR high-level architecture.** For TeNOR, a micro-service based architecture was selected, to ensure a lean and modular implementation and operation of the system. Micro-services are organized in two groups: one dedicated to NSs, which provides services to the upper layers (i.e., green blocks) and requests services from the second group, which is dedicated to VNFs related operations. The micro-services required for the function of TeNOR are: • NS/VNF Manager: it is a facade for the northbound interface -the Marketplace for the NS Manager, the NS Manager for the VNF Manager (VNFM)- and manages the NS/VNF Catalogue. The proposed architecture embraces both the concept of generic VNFM as well as VNF specific VNFMs, as suggested by ETSI WG; • Service Mapping: this module contains the mapping algorithm implementations, which map the required resources to support a NS instance to the best available location in the infrastructure; • NS/VNF Provisioning: it accepts requests for NS instances from the Marketplace (through the NS Manager) and for VNF instances from the VNFM; it also manages the NS/VNF Instances repositories; • NS/VNF Monitoring: it accepts Virtual Machine (VM) based monitoring data from the lower virtualized infrastructure and management (VIM) layer and maps it to the corresponding NS/VNF instances. These data are later provided to the Marketplace, for both Customers and Function Provider dashboards; • SLA Enforcement: in charge of comparing monitoring data to the agreed SLA for every NS instance, and generating alerts for impending SLA breaches. Data associated with a potential breach are passed to the NS Manager, which initiates the necessary actions to guarantee the SLA (it either migrates or scales VNF instances or improves their network connections). More detailed information on the TeNOR architecture and implementation can be found in [Riera-2016] and [T-NOVA-2015].

The above platforms will play a crucial role in the 5G landscape, since they will be deputed to maintain and to manage the lifecycle of any mobile access and core network functionalities. The same design of the 5G ecosystem is currently heavily affected by NFV concepts and paradigms.

As shown in Figure 5.3.13, the upcoming 5G network architecture is going to be much more modularized than previous generations, and each module is going to be provided through well separated (virtual) network function(s). Moreover, with the aim of providing the maximum possible level of flexibility, 3GPP is also focusing on the possibility of allocating such functions in various deployment combinations, and with diverse chaining (i.e., with or without some function) to cope with the peculiarities to address the support of vertical applications (e.g., with or without mobility).

In this respect, it is worth noting that the 5G framework is going to be the first generation of mobile network architectures not only exploiting, but also providing intrinsic virtualization capabilities through the "**network slicing**" concept [Thalanany-2016].

A network slice is defined to be "*a set of network functions and the resources for these network functions which are arranged and configured, forming a complete logical network to meet certain network characteristics. […] A network slice instance (NSI) is complete in the sense that it includes all*

functionalities and resources necessary to support a certain set of communication services thus serving a certain business purpose. […] The NSI contains NFs (e.g., belonging to Access Network and Core Network). If the NFs are interconnected, the 3GPP management system contains the information relevant to connections between these NFs such as topology of connections, individual link requirements (e.g. QoS attributes), etc." [3GPP-2017]. To cope with the above specifications, 5G network slices will be formed by multiple sub-networks (e.g., RAN and Core), which will expose specific configuration and communication interfaces.

The same 3GPP association defined the slice as "*a managed entity* ***in the operator's network***" with its own lifecycle, and whose final consumers are vertical industries and/or Over-The-Top (OTT) players. Communication Service Providers are also introduced as main intermediate players between the Network Operator, owning the slices, and the final consumers.

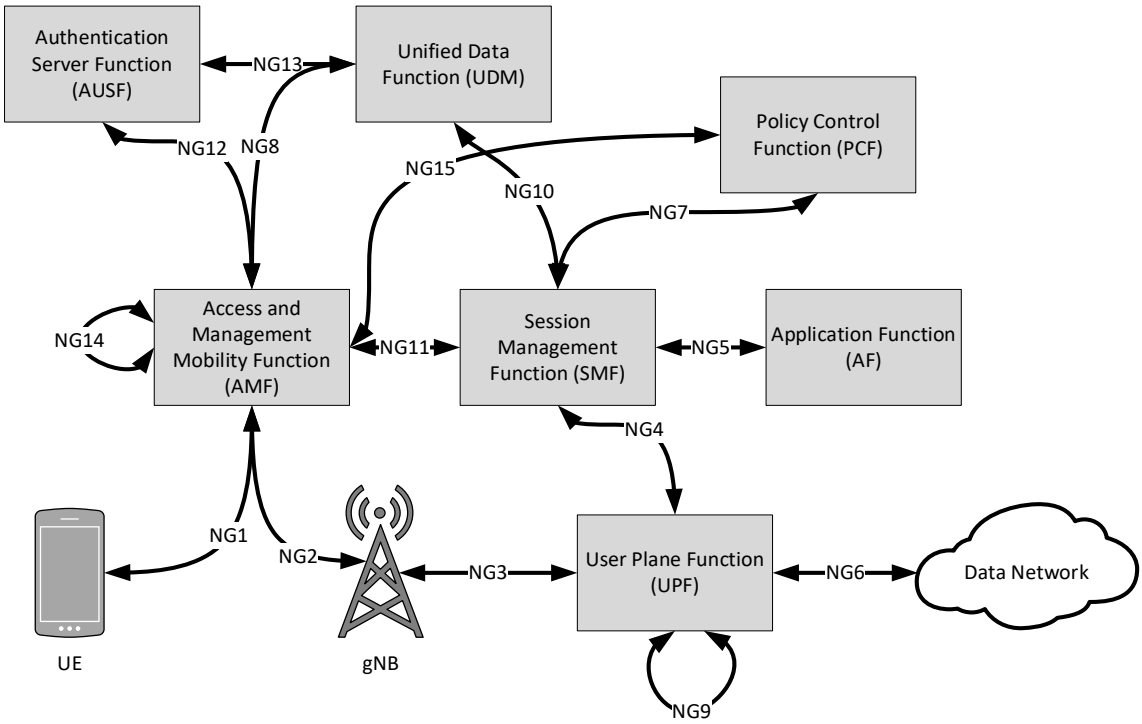


Figure 5.3.13: Candidate 5G architectures under investigation at 3GPP.

Owing to the considerations above and as noted in [Flinck-2013], network slice management functions operate above the NFVO level. In view of 3GPP, as well as ETSI NFV, network slice management functions have been supported by interfaces exposed by Network Service Providers' OSS/BSS to third-party overlying players.

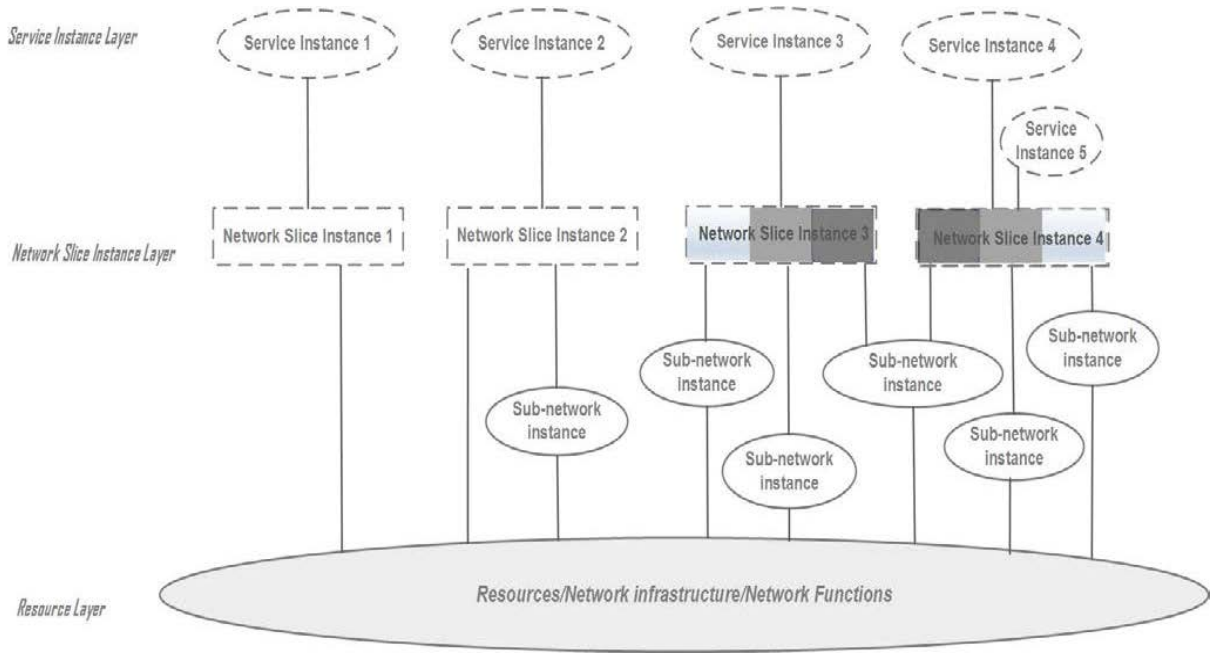


Figure 5.3.14: Network slicing conceptual outline. Source: [Thalanany-2016].

Offloading Application Orchestration over Multi-Site Infrastructures

Today, only few orchestration platforms are designed to support a large number of multi-site VIMs in an effective fashion. Often, the enablement of this feature induces high complexity in the orchestration workflow and operation. For example, when the multi-site capability is enabled, the Open Source MANO platform (OSM) [OSM] requires the external pre-provision of network resources and interconnectivity among VIMs.

From a high-level perspective, multi-site resource management mechanisms are meant as means to offload the complex work of application/network service orchestrators. Existing frameworks have different offload capabilities and objectives, among which the most relevant ones can be summarized as follows:

- Providing the implementation of different VIM APIs, and offering them through simplified abstraction interfaces;
- Aggregating the view of resources and their status across the VIMs acting on different sites;
- Synchronizing and pre-distributing data (e.g., VM images, storage volumes, SSH keys, security groups, etc.) among the VIMs;
- Providing connectivity among tenants' network overlays split across multiple VIMs;
- Increasing the automation level, by providing a simplified north-bound API to orchestrators.

The largest part of multi-site support features has been realized in the context of NFV orchestration projects, like Open Baton, OSM, and OpenStack Tacker, since the need for relying on geographically distributed datacentres is more evident than in classical cloud applications.

Two OpenStack projects, named KingBird [KingBird] and Tricircle [Tricircle], can be considered as the most advanced solutions towards the offloading of orchestration operations in multi-site VIM environments.

Specifically, KingBird (part of the OS OPNFV Multisite sub-project) allows resource synchronization and management for multi-region OpenStack VIM instances. In more detail, it allows aggregating/centralizing the view of resources (e.g., quotas, tenant-level IP/MAC addresses, etc.), as well as pre-distributing data (e.g., security groups, images, SSH keys, flavours, etc.).

As shown in Figure 5.3.15, KingBird runs as a centralized service, relying in its turn on a centralized Keystone module. Therefore, it natively supports multi-site, but not multi-administrative domain scenarios. It also defines its own northbound APIs (which should be supported by orchestrators), and offers a plugin for integration with the (centralized) OpenStack Horizon Dashboard.

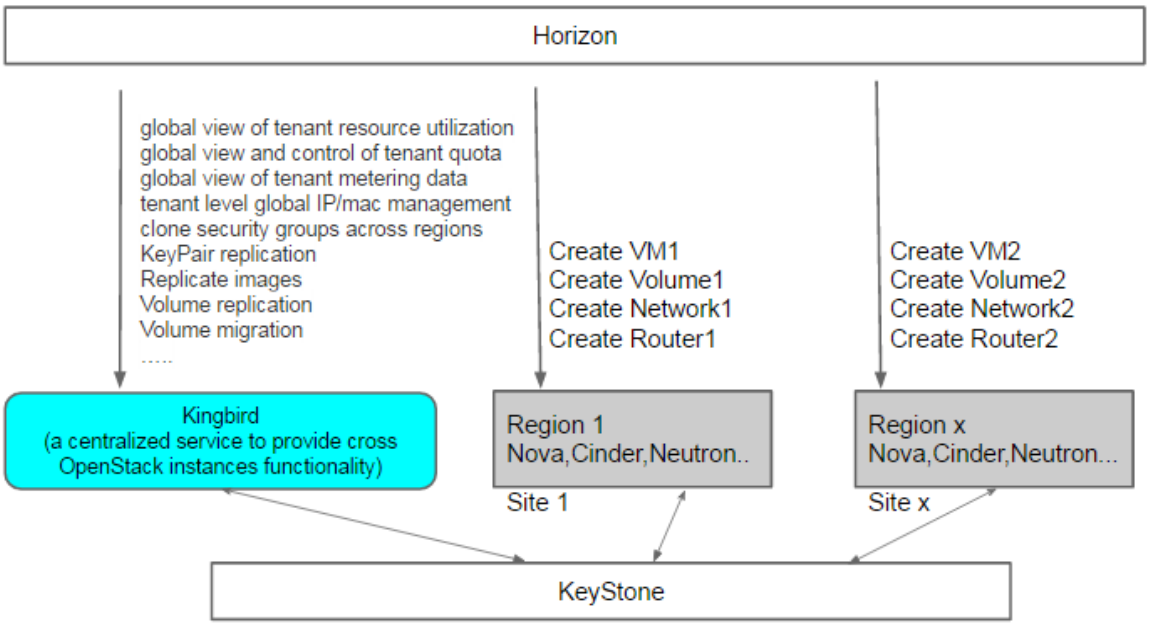


Figure 5.3.15: Kingbird workflow example. Source: [Huang-2015].

The OpenStack Tricircle project has been designed to provide networking automation across OpenStack Neutron instances in multi-region scenarios. In detail, this project fulfils the following objectives/functionalities:

- Leverage Neutron API for cross VIM networking seamless automation.
- Support modularized tenant-level capacity expansion in large scale clouds.
- L2/L3 networking automation across VIM instances (tenant's VMs communicate with each other via L2 or L3 networking across different VIM instances – see Figure 5.3.17).
- Security group applied across OpenStack instances.
- Tenant level IP/MAC addresses management to avoid conflict across OpenStack instances.
- Tenant level quota control across OpenStack instances.

As shown in Figure 5.3.16, similarly to KingBird, it should rely on a central module instance; but, differently from the previous project, it does not define new northbound APIs, since it rather acts as an OpenStack Neutron API gateway. This solution obviously permits faster integrations, since it can transparently expose aggregated Neutron APIs to orchestration platforms in a Neutron-native fashion.

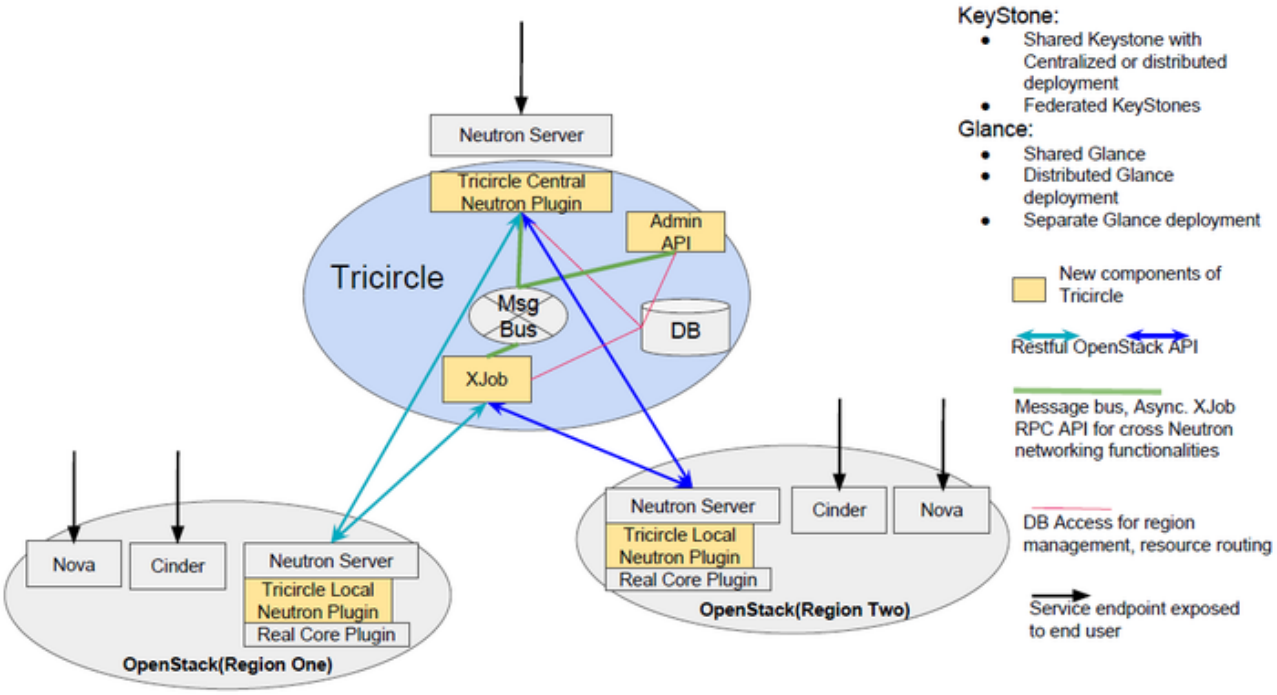


Figure 5.3.16: Tricircle internal architecture. Source: [TricircleWIKI].

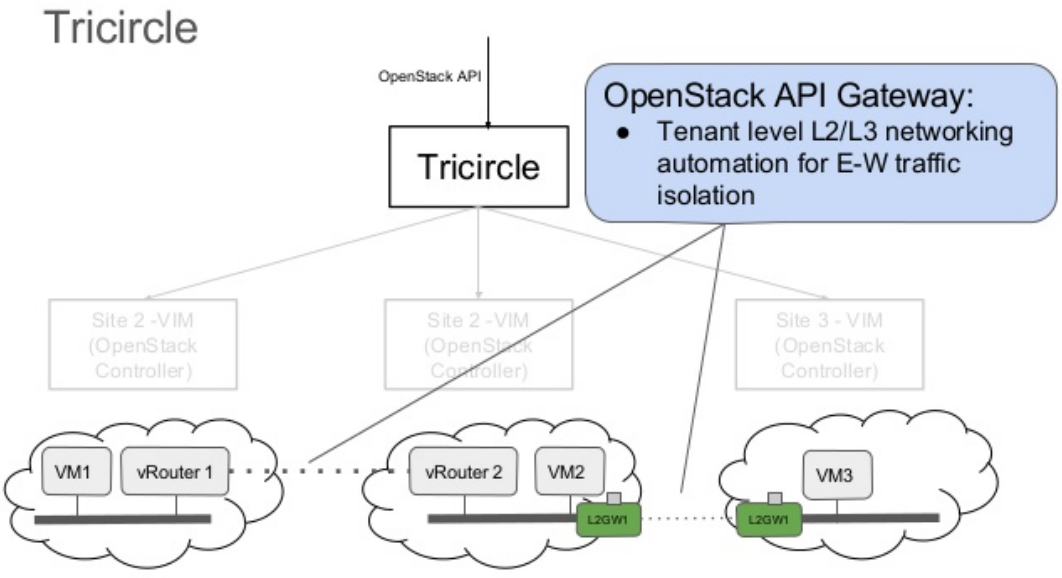


Figure 5.3.17: Multi-site network automation in Tricircle. Source: [TricircleWIKI].

One of the biggest challenges for Multi-VIM Orchestrators is the placement of applications and services (which include the virtual network functions) in order to optimize performance and reduce latency [Mijumbi-2016a]. This task becomes even harder when considering dynamic changes in the network and computing infrastructure triggered during the scaling process. Recent research and development activities have been proposed to surpass the current limitations by exploring new methodologies such as machine learning and other intelligent mechanisms.

As outlined in the Linux Foundation OS-O&M, the future 5G ecosystem will rely on multiple software-driven layers, where various functionalities should require their own lifecycle orchestration and management, but software acting at the various layers should be harmonized to cooperate in the effective management of the entire ecosystem. In detail, as shown in Figure 5.3.2, 5G infrastructures will be abstracted and managed by VIMs, network services by NFV orchestrators (see Table 5.3.4), while vertical applications, residing on the upper OS-O&M layer, should require extensions with respect to the platforms introduced in Table 5.3.3. In this respect, various standardization bodies and industrial forums are focusing on the definition of new interfaces and data models relevant to upcoming 5G architectures. Important examples are the Lifecycle Services Orchestration (LSO) [LSO] framework under specification in the Metro Ethernet Forum (MEF), as well as 5G network slicing in 3GPP [3GPP-2017].

