



5TH GENERATION END-TO-END NETWORK, EXPERIMENTATION, SYSTEM INTEGRATION, AND SHOWCASING

[H2020 - Grant Agreement No. 815178]

Deliverable D3.7

Open APIs, service level functions and interfaces for verticals (Release A)

Editor	ATOS
Contributors	ATOS (Atos Spain SA), UMA (Universidad de Málaga), LMI (L.M. Ericsson Limited), UNIS (University of Surrey), INTEL (INTEL Deutschland GmbH), NEMERGENT (Nemergent Solutions SL)
Version	1.0
Date	October 15 th , 2019
Distribution	PUBLIC (PU)



List of Authors

ATOS	ATOS SPAIN
Elisa Jimeno, Javier Melian, Sonia Castro	
UMA	UNIVERSIDAD DE MALAGA
Almudena Díaz Zayas, Bruno García	
LMI	L.M. ERICSSON LIMITED
Erik Aumayr, Anne-Marie Bosneag	
UNIS	UNIVERSITY OF SURREY
Seiamak Vahid, Klaus Moessner	
NEM	NEMERGENT SOLUTIONS
Egoitz Alonso	
INT	INTEL DEUTSCHLAND GMBH
Valerio Frascolla	

Disclaimer

The information, documentation and figures available in this deliverable are written by the 5GENESIS Consortium partners under EC co-financing (project H2020-ICT-815178) 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

Copyright © 2019 the 5GENESIS Consortium. All rights reserved.

The 5GENESIS Consortium consists of:

NATIONAL CENTER FOR SCIENTIFIC RESEARCH "DEMOKRITOS"	Greece
AIRBUS DS SLC	France
ATHONET SRL	Italy
ATOS SPAIN SA	Spain
AVANTI HYLAS 2 CYPRUS LIMITED	Cyprus
AYUNTAMIENTO DE MALAGA	Spain
COSMOTE KINITES TILEPIKOINONIES AE	Greece
EURECOM	France
FOGUS INNOVATIONS & SERVICES P.C.	Greece
FON TECHNOLOGY SL	Spain
FRAUNHOFER GESELLSCHAFT ZUR FOERDERUNG DER ANGEWANDTEN FORSCHUNG E.V.	Germany
IHP GMBH – INNOVATIONS FOR HIGH PERFORMANCE MICROELECTRONICS/LEIBNIZ- INSTITUT FUER INNOVATIVE MIKROELEKTRONIK	Germany
INFOLYSIS P.C.	Greece
INSTITUTO DE TELECOMUNICACOES	Portugal
INTEL DEUTSCHLAND GMBH	Germany
KARLSTADS UNIVERSITET	Sweden
L.M. ERICSSON LIMITED	Ireland
MARAN (UK) LIMITED	UK
MUNICIPALITY OF EGALEO	Greece
NEMERGENT SOLUTIONS S.L.	Spain
ONEACCESS	France
PRIMETEL PLC	Cyprus
RUNEL NGMT LTD	Israel
SIMULA RESEARCH LABORATORY AS	Norway
SPACE HELLAS (CYPRUS) LTD	Cyprus
TELEFONICA INVESTIGACION Y DESARROLLO SA	Spain
UNIVERSIDAD DE MALAGA	Spain
UNIVERSITAT POLITECNICA DE VALENCIA	Spain
UNIVERSITY OF SURREY	UK

This document may not be copied, reproduced or modified in whole or in part for any purpose without written permission from the 5GENESIS Consortium. In addition to such written permission to copy, reproduce or modify this document in whole or part, an acknowledgement of the authors of the document and all applicable portions of the copyright notice must be clearly referenced.

Version History

Rev. N	Description	Author	Date
1.0	Release of D3.7	Elisa Jimeno, Javier Melian, Sonia Castro, Almudena Díaz Zayas	15/10/2019

LIST OF ACRONYMS

Acronym	Meaning
3GPP	3rd Generation Partnership Project
5G	5th Generation (of mobile communications)
API	Application Programming Interface
B2B	Business to Business
BSS	Business Support System
GA	Grant Agreement
CA	Consortium Agreement
CAPIF	Common API Framework
Dx.y	Deliverable Number
DB	Data Base
ELCM	Experiment Life cycle Manager
EPS	Evolved Packet system i.e. 4G, LTE
E-UTRA	Evolved Universal Terrestrial Radio Access
ID	IDentificator
IT	Information Technology
КРІ	Key Performance Indicators
MANO	Management and Orchestration
MBMS	Multimedia Broadcast Multicast Services
NFV	Network Function Virtualization
NFVO	NFV Orchestrator
NRF	Network Repository Function
NS	Network Service
NSD	Network Service Descriptor
Mx	Month # of the project work plan (e.g. M2)
NEF	Network Exposure Function
NR	New radio
OSM	Open Source MANO
OSS	Operational Support System
REST	Representational State Transfer
SBA	Service Based Architecture
SBI	South-Bound Interface
SCEF	Service Capability Exposure Function
SMF	Session Management Function
Tx.y	Task Number

TR	Technical Report
TS	Technical Specification
UI	User Interface
UE	User Equipment
VNF	Virtual Network Function
VNFD	Virtual Network Function Descriptor
WPx	Work Package Number

Executive Summary

Although the main goal of 5GENESIS is to validate 5G KPIs for various 5G use cases, it also aims at offering to the broadest possible audience (research community and industry, but especially 5G verticals) a Facility to execute 5G and beyond experimentations.

The 5GENESIS facility is composed of five experimental Platforms with complementary features, distributed across Europe. Each one of these Platforms follows a common reference implementation architecture, which exposes an open Application Programming Interface (API) with the goal of offering experimenters an open and common method to interface with the Facility.

The focus of this deliverable is to present the design of such an open API (architecture, exposed features, and interfaces), its initial implementation resulting in Release A and the planned roadmap for its evolution.

The 5GENESIS open API is the main interface for experimenters to define and execute their experiments. However, 5GENESIS also provides a Portal with a friendly Web User Interface (UI) to make interacting with the Facility even easier. Such Portal it plays itself the role of client of the 5GENESIS open API and it is able to display the execution logging output for all execution stages of the experiments (Pre-Run, Run and Post-Run). Besides, for each experiment execution, it provides a link to a Grafana customized experiment-specific dashboard for easy visualization of the data generated by the experiment. Release A of the Portal, as well as its future extensions roadmap, is presented in this deliverable. The Dispatcher and the Validator are also key components in Task 3.5 as shown in Figure 1. The Dispatcher is the component that exposes the open API, while the Validator check the format of the descriptors sent to the platform via the open API.





In summary, this document presents the two ways the experimenters can interact with the 5GENESIS facility, either via the Portal (an abstraction layer and client example of the open API), or directly using the open API.

