5G CITY

Grant Agreement No.761508 5GCITY/H2020-ICT-2016-2017/H2020-ICT-2016-2

D4.4

Final 5GCity Orchestrator Release

Dissemination Level		
☒	PU:	Public
□	PP:	Restricted to other programme participants (including the Commission Services)
□	RE:	Restricted to a group specified by the consortium (including the Commission
□	CO:	Confidential, only for members of the consortium (including the Commission Services)

Grant Agreement no: **761508**	Project Acronym: **5G CITY**	Project title: **5G CITY**

Lead Beneficiary: I2CAT Foundation	Document version: **V1.0**

Work package: WP4

Deliverable title: Final 5GCity Orchestrator Release

Start date of the project: 01/06/2017 (duration 30 months)	Contractual delivery date: M28	Actual delivery date: 01/10/2019

Editor name: Apostolos Papageorgiou (I2CAT Foundation)

List of Contributors

Participant	Short Name	Contributor
I2CAT Foundation	I2CAT	Apostolos Papageorgiou
NEXTWORKS	NXW	Elian Kraja, Leonardo Agueci, Gino Carrozzo
UBIWHERE LDA	Ubiwhere	Ricardo Preto
ITALTEL SPA	ITL	Antonino Albanese, Viscardo Costa, Valerio Vecchio Verderame
University of Bristol	UINIVBRIS	Carlos Colman Meixner
ADLINK TECHNOLOGY SARL	PRISMTECH	Gabriele Baldoni
Virtual Open Systems	VOSYS	Teodora Sechkova, Michele Paolino

List of Reviewers

Participant	Short Name	Contributor
I2CAT Foundation	I2CAT	Shuaib Siddiqui
NEXTWORKS	NXW	Gino Carrozzo

Change History

Version	Date	Partners	Description/Comments
0.1	12-09-2019	I2CAT	First draft including initial ToC and first contents, mostly taken from the interim release
0.2	20-09-2019	I2CAT	Second version including updates to reflect the changes of all components as well as the functional validation section
0.3	23-09-2019	UNIVBRIS	Added Infrastructure Abstraction and checked related parts
0.4	25-09-2019	I2CAT	Consolidated and highlighted delta of features compared to Interim Release QA review
1.0	01-09-2019	I2CAT	Final version

DISCLAIMER OF WARRANTIES

This document has been prepared by 5GCITY project partners as an account of work carried out within the framework of the contract no 761508.

Neither Project Coordinator, nor any signatory party of 5GCITY Project Consortium Agreement, nor any person acting on behalf of any of them:

- makes any warranty or representation whatsoever, express or implied,
    - with respect to the use of any information, apparatus, method, process, or similar item disclosed in this document, including merchantability and fitness for a particular purpose, or
    - that such use does not infringe on or interfere with privately owned rights, including any party's intellectual property, or
- that this document is suitable to any particular user's circumstance; or
- assumes responsibility for any damages or other liability whatsoever (including any consequential damages, even if Project Coordinator or any representative of a signatory party of the 5GCITY Project Consortium Agreement, has been advised of the possibility of such damages) resulting from your selection or use of this document or any information, apparatus, method, process, or similar item disclosed in this document.

5GCITY has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 761508. The content of this deliverable does not reflect the official opinion of the European Union. Responsibility for the information and views expressed in the deliverable lies entirely with the author(s).

Table of Contents

Figures

Tables

Executive Summary

This Deliverable D4.4 documents the software prototype of the 5GCity Orchestrator Platform by presenting the essential features included in the Final Release of all its software components. The work described in this deliverable is related to the main outcomes of tasks T4.1 and T4.2. The overall 5GCity architecture has been described in Deliverable D2.2, the logic of the 5GCity Orchestrator components have been elaborated in Deliverable D4.1, while the Interim Release of the 5GCity Orchestrator software components has been documented in Deliverable D4.2. This document completes the documentation of the 5GCity Orchestrator architecture, mainly focusing on software modules and information necessary to deploy and use the developed prototypes. Results of functional tests for the various modules are also included which serve as validation of the integrated prototypes before their integration in city infrastructures.

The document is organized to describe:

- How the high-level 5GCity Orchestrator architecture is mapped to the actual software components that have been implemented.
- The main features of each component, thus hinting to what can be achieved by using this prototype, which is used for orchestrating the 5GCity Use Cases.
- Hardware and software requirements and installation instructions for the prototype, including additional documentation about installation and usage that is available online wherever necessary.
- Information about the functional tests that validated the proper functionality of the orchestrator.