Despite these efforts, the overall 5G ecosystem and its support to vertical applications is still very fragmented and incomplete.

One of the main challenges and contributions of the MATILDA project to the 5G design will be to offer a top-down vision, for vertical industries and their applications towards the 5G services and infrastructures. To this end, the project will rely on the latest state of the art technologies, paradigms and in-progress standardization activities related to 5G, like NFV, Mobile Edge/Fog computing, as well as innovative concepts like 5G network slicing.

As previously sketched, based on the specifications in [ETSINFV-2014b] and [3GPP-2017], the MATILDA project will consider a 5G architecture composed of a number of different subsystems, which might act in autonomous fashion with diverse objectives. In detail, in a base-line scenario, we can expect having four levels at least of orchestration/control, with the following main roles:

- The vertical application orchestrator (i.e., the MATILDA orchestrator), managing the lifecycle of the chain of application components of the 5G-ready application.
- The 5G Telecom Service Provider OSS/BSS exposing network slices as-a-Service to vertical industries, and mapping them into network services to be instantiated at the NFV orchestrator.
- The NFV orchestrator(s), managing the network services realizing the needed network slices among application components and 5G user equipment (UE).
- The VIM consolidation control element(s), mapping virtual resources acquired as-a-Service by overlying orchestrators onto physical ones, and realizing isolation among different tenants (e.g., network slices, applications, etc.) in each PoP datacentre.
- The WAN network controller(s), realizing the logical interconnectivity among sets of service/application components instantiated in different PoP and/or towards 5G UEs.

It can be noted that all the above subsystems not only have different roles in the 5G ecosystem, but they might be also associated/owned by different stakeholders/organizations, namely vertical industries, network service providers, network (computing and transport) infrastructure providers, respectively. As depicted in Section 5.3.2, the same subsystems are usually designed to provide advanced “multi-tenancy” and “multi-domain” capabilities, in the sense that they are designed to host multiple overlying subsystems to exploit the resources/services from multiple instances of the underlying subsystems.

Unfortunately, the interdependency, positioning, and interfacing of all the subsystems above is still under investigation in standardization bodies and industrial forums, along with the specification of the 5G architectural framework itself. In detail, the interdependency between the (MEC and NFV) orchestration layer and VIMs/WAN controllers is already partially in place in cloud legacy systems, and its evolution partially specified by the ETSI NFV working group [ETSINFV-2014b], but the integration and reference points and interfaces between the vertical applications and network service

orchestrators is still a core open issue, where only preliminary design and integration approaches have been proposed.

In such scenario, the MATILDA project will focus on contributing on and evolving architectural solutions for enabling vertical applications to be flexibly interfaced with 5G network services, and for allowing the direct hosting of application components in the network edge. To this purpose, arising technologies, interfaces, and information/business models will be considered and used as starting base.

In detail, as shown in Figure 5.3.18, the scenario addressed by the project will be composed of the vertical application orchestrator (the MATILDA orchestrator), maintaining a graph of components. The application graph is terminated towards 5G connected things and user terminals. When such components (or a part of them) are deployed onto the (virtualized) telecom infrastructure, the interconnection among them is realized, according to the upcoming 3GPP specifications [3GPP-2017], as 5G network slices. Such network slices can correspond to a heterogeneous set of NFV services to be applied in the 5G radio access network and/or in the core segment. Slices will be offered by the Telecom Service Provider OSS/BSS, and customized by the vertical application orchestrator.

VNFs composing network services in network slices and part of application components will be deployed in PoPs. Each PoP exposes a VIM interface, interconnected through one or more network segments managed by WIMs. WIMs will support the instantiation of virtual point-to-point links able to support resource reservation and QoS metrics (e.g., bandwidth, maximum delay, etc.).

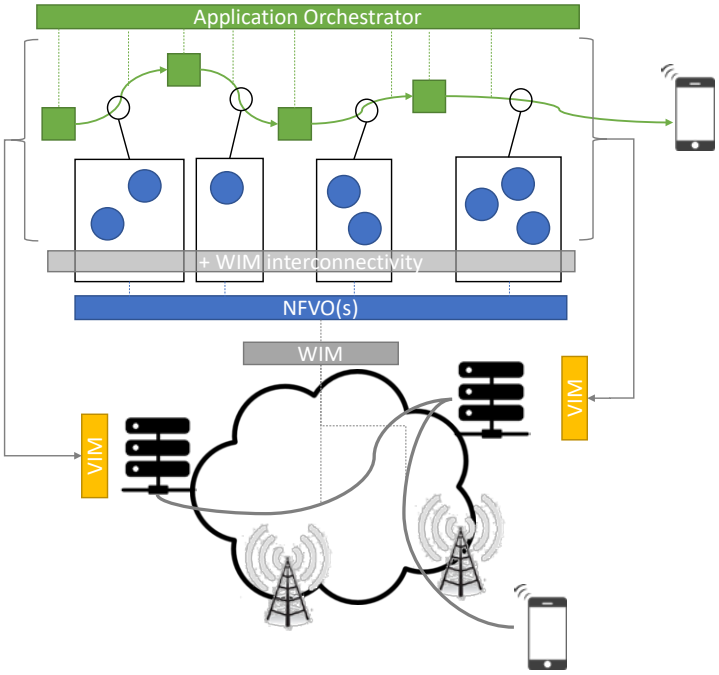


Figure 5.3.18: Main 5G ecosystem considered by the MATILDA project.

5.3.2 Technology Requirements

ID	MO_1
Unique Name/Title	5G Network Slices
Priority	High

Type	Functional
Brief Description	The MATILDA 5G-ready application orchestrator should acquire, update, release 5G network slices as-a-Service from the Telecom Service Provider's OSS. Such a functionality is envisaged to be realised through a Slice Manager.
Rationale	3GPP specification review.
Validation method/Relevant KPI	Conformance with the 3GPP specification (e.g., 3GPP TR 28.801).

ID	MO_2
Unique Name/Title	Vertical Applications inside Telecom Infrastructures
Priority	High
Type	Functional
Brief Description	In order to provide zero-perceived end-to-end latency or scalability levels for mass-scale service, the MATILDA framework has to exploit hosting capabilities provided by 5G Network Service Providers at the network edge. A first example of this support is the ETSI MEC standard.
Rationale	Technology review and Use cases.
Validation method/Relevant KPI	Host application components at the Network Service Provider's edge facilities.

ID	MO_3
Unique Name/Title	Vertical Applications Termination towards 5G Devices
Priority	Medium
Type	Functional
Brief Description	The MATILDA orchestrator should orchestrate the service graph by explicitly representing user equipment and 5G connected things as service terminations.
Rationale	Technology review and Use cases.
Validation method/Relevant KPI	UEs explicitly present in the application graph metamodel.

ID	MO_4
Unique Name/Title	As-a-Service interface for Computing Resources at the Telecom BSS/OSS
Priority	High
Type	Functional
Brief Description	The MATILDA framework must be able to manage the lifecycle of application components deployed in the Telecom Service Provider's facilities through state of the art IaaS/PaaS interfaces, and evolve them, if needed, to carry additional metadata.
Rationale	Technology Review.
Validation method/Relevant KPI	Manage the lifecycle of an application component deployed at the Telecom Service Provider's facilities through (evolved) IaaS/PaaS APIs.

ID	MO_5

Unique Name/Title	Locality and Mobility Awareness
Priority	High
Type	Functional
Brief Description	The Matilda Orchestrator must be able to manage the lifecycle of the application components deployed at the network edge by considering locality of computing resources and of network terminations, as well as mobility of UEs.
Rationale	Technology Review and Use cases.
Validation method/Relevant KPI	Metadata related to locality and mobility are part of the metamodel used by the Matilda Orchestrator.

ID	MO_6
Unique Name/Title	Multi-site support
Priority	Medium
Type	Functional
Brief Description	The Matilda orchestrator must be able to manage the resources and the lifecycle of application components at diverse facilities, like central/remote public/private/hybrid cloud facilities or at the mobile network edge.
Rationale	Technology Review and Use cases.
Validation method/Relevant KPI	VIM support for public cloud and the mobile Edge.

ID	MO_7
Unique Name/Title	Event Reactiveness
Priority	Medium/High
Type	Functional
Brief Description	The MATILDA Orchestrator should be able to react to mobile network events, such as registration of a UE to the RAN, handovers between cells, etc.
Rationale	Technology Review and Use cases.
Validation method/Relevant KPI	Receive control plane messages from the network platform.

5.4 Intelligent Application Orchestration Mechanisms

The orchestrator is going to support a set of intelligent orchestration mechanisms, including deployment and runtime policies enforcement, data monitoring, fusion and analytics, and a context awareness engine for inference of knowledge based on the collected information. In this subsection, the main components of the intelligent orchestrator are described in detail focusing on the existing technologies for each solution, as well as on the advances that MATILDA will bring beyond the state of the art.

5.4.1 Existing Technologies and Progress Beyond

Optimization & Context Awareness Engine

MATILDA entails a strict metamodel that will allow the creation of multiple rules that will be evaluated continuously. Since multiple policies can be defined per network-aware application graph prior to

actual deployment, the service provider has to create a policy or select one from the available set of policies. A policy may consist of multiple expressions and each expression combines several conditions that can trigger multiple actions. The conditions and the actions are bound to the defined normative models like the chainable application component metamodel, the 5G-ready application graph metamodel, the VNF/PNF metamodel and the VNF-FG metamodel. In an analogous manner, the available actions that have to be performed upon the satisfaction of some conditions may vary according to the granularity of the element that has to perform the action. Policies' definition may regard deployment objectives, as well as runtime policies' enforcement actions, both of which are highly interconnected. Deployment objectives are fulfilled based on the preparation of the deployment plan by the MATILDA optimization engine, while runtime policies' enforcement is based on a real-time rule-based management system (MATILDA Context Awareness Engine).

The Optimization Engine interacts with the Deployment Manager in order to get the optimal deployment plan. In particular, the Deployment Manager and the Optimization Engine are the entities that are responsible for "translating" a deployment model into an optimal deployment plan taking under consideration: a) the available programmable resources, b) the current situation in the infrastructures where these resources reside, and c) the selected policy. In MATILDA, the deployment of network-aware applications will be considered as an extension to the highly distributed applications embedding problem [ARCADIA-D.3.1]. Furthermore, the requirement for deployment of the embedded VNF-FGs will be considered as well, complicating even further the overall embedding problem. Policies will be extended accordingly and enforced in conjunction with the optimization of objectives and constraints satisfaction. Implementation of the MATILDA Optimization engine is going to be realized based on a metaheuristics solver engine; OptaPlanner [Optaplanner], which smoothly integrates into the architecture as the component to produce near-optimal placement plans. In OptaPlanner constraints and objectives are illustrated as rules in the Drools rules engine [Drools].

MATILDA optimization engine: As mentioned in the previous subsection, the MATILDA optimization engine will be implemented based on the OptaPlanner solver. OptaPlanner is an *open source* software, released under the Apache Software License 2.0. The specific software solution constitutes a lightweight, embeddable constraint satisfaction engine, which optimizes planning problems and offers the capability to combine optimization heuristics and metaheuristics with a very efficient score calculation. In particular, OptaPlanner supports three families of optimization algorithms: Exhaustive Search, Construction Heuristics and Metaheuristics. Score constraints are written in an Object-Oriented language, such as Java™ code or Drools rules**.** The combination of OptaPlanner with a rule engine (like Drools Expert) is very efficient, as the two engines can complement each other. More specifically, a rule engine, such as Drools Expert, can be very efficient for calculating the score of a solution to a planning problem, making it easy and scalable to add additional soft or hard constraints. However, it tends to be unsuitable to find new solutions. On the contrary, the optimization engine is good at finding new improving solutions for a planning problem, without necessarily brute-forcing every possibility, but it needs to know the score of a solution and offers no support in calculating that score efficiently. The interaction between the two types of engines is also represented in Figure 5.4.1.

Apart from OptaPlanner, there are several proprietary and open-source optimization software solutions, a brief overview of which is presented in Table 5.4.1. More specifically, the category of open-source solutions also includes OpenMDAO, MIDACO and several other software products. However, as OptaPlanner is fully compatible with the rules engine that will be employed in MATILDA (Drools) and it includes a vast range of optimization algorithms that can be adapted in real-time, it has been selected as the most appropriate tool to be used during the project.

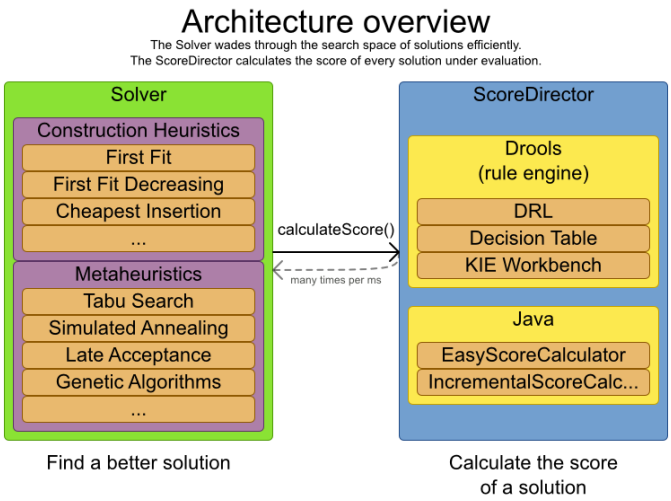


Figure 5.4.1: High-level architecture overview that shows the interaction between the Optimization Engine & the Rule-enforcement engine [Optaplanner].

Table 5.4.1: Comparison of different optimization software solutions.

Name	Language	Academic/ non-commercial use is free	Description
ALGLIB	C++, C#, FreePascal, VBA	Yes	General-purpose library that also includes an optimization package.
APMonitor	Fortran, C++, Python, Matlab, Julia	Yes	A differential and algebraic modelling language for mixed-integer and nonlinear optimization.
Artelys Knitro	C, C++, Python, Java, C#, Matlab, R	No	General-purpose library specialized in nonlinear optimization. Handles mixed-integer problems and mathematical programs with equilibrium constraints.
GNU Linear Programming Kit	C	Yes	Free library for linear programming and mixed integer programming.
GNU Scientific Library	C	Yes	Free library provided by the GNU project.
LIONsolver	C++, Java	Yes	Support for interactive and learning optimization, according to RSO principles.
MIDACO	C++, Python, Matlab, C#, Fortran, R, Java, Excel, VBA	Yes	General-purpose global optimization solver, single- and multi-objective MINLP problems, supporting parallelization and large-scale.
NMath	C#	No	C# numerical library built on top of MKL.
OpenMDAO	Python	Yes	Multidisciplinary Design, Analysis, and Optimization framework, written in the Python programming language.
OptaPlanner	Java	Yes	Lightweight optimization solver in Java
Pagmo/Pygmo	C++ and Python	Yes	Easy coarse parallelization of solvers.

Deployment of a network application service requires assigning and instantiating an execution environment for each software component, while illustrating the communication links among them as required. Assignment of infrastructure resources to execution environments and communication links has to fulfil requirements, satisfy objectives and avoid policy violations. This may be expressed as an optimization problem, which falls into the category of NP-Hard problems; finding an optimal solution is computationally intractable. Thus, an exact solution is appropriate for small instances of the problem and efficient heuristics can be used to produce near optimal solutions in a short time. Several

forms of the problem have been studied in the past including the Virtual Network Embedding (VNE) problem [Chowhury-2009], the Virtual Datacentre Embedding (VDCE) problem [Zhani-2013], [Fischer-2013] and the Cloud Application Embedding (CAE) problem. In the ARCADIA project the Highly Distributed Application Embedding (HDAE) problem is considered, where an application service graph is required to be deployed [ARCADIA-D.3.1]. HDAE differentiates in that the focus is specific on the application component requirements and not on a general purpose virtual machine. Policies can be used to enforce business rules, as well as to specify resource constraints over the programmable infrastructure. Managing the policies' definition, enforcement and conflict resolution, on behalf of the Services Provider, is a crucial task for applications' deployment and operation. Rule-based systems such as Drools Business Rules Management System provide the required ease to illustrate complex policies, flexibility to modify rules and interoperability for efficient policies definition and enforcement during runtime. A Policies' Management Framework (PMF) is providing policies enforcement over the deployed application service graphs following a continuous match-resolve-act approach.

In MATILDA, the deployment of 5G-ready applications will be considered as an extension to the HDAE problem presented in ARCADIA augmented for the deployment of embedded VNF-FGs. Policies will be extended accordingly and enforced in conjunction with optimization objectives and constraints satisfaction. Implementation of the MATILDA Optimization engine is going to be realized based on a metaheuristics solver engine, OptaPlanner, which smoothly integrates into the architecture as the component to produce near-optimal placement plans. In OptaPlanner constraints and objectives are illustrated as rules in Drools rules engine. Implementation of the MATILDA runtime policies enforcement mechanisms will be based on Drools, supporting interoperability among the deployment and runtime policies' definition and enforcement.

MATILDA Policy Manager and Context Awareness Engine: The Policy Manager and Context Awareness Engine (CAE) refer to the entities responsible for managing the collected information/context by the various monitoring streams, the extraction of advanced information and insights upon reasoning over them and the suggestions of actions based on the active policies per network-aware application graph.

The Policy Manager in MATILDA is providing policies enforcement over the deployed graphs following a continuous match-resolve-act approach. Policies' enforcement is realized through a rule-based framework that attempts to derive execution instructions based on the current set of data and the active rules. The CAE consists of three basic functional components (see Figure 5.4.2): a) the **working memory**, where facts based on the provided data are inserted, b) the **production memory**, where predefined-static rules that are bound to the policy exist and c) an **inference engine** that supports **reasoning and conflict resolution over the provided set of facts and rules**, besides triggering the appropriate actions.

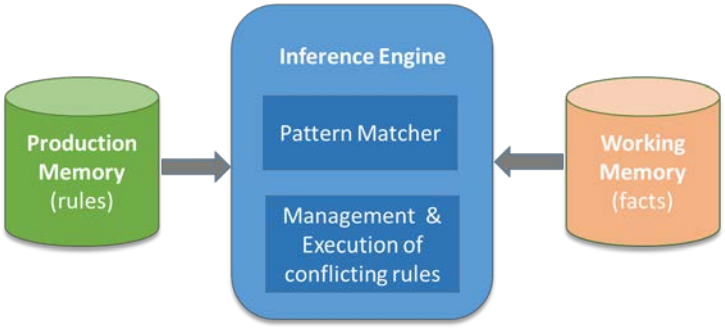


Figure 5.4.2: High-Level View of Context-Awareness Engine.

An extended overview of the different rules engines is provided in [rule_eng], focusing on the ones that are implemented in Java. In MATILDA, the policies' enforcement mechanisms are going to be deployed based on Drools, which is an open-source, highly scalable Rules Management System (BRMS) solution.

Drools is a Java implementation (released under the Apache Software License 2.0), which became integrated with the widely used JBoss Java EE application server. It offers a variety of ways to state rules and connect the rule logic to programs, providing a core Business Rules Engine (BRE), a web authoring and rules management application (Drools Workbench) and an Eclipse IDE plugin for core development. Other similar tools are OpenRules [OpenRules], which is a general purpose BRMS that is also available as an Open Source product[1], OpenL Tablets [OpenL] referring to another BRMS solution that consists of a Business rules engine, WebStudio (a web-based rules' editing and management environment), Web services and a rules' repository.

The Context Awareness Engine in MATILDA is the entity in charge of processing data coming from various data streams and extracting advanced events and insights that may be helpful to the various intelligent orchestration mechanisms and especially to the Policy Manager. Under this perspective, the need for real-time processing on large data streams led to the use of Complex Event Processing (CEP). CEP is a technology for processing events on the fly providing high throughput and low latency and is being used by applications that require real-time or near real-time processing of large data streams (events) like stock exchange, network monitoring, etc. CEP technology contrasts with the traditional processing of data where data is first stored and then processed. The first CEP systems, also known as streaming systems, include Telegraph [Chandrasekaran-2003] and Aurora [Abadi-2003], which were developed at the beginning of the century as centralized systems. Borealis [Abadi-2005] was a distributed version of Aurora, in which each operator of a query can be deployed on a different node. Aurora [Cherniack-2009] and Flux [Shah-2003] were also distributed systems in which queries were run on different nodes; however, the incoming events where routed through a single node, which eventually became a bottleneck, as shown by StreamCloud [Gulisano-2012], an elastic parallel distributed streaming system that provides intra-query, inter-query parallelism and inter-operator parallelism. Open source systems like Apache Storm [ApacheStorm] made CEP popular and widely used. Although Storm provides the same kind of parallelism as StreamCloud, it does not provide a query language for CEP (query languages like the one of Esper [Esper], which eases the task of developers). Other open source systems like Spark streaming [ApacheSpark] provide a declarative query language; however, it cannot be deployed across different datacentres. Apache Flink [ApacheFlink] follows a similar approach.