Table of Contents

LIST OF ACRONYMS	6
1. INTRODUCTION	12
1.1. Purpose of the document	
1.2. Document dependencies	
1.3. Structure of the document	
1.4. Target audience	
2. State of the Art	14
2.1. 3GPP Approach	14
2.1.1. 3GPP Common API Framework	14
2.1.2. 3GPP 5G Service Based Architecture APIs	14
2.2. TM Forum approach	15
2.2.1. TM Forum Ecosystem	15
3. 5GENESIS OPEN API	17
3.1. How to interact with the 5GENESIS Facility	
3.2. Requirements and specifications	17
3.2.1. Requirements	
3.2.2. Specifications	
3.2.2.1. Communication Interfaces	
3.2.2.2. Exposed Functionalities	
3.3. Architecture	
3.4. Components Description	
3.4.1. Dispatcher	
3.4.2. Validator	
3.5. Interfaces Description (ATOS)	
3.5.1. Operations with VNFs and NSs	21
3.5.2. Operations with Experiments	
3.5.3. Operations with Analytics and Monitoring	
3.5.4. Operations with the Privacy / Security Manager	
4. 5GENESIS PORTAL	
4.1. Portal Design	
4.1.1. Portal Users	
4.1.2. Experiment creation	

4.1.3. VNF/NS Management
4.1.4. Execution Logs
4.1.5. Execution Results
4.2. Portal Implementation
4.2.1. Portal internal data model
5. RELEASE A SUMMARY AND FUTURE PLANS
5.1. Release A
5.1.1. Open API 40
5.1.2. Portal
5.2. Release B
5.2.1. Open API
5.2.2. Portal
5.3. Release C
5.3.1. Open API
5.3.2. Portal
6. CONCLUSIONS
REFERENCES
ANNEX 1: PORTAL DATABASE
ANNEX 2: NS DESCRIPTORS
ANNEX 3: EXPERIMENT LOGS
ANNEX 4: EXPERIMENT DESCRIPTOR

1. INTRODUCTION

1.1. Purpose of the document

This deliverable is the first one of a series of two reports on *open API, service-level functions and interfaces for verticals* that will be delivered by the project consortium throughout the project lifetime. The next report, *D3.8,* will be delivered in M30.

The purpose of these two deliverables is to present the exposed open API by the 5GENESIS Facility with the goal of offering, especially to 5G vertical industries, an open and common method for experimentation.

This deliverable includes the design of the open API, the initial implementation (Release A) and the foreseen roadmap for its evolution. It also includes initial design, implementation and foreseen roadmap of the web Portal, which is an alternative method for experimenters to access the 5GENESIS facility; its goal is easing the experimenters' work when using the facility.

D3.8 will present the final version of both the open API and the Portal developed by the project.

The work described in both mentioned deliverables corresponds to the task *T3.4 Open API, service-level functions and interfaces for verticals* included in the *WP3 Openness Framework and Integral Components of the Facility.* The two deliverables are complemented with other 14 ones, corresponding to each of the other tasks included in WP3, that will be also delivered in M15 and M30. All these documents together will provide a complete overview of the work and results delivered by WP3 during the first nine months of its duration.

1.2. Document dependencies

This document is based on specifications, requirements and assumptions as discussed in the first release of the Architecture related deliverables. The table below summarizes the relevance towards the deliverables produced by WP2.

id	Document title	Relevance
D2.1 [1]	Requirements of the Facility	The document sets the ground for the first set of requirements related to supported features at the testbed for the facilitation of the Use Cases.
D2.2 [2]	5GENESIS Overall Facility Design and Specifications	The 5GENESIS facility architecture is defined in this document. The list of functional components to be deployed in each testbed is defined.
D2.3 [3]	Initial planning of tests and experimentation	Testing and experimentation specifications that influence the testbed definition, operation and maintenance are defined.

Table 1: Document dependencies

1.3. Structure of the document

The deliverable is organized in the following manner:

- Section 1 (this section) is an introduction to the deliverable.
- Section 2 provides an analysis of the state of the art of the open APIs world.
- Section 3 is the main section of the deliverable as it presents the 5GENESIS open API, its role, requirements and specifications, and architecture, including a description of its components and interfaces.
- Section 4 introduces the 5GENESIS Portal, a friendly Web UI provided by the Facility to facilitate for experimenters the definition, execution and analysis of the results of their experiments. Its design and current implementation are described.
- Section 5 presents the roadmap for each of the three Releases defined by the 5GENESIS project, for both the open API and the Portal implementation.
- Section 6 provides the conclusions and hint at future work.

1.4. Target audience

This document provides information regarding the open API and the Portal that are being implemented in 5GENESIS with the goal of offering an open and common method for experimentation. In this context, the document is targeted to experimenters, especially to 5G vertical industries, that could use the 5GENESIS infrastructure to run their experiments.

2. STATE OF THE ART

Technologies and concepts such as virtualization, cloud computing, Internet of Things, functionality exposure and self-organizing networks are central building blocks of 5G and will allow seamless communication as well as enable synergies between different industries. In order to expose network functionalities to third-parties, open APIs are used to enable rapid, repeatable, and flexible integration among operations and management systems. This way, making it easier to create, build and operate complex digital services. The following sections provide a brief overview of the two main approaches adopted in telecommunications industry, namely the 3GPP Common API Framework (CAPIF) framework [4], and the TM forum open API programme [5].

2.1. 3GPP Approach

In 3GPP, there are multiple northbound API-related specifications, e.g. APIs for Service Capability Exposure Function (SCEF) functionalities, as defined in TS 23.682 [6], API for the interface between a Multimedia Broadcast Multicast Services Service Provider and a Broadcast Multicast Service Center, as defined in TR 26.981 [7]. To avoid duplication and inconsistency of approach between different API specifications, 3GPP considered the development of a CAPIF that includes common aspects applicable to any northbound service API.

2.1.1. 3GPP Common API Framework

The TS 23.222 "Common API Framework for 3GPP Northbound APIs" [3] specifies the architecture, procedures and information flows necessary for the CAPIF. The aspects of this specification include identifying architecture requirements for the CAPIF (e.g. registration, discovery, identity management) that are applicable to any service APIs when used by northbound entities, as well as any interactions between the CAPIF and the service APIs themselves. The common API framework applies to both Evolved Packet and 5G systems and is independent of the underlying 3GPP access network.

2.1.2. 3GPP 5G Service Based Architecture APIs

In 5G, the RESTful design of API, is used in the 5G Service Based Architecture (SBA) [8]. The basic principles of modern API development are used to address the specific needs of the 5G Core Network as well as application functions requirements. All the different APIs have been defined along a number of common principles which are referred to as REST architectural style [9].

In order to expose network functionlaities to third-parties and system internal communication, the 3GPP chose to make use of the widely established REST architecture design paradigm [10].

Capability exposure¹, i.e. the capability of making 5G Core Network functionalities available to third-parties such as service providers and vertical industries outside the operator's domain, is provided by the application functions, more specifically the Network Exposure Function (NEF). The interface provided by the NEF is defined in a way that it fully aligns with widely accepted design principles for such exposure interfaces.

The so-called RESTful APIs can be understood as an example of 3GPP commitment to tightly integrate 5G in the current and upcoming digital ecosystem of their customer. The APIs offered to third-parties, also known as "northbound APIs" are only applicable to a single interface of the 5G system, whilst the NEF is one of many Network Functions within the completely redesigned 5G Core Network. This redesigned core is the architectural and technical realization of the service-based change design of the 5G system and therefore was named SBA. The 3GPP also decided to use RESTful APIs not only for third-parties functionality exposure but also for the southbound Interfaces (SBI). Therefore the 5G Core Network internal communication obeys the same principles as the functional exposure, thus providing a harmonized approach of the complete 5G system, fully in-line with the progressive paradigms which are at the heart of a wide range of services used by end-customers as well as for the automation of industries.

The 3GPP TS 29.501 [11] specification provides design principles for the 5G Core SBI APIs.

2.2. TM Forum approach

The TM Forum's open API program [15] is a global initiative to enable end-to-end seamless connectivity, interoperability, and portability across complex services. The program has created an open API suite of 50+ REST (Representational State Transfer) based open APIs.