We describe the usage and capabilities of the released prototypes by going through some usage scenarios of the user-facing components of the Orchestrator, namely the 5GCity Dashboard and the 5GCity SDK.

This document represents an update of deliverable D4.2 released in M24, and as such it incorporates changes in all sections in order to reflect the new or modified platform features, as well as a new section dedicated to functional testing.

1. Introduction

The development work executed for the 5GCity Orchestrator reflects the implementation of the architecture described in Deliverable D4.1. Starting from D4.1, the architectural components of the 5GCity Orchestrator have been implemented in terms of software modules resulting in a mapping between functional entities and actual software components as depicted in Figure 1. In order to help the reader go through the rest of the document, the specific software modules of the final release are listed and highlighted in the yellow notes of Figure 1.

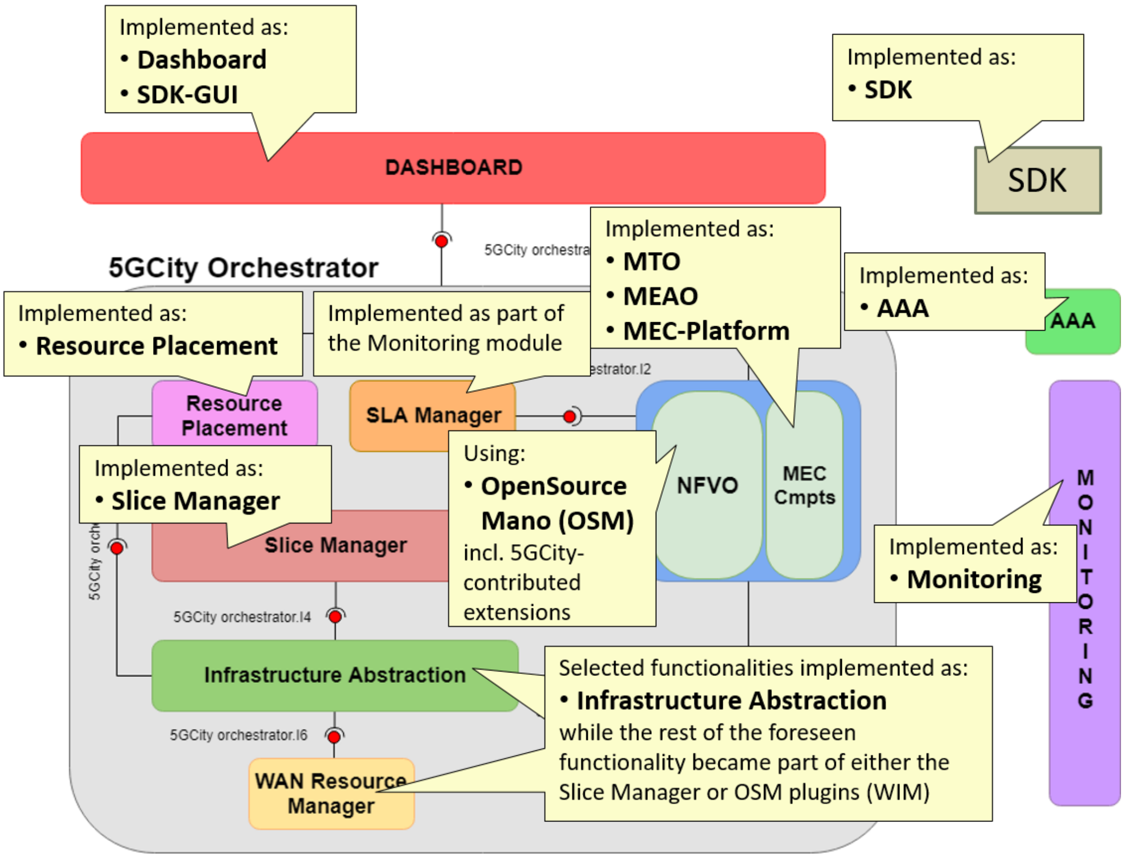


Figure 1. Software modules implementing the 5GCity Orchestrator Final Release.

In line with the depicted status, the subsequent sections will be structured based on the software modules of the Final Release, namely **Dashboard, SDK-GUI, SDK, AAA, Resource Placement, Slice Manager, MTO, MEAO, MEC-Platform, Infrastructure Abstraction, OSM, Monitoring**.

As anticipated in previous Deliverables, the interfaces and their usage have been adjusted to better reflect design choices made during the software implementation, so that the detailed and strictly accurate interfacing between software modules can only be derived by looking into the software documentation of the modules (e.g., swagger files), which are also linked in this document.