As already mentioned, the MATILDA CAE is an intelligent framework able to extract information and insights in order to provide this information to policy enforcement mechanisms to act accordingly. It plays an important role in the Orchestration Mechanism of MATILDA since it:

- Creates knowledge by utilizing machine reasoning techniques to analyse the streaming monitoring data
- Facilitates policy enforcement over the deployed graphs based on policies (i.e. rules)
- Resolves conflicts with respect to resources and the mapped policies

The Policy Manager consists of three main blocks:

- A Reasoning Engine that extracts facts over the streaming data, called Working Memory. The Reasoning Engine uses Complex Event Processing techniques over the streaming monitoring data in order not only to recognize the events (facts), but also to consider the optimum time frame/window for event identification
- A Policies Mapper that maps predefined rules and policies, called Production Memory
- A Conflict Resolution Mechanism that acts based on the facts of the streaming data, the Working and Production Memory and the available resources providing resource information to the Optimization Engine and rules to the Orchestration Mechanisms

[1] OpenRules is dual licensed: GPL for open source projects and commercial license for non-open source projects.

Based on the above, the added value beyond the State of the Art refers to the introduction of CEP techniques in the 5G domain, which provide runtime insights to optimize the management of the 5G infrastructure. These insights are then being analysed to provide more intelligent information regarding the current state of the system, in order to assist the Optimization Engine to take more advanced decisions.

Data fusion, Analysis and Profiling

The estimation of computation, storage and network resources during initial deployment, as well as during the application's lifetime is a complex procedure. Specifically, it involves the real-time monitoring of the workload, the initial and real-time estimation of required resources, while at the same time the offered resources must satisfy QoS constraints. The automation of this process, called "auto-scaling" requires the assignment and release of resources either reactively or proactively in order to avoid under- or over-provisioning.

Auto-scaling schemes involve rule-based approaches using thresholds, queueing theory, control theory, time-series analysis and reinforcement learning [Lorido-2014, Perez-2014]. Prediction of resources needed can also help in the development of proactive solution [Roy-2011]. A major part of these approaches is the profiling [Xu-2012, Ren-2010], i.e. measuring multiple metrics through a profiler tool during a program's runtime, whether at application- or function-level.

Machine learning (ML) will greatly contribute to the analysis and profile modelling. The methods may include various approaches and algorithms. In general, data analysis methodologies -both supervised and unsupervised, separately or combined- may be employed for clustering, classification and regression, such as k-means for clustering, Principal Component Analysis (PCA) for dimensionality reduction, Gaussian Mixtures Models (GMMs), Linear/ Logistic Regression, Decision Trees, Random Forests, etc., for regression and/or classification. Greater benefits are expected from the implementation of deep learning methodologies, which can offer improved prediction results, as shown in past benchmarks surpassing other ML approaches [LeCun-1998]. Some deep learning algorithms are:

- Deep (Stacked) Auto-Encoders with Unsupervised Pre-Training to extract higher level features. The features can be fed to Hidden Markov Models, for sequence labelling and sequence prediction or other classification/regression models.
- Recurrent Networks and its variation Long Short-Term Memory (LSTM) Recurrent Neural Networks for time-series predictions (sequence prediction for regression and classification).
- Deep Belief networks, i.e. stacked restricted Boltzmann machines, used for unsupervised clustering of unlabelled data or in conjunction with an artificial neural network for supervised regression and classification.
- Convolutional Networks (ConvNets) for processing matrices with additional dimensions (e.g., for inclusion of spatial and spectral information in image processing).

In order to develop static and dynamic models, e.g. for the prediction of workload (type and volume), for the estimation of resources needed and for profiling, it is imperative to ensure the collection of large amounts of data and measurements coming from active and passive monitoring of the infrastructure (i.e., resource availability, QoS metrics) and running applications (i.e., real-time profiling, performance metrics hooks). In addition, data aggregation and filtering are required in order to minimize overhead and make data exploitable towards feeding analytics engines (i.e. for predictions) and providing input for complex decision making (i.e. policy enforcement, optimization reconfigurations, auto-scaling). For these reason, data fusion/ingestion, data analysis mechanisms and the latest big data technologies have to be deployed.

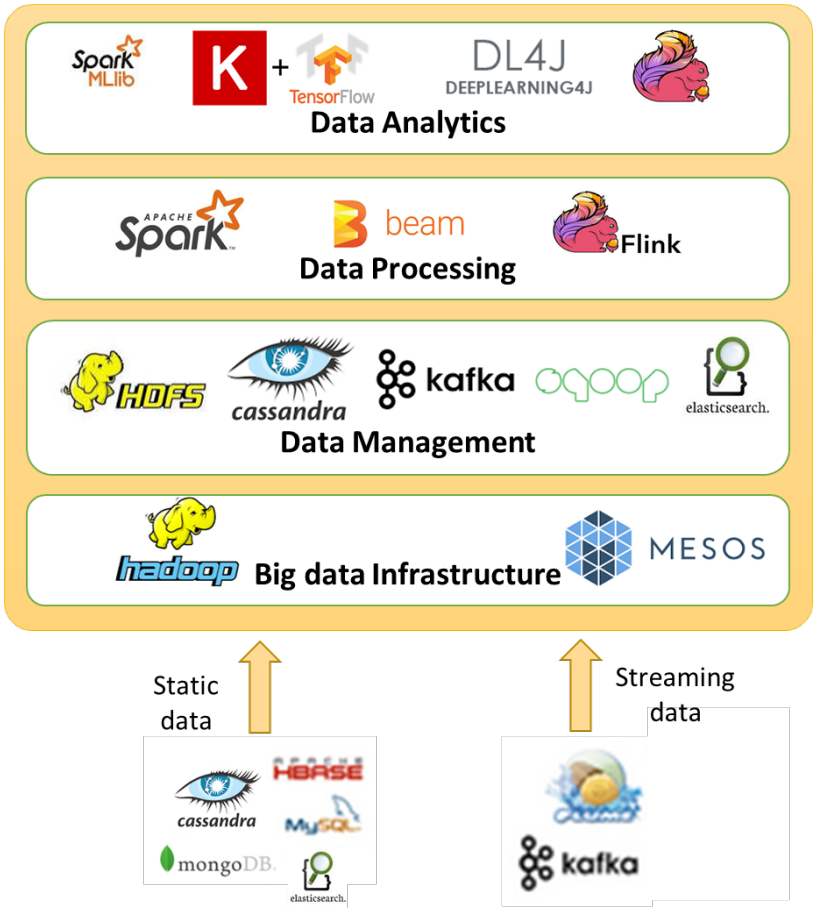


Figure 5.4.3: Representative open source big data technologies.

In recent years, the need for additional resources and scalable approaches has led to the rapid advancement of big data technologies and big data solutions. Some of the most representative solutions are shown in Figure 5.4.3 and involve various aspects, such as infrastructure and resource management, data pipelines, data storage and indexing, processing engines and machine learning libraries. The following text describes the most prominent of those solutions.

Apache Hadoop [ApacheHadoop] is essentially a distributed data infrastructure involving a storage component (Hadoop Distributed File System - HDFS) [HDFS], but also a processing component based on the MapReduce approach, distributing massive data collections across multiple nodes, keeping track of that data and enabling big data processing and analytics. HDFS is the most common solution for distributed data infrastructures as it is designed to scale up, while it provides data replication and automatic recovery to ensure data integrity. Other storage approaches involve NoSQL databased such as **Cassandra** [ApacheCassandra], a recent open source standalone non-relational database system that offers continuous availability and pluggable replication strategies across cluster nodes to ensure reliability and fault tolerance.

In terms of data ingestion:

- **Apache Sqoop** [ApacheSqoop] is employed to import bulk data from a relational database into HDFS.
- **Apache Flume** [ApacheFlume] is a distributed service for collecting and aggregating large amounts of streaming data into HDFS (especially "logs").
- **Apache Kafka** [ApacheKafka] for streaming data is designed to provide scalable, high-throughput persistent messaging. Kafka is a solution for parallel data loads into HDFS, as well

as building (a) real-time streaming data pipelines between systems or applications and (b) real-time streaming applications that transform or react to the streams of data.

The latter is currently the most widely spread, scalable and fault tolerant solution and is deployed in real-life applications. It offers APIs for publishing (Producer API), subscribing and processing streams of records (Consumer API), processing and producing output streams (Streams API) and running reusable producers or consumers connected to existing applications or data systems (Connector API). A schematic representation is available in Figure 5.4.4.

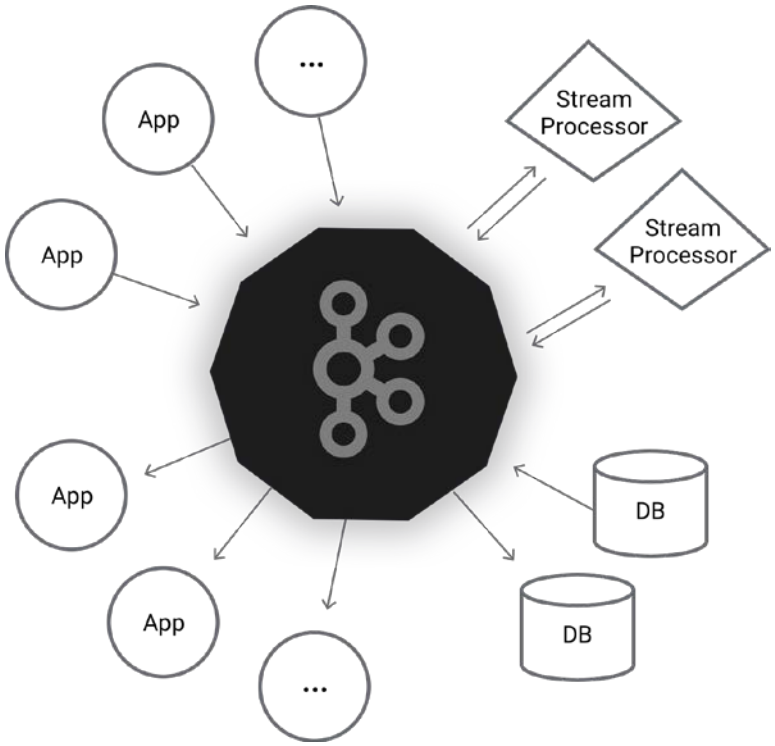


Figure 5.4.4: Schematics presentation of Kafka APIs [ApacheKafka].

The incoming data from the above-mentioned data ingestion mechanisms can be stored and processed at a later time (batch processing) or in real-time through streaming and micro-batch approaches (stream processing). The most common data processing frameworks are mentioned below.

- **Apache Spark** and **Spark Streaming** [ApacheSpark] do not come with a file management system, but can operate on HDFS or another cloud-based data platform. Spark generally works better than the MapReduce approach, as it performs operations in-memory and real-time, while it offers full recovery from faults or failures. Additionally, Spark offers its own machine learning library and is compatible with other libraries that can run within the Spark framework.
- **Apache Flink** [ApacheFlink] is a stream processing framework for distributed, high-performing streaming applications, allowing for fault tolerance over data streams. Flink also bundles libraries for domain-specific use cases: a complex event processing library, a Machine Learning Library and others.
- **Apache Beam** [ApacheBeam] is the programming/ SDK portion of the Dataflow model [Akidau-2015]. A recently graduated Apache Software Foundation incubator project Beam offers a unified programming model for constructing data pipelines executed in multiple computing back-ends, e.g. Spark and Flink Runners are supported along with the corresponding machine learning libraries.

The most popular libraries related with the previously presented processing frameworks are Spark MLib, FlinkML, Deeplearning4J (DL4J) [DL4J] for deep neural nets combined with Keras- TensorFlow [Abadi-2016, KERAS].

Relevant EU projects

Comparing MATILDA with relevant past & on-going EU projects, very few have included functionalities involving advanced data analytics and monitoring. The following ongoing 5G-PPP projects aspire to include a more intelligent orchestration approach based on machine learning modelling.

SelfNet [SelfNet]: SelfNet explores the possibilities for integration of technologies in Software-Defined Networks (SDN), Network Functions Virtualization (NFV), Self-Organizing Networks (SON), Cloud computing, Artificial intelligence, Quality of Experience (QoE), in order to provide efficient self-organized network (SON) management for 5G. It focuses mostly on the SON paradigm in conjunction with use cases referring to Self-Monitoring, Self-Protection, Self-Healing and Self-Optimization. Therefore, this project focuses mostly on the SON paradigm for SDN/NFV Orchestration.

CogNet [COGNET]: CogNet focuses on the application of Machine Learning to resource requirement prediction and autonomic management of resources for NFV, as well as security issues and network resilience, feeding into applications for autonomic network management. This project uses similar ML approaches as the ones to be employed in MATILDA and focuses mostly on network management.

However, compared to the above-mentioned projects, MATILDA has a significantly broader scope aiming to implement an end-to-end operational framework tackling the lifecycle of design, development and orchestration of 5G-ready applications and 5G network services over programmable infrastructure, while at the same time employing state of the art ML mechanisms.

PaaSword [PaaSword]: PaaSword introduces a holistic data privacy and security by design framework enhanced by sophisticated context-aware policy access models and robust policy access, decision, enforcement and governance mechanisms. The goal is to enable the implementation of secure and transparent Cloud-based applications and services that will maintain a fully distributed and totally encrypted data persistence layer, and, thus, will foster customers' data protection, integrity and confidentiality, even in the case wherein there is no control over the underlying third-party Cloud resources utilized. PaaSword plans to extend the CSA Cloud security principles by capitalizing on advances on context-awareness models and policy governance. In particular, PaaSword considers context-aware access control that incorporates the dynamically changing contextual information into novel group policies implementing configurable context-based access control policies and context-dependent access rights to the stored data at various different levels. Furthermore, regarding policy governance, modelling and annotation techniques are employed that allow application developers to specify an appropriate level of protection for the application's data, while the evaluation of whether an incoming request should be granted access to the target data takes dynamically place during application runtime.

While PaaSword directly addresses the critical security issues in cloud technologies, the MATILDA context model will be used in all service lifecycle phases (i.e. development, composition, deployment planning, execution) in order to conceptualize specific aspects of application services that are essential by the architectural components.

Lastly, the MATILDA project will extend the mechanisms and experience derived from ARCADIA [ARCADIA], so as to extend and include more advanced intelligent orchestration functionalities.
For the MATILDA intelligent orchestrator all the extracted measurements will be fed seamlessly in real-time (streaming data) to the analytics toolkit in a reliable fashion using the Kafka distributed

streaming platform [ApacheKafka]. Kafka is used for building real-time data pipelines and streaming apps; it also offers horizontal scalability and, therefore, it will be able to handle the incoming monitoring data and quality metrics expected from multiple sources, i.e. multiple sites, as well as multiple services.

The analytics mechanisms will be based on the IncelliAna, IncelliSim and IncelliOpt software modules developed by Incelligent [Incellligent]. The mechanisms employed will involve machine learning methodologies available in open source projects, as well as proprietary software. In the MATILDA project, various mechanisms –ranging from very simple to highly sophisticated and performance intensive algorithms- will be explored through the above-mentioned modules and the best methods will be included in the MATILDA intelligent orchestration schemes. The most promising algorithms that have shown improved results are deep learning neural nets [LECUN-98, DL4J-ACC], which are incorporated in the Incelligent modules and are supported by DL4J [DL4J], an open-source, distributed deep-learning library -under the Apache 2.0 license- written for Java and Scala, as well as proprietary software. Additionally, DL4J can import models from major frameworks via Keras [KERAS], bridging the gap between the Python ecosystem, and it is designed to be used on distributed GPUs and CPUs. Therefore, combined with the data analytics tools, it is compatible with big data frameworks and can be integrated with Hadoop and Spark.

Apache Spark [ApacheSpark] will be the selected framework for the deployment of predictive and prescriptive analytics, as well as clustering and classification algorithms, and is fully compatible with the above-mentioned libraries and tools. Spark not only offers its own machine learning libraries; it can also perform both batch and stream processing. Furthermore, it has been deployed in production in many companies, offering the appropriate level of technology readiness for the MATILDA project requirements. Lastly, ML mechanisms -ranging from clustering, classifications and regression methods to reinforcement learning methodologies- are going to be applied based on the technologies mentioned above for the production of network level analytics, resources usage analytics, workload prediction, etc., as well as for static and dynamic profiling and driving reconfigurations for meeting policies, objectives and SLAs.

5.4.2 Technology Requirements

ID	Optim_Engine_1
Unique Name/Title	Delivery of Real-Time Deployment Planning
Priority	High
Type	Functional
Brief Description	The MATILDA Optimization Engine should be able to efficiently wade through the incredibly large number of possible solutions and find a near-optimal solution to support the real-time deployment planning, taking under consideration the available programmable resources and the current situation in the infrastructure where these resources reside.
Rationale	Technology review and Use cases.
Validation method/Relevant KPI	Validation based on the required time for adjustment of the Optimization Engine.

ID	Optim_Engine_2
Unique Name/Title	Scalability & Reliability of Optimization Engine
Priority	High
Type	Functional
Brief Description	The MATILDA Optimization Engine should be heavily tested with unit/integration and stress tests to prove that it will be able to support large-

	scale optimization problems with thousands of variables, multiple constraints and numerous constraint matches.
Rationale	Technology review and Use cases.
Validation method/Relevant KPI	Validation based on the maximum number of variables/constraints that can be supported by the Optimization Engine.

ID	Data_Fusion_1
Unique Name/Title	Support of Real-Time Streaming Data
Priority	High
Type	Functional
Brief Description	MATILDA Monitoring & Data Fusion mechanisms should be able to handle parallel data loads from multiple sources and offer real-time streaming data pipelines between systems and components, as well as transform or react to the streams of data.
Rationale	Technology review and Use cases.
Validation method/Relevant KPI	Validation based on extensive testing of pipelines from different sources and of real-time aggregation & filtering.

ID	Data_Fusion_2
Unique Name/Title	Scalability & Reliability of Data Fusion Mechanisms
Priority	High
Type	Functional
Brief Description	MATILDA Monitoring & Data Fusion mechanisms should be able to handle parallel data loads from multiple sources in a scalable, high-throughput persistent fashion.
Rationale	Technology review and Use cases.
Validation method/Relevant KPI	Validation based on guaranteed "at least once" delivery after extensive testing.

ID	Data_Fusion_3
Unique Name/Title	Extraction of advanced insights and events
Priority	High
Type	Non-Functional
Brief Description	The Monitoring & Data Fusion component should support the extraction of advanced insights and events through the processing of the available data.
Rationale	Technology review and Use cases.
Validation method/Relevant KPI	Specification and development of Context Awareness Engine.

ID	Analytics_1
Unique Name/Title	Support of Real-Time Analytics
Priority	High
Type	Functional
Brief Description	MATILDA Analytics and Dynamic Profiling mechanisms must be able to create supervised & unsupervised models for both batch and streaming data, in order to support multiple analysis and dynamic profiling tasks.

Rationale	Technology review and Use cases.
Validation method/Relevant KPI	Validation based on the ability to deploy models using both simulated batch and streaming data.

ID	Analytics_2
Unique Name/Title	Scalability, Reliability & Accuracy of Analytics Mechanisms
Priority	High
Type	Functional
Brief Description	MATILDA Analytics and Dynamic Profiling mechanisms should be able to offer scalable algorithmic deployment, running on multiple nodes and using distributed data sources.
Rationale	Technology review and Use cases.
Validation method/Relevant KPI	Validation based on time required for model development, as well as per-case model metrics, i.e. accuracy, recall, precision, f-score, mean squared error, R-squared, etc.

ID	CAE_1
Unique Name/Title	CAE Infrastructure Metrics
Priority	High
Type	Functional
Brief Description	CAE should obtain infrastructure level metrics from the Monitoring Framework, in order to create facts over the streaming data.
Rationale	Technology review and use cases.
Validation method/Relevant KPI	• Interfaces to monitoring mechanisms implemented and validated. • Infrastructure metrics provided for all services to be analysed. • Error in identification of facts should be less than 10%.