2.2.1. TM Forum Ecosystem

The TM Forum REST based APIs are technology agnostic and can be used in any digital service scenario, including B2B value fabrics, Internet of Things, Smart Health, Smart Grid, Big Data, NFV, Next Generation OSS/BSS, etc.

The TM Forum has brought different stakeholders from across industries to work together to create the APIs. Up to the date, 52 of the world's leading service providers and technology ecosystem participants have signed the open API Manifesto demonstrating their endorsement of TM Forum's suite of Open APIs.

¹ The 3GPP TS 23.501 [8] defines the the system architecture requirements as well as the roles of service consumer and service producer. The service consumer is the VNF which requests the related service, whilst the VNF which exposes the requested service is the service producer. Services exposed by a VNF are further structured into service operations, as defined in 3GPP TS 23.501 [8] and 3GPP TS 29.501 [11]. The basic procedures and flows are defined in 3GPP TS 23.502 [12][1], whilst the more detailed flows and protocol elements which are exchanged, are specified in the related NF specifications, e.g. for the Network Repository Function (NRF) in 3GPP TS 29.510 [13] and for the Session Management Function (SMF) in 3GPP TS 29.502 [14].

The signatories to the manifesto have committed to using the TM Forum open APIs in relevant product applications and to continue to provide feedbacks and extensions so to ensure continuous growth of the community in support of these industry agreed APIs [5].

3. 5GENESIS OPEN API

3.1. How to interact with the 5GENESIS Facility

The 5GENESIS Facility is based on five complementary experimentation Platforms, which expose their underlying capabilities in a harmonised way, as described in deliverable D2.2, which explains the 5GENESIS common reference architecture. This defined 5GENESIS reference architecture exposes an open API with the goal of offering verticals an open and common method to interface with the Facility for experimentation.

The 5GENESIS open API is the interface offered by the coordination layer to experimenters for the definition and execution of the experiments. However, 5GENESIS also provides them with a Portal with a friendly Web Interface to facilitate their task even more. The Portal itself uses the 5GENESIS open API to communicate with the coordination layer component.

Therefore, there are two ways in which the experimenters can interact with the 5GENESIS Facility, (1) via the Portal and/or (2) using the open API directly in the case of more advanced users of the Platform or experienced verticals.

3.2. Requirements and specifications

One of the major goals of 5GENESIS is to offer a common way for experimenters and 5G verticals to interface with the Facility. As defined in Deliverables D2.1 and D2.2, the open API is thus an essential functional requirement to connect experimenters and 5G verticals with the Facility in a consistent way, as shown in Figure 2.





The open API is the interface between the 5GENESIS coordination layer and the experimenter. The coordination layer performs the overall coordination of the experiments on the Platform, achieving overall supervision and end-to-end configuration for service deployment and end-user's management and monitoring. The open API will allow the exposure of these activities towards the experimenters.

3.2.1. Requirements

The foremost requirement of the open API is to present a common and open method of interaction between the 5G vertical or the experimenter – both commercial as well as experimental UEs - and the 5GENESIS facility in order to:

- Gain access to the facility
- Flexibly define experiments
- Configure the experiments
- Conduct experiments
- Get information about the status of experiments
- Retrieve the experimental results.

3.2.2. Specifications

3.2.2.1. Communication Interfaces

The open API is the interface between the experimenter or Portal and the dispatcher in the coordination layer. As detailed in Section 3.4.1, the dispatcher main role is to coordinate with the underlying components to offer towards the upper layer a uniform view on authenticating users, as well as running and managing experiments.

Therefore, the dispatcher will interface with the other components through internal interfaces, as follows:

- Dispatcher ↔ Validator
- Dispatcher \leftrightarrow NFVO (NVF Catalogue)
- Dispatcher ↔ Result catalogue
- Dispatcher ↔ Privacy/Security Manager
- Dispatcher ↔ Experiment Live Cycle Manager

These internal interfaces are needed to support the dispatcher role towards the experimenters or the Portal. However, these interfaces are not directly seen by the experimenter and are therefore defined outside the open API.

A separate interface between the dispatchers of the different platforms will enable interconnected experiment setups. This interface is yet to be defined. As detailed in section 5, this be developed as part of Release C.

3.2.2.2. Exposed Functionalities

In order to make 5GENESIS platform features available, such as definition and execution of the experiments, the open API will expose the following procedures:

- Access control and authorisation
- Define an experiment
- Manage or check experiment lifecycle, e.g., start, stop, check if active, check logs, etc.
- Onboarding and retrieval of NS and VNF Descriptors as part of an experiment
- Retrieval of historical data, as well as measurements and test reports

These operations will be further detailed in Section 3.5.

3.3. Architecture

The main priority of the open API architecture is to define the interface that can be easily consumed and accessed by the experimenter' client. The result offers protocols and custom data format to facilitate the interaction with the 5GENESIS system.

During the design process, the requirements and needed exposed features to communicate the different modules in the 5GENESIS facility were taken into account, specially the interfaces between the coordinator layer and the MANO layer. Beyond the functionality, the design of the API considers the exposed functionalities and the end-user experience.

The following picture depicts the architecture and interfaces between the components and modules involved in the creation of the experiment from the 5GENESIS portal.



Figure 3: Open APIs architecture

The open API architecture might evolve to a more complex system if during the evaluation with the experiments and experimenters more needs or features are identified. Therefore, the design presented in this document could be enhanced in the next quarters.

3.4. Components Description

The 5GENESIS open API is a virtual module which encompasses two main modules: The *Dispatcher* and the *Validator*.

The *Dispatcher*, the core of the open APIs, receives all the requests from external users and analyses which internal component should deal with it. It leans on the *Validator* to support some of the features of the open APIs (Figure 3).

3.4.1. Dispatcher

The Dispatcher is located in the Coordination Layer (see Figure 3). It is the entry point of any external user to the platform and is responsible for receiving all the requests sent by the experimenter (from the portal or using any other client). It agglutinates the whole set of the 5GENESIS features, exposing the appropriate functionalities from the lower layers that are necessary to run the platform securely from the outside through a single (open) API. The Dispatcher will interact with the underlying modules (see Figure 3) to offer the experimenter all the capabilities of the 5GENESIS system as a single interface with all the available features without the need of knowing the details of the Facility.

It is expected that the features the Dispatcher exposes are:

- Secure the communications with the open APIs, authenticate the experimenter (end user) to access the different platforms and provide the expected level of authorization.
- Manage the lifecycle of the NFVO VNF catalogue, including a pre-onboarding validation.
- Manage the lifecycle of an experiment: run, stop, etc.
- View information about the experiments: composition, statistics, historical data, etc.
- Retrieve experiment results.

Details of the exposed API and functionalities are given in section 3.5.

The operations related to the interconnection between Platforms are also part of the Dispatcher's functions although not part of the open APIs. How this interconnection will occur in technical terms is yet to be defined as that is a rather complex operation and the platforms are not ready at this stage to consider this sort of interface.

3.4.2. Validator

The Validator is an intermediate module between the Dispatcher and the **NFVO**, which is inside the MANO layer. It is also an intermediary between the Dispatcher and the **ELCM**. In both cases, it is in charge of validating the descriptors that are going to be used throughout the platform:

• The VNF and NS descriptors (VNFD and NSD) that are going to be onboarded in the Platform NFVO: 5GENESIS does not have its own VNF and NS descriptors and uses the NFVO default ones. The Validator will support different types of NFVO like OSM [16] or Open Baton [17],

therefore, the Validator should be able to identify the format of the descriptor to be onboarded as well as distinguishing the right NFVO to validate the descriptor with the proper schema. Regardless of the chosen NFVO, the behaviour and the interaction with the neighbour components is the same, as it relies on common components throughout the 5GENESIS Platforms.