2. Main features of the 5GCity Orchestrator – Final Release

This section lists and explains the main features offered by the released 5GCity Orchestrator software in its final release. The goal here is to list features that have been implemented based on the specifications provided in Deliverable D4.1. Many of these features were present in the interim release (Deliverable D4.2) and have been further tested or enhanced for the final release. Components and features that are completely new in this final release are annotated as (*NEW)*. The rest of listed features have undergone a refinement and bug fixing process thus incorporating updates aimed at stabilizing operation.

Dashboard

Browse infrastructure

- View physical resources available in the 5GCity platform:
  - Computes.
  - Physical Networks.
  - Access networks.

Manage infrastructure

- Create and Delete physical resources in the 5GCity platform:
  - Computes.
  - Physical networks.
  - Access networks.

Manage slices

- Create chunks (i.e., partitions of the previously listed resources).
- View, create, edit, and delete slices as collections of chunks.

RAN activation ***(NEW)***

- Trigger the activation of the RAN components of a slice. *(NEW)*
- Respective options are given to handle slices that include no RAN, only WiFi, only LTE/5G, or both RAN technologies. *(NEW)*

Manage services instances ***(NEW)***

- View, create and delete network services instances currently deployed in 5GCity *(NEW)*

Authenticate users

- login/logout as well as featuring data/features based on the permission level of the user, supporting two roles ("Infrastructure Owner" and "Slice User") as well as multi-tenancy.

SDK-GUI

List SDK services

- List SDK services available in the local database

Manage services

- Add, edit, or delete an SDK service by using a graphical interface (5GCity dashboard dedicated pages):
  - The SDK service topology is edited through a drag and drop interface.
  - SDK Functions and service configurations are done through menu definition (no json or XML editing is required)

Publish services

- Publish created services to the 5G App & Service Catalogue (3rd party external software), directly interfaced to OSM NFVO and catalogue service

List SDK functions ***(NEW)***

- List SDK functions available in the local database *(NEW)*

Manage functions ***(NEW)***

- Add, edit, or delete an SDK functions by using a graphical interface (5GCity dashboard dedicated pages) *(NEW)*:
  - The SDK function topology is edited through a drag and drop interface. *(NEW)*

Publish functions ***(NEW)***

- Publish created functions to the 5G App & Service Catalogue (3rd party external software), directly interfaced to OSM NFVO and catalogue service *(NEW)*

SDK core

SDK Functions

- Create a new SDK function.
- Retrieve a list of available SDK functions. *(NEW)*
- Retrieve a specific SDK function.
- Delete a specific SDK function.
- Modify existing SDK functions. *(NEW)*
- Publish/Unpublish a specific SDK function to/from 5G App & Service Catalogue *(NEW)*
- Import SDK functions from VNFDs in 5G App & Service Catalogue at start-up and periodically:
  - Create SDK function when new VNF packages are present in 5G app & service catalogue
  - Remove SDK function/s when corresponding VNF package is not found (e.g. VNFpackage removed on 5G app & catalogue and to be removed from SDK as well).
- Define a list of monitoring parameters related to an SDK function: *(NEW)*
  - Creation of parameters. *(NEW)*

    - Modification of parameters. *(NEW)*
    - Deletion of parameters. *(NEW)*

SDK Services

- Create a new SDK service.
    - A service might be composed by different functions and/or other services.
- Retrieve a list of available SDK services.
- Retrieve an SDK service.
- Delete an SDK service.
- Modify existing SDK services.
- Publish/Unpublish SDK services to/from 5G App & Service Catalogue.
- Define a list of monitoring parameters related to an SDK service:
    - Creation of parameters.
    - Modification of parameters.
    - Deletion of parameters.
- Define a list of action rules related to an SDK service:
    - Creation on action rules.
    - Modification of action rules.
    - Deletion of action rules.

Multi-tenancy management *(NEW)*

- *SDK services and SDK functions can now belong to a project mapped 1:1 with a slice created by slice-manager (NEW)*
    - Create a new project. *(NEW)*
    - Retrieve a list of available projects. *(NEW)*
    - Retrieve a specific project. *(NEW)*
    - Delete a specific project. *(NEW)*
    - Add/remove a user to/from a project. *(NEW)*
    - Retrieve a list of available projects per user. *(NEW)*

AAA

5GCity platform component for Authentication and Authorization of Platform users.

- Integration of state-of-the-art **authentication and authorization** in front of all interfaces exposed by the platform components, based on Keycloak and Gravitee.
- Support **rules for data access security** required by multi-tenancy features of the neutral hosting model.

Slice Manager

Manage infrastructure owner resources.