ID	CAE_2
Unique Name/Title	CAE QoE Assurance
Priority	High
Type	Non-Functional
Brief Description	The CAE should interact with the Monitoring Mechanism in order to propose actions to guarantee QoE.
Rationale	Technology review and use cases.
Validation method/Relevant KPI	• Interface to Monitoring Mechanisms to obtain feedback and QoE information. • The latency of the response of CAE should be less than 10 seconds.

6 MATILDA Architectural Approach

6.1 MATILDA Reference Architecture

The MATILDA reference architecture, designed according to the requirements in sections 4 and 5, is depicted on
Figure 6.1.1. As illustrated, the architecture is divided in three distinct layers; namely, ***a)*** the **Development Environment** and **Marketplace**, ***b)*** the ***5G-ready Application Orchestrator*** and **c)** the **Programmable 5G Infrastructure Slicing and Management**. In a nutshell, the development environment is responsible for packaging a cloud-native component in a proper format, making it usable by the Control Plane architectural components. Beyond that, the combination of the components in the form of complex graphs is performed by editors that will be provided in this layer. Cloud-native components and application graphs will be persisted in a marketplace, so as to be searchable by application developers. On the other hand, the logically centralized service mesh control plane is the layer that is responsible for the orchestration, monitoring and policy enforcement of a 5G-enabled application, while the programmable 5G infrastructure slicing and management is responsible for the configuration and management of all underlying resources based on the requirements of the active policy. We will elaborate on these distinct layers in the following sections.

6.1.1 Development Environment and Marketplace

As already mentioned, the scope of the development environment and marketplace is to support all pre-deployment steps of a 5G-enabled application. Such steps include the proper packaging and the proper combination of cloud-native components. The main modules of this layer include the **Component Wrapping Toolkit**, an **Application Graph Editor**, a **Component Repository**, an **Application Graph Repository** and a **Policy Editor**.

Module: Component Development/Wrapping Toolkit

The aim of the Component Wrapping Toolkit is to assist the software developer to wrap cloud native components in a proper format to be publishable in the Component Repository and reusable in the frame of complex Application Graphs. As already thoroughly analysed, every cloud-native component has to comply with a specific metamodel. This metamodel compliance guarantees that a component will be "orchestratable" during its deployment. The wrapping toolkit will provide design-time validation. The component model shall contain all **facets** that are necessary in order for a component to be operational. Such facets include a) **minimum infrastructural requirements**, b) **deployment preferences**, c) **configuration parameters during component initialization**, d) mutable configuration parameters during runtime, e) **exposed and required interfaces**, f) **exposed metrics** and g) **link metrics**.

Module: Application Graph Editor

While the component wrapping toolkit is a developer-centric environment, the aim of the application graph editor is to help a service provider to create application graphs that combine the chainable components which are themselves released by the component wrapping toolkit. The following functionalities that are offered by the application graph editor:

.

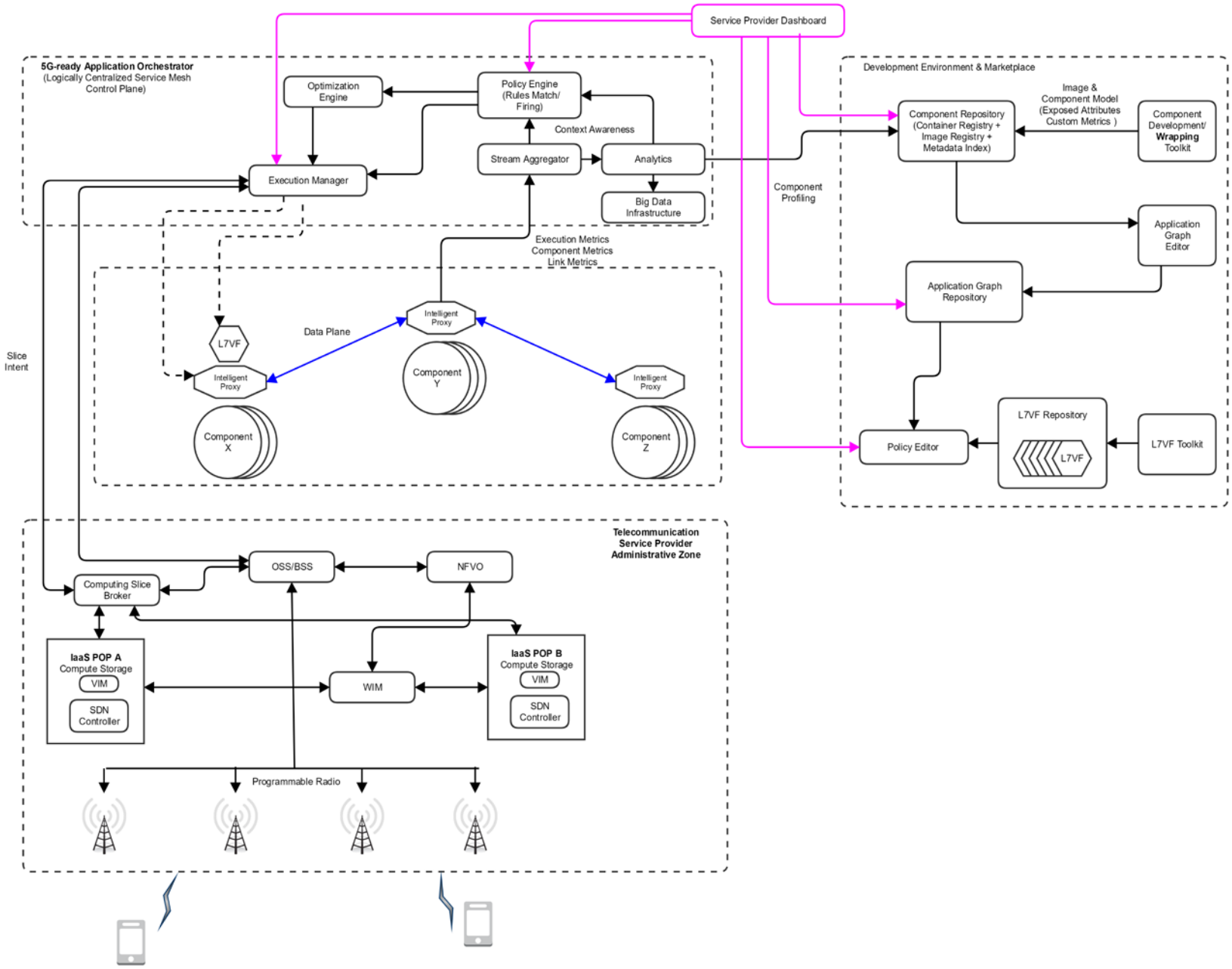


Figure 6.1.1: MATILDA Architecture

a. ***Application Graph Creation and Validation***: When an application graph is formulated, the most critical issue that must be tackled is the **assurance of complementarity** regarding the components that have to be chained. This complementarity is achieved by selecting proper 'binding' interfaces between "**interface-requestors**" and "**interface-publishers**". Furthermore, a crucial aspect of the graph formulation is the definition of graph metrics. As already discussed, the components are accompanied by a specific set of metrics that will be measured during runtime. However, each application graph may be characterized by **additional metrics** that are not exposed by the components (e.g. end-to-end delay). The role of the application graph editor is to **define** these additional metrics, along with the **proper probes** that will quantify these metrics.
b. ***Graph Serialization***: Every application graph that is created should be serialized according to the application graph metamodel. The serialized model is saved to the Application Graph Repository to be either edited or instantiated by a service provider.

Module: Component Repository

This module acts as a persistency layer for the (wrapped) cloud-native components. As a persistency layer, the main functionalities that must be supported are the following:

a. ***Scalable Storage****:* The wrapped components along with their metadata have to be stored in a scalable storage engine. Since the execution-ware technology will rely (for most of the cases) on Linux containers, a scalable container storage engine will undertake the task of storing container images.
b. ***Searchability****:* In order for a component to be identified as a candidate component for chaining, it has to be searchable. Searchability will be achieved using a highly responsive and scalable indexing engine, which will index the entire serialized component model. Upon indexing a component can be identified using keywords or facets.

Module: Application Graph Repository

In an analogous manner to the component repository, the application graph repository is in charge of storing and searching composed application graphs. The significant difference with the component repository is that actual (container) images are not persisted at all, since the composed application graph comprises a serialization model that combines several component models. Images per se are resolvable through references (soft links) that exist in the component model.

Module: Policy Editor

Every application graph that is deployed can be subjected to runtime changes/reconfiguration. This reconfiguration aims to the satisfaction of a set of business goals that are bundled in the form of a Service Level Agreement (SLA). Towards business goals satisfaction, many types of "actions" may be required. Indicative actions include allocating more resources, spawning new instances of cloud-native components, migrating live instances, etc.

All these actions will be "instructed" by proper rules that take under consideration the capabilities of instrumentation (i.e., collection of measurements) and the programmability layer of the virtualized substrate environment. These instructions should be provided in a normative way, so as to be executed upon instantiation of a 5G-enabled application. This specific module will be in charge of authoring these instructions in a formal rules format. Beyond authoring rules, the Policy Editor will be responsible for ensuring the structural validity of provided rules and verifying their applicability. Verification of applicability will be performed by examining the existence of proper "enablers" that will facilitate the execution of a rule.

6.1.2 5G-ready Application Orchestrator

As previously discussed, a 5G-enabled application relies on an **abstracted network layer, which is materialized by a service mesh**. The approach of network abstraction is considered as a state of the art approach from industrial giants of the cloud industry. Indicatively, Google, HP, Red Hat, Twitter have converged to a high-level architecture regarding the network abstraction of cloud-native applications. A **crucial element** of this architecture is the **component-proxying**, i.e. the fact that each cloud-native component is interacting with other components through a proxy. The **interaction between the proxies** constitutes the **data plane**, while the configuration actions of the proxy are based on information gathering that is performed by the proxies per se. The information gathering, along with the actions' enforcement, is addressed as **service mesh control plane**.

Module: Intelligent Proxy

The intelligent proxy is responsible to proxy all inbound and outbound traffic of the cloud-native applications. The proxy undertakes several tasks, such as ***dynamic service discovery***, ***load balancing***, ***TLS termination***, ***circuit breaking***, ***health checking***, ***traffic shaping (Layer 7)***, ***publication of metrics***. In order to perform dynamic service discovery, the proxy assumes that the entire 5G-enabled application is supported by a service registry (such as Consul [Consul]) to keep track of the existing instances. It also assumes that new instances are automatically registered with the service registry and unhealthy instances are automatically removed. Operationally, these functionalities will be covered by the Application Graph Orchestrator.

By combining the knowledge of healthy nodes and existing resources, the orchestrator can achieve load balancing and traffic shaping.

Regarding circuit breaking, it is common for distributed components to make remote calls to components running in different processes, probably on different machines across a network. One of the big differences between in-memory calls and remote calls is that **remote calls can fail, or hang without a response until some timeout limit is reached**. What's worse, if one has many callers on an unresponsive supplier, then one can run out of **critical resources leading to cascading failures across multiple systems**. The circuit breaker pattern has been introduced to prevent this kind of catastrophic cascade. The basic idea behind the circuit breaker is very simple. A protected function call is wrapped in a circuit breaker object, which monitors for failures. Once the failures reach a certain threshold, the circuit breaker trips, and all further calls to the circuit breaker return with an error, without the protected call being made at all. Having this functionality transparent to the invoker is extremely crucial in the frame of MATILDA.

Finally, the Intelligent Proxy will be responsible to **dynamically load and apply layer 7 filters**. Such filters will act as **enablers of several actions**. Thus, during policy formulation, the available filters that can be potentially applied determine the type of layer 7 actions that can be performed. These filters will be hereinafter addressed as **layer-7-VFs** (or simply **L7VFs**). The available filters will be hosted in a special repository within the MATILDA marketplace.

For the sake of implementation, one of the existing open source transparent proxies will be extended to cover the operational needs of the project. Such prominent proxies include Envoy [Envoy] and Linkerd [Linkerd].

Module: Execution Manager

The Application Graph Orchestrator is one of the most crucial modules of the MATILDA framework, since it is the component that interacts with ***a)*** the intelligent proxies, in order to properly configure them; ***b)*** the Policy Engine, which will infer appropriate actions that are required in order for selected policies to be enforced, and ***c)*** the programmability layer of the telecommunications provider.

The basic input to this component is a 5G-enabled application (in the form of an application graph) and one of the applicable policies that are applicable for this graph. The orchestrator is aware of the programmable resources that are "advertised" by the telecommunication provider. The first task that it will perform is infer a deployment plan i.e. take under consideration the constraints that may have been declared by the selected policy and coordinate the deployment of all cloud-native components (see indicative signalling on Figure 6.1.2).

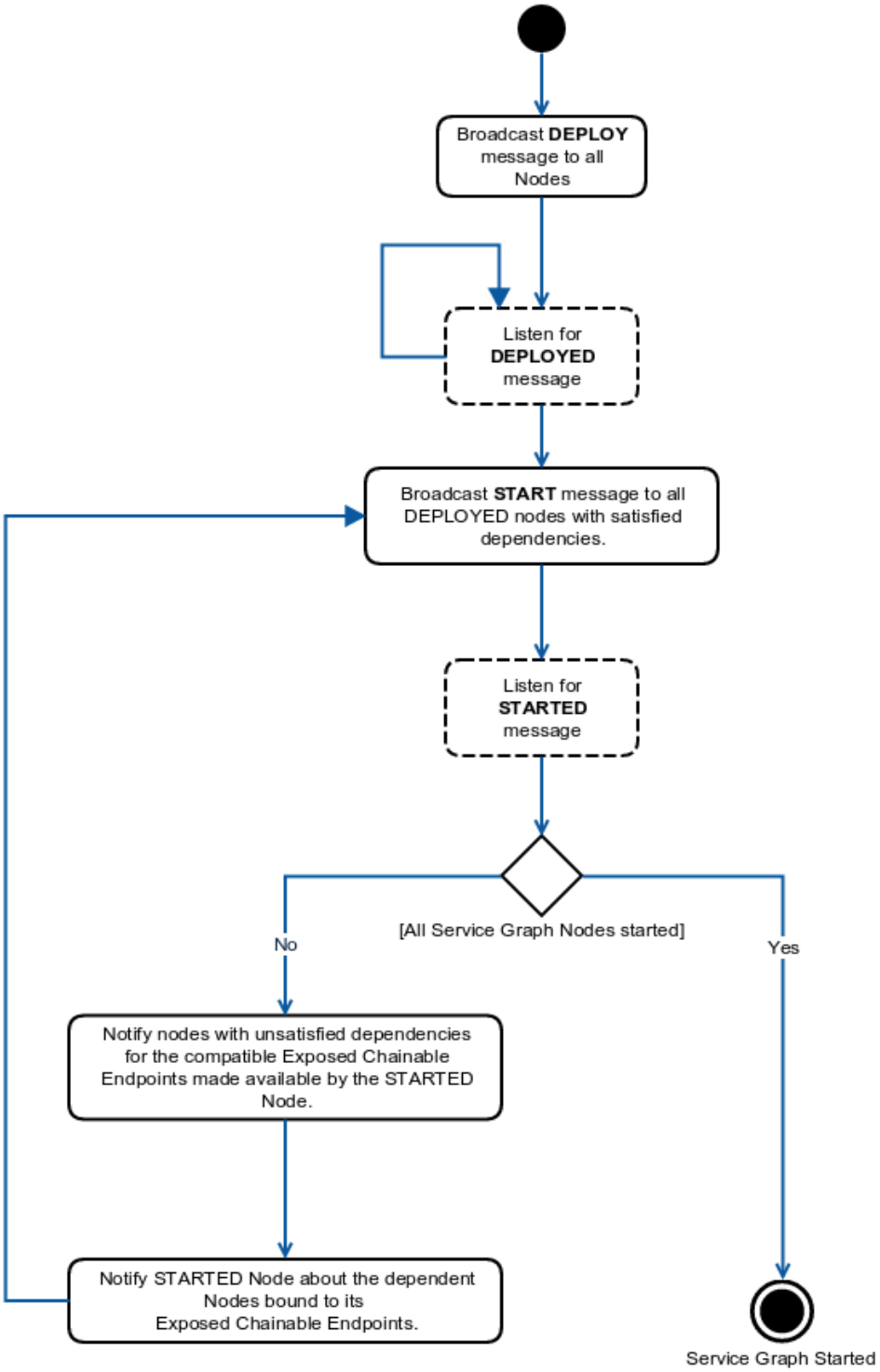


Figure 6.1.2: Indicative deployment business logic.

Upon deployment, the proper configuration of the intelligent proxies will be performed. In the frame of the configuration, some L7VFs may be configured. Finally, the orchestrator will guarantee that the monitoring streams are operational and then it will make a transition to a continuous loop, waiting for any potential actions that are instructed by the Policy Execution Engine.

This continuous loop is broken when the 5G-enabled application has to be undeployed.

Module: Stream Aggregator

Each of the cloud-native components that comprise the 5G-enabled service is deployed on its own execution context. Data monitoring and management processes are supported through a set of active and passive monitoring probes. Such probes collect **a)** information regarding availability and usage of physical resources (compute, storage and network resources) over the programmable infrastructure, **b)** information regarding resource usage per deployed component and **c)** information regarding custom metrics of the deployed application graphs and/or components, as defined in the model. Collection and consumption of monitoring streams is based on a scalable **publish/subscribe framework**, where metrics related to **components**, **application graphs** and infrastructure are provided based on **dedicated topics** in the frame of this module.

Apache Kafka [ApacheKafka] will be used as a pub/sub framework. Kafka is widely used for building real-time data pipelines and streaming apps. It is horizontally scalable, fault-tolerant, extremely fast, and runs in production in thousands of companies. The potential use of a complementary stream processing framework (such as Apache Storm [ApacheStorm]), in order to perform data reduction or alignment of monitoring data, will be examined during the implementation phase. The functionality that has to be performed in real-time fashion relates to the real-time analytics functionality and will be described below.

Module: Analytics

The purpose of this module is to process the aggregated data in order to infer several operational aspects of the running application graph. These aspects may include performance degradation, load prediction, resource usage prediction, component profiling, etc. In order to perform such analysis several types of analytics engines have to be utilized. In general, MATILDA will support **a) real-time** techniques, **b) micro-batching** techniques, and **c) batching** techniques. Real time techniques will be used when there is a hard constraint regarding the outcome of the analysis. Although there is no official definition, streaming frameworks (e.g., Kafka) set the barrier of real-time responsiveness to some milliseconds. At this point it should be clarified that **stream processing techniques are absolutely distinct from batch processing techniques**. A batch processing system **a)** has access to all data, **b)** might **compute** something big and **complex**, **c)** is generally more **concerned with throughput than latency** of individual components of the computation, and **d)** has **latency measured in minutes** or more.

It should also be clarified that beyond stream and batch processing there is an emerging processing type addressed as **micro-batching**. This hybrid approach attempts to offer a general solution for data processing through the provision of various programming paradigms. Such systems provide **a classic batch processing model, which processes all data in-memory and provide streaming capabilities, by buffering the stream in sub-second increments**. These are sent as small fixed datasets for batch processing. In practice, this approach works fairly well, but it does lead to a different performance profile than true stream processing frameworks. The necessity of all these types of processing is imperative in order to achieve advanced network performance and security analytics.
In the frame of MATILDA, the utilized algorithms will be classified in three categories:

- **Regression Analysis Algorithms**: Regression analysis is a statistical process for estimating the relationships among variables. It includes many techniques for modelling and analysing several variables, when the focus is on the relationship between a dependent variable and one or more independent variables (or 'predictors'). There are several algorithms that can be used

such as **linear regression, multiple linear regression**, **non-linear regression**, **logistic regression**, etc. In the frame of MATILDA, such algorithms may be used for **prediction and anomaly detection**, although their usage has substantial overlap with the field of machine learning.

- **Predictive & Prescriptive Analytics Algorithms**: Predictive analytics is a scientific area that deals with extracting information from timestamped data and using it to predict trends and behavioural patterns. Generally, the term predictive analytics is used to mean predictive modelling. In the frame of MATILDA all these models can be used in optimization, maximizing/minimizing certain outcomes.
- **Data Mining & Machine Learning Algorithms**: Generally, data mining is the process of analysing data from different perspectives and summarizing them into useful information. Data mining algorithms allow users to analyse data from **many different dimensions or angles, categorize them, and summarize the relationships identified**. Technically, data mining is the process of finding correlations or patterns among dozens of fields in large data sets. In the frame of MATILDA, machine learning is very crucial for **component profiling**. The purpose of profiling is to identify patterns of resource usage based on the handled load. Component profiling is an important technique for **efficient resource management**. The decision making of scheduling and resource allocation typically takes great advantage of such a technique primarily to improve resource utilization.