The Validator carries out several steps during the descriptor validation process:

1. Receives the validation request from the Dispatcher including a file in the same format that would be accepted by the platform NFVO (usually a package containing several files).

- 2. Identifies the NFVO it belongs to.
- 3. Extracts the file that contains the core descriptor information from the package.

4. Performs a syntax validation over the text file to identify potential issues prior to the final onboarding.

5. If the descriptor is not syntactically valid, an error message will be returned and the descriptor will not be available in the 5GENESIS platform.

Details of the exposed API are given in section 3.5.1.

• The experiment descriptor is created by the experimenter and it contains all the necessary information for the platform to run and test a full experiment. The Validator will perform again a syntactical validation over the descriptor, verifying whether it is correct before sending it to the ELCM, returning an error message otherwise.

Details of the exposed API are given in section 0.

3.5. Interfaces Description (ATOS)

This section describes the operations that we foresee might be useful for the experimenter to have, so to be able to properly run the experiment over the Platform. The list can be changed or enhanced as the project continues its work, so to provide experimenters with up-to-date operations, which are linked to the latest results coming out of the project enhancements. These operations are classified by types, grouped according to the module that will finally attend the request.

3.5.1. Operations with VNFs and NSs

This type of operations involves mainly the MANO component, which is where the VNF and NS catalogues are included (via the internal interface Oa-Or, see Figure 3). When dealing with descriptors onboarding, these operations implicate also the Validator component (using internal interface Va-Or), performing a syntax validation over the descriptor prior to the final onboarding, although this bypass is transparent to the user.

The planned operations are:

Onboarding a VNFD

Description
Onboards a VNFD package in the NFVO NFV catalogue
Request endpoint format

/dispatcher/vnfd
Payload
VNF Descriptor file package
Method
POST
Response
ID of the VNFD recently uploaded and assigned by the NFVO or error message
Response codes
 201: Created – In case everything is correct 400: Bad request – plus details of what was wrong
- 401: Unauthorized
Response example

"id": "ea79077c-97c9-41ff-b310-e996ef66fa6c"

Table 2: VNFD Onboarding

Upload a VNF image

Description
Uploads an image file in the VIM
Request endpoint format
/dispatcher/image/{vim_type}
Payload
Image file
Parameters
- vim_type: edge/core
Method
POST
Response
ID of the image recently uploaded assigned by the VIM or error message.
Response codes
 201: Created – In case everything is correct 400: Bad request – plus details of what was wrong 401: Unauthorized 503: Service unavailable
Response example
{ "id": "ea79077c-97c9-41ff-b310-e996ef66fa6c" }

Table 3: VNF image uploading

Retrieve all VNFDs in the catalogue

Description
Consult the available VNFDs in the VNF catalogue
Request endpoint format
/dispatcher/vnfd
Method
GET
Response
List of VNFDs stored in the catalogue
Response codes
 200: OK – In case everything is correct 400: Bad request – plus details of what was wrong 401: Unauthorized 503: Service unavailable
Response example
<pre>vnfs: [{</pre>

Table 4: VNFDs retrieval

Delete a specific VNFD from the catalogue

Description

Delete an existing VFND from the NFVO NFV catalogue specified by its vnf_id

Request endpoint format

/dispatcher/vnfd/{vnfd_id}

Parameters

- vnf_id: ID of an existing VNFD within the NFVO NFV catalogue

```
Method
```

DELETE

Response

Response codes

- 204: OK In case everything is correct
- 400: Bad request plus details of what was wrong
- 401: Unauthorized
- 404: Not found
- 503: Service unavailable

Response format	
-	

Table 5: VNFD deletion

Onboarding an NSD

Description

Onboards a NSD package in the NFVO NS catalogue

Request endpoint format

/dispatcher/nsd

Payload

NS Descriptor file package

Method

POST

Response

ID of the NSD recently uploaded and assigned by the NFVO or error message

Response codes

- 201: Created In case everything is correct
- 400: Bad request plus details of what was wrong
- 401: Unauthorized
- 503: Service unavailable

Response example

"id": "ea79077c-97c9-41ff-b310-e996ef66fa6c"

Table 6: NSD onboarding

Retrieve all NSDs in the catalogue

Description

Consult the available NSDs in the NFVO NFV catalogue to create experiments based on them

Request endpoint format

/dispatcher/nsd

Method

GET

{

Response

List of NSDs stored in the NFVO NFV catalogue

Response codes

- 200: OK In case everything is correct
- 400: Bad request plus details of what was wrong
- 401: Unauthorized
- 503: Service unavailable

Respons	e example
nsds:	
	"id": "xxxx",
	"name": "xxxx",
	"description": "xxxx",
]	

Table 7: NSDs retrieval

Retrieve a specific NSD from the catalogue

Description
Retrieve a single full NSD from the NFVO NFV catalogue. This is done for the Experimenter to have more information about the NS than the one provided when consulting the complete list.
Request endpoint format
/dispatcher/nsd/{nsd_id}
Parameters
 nsd_id: ID of an existing NSD within the NFVO NFV catalogue
Method
GET
Response
NSD corresponding with the nsd_id specified in the query. See Annex 2: NS Descriptors
Response codes
 200: OK – In case everything is correct 400: Bad request – plus details of what was wrong 401: Unauthorized 404: Not found 503: Service unavailable
Response example
nsds: [{}]

Table 8: Specific NSD retrieval

Delete a specific NSD from the catalogue

Description

Delete an existing NSD from the NFVO NFV catalogue specified by its nsd_id

Request endpoint format

/dispatcher/nsd/{nsd_id}

Parameters

- nsd_id: ID of an existing NSD within the NFVO NFV catalogue

Method

DELETE

Response codes

Response

- 204: OK – In case everything is correct

- 400: Bad request plus details of what was wrong
- 401: Unauthorized
- 404: Not found
- *503*: Service unavailable

Response format

Table 9: NSD deletion

3.5.2. Operations with Experiments

The set of operations the experimenter will use to interact with the 5GENESIS experiments involve the ELCM in the Coordination Layer (internal interface Oa-El, see Figure 3Figure 3), whose main functionalities are exposed through the Dispatcher as part of the open APIs.

The planned operations are:

Execute an experiment

Description
Launch the execution of an experiment and run the tests included in the query.
Request endpoint format
/dispatcher/experiment
Payload
- Experiment data model [Annex 4: Experiment Descriptor]
Method
POST
Response
ID assigned to the experiment just created
Response codes
 201: Created – In case everything is correct 400: Bad request – plus details of what was wrong 401: Unauthorized 503: Service unavailable
Response format
<pre>{ 'execution_id': `xxxx', 'success': True/False, 'message': `Status or error description' }</pre>

Table 10: Experiment execution

List active experiments executions

Description
Retrieve current experiment executions.
Request endpoint format
/dispatcher/executions
Parameters
-
Method
GET
Response
List of active experiments executions
Response codes
 200: OK – In case everything is correct 400: Bad request – plus details of what was wrong 401: Unauthorized 404: Not found 503: Service unavailable
Response format
<pre>[{ 'execution_id': 'xxxx', 'experiment_id': 'xxxx'}, { 'execution_id': 'xxxx', 'experiment_id': 'xxxx'}]</pre>