- Create, read, and delete "partitionable" resources that shall be offered by the infrastructure owner/neutral host for the creation of slices, namely:
    - Compute resources.
    - Physical network resources (for the network interfaces of the above compute resources)
    - Access network resources (currently only WiFi APs)
    - Network link resources (for the links that interconnect compute and access network resources)

Manage chunking/partitioning of infrastructure owner resources

- Create, read, and delete chunks (i.e., partitions) of the previously listed resources, by specifying quotas, i.e., what part/percentage of the whole resource a chunk shall include/reserve. According to the supported resources (see previous feature), the following chunks are currently supported:
    - Compute chunks (provisioned based on OpenStack projects)
    - Physical network chunks (provisioned based on OpenStack physical networks)
    - Access node chunks (currently only for WiFi access)
    - Physical link chunks (as portion of the bandwidth offered by a physical link, currently not enforced by an SDN controller)

Manage slices

- Create, read, and delete slices, which are defined as collections of previously listed chunks, belonging to a specific slice user and potentially having specific characteristics.
- Activate of the RAN ingredients of a slice on-demand after its creation. This involves: *(NEW)*
    - The automated launching of vEPC (virtual Evolved Packet Core) instances if the slice includes an LTE/5G RAN part. *(NEW)*
    - The automated launching of a DHCP service if the slice includes a WiFi part. *(NEW)*
    - The appropriate matching of VLAN tags used in the core network with those used in the RAN. *(NEW)*

Manage slice users

- Create, read, update, delete (CRUD) slice users, who can manage slices and deploy network services on them.

Manage network services (on slices) *(NEW)*

- Create, read, and delete network services, including deploying and un-deploying network service instances on specific slices. *(NEW)*

Integrate intelligent resource placement for service instances *(NEW)*

- For the aforementioned network services an optimal deployment is retrieved by providing slice and service info from the placement module, and then appropriately triggering the placement-aware API of the MTO, including all the required placement details. *(NEW)*

Infrastructure Abstraction *(NEW)*

Provide abstraction of multiple underline RAN controllers and technologies to the Slice Manager. ***(NEW)***

- Create, read, update, delete (CRUD) RAN controller profiles. *(NEW)*

Manage a catalogue of virtualized network resources. ***(NEW)***

- Create, read, update, delete (CRUD) a dictionary of virtualized resources. *(NEW)*

Delegate virtualized network resources requests from the Slice Manager to the controllers. ***(NEW)***

- Receive and retransmit requests of virtualized network resource to/from different RAN controllers. *(NEW)*

MTO

Provide abstraction in front of multiple underlying orchestrators:

- Delegate service lifecycle management calls to ETSI OpenSourceMano (OSM).
- Retrieve MEC application descriptors associated with instantiated network services from a catalogue and pass them to a MEC platform to be applied.

Implement fine-granular placement instructions for the service instances: ***(NEW)***

- Translation of placement instructions into configuration commands as required by the NFVO in order to realize the desired placement. *(NEW)*

MEAO

Manage Mobile Edge (ME) platforms

- Create, read, update, delete (CRUD) operations, allows the definition and lifecycle management of ME platforms running on different mobile edge hosts

Manage ME applications

- Create, read, update, delete operations enabling:
  - Registration of an ME app in any ME platform registered with this MEAO.
  - Removal of a given ME app from any ME platform registered with this MEAO.
  - Update of a given ME app from any ME platform registered with this MEAO.
  - Discovery of ME Apps inside any ME platform registered with this MEAO.

Manage MEC services across ME platforms

- Create, read, update, delete operations enabling:
  - Register a service to any ME platform registered with this MEAO.
  - Removal of a given service from any ME platform registered with this MEAO.
  - Update of a given service from any ME platform registered with this MEAO.
  - Service discovery within any ME platform registered with this MEAO.

Manage the Domain Name Service (DNS) rules across ME platforms

- Create, read, update, delete operations, providing a dynamic DNS service for a given ME platform.

MEC-Platform

Manage MEC services

- Create, read, update, delete operations exposed by the ME platform enabling:
  - Register a service in this platform
  - Removal of a given service from this platform
  - Update of a given service in this platform
  - Service discovery within this platform

Manage DNS (Domain Name Service) rules

- Create, Read, Update, Delete operations, providing a dynamic DNS service for all ME apps connected to this ME Platform.

Manage ME app notifications

- The ME Platform is able to notify other ME Apps or Services about lifecycle management changes of a given ME app.

OSM

As NFVO core engine, we use OSM Release 6 *(NEW)* by referring the official documentation in https://osm.etsi.org/ to describe its features.