Supportive Tools & Big Data Infrastructure

Streaming, micro-batching and batching will be performed with state of the art open source tools. In order to achieve efficiency, effectiveness and scalability for the aforementioned algorithms, **big data technologies are going to be applied**. Big data will be based on the support of a scalable distributing computing framework, along with the integration of a set of natively parallelized algorithms in this framework. Hence, each organization should rely on a **privately hosted big-data infrastructure (e.g., Spark cluster)**, so as to invoke these algorithms. Furthermore, a **map-reduce version of the aforementioned algorithms** is going to be used, especially for massive batch operations.

The necessity of a polyglot persistence engine

With regard to persistence, the analytics algorithms, which are extremely computationally intensive, **require access to the entire dataset** on demand. Some algorithms require efficient querying at the row level and transactional guarantees during writing back to the database. In general, atomicity, consistency, isolation and durability (a.k.a., ACID) guarantees are meaningful in a wide variety of algorithms.

On the other hand, it is proven that **ACID guarantees have a severe penalty, which is linear scalability.** That is the reason why a real SQL-oriented database cannot scale [Brewer-2000]. Therefore, such databases cannot be used as a storage engine for massively increasing datasets (e.g., petabytes of raw TCP/IP packets). The solution to this problem is the utilization of NoSQL repositories. There are many families of NoSQL engines that offer competitive capabilities. These families vary according to the structure of stored information, since some engines support schema-less structures (e.g. MongoDB [MongoDB]), while some others only key-value stores (e.g. Cassandra [ApacheCassandra]). Modern software architectural blueprints employ both types. Relational databases are used where absolute transactional capabilities are needed, while NoSQL engines are used to store non-computing complex data structures that just need to be pulled up when required.

However, these types of storage engines are not enough for a very critical part of analytic algorithms. Many algorithms require extremely efficient computations at the graph-level instead of efficient set-oriented functions (e.g., select *). Such computations entail the pre-requisite of a special organization of the stored information. More specifically, information should be persisted as a directed acyclic graph. These types of persistence engines are addressed as Graph Databases.

In the frame of MATILDA all types of databases will be used. More specifically, an **abstraction layer will be developed** on top of **a)** a high available **transactional relational database**; **b)** a linear scalable key-value store; **c)** a linear scalable schema-less storage; **d)** a scalable graph engine; and **e)** an

index that will be used for efficient searching even if the primary datasets are not persisted. **This abstraction layer will constitute the polyglot storage API and will be used seamlessly by the process templates through the MATILDA development kit.**

Module: Policy Engine

The Policy Engine module is responsible for the enforcement of specific policies over the deployed 5G-enabled applications following a continuous **match-resolve-act** approach. Specifically, the **match** phase regards the mapping of the set of applied rules, which are satisfied based on the **data streams coming from the monitoring infrastructure**, the **resolve** phase regards the process of **conflict resolution -if any- among the satisfied rules** taking into account the pre-defined salience of each rule, while the **act** phase regards the **provision of a set of suggested actions** to the orchestration module. Therefore, policies enforcement is realized through a **rule-based framework** that attempts to derive execution instructions based on the current set of data and the active rules associated with the deployed application graphs at each point in time.

The **rules engine** consists of (i) the **working memory** (WM); facts based on the provided data, (ii) the **production memory** (PM); a set of defined rules, and (iii) an **inference engine** (IE) that supports **reasoning and conflict resolution** over the provided set of facts and rules, as well as triggering of the appropriate actions.

As already mentioned, data is fed to the Policy Execution through the Data Aggregation module, which has the task to collect data based on a set of active monitoring probes, as well as to support a set of data management operations (e.g., calculation of average values in specific time windows).

As already mentioned, the definition of **rules per policy** will be provided by the Policy Editor based on the concepts represented in the Context model. An application graph may be associated with a set of policies; however, **only one can be active** during its deployment and execution time. Each **policy consists of a set of rules**. Each **rule consists** of the **a) expressions** part, denoting a set of conditions to be met and **b) the actions part**, denoting actions upon the fulfilment of the conditions.

Expressions may regard custom metrics of an application graph or a component/microservice. An indicative expression is as follows: “if componentX.avg_cpu_usage is greater than 80%”, and it can be combined in various ways (and/or/grouping/subgrouping) with other expressions. The potential actions of a policy will be classified in three categories, where each one regards: **(i)** the components/microservices **lifecycle management** (e.g., start, stop, destroy), **(ii)** the configuration of the **intelligent proxy**, **(iii)** the management of the underlying resources, i.e. alter slice, etc. Finally, each rule will be associated with a priority indicator based on importance during conflict resolution. A time window is specified per rule for the examination of the provided expressions and the support of inference functionalities during that.

Module: Optimization Engine

The Optimization Engine has the task of **producing results in terms of optimized deployment plans** to support **pro-active adjustments of the running configuration,** as well as **re-active re-configurations of deployments**, based on measurements that derive from the monitoring components of the Applications Orchestrator (Monitoring and Analysis Engine). The ultimate goal of the Optimization Engine is to produce deployment plans to satisfy: i) **zero-service disruption** and ii) **optimal configuration across time**.

Several common pitfalls should be avoided and expressed instead as constraints when placing and assigning resources to application graphs in an optimal way. Such indicative factors to be avoided are **Resource contention**: a single resource or utility is being accessed by two different applications at the same time; **Scarcity of resources**: there are not enough resources to deal with the workload; **Resource fragmentation**: valuable resources lie around in a highly disorganized manner; **Over-**

provisioning: more resources are being assigned than required; and **Under-provisioning**: not adequate resources assigned.

Furthermore, **objectives and satisfiable constraints** should express developers' requirements regarding the deployment and the execution of an application graph, as well as Service Provider's policies and objectives. A **methodology needs to be devised to filter all requirements** and map them to objectives and constraints according to priorities set, or drive the procedure till a deployment model instance is created.

The optimization framework component in fact is the "engine" to solve constrained programming problems. The problems solved by it are related to: i) Producing a deployment plan for the **initial deployment** of an application graph and ii) Producing a **new deployment plan**, incremental, partially or completely different, for an already running application graph. The latter may be triggered either a) to satisfy the scalability needs of an application graph in order to cope with high demand or to sustain quality requirements, b) to prevent/deal with a Service Provider's policy violation or sustain meeting Service Provider's objectives, c) to satisfy the requirements of a newly deployed application graph, which could not be satisfied otherwise.

The Optimization Framework component is in charge of providing fast solutions to the problem of producing deployment plans. A deployment plan indicates which of the available resources are to be reserved and utilized for the execution of an application graph. Such an "embedding" of an application graph to an infrastructure may be considered as an extension of problems already studied in the literature, such as the **Virtual Network Embedding (VNE) problem**, the **Virtual Datacentre Embedding (VDCE) problem** and the **Cloud Application Embedding (CAE) problem.** All these - along with the considered one as an extension of them- fall into the category of **NP-hard** problems. Thus, obtaining a solution to the problem is expected to be computationally intensive and time consuming, especially as the input to the problem gets large. This is not desired in our case where decisions should be taken on the fly, especially when an application graph is in operation mode and continuous operation with quality characteristics should be maintained. This strict restriction may relax when the initial deployment of an application graph (application graph is not yet in operation phase) is considered, since an initial startup latency maybe tolerated. Efficient heuristics have to be devised to produce near optimal solutions within acceptable times.

For the initial deployment plan of an application graph, the input to the considered problem is the **offered infrastructure resources' view, along with resources availability,** pulled from the Resources Manager, while the output is the **resources reservation plan** to be illustrated by the Resources Manager. Actual **instantiation of all the needed components and execution** is handled by the **Execution Manager**. In some cases, an initial deployment may trigger the need for producing several other deployment plans of already running application graphs. A new deployment plan may be needed, as well, during an application's lifetime, in order to meet the objectives and requirements as a decision of the **Stream Aggregation and Analytics Engine** initiated by the **Execution Manager** as a request to the Deployment Manager. In this case, also other reconfigurations for already deployed application graphs may be triggered. When initially producing a deployment plan for an application graph, the problem considered is an online problem; subsequent deployments are not a priori known. When considering running application graphs and seeking a better placement solution, the problem may be considered as static; all the required deployments are already known, and their placement needs to be produced.

6.1.3 5G Programmable Infrastructure Slicing and Management

Up to now, we have examined the operational aspects of the 5G-ready applications, ignoring the infrastructure programmability, which is required in order to support the afore-analysed service mesh. The purpose of this layer is, on one hand, to facilitate the operational requirements of the service mesh and, on the other hand, to provide feedback from the infrastructure, which will be taken under consideration by the policy engine. Thus, in the current section, the proposed approach by

MATILDA for supporting vertical applications placement over programmable infrastructure on behalf of the telecommunication service providers and infrastructure service providers is detailed.

As already mentioned, the operational behaviour of a 5G vertical application is affected by a number of different subsystems owned by different stakeholders, which might act in autonomous fashion, at different layers, and with diverse objectives. As outlined in [3GPP-2017] and depicted in Figure 6.1.3, the main stakeholders actively involved will be three: the *vertical industry* owing the application, the *telecom service provider(s)* (TSPs) offering 5G services, and the *telecom infrastructure provider(s)* providing computing and communication facilities.
The main control and management blocks, acting in the domains of the different stakeholders above, are:

- The *Vertical Application Orchestrator (VAO)*, managing the lifecycle of the graph of microservices composing the application, and acquiring network and computing resources *as-a-Service* from the underlying blocks.
  - The *Business and Operational Support Systems* (BSS/OSS), providing resources of the telecom service providers as-a-Service to vertical industries. According to the latest specifications in 3GPP [3GPP-2017], these modules will expose the 5G network to verticals in terms of 5G network slices. At the southbound, as specified in [ETSINFV-2014b], the BSS/OSS are supposed to interface to the NFV Orchestrator (NFVO) to request the activation/deactivation/modification of NFV services, and to the VNF instances for configuration purposes.
- The *NFV Orchestrator(s)* (NFVO), managing the network services composing the network slices activated by the BSS/OSS.[2]
  - The *Mobile Edge Orchestrator* (MEO), managing the embedding of mobile edge applications, and the management of their lifecycle. As specified in [ETSIMEC-2016b], the MEO is triggered by the BSS/OSS.
- The *Virtualization Infrastructure Manager(s)* (VIM), exposing the resources (especially computing and storage) of PoP datacentres mainly to the NFVO and to the MEO/VAO.
- The *Wide-area Infrastructure Manager(s)* (WIM), realizing the logical interconnectivity among sets of service/application components instantiated in different PoPs and/or towards 5G UE.

It can be noted that the VAO is acting in the Vertical Industry domain; the BSS/OSS and the NFVO in the one of Network Service Providers; the VIM and the WIM in the Infrastructure Provider's domain, respectively. To this end, the aforementioned control systems are usually designed to provide advanced "multi-tenancy" and "multi-domain" capabilities, in the sense that they are designed to host multiple overlying systems to exploit the resources/services from multiple instances of the underlying systems.

While the interdependency between the (MEC and NFV) orchestration layer and VIMs/WAN controllers is already partially in place in cloud legacy systems, and its evolution well-specified by the ETSI NFV working group [ETSINFV-2014b], the integration and reference points and interfaces between the vertical application and network service orchestrators are still a core open issue, where only preliminary design and integration approaches have been proposed.

In order to cope with the main drawbacks of the integration approaches previously introduced, and leverage on their advantages, we designed a third integration scenario. As depicted in Figure 6.1.4, our proposal introduces a further element in the reference architecture, named "Computing Slice Broker" (CSB) and residing in the Telecom Service Provider domain.

[2] For the sake of simplicity, ETSI MANO VNF managers and service/VNF catalogues are meant to be part of the NFVO block.

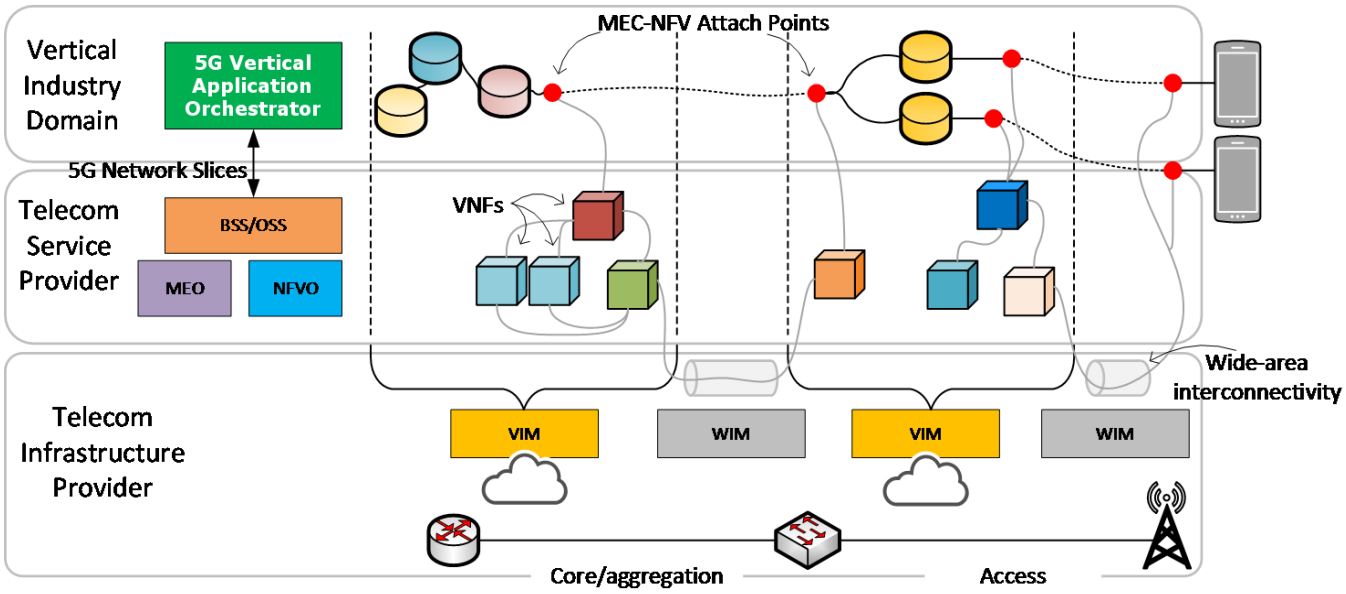


Figure 6.1.3: Example of deployment of a vertical application into a 5G infrastructure, main involved stakeholders and related architectural key building blocks. The picture also shows how application graph links are exploded into sets of VNFs and wide-area interconnectivity means.

The main role of this element is to expose a sort of "virtualized" computing infrastructure to vertical industries, able to host application components without exposing the details of which PoP(s) in the infrastructure will run them. To this end, the broker has to provide a VIM-like interface to the VAO, to enable it to upload images of components and to directly manage their complete lifecycle. Under this perspective, it can be noted that existing projects like OpenStack Tricircle are commonly used to accomplish these goals, since they allow to "proxy" a multi-domain infrastructure into a single IaaS interface.

In addition to those projects, the CSB has to cope with the issue of (dynamic) attach points between the application and the NFV domain. This is accomplished by extending the interface exposed towards the VAO with metadata describing the "relative" position of an application component. In detail, instead of fixing the physical PoP where a component has to run, the metadata model of such interfaces will allow associate the end-points of network slices to components' network ports to which they have to be attached.

The network slice end-point identifiers and related information will be provided by the NFVO and the VNF EMs by means of the OSS or even directly. Differently from the case of standalone orchestrators, direct communication among the CSB, EMs and NFVO is possible, since all these elements are all into the domain of the Telecom Service Provider.

Thus, the Computing Slice Orchestrator will operate by exposing the existing attach points, and creating one "virtual" PoP for each of them. Multiple virtual PoPs can be mapped onto a single physical PoP.

Whenever an attach point moves from one physical PoP to another, the CSB will be in charge of migrating all the application components to the new physical PoP, in a transparent fashion with respect to the VAO. In the case of creation/deletion of a new attach point (e.g., a new UE is activated /deactivated in the network slice, a new configuration of the network slice is requested by the VAO), the CSB will notify this change to the VAO, which, in its turn, will manage the whole lifecycle of application components over such virtual PoPs in a similar fashion with respect to standard operations in today's cloud. The CSB will forward the VAO commands to the right physical VIMs, and it will provide access to any measured performance indexes.

According to this approach, no sensible information of Telecom Service Providers will need to be exposed to third-parties, and vertical industries will not have to deal with multiple (service +

infrastructure) providers, but they will have a single reference 5G provider. Also, handling of notifications will be better managed, since the CSB could be directly interfaced to the NFVO and EMs through standard interfaces internally to the Telecom Service Provider domain, avoiding the overloading of the OSS and BSS modules, and their external interfaces towards the VAO. The last and not least advantage of such solution consists of a better automation of operations to be performed when attach points move.

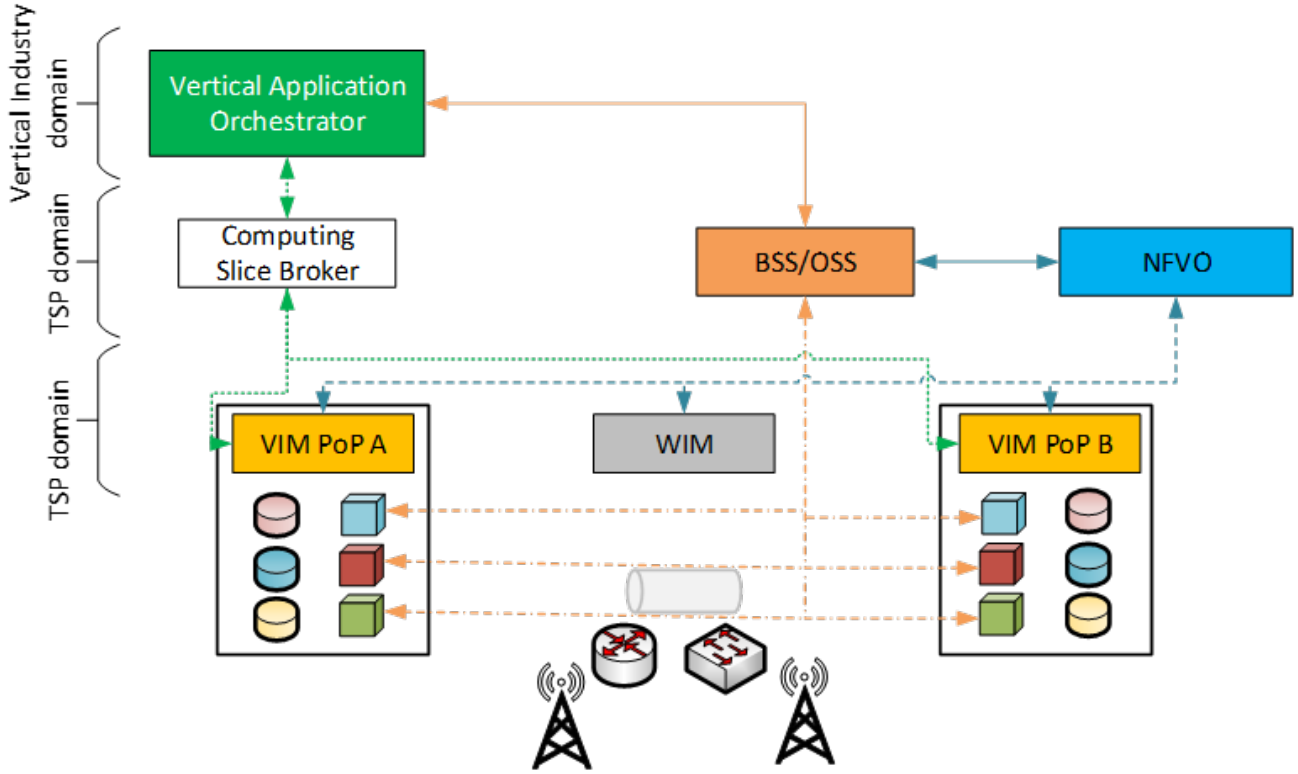


Figure 6.1.4: "MEC Computing Slices" architectural solution.

6.2 Mapping of Requirements to Architectural Components

In this section, a mapping of the requirements provided in section 4 and section 5 to the architectural components detailed in Section 6.1 is provided. The mapping is realised taking into account the priority declared per requirements. The current mapping, as depicted at Table 6.2.1, is going to be revisited throughout the specifications and implementations realised in WP2, WP3, WP4 and WP5, aiming at validation of the successful addressing of all the requirements.