Table 11: Active experiments executions listing

Check execution status

Description
Retrieve the status of a specific experiment execution.
Request endpoint format
/dispatcher/execution/{id}
Parameters
- Id: ID of an existing experiment execution
Method
GET
Response
Experiment execution status
Response codes
 200: OK – In case everything is correct 400: Bad request – plus details of what was wrong 401: Unauthorized 404: Not found

- 503: Service unavaila	able
-------------------------	------

Response format

`running': True/False

Table 12: Experiment execution status

Retrieve Experiment Descriptor

Description

{

Retrieve the experiment descriptor of an experiment

Request endpoint format

/dispatcher/experiment/{id}

Parameters

- Id: ID of an existing experiment

Method

GET

Response

Experiment descriptor that corresponds with the specified id in the query [Annex 4: Experiment Descriptor]

Response codes

- 200: OK In case everything is correct
- 400: Bad request plus details of what was wrong
- 401: Unauthorized
- 404: Not found
- 503: Service unavailable

Response format

{...}

Table 13: Experiment Descriptor retrieval

Stop an experiment execution

Description
Stop an experiment execution, keeping the data for the history
Request endpoint format
/dispatcher/execution/{id}
Parameters
- Id: ID of an experiment execution (execution id)
Method
DELETE
Response
Experiment detail

Response codes

- 204: OK In case everything is correct
- 400: Bad request plus details of what was wrong
- 401: Unauthorized
- 404: Not found
- 503: Service unavailable

Response format

```
'success': True/False,
'message': 'Status description'
```

Table 14: Experiment execution stop

Retrieve execution logs

Description
Retrieve the logs generated by a given experiment execution
Request endpoint format
/dispatcher/execution/{id}/logs
Parameters
- Id: ID of an experiment execution in the DB
Method
GET
Response
The collection of logs generated by an experiment execution
Response codes
 200: OK – In case everything is correct 400: Bad request – plus details of what was wrong 401: Unauthorized 404: Not found 503: Service unavailable
Response format
See Annex 3: Experiment Logs

Table 15: Execution logs retrieval

Historical data of the executed experiments

Description

Retrieve the list of experiment executions registered for a given experiment id

Request endpoint format

/dispatcher/experiment/history

Payload

- Filters: date, experiment, etc.

Method
GET
Response
List of experiment executions
Response codes
 200: OK – In case everything is correct 400: Bad request – plus details of what was wrong 401: Unauthorized 404: Not found 503: Service unavailable
Response format
<pre>[{ 'execution_id': 'xxxx' }, { 'execution_id': 'xxxx' }]</pre>

Table 16: Executed experiments historical information

3.5.3. Operations with Analytics and Monitoring

In this case, the Dispatcher will contact the Result Catalogue module to fulfil the request (Oa-An interface in Figure 3).

The planned operations are:

```
List of KPIs measured for a given execution
```

Description
Retrieve the list of KPIs generated by a given experiment execution
Request endpoint format
/dispatcher/analytics/{exp_id}
Parameters
 exp_id: ID of an existing execution
Method
GET
Response
List of KPIs generated
Response codes
 200: OK – In case everything is correct 400: Bad request – plus details of what was wrong 401: Unauthorized 503: Service unavailable
Response format
<pre>[{ `kpi_name': `xxxx', `value': `xxxx', `unit': `xxxx'},</pre>

{ 'kpi_name': 'xxxx', 'value': 'xxxx', 'unit': 'xxxx'}

Table 17: List of KPIs measured for a given execution

Analytics report for an existing execution

Description
Retrieves the generated PDF report of a given experiment execution
Request endpoint format
/dispatcher/report/{exp_id}
Parameters
 exp_id: ID of an existing execution
Method
GET
Response
PDF file with the requested report
Response codes
 200: OK – In case everything is correct 400: Bad request – plus details of what was wrong 401: Unauthorized 503: Service unavailable
Response format
- PDF report file

Table 18: Analytics report for an existing execution

3.5.4. Operations with the Privacy / Security Manager

For Release A of the project, in order to avoid potential issues, the security and privacy topic has been taken out of the open APIs and it is handled temporarily by the Portal, performing user authentication before accessing the platform. In the next phases of the implementation, these will be part of the available operations of the open APIs, securing the rest of the operations.

4. 5GENESIS PORTAL

In addition to the option of using the open API in a direct way, the 5GENESIS Portal provides a user-friendly web interface for experimenters to interact with the 5GENESIS Facility.

In Release A, the exposed functionalities by the Portal are a subset of the planned functionalities of the open API. In the future, the Portal will be a client of the open API.

Users can define new experiments, examine the results and logs of previous executions or manage deployed VNFs among other features. Furthermore, experimenters can view information about their latest performed actions and access to system notices.

Experimenters can define experiments to generate KPIs and obtain results of the execution. For that purpose, they can configure several parameters such as the test cases to perform, and Network Services (previously uploaded) to deploy. Afterwards, all the logs written during the different parts of the execution can be retrieved and reviewed. In addition to this, a link to a custom Grafana dashboard, that displays the execution results, is provided.

5Genesis	Home Create Experiment VNF/NS	Management		5Genesis - Logout
	EXPER	RIMENTS	6	NOTICES
Experiment ID	Name	Туре	Action	System-wide notices appear here
4	Round Trip Time Test - Ideal	Standard	Run Experiment Executions	
3	Maximum User Data Rate Test - Ideal	Standard	Run Experiment Executions	ACTIONS
2	Service Creation Time Tests - Ideal	Standard	Run Experiment Executions	19 September 2019, 12:14:29 Ran experiment: Maximum User Data Rate Test - Ideal
1	MCPTT Tests - Ideal	Standard	Run Experiment Executions	19 September 2019, 12:11:06 Ran experiment: Maximum User Data Rate Test - Ideal
				19 September 2019, 12:06:22 Ran experiment: Maximum User Data Rate Test - Ideal
				19 September 2019, 12:02:50 Ran experiment: Maximum User Data Rate Test - Ideal v

Figure 4: Main page of the 5Genesis Portal

4.1. Portal Design

The main sections of the Portal are:

- **Register/Login**: Registration and login pages for accessing the Portal.
- **Main page:** The main page shows all the experiments that the user has created, as well as the system notices and a log of the latest performed activities.
- **Create Experiment:** The experimenters can create their own experiments by configuring the available parameters. After that, the experiment can be run and examined, either by checking the status and execution logs or visualizing the results in the Grafana dashboard.

- VNF/NS Management: Users can upload their VNFs and NS to later use in the experiments. VNFs require an image and a VNFD (VNF Descriptor), while network services require an NSD file. This section also allows users to remove their previously uploaded VNFs and NS. This functionality is not yet connected with the lower layers and appears as a proof of concept of the UI.



Figure 5: Overview of Portal organization

4.1.1. Portal Users

Once the users are registered in the Portal, they can create different experiments in which they can configure certain parameters; experimenters can also upload VNFs and NS that will be linked to their accounts and can be used in their experiments. All the experiments created by the user are displayed in the dashboard (see Figure 4).

4.1.2. Experiment creation

Experimenters provide information about their experiments through the Portal. Furthermore, when either uploading a VNF/NS or creating an experiment, they are asked for additional information.

Experiments need the following information (see Figure 6):

- **Type:** The type of experiment that can be selected is Standard or Custom, though only Standard experiments can be created in the current release of the Portal.
- **Test Cases:** The experimenters can select from a list of pre-defined test cases the ones that apply to the experiment. At least one is required.
- **UEs:** The experimenters may choose multiples devices for monitoring and perform automation tasks during the experiment.
- Slice: A pre-configured network slice can be selected and deployed during the experiment.
- **Network Services:** Experiments can use multiple NS (or none) during the experiment execution.