Table 6.2.1: Mapping between requirements and architectural components

ID	Subject	Layer			Relevant Components
		Applications Development	Applications Orchestration	Programmable Infrastructure Slicing and Management	
UC1_1, UC4_2, UC5_1, UC7_7, UC8_6, UC9_6	Network Slicing Capability			√	Computing Slice Manager, OSS/BSS
UC1_2, UC2_1, UC4_3, UC5_2, UC7_1, UC8_1	Adjustable Bandwidth Allocation			√	OSS/BSS, NFVO, WIM
UC1_3, UC2_2, UC4_8, UC5_3, UC7_2, UC8_2,	Low Delay / Latency Guarantees			√	OSS/BSS, NFVO, WIM

UC9_4					
UC1_4	Resource Usage Monitoring		√	√	Stream Aggregator, Monitoring Mechanisms, Policy Manager, VIM
UC1_5	Policy Enforcement		√	√	Policy Manager, NFVO
UC1_6	VNF Scalability (in/out)		√	√	Stream Aggregator, Monitoring Mechanisms, Policy Manager, NFVO, VIM
UC2_3, UC3_19, UC4_1, UC5_5, UC7_3, UC8_3, UC9_1	High Availability and Reliability	√	√	√	All the components
UC2_4, UC3_18, UC4_6, UC6_11, UC7_4, UC8_4, UC9_2	Interoperability with Various Access Networks			√	WIM, NFVO, OSS/BSS
UC2_5, UC3_14, UC3_15, UC4_4, UC5_4, UC6_4, UC9_3	Security & Privacy, Isolation	√	√	√	Applications Orchestrator, Computing Slice Manager, OSS/BSS, NFVO, VIM
UC2_6, UC3_11, UC6_5, UC7_5, UC9_5	Dynamic QoS Provisioning	√	√	√	Applications Orchestrator, Computing Slice Manager, OSS/BSS, NFVO
UC2_7, UC3_17, UC7_6, UC8_5	Network Programmability			√	WIM, NFVO
UC2_8, UC3_8, UC3_12, UC4_7, UC6_8, UC9_9	Network and Service Monitoring		√	√	Stream Aggregator, Monitoring Mechanisms, WIM, OSS/BSS
UC3_1, UC6_9, UC7_8	Distributed application components	√			Development Toolkit, Metamodels
UC3_2, UC4_5, UC5_6, UC6_3, UC7_9, UC9_7	Support vertical/horizontal scalability	√	√	√	Applications Orchestrator, Computing Slice Manager, OSS/BSS, NFVO, VIM
UC3_3	Chainability of components	√			Development Toolkit, Metamodels
UC3_4	Context-based application orchestration	√	√		Development Toolkit, Applications Orchestrator
UC3_5, UC6_6	Policy-based dynamic reconfiguration		√	√	Policy Manager
UC3_6, UC6_7, UC9_8	Redundancy and resilience mechanisms		√	√	Applications Orchestrator, OSS/BSS, NFVO

UC3_7	Support of legacy applications and services	√			Development Toolkit
UC3_9, UC6_10	Support of VNF and PNF	√		√	Development Toolkit, NFVO
UC3_10	Context-based network orchestration	√		√	Development Toolkit, NFVO, OSS/BSS
UC3_13	Networking across IaaS			√	WIM, OSS/BSS
UC3_16	SLA and Service Level Policy	√	√	√	Policy Manager
UC6_1	Low power consumption		√	√	Policy Manager, Computing Slice Manager, OSS/BSS
UC6_2	High density signalling			√	OSS/BSS, NFVO
UC8_7	Optimised Storage Location in Distributed Storage Facilities	√	√		Development Toolkit, Policy Manager, Execution Manager
GEN_1	Modularity	√	√	√	All the components
GEN_2	Extensibility/Upgradability	√	√	√	All the components
GEN_3	Maintainability	√	√	√	All the components
GEN_4	Openness	√	√	√	All the components
GEN_5	User friendliness	√	√	√	All the components
Dev_Tool_1	Applications denoted in the Form of a Service Graph	√			Development Toolkit, Metamodels
Dev_Tool_2	Adherence to the MATILDA metamodels	√			Development Toolkit, Metamodels
Dev_Tool_3	Web Based and Collaborative Development Environment	√			Development Toolkit
Dev_Tool_4	Repositories for Sharing of Developed Software	√			Development Toolkit, Marketplace
Dev_Tool_5	Application Components and VNFs Configurability	√			Development Toolkit, Metamodels
Dev_Tool_6	Application Components and VNFs Chainability	√			Development Toolkit, Metamodels
Dev_Tool_7	Application Components and VNFs QoS Awareness	√			Development Toolkit, Metamodels
Dev_Tool_8	Application Components and VNFs Scalability	√			Development Toolkit, Metamodels
Dev_Tool_9	Infrastructure Agnostic Software Development	√			Development Toolkit, Metamodels
Dev_Tool_10	Application Components and VNFs Performance Profile	√			Development Toolkit, Metamodels
Dev_Tool_11	5G-ready Applications and Network Services Composition through the	√			Development Toolkit, Graph Composer

	Graph Composer				
Dev_Tool_12	Formal Language Expressing Networking Requirements	√			Development Toolkit, Metamodels
Policies_Editor_1	Policies Assigned to 5G-ready Application Graphs	√	√		Development Toolkit, Policy Manager
Policies_Editor_2	Assigning Predefined Policies to 5G-ready Application Graphs	√	√		Development Toolkit, Policy Manager
Policies_Editor_3	Policy Levels for 5G-ready Application Graphs	√	√		Development Toolkit, Policy Manager
Policies_Editor_4	Pushing Service Graph Deployment Policies to the Deployment Manager	√	√		Development Toolkit, Policy Manager, Applications Orchestrator
Policies_Editor_5	Pushing Service Graph Runtime Policies to the Policy Manager	√	√		Development Toolkit, Policy Manager, Applications Orchestrator
MP_1	Graphical User Interface for Stakeholders	√			Marketplace
MP_2	Service Trading from Third Party Developers	√			Marketplace
MP_3	Support of Various, Different Profiles/Functions for Different Users/Stakeholders/Roles	√			Marketplace
MP_4	Authentication, Authorization and Access Control	√			Marketplace
MP_5	Web Access	√			Marketplace
MP_6	Parallel Access and Synchronisation	√			Marketplace
MP_7	Availability				Marketplace
MP_8	Interface GUI - Catalogues	√			Marketplace
MP_9	Interface Marketplace - Orchestrator	√			Marketplace
MP_10	Concurrency and User Isolation (for the Marketplace)				Marketplace
MP_11	App Repository	√			Marketplace
MP_12	VNF Repository	√			Marketplace
MP_13	L7VFs	√			Marketplace
MP_14	Policies Definition	√			Marketplace
MP_15	Repository Operations	√			Marketplace
MP_16	Concurrency and Synchronisation (of Operations on Marketplace Repositories)	√			Marketplace
MP_17	Security/Integrity of Repositories' Data	√			Marketplace
Optim_Engine_1	Delivery of Real-Time Deployment Planning		√		Optimisation Engine
Optim_Engine_2	Scalability & Reliability of Optimization Engine		√		Optimisation Engine
MO_1	5G Network Slices	√	√	√	Computing Slice Manager, OSS/BSS

MO_2	Vertical Applications inside Telecom Infrastructures		√	√	Computing Slice Manager, OSS/BSS
MO_3	Vertical Applications Termination towards 5G Devices		√	√	Computing Slice Manager, OSS/BSS, WIM, NFVO
MO_4	As-a-Service interface for Computing Resources at the Telecom BSS/OSS		√	√	Applications Orchestrator, OSS/BSS
MO_5	Locality and Mobility Awareness			√	OSS/BSS, NFVO
MO_6	Multi-site support			√	Computing Slice Manager, OSS/BSS, WIM, VIM
MO_7	Event Reactiveness		√	√	Policy Manager, Monitoring Mechanisms
Data_Fusion_1	Support of Real-Time Streaming Data		√		Stream Aggregator, Monitoring Mechanisms
Data_Fusion_2	Scalability & Reliability of Data Fusion Mechanisms		√		Data Analytics Toolkit
Data_Fusion_3	Extraction of advanced insights and events		√		Data Analytics Toolkit
Analytics_1	Support of Real-Time Analytics		√		Data Analytics Toolkit
Analytics_2	Scalability, Reliability & Accuracy of Analytics Mechanisms		√		Data Analytics Toolkit
CAE_1	CAE Infrastructure Metrics		√		Context Awareness Engine, Data Analytics Toolkit
CAE_2	CAE QoE Assurance		√		Context Awareness Engine, Data Analytics Toolkit

7 Conclusions

This deliverable presented the overall system architecture of the MATILDA holistic framework for developing, deploying, orchestrating and managing 5G-ready applications and network services over sliced programmable infrastructure.

For the deployment of this architecture, several stakeholders were identified alongside with their associated roles, namely *Infrastructure Provider*, *Application Developer*, *VNF/PNF Developer*, *Service Provider*, *Application Store/Marketplace and Service Consumer*. Different levels of involvement were discussed, as one stakeholder may undertake more than one role depending on the nature of the 5G application and industry vertical. In order to identify the key expectations of the stakeholders regarding MATILDA exploitation and business plans, a survey was conducted with key partners in the Romanian market covering a wide industry spectrum.

Guided by this market prospect, a set of seven important use cases were selected to highlight various verticals:

- **Media and Entertainment** needs were addressed throughout a High-Resolution Media on Demand use case that proposes immersive 5G video services.
- **Automotive Industry** was represented by a Distributed System Testing use case that proposes the usage of an application graph over Wireless Wide Area Networks to interconnect industrial bus signals in geographically distributed facilities.
- **Industrial** verticals that involve **manufacturing** and **logistics** were covered by two Smart Factory use cases that propose 5G-ready applications to support inter and intra enterprise integration.
- **Smart Cities** were illustrated by an Intelligent Lighting System use case that focuses on energy consumption optimization and is based on the infrastructure available in Alba Iulia, a middle size city in Romania.
- **Security and Surveillance** were exemplified by a Mobile Night Safeguard Systems use case involving Ultra-High-Definition Video Streaming/Recording and real-time video transmission.
- **Finance** vertical was represented by a Banking on the Cloud use case that proposes the usage of network slicing and cloud computing techniques to create a more flexible, agile business model that meets the growing business needs in a dynamic and competitive landscape.

Two horizontal use cases that can be shared among several industries were also selected:

- A 5G **Emergency Infrastructure** and Services Orchestration **with Service Level Agreement (SLA)** Enforcement use case that proposes network-aware applications to help emergency response teams both in day-to-day operations and during extreme situations requiring large on-site interventions; and
- A Provisioning of **Distributed Application Services** (e.g. CRM, ERP) use case based on the so called “Mobile/Virtual Office” approach, that uses cloud/edge infrastructure instead of on-premises deployments.

Moreover, existing technology related to the *design and development of 5G-ready applications*, *marketplaces and application/component collaborative repositories*, *multi-site resource management and orchestration* and *intelligent application orchestration* were investigated by reviewing the state of the art and discussing the current capabilities and challenges according to MATILDA vision.

Based on the different requirements elicited by the created use cases and the investigated state of the art, a design based on the concept of "Service Mesh" is proposed, combining the behaviour of cloud-native applications with a powerful abstraction for defining dedicated network infrastructure. This

design foresees not only a development environment and marketplace for 5G-ready applications, its components and VNFs, but also a logically centralized control plane and a programmable 5G infrastructure slicing and management mechanism. The proposed architecture comprises three distinct layers: the Development Environment and Marketplace, the 5G-ready Application Orchestrator, and the Programmable 5G Infrastructure Slicing and Management.

The Development Environment and Marketplace supports all pre-deployment steps of a 5G-enabled application, through proper packaging and combination of cloud-native components. It provides developer-centric tools to create and publish reusable components, an Application Graph Editor that helps service providers to combine them according to a chainable approach, several searchable repositories and a Policy Editor that allows the definition of runtime reactive behaviour.

The 5G-ready Application Orchestrator layer uses component-proxying to materialize a service mesh. The proxies constitute the data-plane and abstract network traffic management aspects, by performing tasks as dynamic service discovery, load balancing, TLS termination, circuit breaking, health checking, traffic shaping (Layer 7), publication of metrics, etc. The service mesh is constantly monitored, analysed and optimised, regarding the installed components and allocated infrastructure, in order to guarantee the optimal usage of resources and enforce that network slice specifications are met.

The Programmable 5G Infrastructure Slicing and Management aims, on one hand, to facilitate the operational demands of service meshes to be handled and, on the other hand, to retrieve feedback from the infrastructure. Therefore, it is responsible for managing the lifecycle of the application graph deployment, acquiring network and computing resources as-a-Service from the underlying blocks, managing the network services that compose the network slices and realizing the logical interconnectivity among geographically distributed points of presence.

The outputs of this deliverable will serve as input for the implementation work that will be performed in the WP2, WP3 and WP4 work packages. Additionally, the referenced metamodels will be further discussed in tasks T1.3, T1.4, T1.5 and T1.6, and described in detail in the deliverables D1.2, D1.3, D1.4 and D1.5. The gathered use cases will be also developed in task T1.7 and presented by the deliverable D1.6.

References

[12Factor]	The Twelve-Factor App. URL: https://www.12factor.net/.
[3GPP-2017]	3GPP, "Study on management and orchestration of network slicing for next generation network," TR 28.801, version 15.0.0, Sept. 2017.
[5GPPP-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
[5GPPP-2016]	5GPPP, "5G for verticals white paper", February 2016. URL: https://ec.europa.eu/digital-single-market/en/news/more-smartphones-white-paper-shows-how-5g-will-transform-eu-manufacturing-health-energy
[Abadi-2003]	D. Abadi, D. Carney, U. Çetintemel, "Aurora: a new model and architecture for data stream management", The VLDB Journal - The Int. Journal on Very Large Data Bases, 12(2):120–139, 2003
[Abadi-2005]	D. Abadi, Y. Ahmad, M. Balazinska, "The design of the borealis stream processing engine", In CIDR, volume 5, pp. 277–289, 2005
[Abadi-2016]	M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, X. Zheng (2016). TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. Retrieved from http://arxiv.org/abs/1603.04467
[ADVA]	ADVA Ensemble Orchestrator. URL: http://www.advaoptical.com/en/products/network-virtualization/ensemble-orchestrator.aspx.
[Akidau-2015]	T. Akidau, R. Bradshaw, C. Chambers, S. Chernyak, R. J. Fernández-Moctezuma, R. Lax, S. McVeety, D. Mills, F. Perry, E. Schmidt, and S. Whittle. 2015. The dataflow model: a practical approach to balancing correctness, latency, and cost in massive-scale, unbounded, out-of-order data processing. Proc. VLDB Endow. 8, 12 (August 2015), 1792-1803. DOI: http://dx.doi.org/10.14778/2824032.2824076
[Andrushko-2017]	D. Andrushko, G. Elkinbard, "What is the best NFV Orchestration platform? A review of OSM, Open-O, CORD, and Cloudify," Open Cloud Digest, Mirantis, March 2017. URL: https://www.mirantis.com/blog/which-nfv-orchestration-platform-best-review-osm-open-o-cord-cloudify/
[ApacheBeam]	Apache Beam. URL: http://beam.apache.org/
[ApacheCassandra]	Apache Cassandra. URL: http://cassandra.apache.org/
[ApacheCloudStack]	Apache CloudStack. URL: https://cloudstack.apache.org/.
[ApacheFlink]	Apache Flink. URL: https://flink.apache.org/
[ApacheFlume]	Apache Flume. URL: https://flume.apache.org/
[ApacheHadoop]	Apache Hadoop. URL: http://hadoop.apache.org/
[ApacheJCloud]	Apache JCloud. URL: https://jclouds.apache.org
[ApacheKafka]	Apache Kafka. URL: http://kafka.apache.org/
[ApacheSpark]	Apache Spark. URL: https://spark.apache.org/
[ApacheSqoop]	Apache Sqoop. URL: http://sqoop.apache.org/

[ApacheStorm]	Apache Storm. URL: http://storm.apache.org/
[ARCADIA]	ARCADIA project. URL: http://www.projectarcadia.eu/
[ARCADIA-D.2.2]	ARCADIA project deliverable, D2.2 Definition of the ARCADIA context model -, Online: http://www.arcadia-framework.eu
[ARCADIA-D.3.1]	ARCADIA project deliverable, D3.1 - Implementation of the discrete components of the Smart Controller. URL: http://www.arcadia-framework.eu
[Blazheski-2016]	F. Blazheski, "Cloud banking or banking in the clouds?", 2016, U.S. Economic Watch, BBVA Research. URL: https://www.bbvaresearch.com/wp-content/uploads/2016/04/Cloud_Banking_or_Banking_in_the_Clouds1.pdf
[Brewer-2000]	E. A. Brewer, "Towards robust distributed systems." PODC. Vol. 7. 2000.
[Brown-2014]	K. Brow and M. Capern, "Top 9 rules for cloud applications". URL: https://www.ibm.com/developerworks/websphere/techjournal/1404_brown/1404_brown.html
[Bruschi-2016]	R. Bruschi, P. Lago, G. Lamanna, C. Lombardo, S. Mangialardi, "OpenVolcano: An Open-Source Software Platform for Fog Computing," in Proc. of 2016 28th Internat. Teletraffic Congress (ITC 28), Wurzburg, Germany, Sept 2016, pp. 22-27.
[Calçado-2017]	P. Calçado, "Pattern: Service Mesh". URL: http://philcalcado.com/2017/08/03/pattern_service_mesh.html
[ChaoHu-2015]	Y. Chao Hu, M. Patel, D. Sabella, N. Sprecher, V. Young, "Mobile Edge Computing A key technology towards 5G," ETSI White Paper, no. 11, 1st edition, Sept. 2015, ISBN: 979-10-92620-08-5, URL: http://www.etsi.org/images/files/ETSIWhitePapers/etsi_wp11_mec_a_key_technology_towards_5g.pdf.
[Chandrasekaran-2003]	S. Chandrasekaran, O. Cooper, A. Deshpande, M. J. Franklin, J. M. Hellerstein, W. Hong, S. Krishnamurthy, S. R. Madden, F. Reiss, and M. A. Shah, "Telegraphcq: continuous dataflow processing", Proceedings of the 2003 ACM SIGMOD Int.Conference on Management of Data, pp. 668–668, ACM, 2003
[Chappell-2015]	C. Chappell, "NFV MANO: What's Wrong & How to Fix It," Heavy Reading Snapshot, vol. 13, no. 2, Feb. 2015.
[Cherniack-2009]	M. Cherniack, H. Balakrishnan, M. Balazinska, D. Carney, U. Çetintemel, Y. Xing, and S. B. Zdonik, "Scalable distributed stream processing", CIDR, 2003
[Chowhury-2009]	N. M. M. K. Chowdhury, R. Boutaba. Network Virtualization: State of the Art and Research Challenges, IEEE Communications Magazine, Jul. 2009.
[CiscoNFVI]	The Cisco NFV Infrastructure. URL: http://www.cisco.com/go/nfvi.
[Cloudify]	Cloudify. URL: http://cloudify.co.
[CloudifyTelco]	Cloudify Telecom Edition. URL: http://cloudify.co/downloads/TelcoEdition.html.
[COGNET]	Cognet project, http://www.cognet.5g-ppp.eu/,
[Consul]	Consul. URL: https://www.consul.io/
[DATA.GOV.RO]	DATA.GOV.RO [Online]. Available: http://data.gov.ro.