5Genesis Home	Create Experiment VNF/NS Management		5Genesis - Logout
	CREATE EXPERIME	NT	
	Name	Туре	
	Mission Critical	Standard ~	
	Test Cases		
	Round Trip Time Test Maximum User Data Rate Test MCPTT Tests Service Creation Time Tests		
	UEs		
	Galaxy S9		
	Slice		
	Slice1	\sim	
	Network Services		
	Select VNF		
	1 MalagaPlatform	~	
	Add Experiment		

Figure 6: Experiment configuration parameters

4.1.3. VNF/NS Management

The VNF/NS Management dashboard provides a centralized view where users can see all the available VNFs and NS in their account, as well as upload new instances.

VN	F REPOSITORY	
Upload VNF		
	VNF: Test VNF	
Description This is a test VNF		Delete
VNFD Path	Image Path	
test.vnfd	test.image	
	S REPOSITORY	
	NS: Test NS	
Description This is a test NS		Delete
1. Company of the second secon		
NSD Path		

Figure 7: VNF/NS management dashboard

5Genesis	Home	Create Experiment VNF/NS Management	5Genesis - Logout
		UPLOAD VNF	
		Name	
		Test VNF	
		Description	
		This is a test VNF	н
		Upload VNFD Browse Selected: test.vnfd	
		Upload Image Browse Selected: test.image	
		Upload VNF	

Figure 8: VNF upload screen

When uploading a VNF, the user must provide the following information (see Figure 8):

- Description: Extra information about the uploaded VNF.
- Image: The user must upload an image file for the image service.
- VNFD: The user must upload a VNF Descriptor which contains the behavioural and deployment information of a VNF.

5Genesis	Home Create Experiment VNF/NS Management	5Genesis - Logout
	UPLOAD NS	
	Name	
	Test NS	
	Description	
	This is a test NS	
	Upload NSD Browse Selected: test.nsd	
	Upload NS	

Figure 9: NS upload screen

Uploading a NS is very similar: When uploading a NS, the user must provide the following information (see Figure 9):

- Description: Extra information about the uploaded VNF.
- NSD: The user must upload a network experiment descriptor.

4.1.4. Execution Logs

The Portal is able to display the execution's logs for all execution stages of the experiments: Pre-Run, Run and Post-Run. The user can filter the messages displayed depending on the severity level (see Figure 10). Those are: Info, in white colour (displayed by default); debug, in blue; warning, in yellow; error, in red; and critical, in purple. Each line of the log will show the timestamp of when the message was produced, the severity level and the message itself.

SGenesis Home Create Exp	5Genesis - Logout			
		Execution 2	18	
			40	
Status	Start Time	End Time	Experiment	Action
Finished	19 September 2019, 9:59:54	19 September 2019, 10:03:04	MCPTT Tests - Ideal	Results
Pre-Run Log				
	Debug 13	Info 22 Warning	Error Critical	
2013-09-19 00159:54,203 DEBUG C [F41] 2019-09-19 00159:54,203 DEBUG C [Uxi] 2019-09-19 09159:54,203 INFO Start 2019-09-19 09159:54,206 INFO Start 2019-09-19 09159:54,206 INFO FReque 2019-09-19 09159:54,206 INFO Reque 2019-09-19 09159:54,206 INFO Reque 2019-09-19 09159:54,207 INFO Reque 2019-09-19 09159:54,208 DEBUG Parat 2019-09-19 09159:55,228 INFO [Star 2019-09-19 09159:55,228 INFO Reque 2019-09-19 09159:55,228 INFO Reque 2019-09-19 09159:55,228 INFO Resou 2019-09-19 09159:56,231 INFO [Star 2019-09-19 09159:56,231 INFO Reque 2019-09-19 09159:56,232 INFO Reque 2019-09-19 09159:56,232 INFO<	<pre>s Opened] ng temporal folder: Temp\tm ed ting Task Check Availabilit ns: ('Id': 248, 'Available' sting availability frees not available Check Availability finishle ns: ('Id': 248, 'Available' ting Task Check Availabilit ns: ('Id': 248, 'Available' ting Task Check Availability frees not availability frees availability frees availability frees availability frees availability frees availability fing Task Add Experiment En mation sent Add Experiment Entry finis frees free the finisher free free free free free free free fr</pre>	<pre>hphqyhm_1f] xy] xy] xy] xi False} xi] xy] xy] xi False} xi] xi] xi] xi] xi] xi] xi] xi] xi] xi]</pre>		

Figure 10: Log executions screen

4.1.5. Execution Results

For each experiment execution, the Portal provides a link to a Grafana dashboard for easy visualization of the data generated by the experiment. A customized dashboard is generated for each type of experiments, which includes the most important results that have been obtained.

Grafana offers the experimenter a user-friendly interface with multiple graphs and different ways for visualizing data. The user can split views in order to compare different time ranges and metrics side by side. Although the dashboard is configured to show the complete data obtained by default, the user can zoom in the graphs in order to see a detailed view of the measurements in a certain time range.



Figure 11: RTT experiment dashboard generated by the ELCM

4.2. Portal Implementation

The Portal has been implemented using the Flask framework for Python. For the frontend, the Bootstrap framework is used.

The database model is based on SQLAlchemy, which allows every 5GENESIS Platform to decide between multiple backends. The data is stored in the backend following the structure described in the following subsection. For more information about the stored tables, refer to Annex 1.

4.2.1. Portal internal data model

For supporting all the features previously outlined, the Portal must store some information in its database. The main entities managed by the Portal are shown in Figure 12:



Figure 12: Entities of the 5GENESIS Portal

The **User Model** contains the username, organization, email, and hashed password of a user. The password is saved encrypted so that not even Platform administrators can know this value by inspecting the database.

id	username	email	password_hash	organization
1	TestUser	test@5genesis.com	pbkdf2:sha256:50000\$FlcGt[]	5GENESIS

Table 19: User Model

The Experiment and VNF Models store the information required for running execution. The **Experiment Model** contains a reference to the experiment owner, the type of experiment (currently only standard experiments are supported), a list of test cases and UEs and a network slice identifier (optional). The list of network services used in the experiments is saved on an auxiliary table called 'experiment_ns' (not pictured) in order to support the many to many relations.

id	user_id	name	type	test_cases	ues	slice
1	1	TestExperiment	Standard	["RTT"]	["Galaxy S7"]	Test Slice

Table 20: Experiment Model

The **VNF and NS Models** are very similar. Both include the name of the instance, a description and a reference to the owner. In the case of the VNF model, there are columns for saving the image and VNFD file names, while on the NS model the NSD file name is stored.

id	user_id	name	description	VNFD	image
1	1	Test VNF	This is a test VNF	test.vnfd	test.image

Table 21: VNF Model

id	user_id	name	description	NSD
1	1	Test NS	This is a test NS	test.nsd

Table 22: NS Model

The **Execution Model** stores the start and end time of an execution, the dashboard URL for the results page, the status and current percentage of the execution and a message that informs the user about the current execution step. The last three attributes are continuously updated during the execution and are displayed to the user in the dashboard.

id	experiment_id	start_time	end_time	dashboard_url	status	percent	message
1	1	2019-05-29 10:34:57.25	2019-05-29 10:36:17.28	/d/Run1/experiment- run-1	Finished	100	Finished

Table 23: Execution Model

The **Action Model** is used for recording the different actions that a user performs. This information is used for generating a feed that will be displayed in the user's dashboard. This feed provides direct access to the results of the latest experiment executions, for example. In order to generate this information, the Portal stores a reference to the user, the timestamp of the event and a HTML encoded message that includes the relevant links.

id	user_id	timestamp	message			
1	1	2019-05-28 09:08:33.40	Uploaded VNF: TestVnf			

Table 24: Action Model

5. RELEASE A SUMMARY AND FUTURE PLANS

In order to achieve a smooth evolution and concurrent validation of the project results, 5GENESIS adopts an iterative integration and upgrade development methodology resulting of three Releases.

In the following subsections, we summarise the work done during the first of these Releases, release A, as well as the foreseen plans for the following ones.

5.1. Release A

5.1.1. Open API

During this first period of the project, the focus has mainly been on the design of the open API. However, a first basic version of the API, Release A, is already available based on the collected requirements. This version will evolve as described in Section 3 in the next Releases.

The validator is now able to validate VNFDs and NSDs. As there is no specific unified VNFD/NSD in the project, the validation depends on the MANO platform used in each of the five Platforms that compose the 5GENESIS facility, namely OSM and Open Baton. The validation of the experiment descriptor will be faced in Release B, once its definition is more precise.

5.1.2. Portal

In Release A, the Portal does not make use of the Dispatcher element, communicating directly with the ELCM in order to execute the experiments. The ELCM and the Portal use a temporary REST API that provides a subset of the functionality of the open APIs.

The Portal exposes a REST API that provide access to its backend. This allows external services to request information stored in the backend database or update it. Currently, this API is used by the ELCM, which makes use of the available endpoints in order to update the status, completion percentage, latest messages and Grafana dashboard URL of each experiment execution. The Portal itself also makes use of this API in order to retrieve the latest experiment execution information from the database, which is used for automatically updating the progress bar and status information that users can see in the interface. All API calls return a JSON object with the data requested and a status message.

Endpoint	Description	Return format (JSON)		
/execution/ <executionid></executionid>	Update the execution information with the values provided by the ELCM. The values updated are: status, dashboard, percent and message.	{ 'Status': <inform about="" the<br="">success of the update> }</inform>		

The endpoints exposed by the REST API are detailed in the following table:

/ <int:executionid>/json</int:executionid>	Return the execution information (status, percent and message) in JSON format. This method is used for updating the progress bar during execution.	{ 'Status': <current of<br="" status="">the experiment>, 'PerCent': <current progress of the step>, 'Message': < A message describing the current step in the execution> }</current </current>
/nextExecutionId	Return next execution Id	{ 'NextId': <the next<br="">execution ID that will be generated> }</the>

Table 25: Endpoints exposed by the Portal REST API

5.2. Release B

5.2.1. Open API

In Release B, the effort will be oriented to evolve the current version of the open API to the design described in section 3. If any additional requirement shows up during the period, it will be also analysed and incorporated.

One of the main tasks for release B will be the integration of the Portal and the Dispatcher, as well as the development of the experiment descriptor validation by the Validator.

Regarding design, during this period, we also expect to start defining the operations related to the interconnection between platforms that, we foresee, will be are part of the Dispatcher's functions.

Security and privacy are other topics to be considered in Release B. As said, this kind of actions are currently handled by the Portal. The idea is that they are part of the available operations of the open APIs, securing the rest of the operations.

5.2.2. Portal

Once the Dispatcher is integrated in the different platforms, the Portal will be updated in order to replace the direct communication that has been implemented in Release A with the use of the open APIs.

5.3. Release C

5.3.1. Open API

As the number of executions of experiments increase, we expect new requirements to be incorporated to the final version of the open API to be released at the end of the project.

The focus of this task in Release C will centre towards the interconnection of Platforms. Although how the technical interconnection between the platforms will occur is still to be defined, we foresee that this will require complex operations that will be faced in the last Release of the project, once the Platforms are ready to consider this sort of interface.

5.3.2. Portal

In Release C, the Portal will be improved in order to support the new functionality developed in other components of the platforms, as well as to fix any issues discovered during the previous experimentation cycles.

6. CONCLUSIONS

This document has described the two possible ways the 5GENESIS Facility can be accessed by a vertical user or project that wants to use it to execute and validate (5G) experiments.

With the goal of offering an open and common method to interact with the Facility for experimentation, 5GENESIS exposes an open API. The initial design of this open API as well as the roadmap that we plan to follow for its implementation has been described in this deliverable.

For less advanced users, 5GENESIS also offers a Portal with a friendly Web UI as an alternative method to use the Facility. The Portal is also able to display the execution's logs of the experiments and provides a link to a Grafana customized experiment-specific dashboard for easy visualization of the data generated in the execution of each experiment. The current design and implementation of the Portal has been also presented in this document.

During this period, the 5GENESIS project has been mainly focused in building and delivering an initial version of the Facility, Release A which incorporates basic management of VNFs, NSs and experiments among other features. A more intensive use of the Facility in relation to experimentation is expected to occur from now on. Any new requirement that may arise from both internal and external experimenters as a result of the use of the Facility, will be analysed and incorporated in the next releases of either the open API or the Portal, or both of them if needed.

REFERENCES

- [1] 5GENESIS Consortium, D2.1 Requirements of the Facility: <u>https://5genesis.eu/wp-content/uploads/2018/11/5GENESIS D2.1 v1.0.pdf</u>
- [2] 5GENESIS Consortium, D2.2 Initial overall facility design and specifications: https://5genesis.eu/wp-content/uploads/2018/12/5GENESIS D2.2 v1.0.pdf
- [3] 5GENESIS Consortium, D2.3 Initial planning of tests and experimentation: https://5genesis.eu/wp-content/uploads/2018/12/5GENESIS D2.2 v1.0.pdf.
- [4] TS 23.222 V16.4.0 (2019-06) 'Functional architecture and information flows to support Common API Framework for 3GPP Northbound APIs; Stage 2 (Release 16)'
- [5] TMForum Open APIs program: <u>https://www.tmforum.org/open-apis/about-the-program/</u>
- [6] 3GPP TS 23.682 V16.3.0 (2019-06) 'Architecture enhancements to facilitate communications with packet data networks and applications', (Release 16)
- [7] 3GPP TR 23.981 V15.0.0 (2018-06) 'Interworking aspects and migration scenarios for IPv4-based IP Multimedia Subsystem (IMS) implementations (Release 15)'
- [8] 3GPP TS 23.501 V16.1.0 (2019-06), 'System Architecture for the 5G System; Stage 2 (Rel-16)'
- [9] R. Fielding, 'Architectural Styles and the Design of Network-based Software Architectures', Dissertation, University of California, Irvine, https://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf.
- [10] RESTful APIs for the 5G Service Based Architecture: https://www.riverpublishers.com/journal/journal_articles/RP_Journal_2245-800X 617.pdf
- [11] 3GPP TS 29.501 V16.0.0 (2019-06), '5G System; Principles and Guidelines for Services Definition; Stage 3 (Release 16)'
- [12] 3GPP TS 23.502 V16.1.1 (2019-06), 'Procedures for the 5G System; Stage 2 (Release 16)'
- [13] 3GPP TS 29.510 V16.0.0 (2019-06), 'Network Function Repository Services; Stage 3 (Release 16)'.
- [14] 3GPP TS 29.502 V16.0.0 (2019-06), '5G System; Session Management Services; Stage 3 (Release 16)'
- [15] TMForum Open APIs: <u>https://www.tmforum.org/open-apis</u>