[DL4J]	Deep Learning for Java, http://deeplearning4j.org/,
[DL4J-ACC]	Deep Learning's Accuracy, http://deeplearning4j.org/accuracy.html
[Docker Compose]	Docker docs, https://docs.docker.com/compose/startup-order/
[Drools]	Drools Business Rules Management System, Online: https://www.drools.org/
[EclipseChe]	Eclipse Che. URL: https://www.eclipse.org/che/
[Envoy]	Envoy Proxy. URL: https://www.envoyproxy.io/
[EricssonNFVI]	The Ericsson NFVi solution. URL: http://www.ericsson.com/ourportfolio/telecom-operators/nfvi.
[Esper]	Esper, Online http://www.espertech.com/products/esper.php
[ETSIMEC]	The ETSI Multi-access Edge Computing Working Group. URL: http://www.etsi.org/technologies-clusters/technologies/multi-access-edge-computing.
[ETSIMEC-2016a]	ETSI GS MEC 002, "Mobile Edge Computing (MEC); Technical Requirements", version 1.1.1, March 2016. URL: http://www.etsi.org/deliver/etsi_gs/MEC/001_099/002/01.01.01_60/gs_MEC002v010101p.pdf
[ETSIMEC-2016b]	ETSI GS MEC 003, "Mobile Edge Computing (MEC); Framework and Reference Architecture," version 1.1.1, March 2016. URL: http://www.etsi.org/deliver/etsi_gs/MEC/001_099/003/01.01.01_60/gs_MEC003v010101p.pdf.
[ETSINFV-2014a]	ETSI GS NFV-MAN 001, "Network Functions Virtualisation (NFV); Management and Orchestration", version 1.1.1, Dec. 2014. URL: http://www.etsi.org/deliver/etsi_gs/NFV-MAN/001_099/001/01.01.01_60/gs_NFV-MAN001v010101p.pdf
[ETSINFV-2014b]	ETSI GS NFV 002, "Network Functions Virtualisation (NFV); Architectural Framework," version 1.2.1, Dec. 2014. URL: http://www.etsi.org/deliver/etsi_gs/NFV/001_099/002/01.02.01_60/gs_NFV002v010201p.pdf.
[ETSINFV-2014c]	ETSI GS NFV 003, "Terminology for Main Concepts in NFV," version 1.2.1, Dec. 2014. URL: http://www.etsi.org/deliver/etsi_gs/NFV/001_099/003/01.02.01_60/gs_nfv003v010201p.pdf.
[ETSINFV-2015]	ETSI GS NFV-INF, "Network Functions Virtualisation (NFV); Infrastructure Overview", version.1.1, Jan. 2015. URL: http://www.etsi.org/deliver/etsi_gs/NFV-INF/001_099/001/01.01.01_60/gs_nfv-inf001v010101p.pdf.
[Eucalyptus]	The Eucalyptus Cloud-computing Platform. URL: https://github.com/eucalyptus/eucalyptus.
[EUODP]	EUODP, ''European Union Open Data Portal'' [Online]. Available: http://data.europa.eu/euodp/en/data.
[Fernando-2013]	N. Fernando, S. W. Loke, W. Rahayu, "Mobile cloud computing: A survey," In Future Gen. Comp. Systems, vol. 29, no. 1, 2013, pp. 84-106.

[Fischer-2013]	A. Fischer, J. F. Botero, M. T. Beck, H. de Meer, X. Hesselbach. "Virtual Network Embedding: A Survey, Communications Surveys & Tutorials," IEEE (Volume: 15, Issue: 4), Feb. 2013.
[Flinck-2013]	H. Flinck, C. Sartori, A. Andrianov, C. Mannweiler, N. Sprecher, "Network Slicing Management and Orchestration," IETF Internet-Draft, July 2017.
[Gigch-1991]	J. P. van Gigch, "System Design Modeling and Metamodeling", chap. 11, Plenum Press, New York, 1991.
[Gohan]	The Gohan project. URL: http://gohan.cloudwan.io/.
[Gulisano-2012]	V. Gulisano, R. Jimenez-Peris, M. Patiño-Martinez, C. Soriente, P. Valduriez, "StreamCloud: An Elastic and Scalable Data Streaming System", IEEE Transactions on Parallel and Distributed Processing (TPDS), Vol 23, issue 12. pp: 2351-2365, December 2012.
[Hattachi-2015]	R. El Hattachi, J. Erfanian, "NGMN 5G White Paper," Feb. 2015, URL: https://www.ngmn.org/uploads/media/ NGMN_5G_White_Paper_V1_0.pdf
[HDFS]	HDFS Architecture Guide, https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html
[Heat]	Openstack Heat. https://wiki.openstack.org/wiki/Heat.
[Hoban-2017]	A. Hoban, A. Tierno Sepúlveda, F. J. Ramon Salguero, G. García de Blas, K. Kashalkar, M. Ceppi, M. Shuttleworth, M. Harper, R. Velandy, S. Almagia, V. Little, "OSM Release Two: a Technical Overview," ETSI OSM Community White Paper, Ed. C. Buerger, Apr. 2017. URL: https://osm.etsi.org/images/OSM-Whitepaper-TechContent-ReleaseTWO-FINAL.PDF.
[Huang-2015]	C. Huang, D. Mazmanov, Z. Huang, "OpenStack Kingbird," October 2015, specification document, publicly available, URL: https://docs.google.com/document/d/17c8IOjegK _CDUQeDfy0Z_vN-5aahv_Xrsd70AJ8pnoE
[IHS-2017]	IHS Markit, "The 5G Economy: How 5G technology will contribute to the global economy". URL: https://www.ihs.com/Info/0117/5g-technology-global-economy.html
[Incellligent]	Incellligent's Official Website, http://incelligent.net/
[INPUT]	The H2020 INPUT project. URL: http://www.input-project.eu.
[ISO 37120]	ISO 37120, ''Sustainable development in communities - Indicators for city services and quality of life''(ISO 37120: 2014) [Online]. Available: https://www.iso.org/standard/62436.html.
[Juju]	Juju Orchestrator, Online: https://jujucharms.com/
[Kavanagh-2015]	A. Kavanagh, "OpenStack as the API framework for NFV: the benefits, and the extensions needed," Ericsson Review, 2015, no. 3, pp. 1-7. ISSN: 0014-0171.
[KERAS]	Keras: The Python Deep Learning library, https://keras.io/,
[Kingbird]	The OpenStack KingBird project, URL: https://wiki.openstack.org/wiki/Kingbird
[KVM]	Kernel Virtual Machine, https://www.linux-kvm.org/page/Main_Page

[LeCun-1998]	Y. LeCun, L. Bottou, Y. Bengio & P. Haffner, "Gradient-Based Learning Applied to Document Recognition," Proceedings of the IEEE, 86(11):2278-2324, November 1998
[LF-2017]	Linux Foundation, "Harmonizing Open Source and Standards in the Telecom World," Networking and Orchestration White Paper, May 2017.
[Linkerd]	Linkerd, Resilient service mesh for cloud native apps. URL: https://linkerd.io/
[Lorido-2014]	T. Lorido-Botran, J. Miguel-Alonso, J. A. Lozano, "A Review of Auto-scaling Techniques for Elastic Applications in Cloud Environments," Journal of Grid Computing, December 2014, Volume 12, Issue 4, pp. 559-592.
[LSO]	The Metro Ethernet Forum (MEF) Lifecycle Services Orchestration (LSO) specifications. URL: https://www.mef.net/third-network/lifecycle-service-orchestration.
[Manvi-2014]	S. S. Manvi, G. K. Shyam, "Resource management for Infrastructure as a Service (IaaS) in cloud computing: A survey," In J. of Netw. and Computer App., vol. 41, 2014, pp. 424-440.
[Mell-2011]	P. Mell, T. Grance, "The NIST Definition of Cloud Computing," Recommendations of the National Institute of Standards and Technology, Special Publication 800-145, Sept. 2011. URL: http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf.
[Mijumbi-2016a]	R. Mijumbi, J. Serrat, J. l. Gorricho, S. Latre, M. Charalambides and D. Lopez, "Management and orchestration challenges in network functions virtualization," in IEEE Communications Magazine, vol. 54, no. 1, pp. 98-105, January 2016.
[Mijumbi-2016b]	R. Mijumbi, J. Serrat, J. L. Gorricho, N. Bouten, F. De Turck and R. Boutaba, "Network Function Virtualization: State-of-the-Art and Research Challenges," in IEEE Communications Surveys & Tutorials, vol. 18, no. 1, pp. 236-262, Firstquarter 2016.
[MongoDB]	MongoDB. URL: https://www.mongodb.com/
[nodeRED]	https://nodered.org/
[NokiaLiquid]	R. McManus, "Liquid Applications – coming to a cloud near you," Nokia Blog, Sept. 2015, URL: https://blog.networks.nokia.com/telco-cloud/2015/09/07/liquid-applications-coming-to-a-cloud-near-you/
[ONAP]	The Linux Foundation ONAP project. URL: http://www.onap.org.
[ONAPWIKI]	The ONAP Architecture. URL: https://wiki.onap.org/display/DW/Architecture.
[Openbaton]	The OpenBaton project. URL: https://openbaton.github.io/.
[OpenL]	OpenL Tablets, Online: http://openl-tablets.org/
[OpenNebula]	The OpenNebula project. URL: https://opennebula.org/.
[OpenRules]	Open Rules Business Rules Management System, Online: http://openrules.com/index.htm
[OpenStack]	The Openstack project. https://www.openstack.org/.

[OpenVIM]	The Open Vim Project, URL: https://github.com/nfvlabs/openvim
[OpenVolcano]	The Open Volcano open-source project. URL: http://www.openvolcano.org.
[Optaplanner]	Optaplanner, Online: https://www.optaplanner.org/
[OSM]	The Open Source Mano project, URL: https://osm.etsi.org/
[Perez-2014]	J.F. Perez, G. Casale, S. Pacheco-Sanchez. Estimating Computational Requirements in Multi-Threaded Applications, IEEE Transactions on Software Engineering, December 2014, Volume 41, Issue 3, pp. 264-278.
[Philips-2017]	Philips Lighting, World Council of City Data, "The Citywide Benefits of Smart & Connected Public Lighting" report assessed through ISO 37120, 2017 [Online]. Available: http://news.dataforcities.org/2017/03/wccd-and-philips-lighting-publication.html.
[Puppet]	https://puppet.com/
[Ren-2010]	G. Ren, E. Tune, T. Moseley, Y. Shi, S. Rus, R. Hundt. "Google-Wide Profiling: A Continuous Profiling Infrastructure for Datacentres," IEEE Micro (2010), pp. 65-79.
[ResourceSupply-2008]	Resource Supply "IP67, What Does That Mean?," 2008. URL: http://www.resourcesupplyllc.com/PDFs/WhatDoesIP67Mean.pdf
[Riera-2016]	J. F. Riera, J. Batall, F. Liberati, A. Giuseppi, A. Pietrabissa, A. Ceselli, A. Petrini, M. Trubian, P. Papadimitrou, D. Dietrich, A. Ramos, and J. Meli, "TeNOR : Steps Towards an Orchestration Platform for Multi-PoP NFV Deployment," in Proc. of the 2016 2nd IEEE Conference on Network Softwarisation (NetSoft), 2016
[Roy-2011]	N. Roy, A. Dubey, A. Gokhale, "Efficient Autoscaling in the Cloud using Predictive Models for Workload Forecasting," 2011 IEEE 4th International Conference on Cloud Computing.
[rule_eng]	http://www.manageability.org/blog/stuff/rule_engines
[SAMBA]	Standard Windows interoperability suite of programs for Linux and Unix. URL: https://www.samba.org
[Samdanis-2016]	K. Samdanis, X. Costa-Pérez, V. Sciancalepore, "From network sharing to multi-tenancy: The 5G network slice broker," IEEE Comm. Magazine, vol. 54, no. 7, pp. 32-39, July 2016
[SaSCh]	Digital Services for shaping agile Supply Chains Project. URL: http://www.biba.uni-bremen.de/dbm/pdf/projects/sasch_ger.pdf
[SDxCentral-2017a]	SDxCentral, "2017 NFV Report Series Part I: Foundations of NFV: NFV Infrastructure and VIM," Market Report, Rev. A, Apr. 2017. URL: https://www.sdxcentral.com/reports/nfv-infrastructure-vim-download-2017/.
[SDxCentral-2017b]	SDxCentral, "2017 NFV Report Series Part 2: Orchestrating NFV – MANO and Service Assurance," Market Report, Rev. A, Apr. 2017. URL: https://www.sdxcentral.com/reports/nfv-mano-and-service-assurance-download-2017/.
[Shah-2003]	A. Shah, J. M. Hellerstein, S. Chandrasekaran, and M. J. Franklin, "Flux: An adaptive partitioning operator for continuous query systems,", ICDE, pp. 25–36, 2003.

[SelfNet]	EU 5G-PPP SELFNET Project, “SELFNET: Framework for Self-Organised Network Management in Virtualized and Software Defined Networks” (H2020-ICT-2014-2/671672) [Online]. Available: https://selfnet-5g.eu/.
[Shahzadi-2017]	S. Shahzadi, M. Iqbal, Z. U. Qayyum and T. Dagiuklas, "Infrastructure as a service (IaaS): A comparative performance analysis of open-source cloud platforms," 2017 IEEE 22nd Internat. Ws. on Computer Aided Modeling and Design of Comm. Links and Netw. (CAMAD), Lund, Sweden, 2017, pp. 1-6.
[Shanhe-2015]	Shanhe Yi, Cheng Li, and Qun Li, “A Survey of Fog Computing: Concepts, Applications and Issues,” In Proc. of the 2015 ACM Ws. on Mobile Big Data (Mobidata '15), pp. 37-42.
[Soldani-2015]	D. Soldani and A. Manzalini, "Horizon 2020 and Beyond: On the 5G Operating System for a True Digital Society," in IEEE Vehicular Technology Magazine, vol. 10, no. 1, pp. 32-42, March 2015.
[SONATA]	Sonata Project. URL: http://sonata-nfv.eu/
[Sotomayor-2009]	B. Sotomayor, R. S. Montero, I. M. Llorente and I. Foster, “Virtual Infrastructure Management in Private and Hybrid Clouds,” in IEEE Internet Computing, vol. 13, no. 5, pp. 14-22, Sept.-Oct. 2009. DOI: 10.1109/MIC.2009.119.
[Syncthing]	Open Source Continuous File Synchronization. URL: http://forum.syncthing.net/
[Szabo-2015]	D. Szabó, F. Németh, B. Sonkoly, A. Gulyás, and F. H.P. Fitzek. “Towards the 5G Revolution: A Software Defined Network Architecture Exploiting Network Coding as a Service,” In Proc. of the 2015 ACM SIGCOMM Conf., pp. 105-106, Aug. 2015.
[Tacker]	Tacker, Online: https://wiki.openstack.org/wiki/Tacker
[Taleb-2017]	T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta and D. Sabella, “On Multi-Access Edge Computing: A Survey of the Emerging 5G Network Edge Cloud Architecture and Orchestration,” in IEEE Communications Surveys & Tutorials, vol. 19, no. 3, pp. 1657-1681, third quarter 2017.
[TCS]	The Tata Cloud OpenVNFManager project. URL: https://github.com/TCS-TelcoCloud/OpenVNFManager.
[Thalanany-2016]	S. Thalanany, P. Hedman, “Description of Network Slicing Concept,” NGMN 5G P1 Requirements & Architecture, Work Stream End-to-End Architecture, version 1.0.8, Sept. 2016. URL: https://www.ngmn.org/uploads/media/161010_NGMN_Network_Slicing_framework_v1.0.8.pdf.
[TMF]	https://projects.tmforum.org/wiki/pages/viewpage.action?pageId=71177320
[T-NOVA]	EU project T-NOVA, Online: http://www.t-nova.eu/
[T-NOVA-2015]	T-NOVA project - Deliverable 3.41: Service Provisioning, Management and Monitoring – Interim, December 2015
[TOSCA]	OASIS Topology and Orchestration Specification for Cloud Applications (TOSCA), Online: https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=tosca

[TricircleWIKI]	Tricircle, OpenStack Wiki, URL: https://wiki.openstack.org/wiki/Tricircle
[US DATA.GOV]	US DATA.GOV, ''The Home of the U.S. Government's open data'' [Online]. Available: https://www.data.gov.
[vCloud]	The VMWare vCloud NFV Platform. URL: http://www.vmware.com/industry/telco/overview.
[Vilalta-2017]	R. Vilalta et al., "TelcoFog: A Unified Flexible Fog and Cloud Computing Architecture for 5G Networks," in IEEE Communications Magazine, vol. 55, no. 8, pp. 36-43, 2017.
[Virtuora]	Fujitsu, "Virtuora Product Suite". URL: http://www.fujitsu.com/us/products/network/products/virtuora/
[VITAL]	VITAL Project. URL: http://www.project-vital.eu/en/live-demo/
[Vogel-2016]	A. Vogel, D. Griebler, C. A. F. Maron, C. Schepke and L. G. Fernandes, "Private IaaS Clouds: A Comparative Analysis of OpenNebula, CloudStack and OpenStack," 2016 24th Euromicro Internat. Conf. on Parallel, Distributed, and Network-Based Processing (PDP), Heraklion, 2016, pp. 672-679.
[Weerasiri-2012]	D. Weerasiri, M. Chai Barukh, B. Benatallah, Q. Z. Sheng, R. Ranjan, "A Taxonomy and Survey of Cloud Resource Orchestration Techniques," in ACM Comput. Surv., vol. 50, no. 2, Article 26, May 2017. DOI: 10.1145/3054177.
[Xu-2012]	Kuai Xu, Feng Wang, Lin Gu, "Profiling-as-a-Service in Multi-Tenant Cloud Computing Environments," The International Workshop on Security and Privacy in Cloud Computing, Macau, China, June 18-21 2012.
[YANG]	http://www.yang-central.org/twiki/bin/view/Main/WebHome
[Zhani-2013]	M. F. Zhani, Q. Zhang, G. Simon and R. Boutaba. "VDC Planner: Dynamic Migration-Aware Virtual Datacentre Embedding for Clouds," IFIP/IEEE International Symposium on Integrated Network Management (IM 2013), May 2013.
[Zhou-2016]	X. Zhou, R. Li, T. Chen, H. Zhang, "Network Slicing as a Service: Enabling Enterprises' Own Software-Defined Cellular Networks," IEEE Comm. Magazine, vol. 54, no. 7, pp. 146-153, July 2016.

Annex 1: Orange Romania Survey Questionnaire

The following images were extracted from a questionnaire distributed by Orange Romania among its clients (potential stakeholders for MATILDA), as part of a survey.

5G Questionnaire – Gathering Business Requirement

The purpose of this questionnaire is to gather user requirements and expectation in the context of the new 5G technology introduction.

Orange Romania Mission

Orange Romania(*www.orange.ro*) is continuously involved in the development and the founding of new business opportunities, leveraging on the all new 5G ecosystem. Orange anticipates the implementation of the new 5G technology by being present in two projects, SLICENET (*www.slicenet.eu*) and MATILDA, part of EC H2020 research and innovation actions, and also in Working Groups that sustain the standardization for 5G. Orange strongly believes that 5G will be rather revolutionary that evolutionary and will disrupt the market as we know it. We are moving from the era in which the Telco provider is providing ubiquitous connectivity to the era in which the operator provides ubiquitous services and application sustained through strategic partnerships with the relevant stakeholders (application providers, verticals, etc.).

5G short overview:

5G technologies have the potential to offer ubiquitous access, rapid provision of end-to-end services and increased data speed. 5G also adds a key technological capability: definition of a dedicated and adaptive virtualized network and IT infrastructure with embedded security tailored to the need of the business (higher bandwidth, lower latency and reduced jitter) through the whole lifecycle.
This truly enables the digital transformation that will provide to the society unprecedented capabilities for communication supporting very high bit rates, low latencies, huge device densities, ready for cloud-based services, virtual reality, augmented reality, artificial intelligence, factory automation, utilities, agriculture, self-driving and self-slice network management. 5G will enable new use cases through new business models and verticals with a fundamental reduction in overall costs and operational efficiency increase, making the technology sustainable through enabling value creation for customers.
The 5G Key Performance Indicators are the following:

- ✓ 1000 times higher mobile data volume per geographical area.
- ✓ 10 to 100 times more connected devices.
- ✓ 10 times to 100 times higher typical user data rate.
- ✓ 10 times lower energy consumption.
- ✓ End-to-End latency of < 1ms.
- ✓ Ubiquitous 5G access including in low density areas

5G is intended to deliver solutions, architectures and technologies for the coming decades with huge potential of creating new markets, business models and innovation opportunities and actions in areas such as Smart Cities, e-Health, Intelligent Transport, Education, Agriculture, Media and Entertainment.
A European Commission[1] study reveal that the benefits of 5G for automotive, healthcare, transport and utilities sector in Europe starting from 2025 are estimated at 113 billion per year.

[1] Identification and quantification of key socio-economic data to support strategic planning for the introduction of 5G in Europe, A study prepared for the European Commission DG Communications Networks, Content & Technology

Figure A1.1: Stakeholder questionnaire, page 1

Questionnaire:

1. **Company description and activities, yearly turnover estimate:**

Turnover: [_____]

Select number of employees:

<10 10 ÷ 100

101÷1000 >1000

2. **Communication solution:**
   a. Please select the key features for the communication solution in order to meet your current or/and future needs:
      - Connections with guaranteed quality of service (e.g., bandwidth, delay, jitter)
      - Connections with best effort quality of service
      - Connections with rapid provision of new services
      - Connections with ultra-low latency (< 10 ms)
      - Connections with broadband access everywhere
      - Connections with high user mobility
      - Connections with ultra-reliable communications
      - Others: please fill below

   b. Please select the communication solution(s) you have in place or intend to use:
      - Internet/Intranet for office use (daily business)
      - Wi-Fi Hot Spots for visitors Internet Access
      - VPN solution with encryption capabilities
      - Smart City Platform (please further specify the application(s), e.g., smart lighting, smart metering, smart grid etc.)
      - M2M communication platform
      - e-Health or mobile health applications
      - Others: please fill in below

Figure A1.2: Stakeholder questionnaire, page 2

Please add here any other relevant details in case of "Smart City Platform", "M2M communication platform", "e-Health"or "Others"

3. Please rate the relevance of the 5G technology for your business:

1 2 3 4 5

1) Very low, as we don't see any 5G business opportunities

2) Low, as we may consider 5G business opportunities

3) Medium, as we expect 5G business opportunities

4) High, as we shall consider 5G as an enabler for our business development

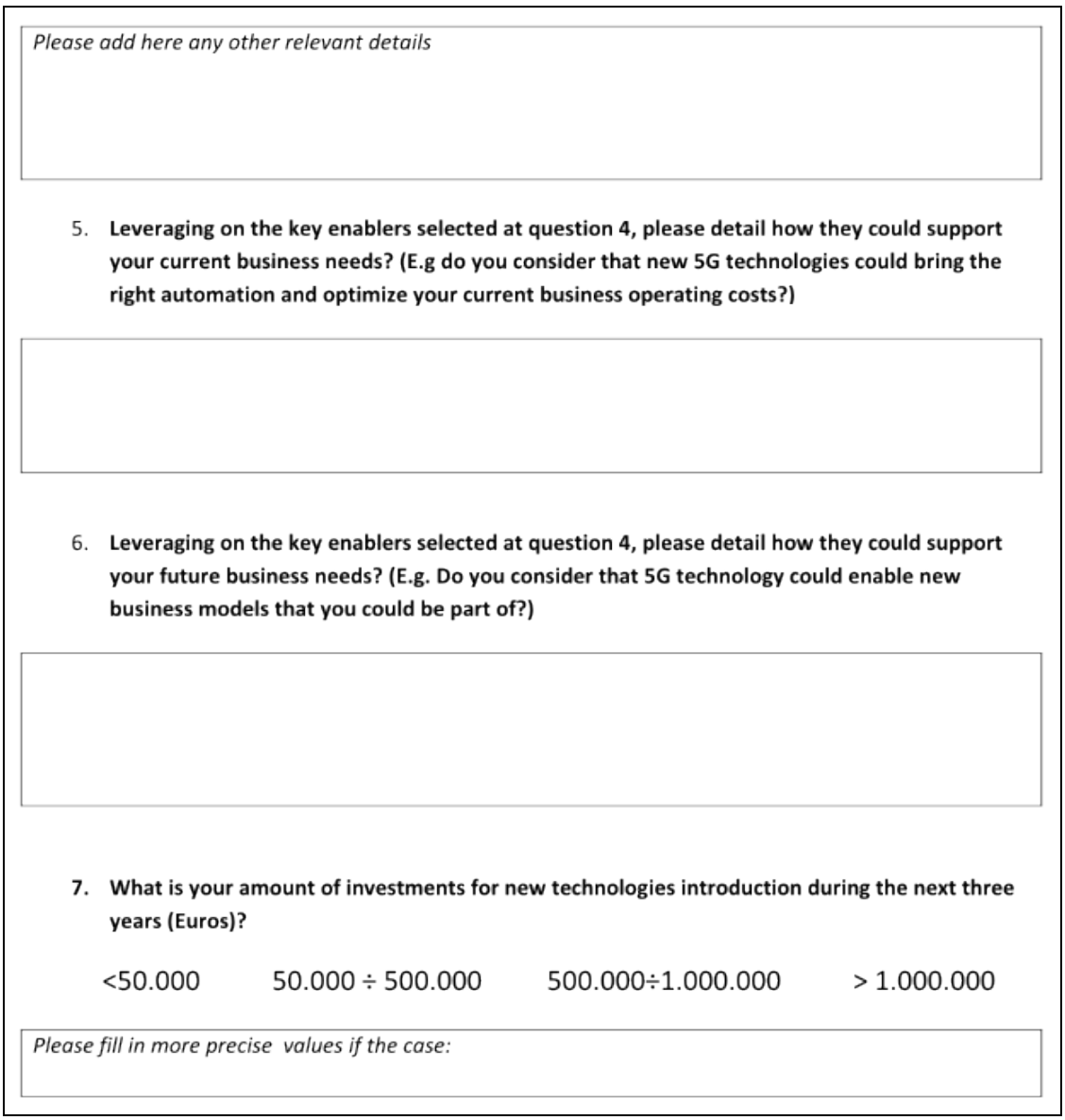
5) Very High, as we will contribute to 5G services definition

Please respond to questions 4-7 if you rated with grades 3-5 question 3

4. Thinking of your current and future business which is the key enabler for migrating towards 5G solutions?

	Degree of importance			
	Not important	**Low**	**Medium**	**High**
Cost efficiency				
new business opportunities				
Opportunity to enhance your current business model				
Faster time to market for your products				
No real key enabler, it will be market trend				

Figure A1.3: Stakeholder questionnaire, page 3



Please add here any other relevant details

5. **Leveraging on the key enablers selected at question 4, please detail how they could support your current business needs? (E.g do you consider that new 5G technologies could bring the right automation and optimize your current business operating costs?)**

6. **Leveraging on the key enablers selected at question 4, please detail how they could support your future business needs? (E.g. Do you consider that 5G technology could enable new business models that you could be part of?)**

7. **What is your amount of investments for new technologies introduction during the next three years (Euros)?**

<50.000 50.000 ÷ 500.000 500.000÷1.000.000 > 1.000.000

Please fill in more precise values if the case:

Figure A1.4: Stakeholder questionnaire, page 4

The questionnaire used in this survey is structured in 3 parts:

1. Overview of Orange Romania mission in the context of 5G
2. Overview of 5G main benefits
3. Set of question to be answered by the interviewees which are depicted in Table A.1 together with their relevance.

Table A.1: 5G Survey Questionnaire and relevance

Question	Relevance
1. Company description and activities, yearly turnover estimate, number of employees	Size of the company & domain in which the company is present
2. Communication solution: a. Please select the key features for the communication solution in order to meet your current or/and future needs b. Please select the communication solution(s) you have in place or intend to use	a. Understand which are the most relevant 5G performance advantages for the interviewees b. Overview of the status quo in terms of communication solution.
3. Please rate the relevance of the 5G technology for your business.	Rate the relevance of 5G technology for the interviewee (criteria for passing forward)
4. Thinking of your current and future business which is the key enabler for migrating towards 5G solutions?	Understand the business rationale for the interviewee to migrate to 5G technologies
5. Leveraging on the key enablers selected at question 4, please detail how they could support your current business needs? (E.g do you consider that new 5G technologies could bring the right automation and optimize your current business operating costs?)	Complement the business rationale of question 4 with details in order to extract better the advantages the interviewees expect from 5G technology for their current business.
6. Leveraging on the key enablers selected at question 4, please detail how they could support your future business needs? (E.g. Do you consider that 5G technology could enable new business models that you could be part of?)	Complement the business rationale of question 4 with details in order to extract better the advantages the interviewees expect from 5G technology for their future business needs.
7. What is your amount of investments for new technologies introduction during the next three years (Euros)?	Understand the potential of investing in new technologies (not necessarily 5G)

The questionnaire was sent to different businesses in Romania that are active in various domains. The name of the business partner is not disclosed; however, Table A.2 depicts the domains and few details about the company objectives. To clarify, in Table A.2, only the companies that responded to the survey are included. Also, it is relevant to note that the survey was run in parallel using three communication channels: email, phone and face to face meetings. This assures that each of the companies had all the details and clarification needed in order to provide relevant responses for the survey. This action also was important towards disseminating the MATILDA project itself, as there were several discussions with all these key partners in order to clarify the main benefits that 5G technology could provide.

Table A.2: Group of responders

Responder	Domain	Objective
Company 1	Cyber Security	Provides a variety of services including penetration testing, trainings and competitions.
Company 2	Device Integrator	Provides integration services to businesses mainly based on Cisco equipment
Company 3	Cyber Security	Provides penetration tests as a service
Company 4	Equipment Provider	Multinational company offering full solutions for FTTA (fiber to the antenna) connectivity, overvoltage protection and outdoor power cabinets for telecommunication operators and OEM companies.
Company 5	Automotive	Multinational company providing automotive parts
Company 6	e-Health	Researching and developing different 3D machinery for additive manufacturing with plastic, metal and biomass.
Company 7	Air Transport Services	Flight operator owning aircrafts and part of International Air Transport Association (IATA)
Company 8	Smart City	Service Consumer for Smart City use case

Table A.3 depicts the gathered responses. As summarized in Figure A.1, more than 60% of the responders consider that the following features are key in the context of 5G technology:

- Connection with high user mobility
- Connection with broadband access everywhere
- Connection with guaranteed quality of service

Table A.3 Questionnaire centralized responses

	Comp. 1	Comp. 2	Comp. 3	Comp. 4	Comp. 5	Comp. 6	Comp. 7	Comp. 8
1. Company description and activities, yearly turnover estimate, number of employees	small enterprise	medium enterprise	small enterprise	large enterprise	large enterprise	small enterprise	large enterprise	public institution
2a. Please select the key features for the communication solution in order to meet your								
Connections with guaranteed quality of service	yes	yes		yes	yes	yes		yes
Connections with best effort quality of service						yes		
Connections with rapid provision of new services						yes		
Connections with ultra-low latency (< 10 ms)					yes	yes	yes	
Connections with broadband access everywhere		yes	yes	yes		yes	yes	yes
Connections with high user mobility				yes	yes	yes	yes	yes
Connections with ultra-reliable communications				yes	yes			
Others: please fill below								
2 b. Please select the communication solution(s) you have in place or intend to use:								
Internet/Intranet for office use (daily business)	yes	yes	yes	yes	yes	yes	yes	yes
Wi-Fi Hot Spots for visitors Internet Access	yes	yes		yes	yes	yes	yes	yes
VPN solution with encryption capabilities	yes			yes	yes	yes	yes	yes
Smart City Platform								yes
M2M communication platform		yes		yes		yes		
e-Health or mobile health applications						yes		
Others: please fill in below								
3. Please rate the relevance of the 5G	1	4	1	4	5	5	3	3
4. Thinking of your current and future business which is the key enabler for migrating towards 5G								
Cost efficiency		medium		medium	N/A	medium	medium	medium
new business opportunities		high		high	high	high	low	high
enhance current business model		medium		high	N/A	high	medium	medium
Faster time to market		medium		medium	N/A	high	low	low
No real key enabler, market trend		N/A		not important	N/A	high	not important	medium
5. Leveraging on the key enablers selected at question 4, please detail how they could support your current business needs? (E.g do you consider that new 5G technologies could bring the right automation and optimize your current		low latency, high bit rate		M2M communication enhancements	N/A	remote operation of medical equipment	robotic process automation	N/A
6. Leveraging on the key enablers selected at question 4, please detail how they could support your future business needs? (E.g. Do you consider that 5G technology could enable new		virtualization		smart cities	autonomous driving	working collaborative on medical equipment	N/a	N/A
7. What is your amount of investments for new technologies introduction during the next three years (Euros)?		50.000-500.000		500.000-1.000.000	over 1mEur	50.000-500.000	50.000-500.000	<50.000

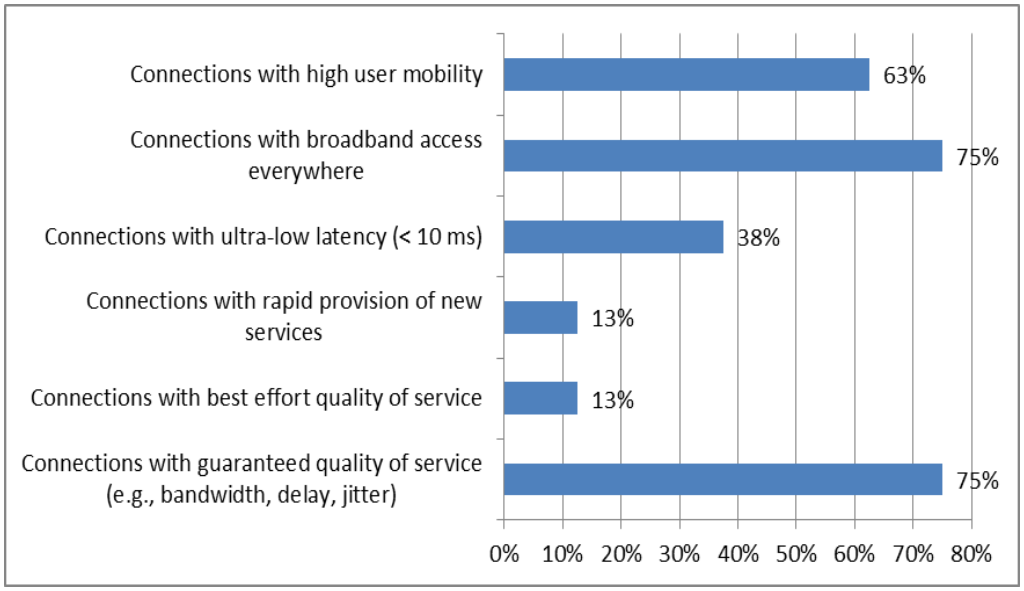


Figure A.1: Key features and relevance for responders (% of responders selected the key features)

Out of 8 responders only 6 considered that 5G technology will be relevant for their business, hence the results presented hereinafter refer to these 6 companies. Both companies responding that 5G technology is not relevant are small enterprises active in the domain of cyber security, mainly handling penetration tests. In this context, they do not consider that 5G shall change or affect their operating model, which is not necessarily true considering the high degree of virtualization that 5G brings.

The 6 responders graded on an importance scale (not important, low, medium, high) the key enablers for migrating to 5G solutions. The statistics can be found in the graphs presented below in Figure A.2, Figure A.3, Figure A.4 and Figure A.5.

Figure A.2: 5G Key enabler – business model & cost efficiency statistics

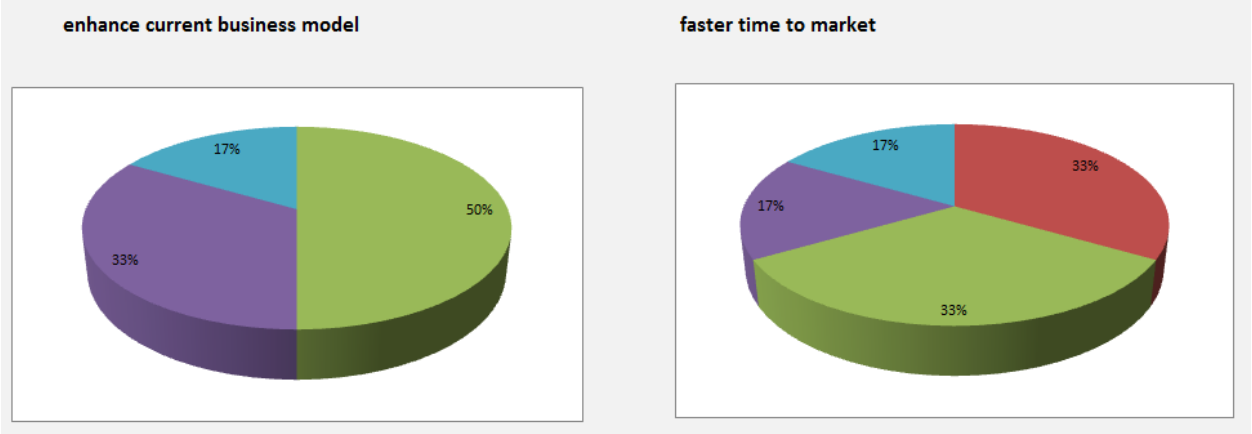


Figure A.3: 5G Key enabler – enhanced business model & time to market

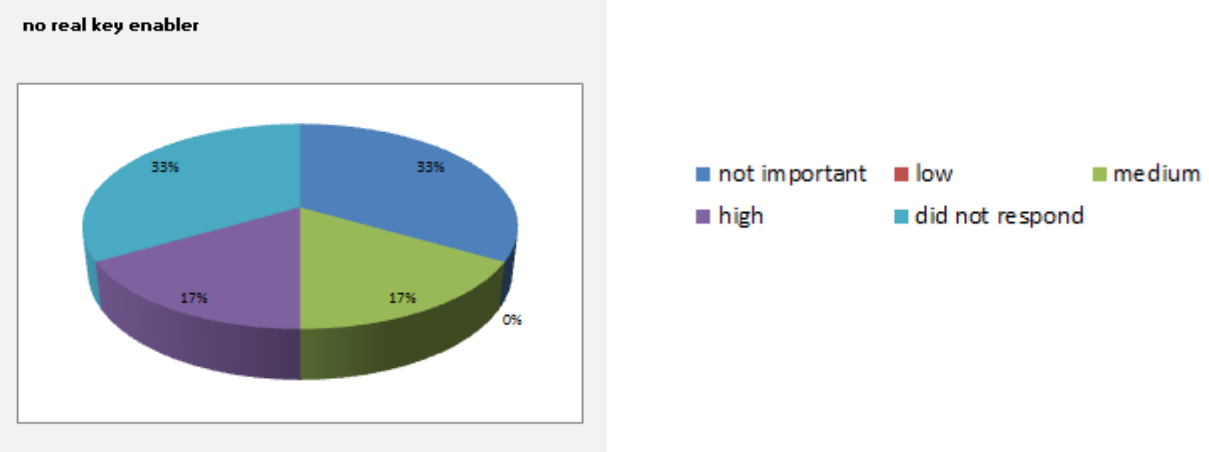


Figure A.4: 5G market trend relevance

Over 80% of the responders consider that 5G will enable new business models and hence new lines of revenues. This is key for a business in a competitive market; it could be observed that this enabler was preferred to the detriment of cost efficiency. Leveraging on 5G technology, the responders see a high potential in the following areas: virtualization, smart cities, autonomous driving, collaborative working on medical equipment (regardless of the location).

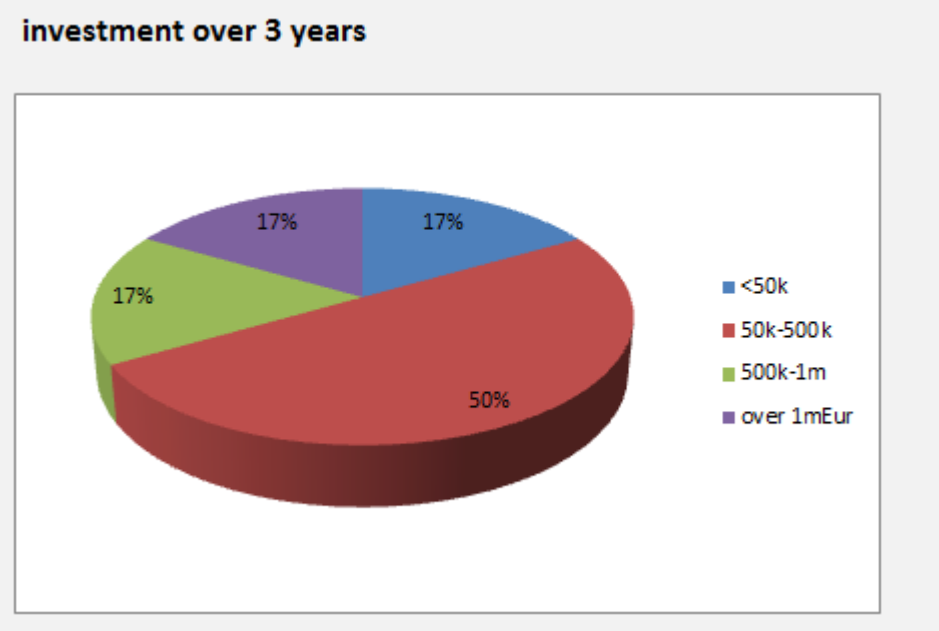


Figure A.5: New technology investments estimation

In terms of investments, the responders are reserved, only one considers an investment of over 1M Eur. However, this should not be taken as reference, as there are many unknown aspects at this point in time related to costs and solutions.