- [16] OSM website: <u>https://osm.etsi.org</u>
- [17] Open Baton GitHub repository: <u>https://openbaton.github.io</u>

ANNEX 1: PORTAL DATABASE

The Portal uses an SQLAlchemy database as backend. This annex includes the current definition of the database tables, using SQL dialect.

```
CREATE TABLE "vnf location" (
           INTEGER NOT NULL,
    "id"
    "location" VARCHAR(32),
"VNF id" INTEGER,
    "experiment id" INTEGER,
    PRIMARY KEY ("id"),
    FOREIGN KEY("VNF_id") REFERENCES "VNF"("id"),
    FOREIGN KEY ("experiment id") REFERENCES "experiment" ("id")
);
CREATE TABLE "VNF" (
    "id" INTEGER NOT NULL,
"name" VARCHAR(64),
    "description" VARCHAR(256),
    "VNFD" VARCHAR(256),
    "image" VARCHAR(256),
    "user id" INTEGER,
    PRIMARY KEY("id"),
    FOREIGN KEY("user_id") REFERENCES "user"("id")
);
CREATE TABLE "action" (
    "id" INTEGER NOT NULL,
    "timestamp" DATETIME,
    "message" VARCHAR(256),
"user id" INTEGER,
    PRIMARY KEY("id"),
    FOREIGN KEY("user id") REFERENCES "user"("id")
);
CREATE TABLE "execution" (
    "id" INTEGER NOT NULL,
    "experiment_id" INTEGER,
    "start time" DATETIME,
    "end_time" DATETIME,
"status" VARCHAR(32),
    "dashboard_url" VARCHAR(64),
    "message" VARCHAR(
"percent" INTEGER,
                 VARCHAR(128),
    PRIMARY KEY("id"),
    FOREIGN KEY ("experiment id") REFERENCES "experiment" ("id")
CREATE TABLE "experiment" (
    "id" INTEGER NOT NULL,
    "user id" INTEGER,
    "name" VARCHAR(64),
    "type" VARCHAR(16),
    "unattended"
                     BOOLEAN,
    "test cases"
                     VARCHAR,
    "ues" VARCHAR,
"NSD" VARCHAR(256),
    "ues"
    "slice" VARCHAR(64),
    PRIMARY KEY("id"),
    FOREIGN KEY("user_id") REFERENCES "user"("id")
):
CREATE TABLE "user" (
    "id" INTEGER NOT NULL,
    "username" VARCHAR(64),
    "email" VARCHAR(120),
    "password_hash" VARCHAR(128),
"organization" VARCHAR(32),
    PRIMARY KEY("id")
);
```

Figure 13: Portal Database

ANNEX 2: NS DESCRIPTORS

OSM NSD example with 2 VNFs:

```
nsd:nsd-catalog:
    nsd:
        id: hackfest2-ns
        name: hackfest2-ns
        short-name: hackfest2-ns
        description: NS with 2 VNFs connected by datanet and mgmtnet
VLs
        version: '1.0'
        logo: osm.png
        constituent-vnfd:
           vnfd-id-ref: hackfest2-vnf
            member-vnf-index: '1'
           vnfd-id-ref: hackfest2-vnf
            member-vnf-index: '2'
        vld:
           id: mgmtnet
            name: mgmtnet
            short-name: mgmtnet
            type: ELAN
            mgmt-network: 'true'
            vim-network-name: mgmt
            vnfd-connection-point-ref:
                vnfd-id-ref: hackfest2-vnf
                member-vnf-index-ref: '1'
                vnfd-connection-point-ref: vnf-mgmt
                vnfd-id-ref: hackfest2-vnf
                member-vnf-index-ref: '2'
                vnfd-connection-point-ref: vnf-mgmt
            id: datanet
            name: datanet
            short-name: datanet
            type: ELAN
            vnfd-connection-point-ref:
                vnfd-id-ref: hackfest2-vnf
                member-vnf-index-ref: '1'
                vnfd-connection-point-ref: vnf-data
                vnfd-id-ref: hackfest2-vnf
                member-vnf-index-ref: '2'
                vnfd-connection-point-ref: vnf-data
                     Figure 14: OSM NSD example
```

ANNEX 3: EXPERIMENT LOGS

The response to a request for retrieving the logs generated during an experiment execution includes a general status message ('Success' or 'Not Found') and one instance of LogInfo for each of the execution stages of an experiment.

LogInfo includes a dictionary called 'Count' that details the number of messages on each independent severity, as well as a list of tuples that contains every message along with their severity in the 'Log' field.

```
Response format:
{
  Status: 'Success' or 'Not Found'
 PreRun:<LogInfo>
 Executor:<LogInfo>
  PostRun:<LogInfo>
}
LogInfo format:
      {
        Count: {
         Debug: <int>,
          Info: <int>,
          Warning: <int>,
         Error: <int>,
          Critical: <int> }
        Log: <List[Tuple[str, str]]:List of pairs</pre>
      (<str:Severity>, str:Message>)>
}
```

Figure 15: Logs retrieval request response

ANNEX 4: EXPERIMENT DESCRIPTOR

The Experiment Descriptor is the data model that includes all the information required for the execution of an experiment and the computation of the associated KPIs from the obtained results.

The information included in the Experiment descriptor can be seen in the table below:

Experiment Descriptor				-ID number-		
#	Description of the fields to be completed			t Values	Importance	
	Experiment details Information required to uniquely identify the experiment. Note 1: A Security Manager is used for editing with safety and privacy data related to the experimenter)		per	riment ID		
1			Owner ID		Mandatory	
			Organization ID			
	Note 2: Each experiment shall include all the combinations of the target metrics/test cases/scenarios/slice configurations listed in the following fields of this form. (one target metric linked to one test case, for a		Organization ID			
			Platform ID			
	specific scenario and a slice configuration is the minimum requirement for a complete experiment).			experiment		
	List of the Target Metric(s)		Me	tric ID1		
2	Selection of the metrics (identified by IDs) that the experiment targets at.				Mandatory	
	(see the Metric Template)					
	List of Test Case(s) to be executed		Test Case ID1		Mandatory	
	Selection of the test cases (identified by IDs) to be used in the experiment.			Case ID2		
3	Note: A test case includes KPI-associated Information (KPI definition.					
	measurement methodology, complementary monitoring needed, etc) linked to a metric from the list in the field above.		Test Case IDi			
	(see the Test Case Template)					
	List of Scenarios to be considered		Scenario ID1		Mandatory	
4	Selection of the Scenarios (identified by IDs) for which the test cases (selected in the previous field) will be executed.		Scenario ID2			
	Note: A scenario includes information related to all the parameters that affect the values of the KPIs to be measuredNetwork deployment and environment conditions, etc.)					
			Scenario IDi			
	(see the Scenario Description Template)					
	List of Slice Configurations to be established	Cline		NSD ID 1	Mandatawa	
5	LIST OF SHILE COMING I ATIONS TO DE ESTADIISTIEU	SIICE I	D	Radio Conf.	wandatory	

	Definition of the Slice templates (identified by IDs) that are required for the experiment(s). (see the Slice Configuration Template)			Extra parameters IDi Confia	
				···· · IDi Conf ···	
	Traffic Description Template (at least one traffic source or service type should be specified)			c sources	Optional
				ісе Туре	Optional
5		UEs identification			Mandatory (unattended experiments)
	Secondary input required for custom experiments	Application under test			Mandatory (unattended experiments)
		Intermediate reporting of KPIs and Time between intermediate reports			Optional

Figure 16: Experiment Descriptor Data Model