- Note 1: The 5GCity extensions to ETSI OSM release 5 (which can be found in the modified OSM version of the 5GCity GitHub repository) enabled the support of fog05 (https://projects.eclipse.org/proposals/eclipse-fog05) as a VIM, in order to add MEC features required by the 5GCity platform. These additions (5GCity contributions) have been integrated in OSM release 6, so now we are just using release 6 unmodified.
- Note 2: 5GCity partners have contributed the OSM WIM plugin, which was already included also in the official OSM Release 5 and can be used to implement part of the Infrastructure abstraction and WAN resource manager of 5GCity.

Monitoring

Manage monitored nodes

- Create, read, update, and delete operations allow the definition of a resource that is exposing monitoring parameters.

Manage monitored jobs (exporters)

- Create, read, update, and delete operations allow the definition of a job as the group of monitoring parameters to be collected, e.g.:
  - NODE (VM related metrics).

Manage monitored services

- Create, Read, Update, Delete operations that allow the definition of the set of resources and jobs that are used to monitor a service.

Expose collected metrics

- Query monitoring parameters, e.g.:
  - Read all system (VM) metrics for a resource
  - Read all system (VM) metrics for all the resources of service

Support for custom metrics

- Register and query custom (service-specific) metrics.

View collected metrics

- Expose monitoring parameters through a local Dashboard, grouped by service and/or resource.

Alerting support (partly implementing the 5GCity SLA Manager functionality) ***(NEW)***

- Create Alert Rules to define thresholds and alerts *(NEW)*
- Send Alerts to a custom endpoint (e.g.: Dashboard) *(NEW)*
- Get the list of active alerts *(NEW)*

Resource placement *(NEW)*

Calculate the **optimal placement of VNFs** that compose a network service by taking into consideration the network topology and resources. *(NEW)*

- Register a placement request with a given network service and slice topology *(NEW)*
- Retrieve a list with optimal VDU to chunk mapping for a network service *(NEW)*

3. Installation requirements and procedures

Pre-requisites

The 5GCity Scalable Orchestrator software components depend on lower-layer controllers as documented in the WP3 deliverables. As a result, the 5GCity Orchestrator requires:

- OpenStack as the core Virtual Infrastructure Manager (VIM). In case this is not already installed, please follow the official installation instructions for the "Queens" version to install and test it properly. After having installed OpenStack, the Slice Manager needs to be configured in order to use the installed instance(s) by following the Slice Manager documentation.
- An SDN RAN controller, currently using a proprietary implementation of i2CAT, which needs to be installed by following the procedure described at https://github.com/5GCity/5GCity-infrastructure-abstraction/blob/1.0/RAN-Controller.md.

Requirements

The installation of the 5GCity Orchestrator requires 7 Virtual Machines (in addition to any VMs used for the VIM and the RAN Controller).

The hardware requirements of each VM and the software modules that will be installed on it are listed in Table 1. Note that the components of VM5 can easily run on separate VMs (following the documentation of the 5GCity platform), but the table intends to reflect the minimal requirements. For example, in our testbed they have been deployed on the same VM in the Lucca infrastructure but on separate VMs in the Barcelona and Bristol infrastructures.

Table 1. Profiling of the 5GCity Scalable Orchestrator Virtual Machines.

Host	*5GCity modules hosted*	*Requirements (recommended)*
VM1	**Dashboard** **SDK-GUI** **AAA**	Ubuntu 16.04 LTE, 2 vCPU, 4GB RAM, 20GB vHDD
VM2	**SDK**	Ubuntu 16.04 LTE, 2 vCPU, 4GB RAM, 20GB vHDD, with the following software installed: • *systemd, git, postgresql-server, and java 8 JDK* • Maven 3.3.9
VM3	**Infrastructure Abstraction**	Ubuntu 16.04 LTE, 2 vCPU, 4GB RAM, 20GB vHDD
VM4	**Monitoring**	
VM5	**Slice Manager** **MTO**	

	Resource Placement	
VM6	MEAO MEC-platform	
VM7	OSM	

Installation Steps

1. Install OSM in VM7 by following the instructions at: https://osm.etsi.org/wikipub/index.php/OSM_Release_SIX#Install_OSM_Release_SIX
   - NOTE: If you use release 5 (NOT RECOMMENDED), then in order to support the fog05/MEC connector (optional), please build the container images for RO, NBI and Keystone using the (modified) code in the 5GCity-OpenSourceMano GitHub repository (using the RO and NBI branches at https://github.com/5GCity/5GCity-OpenSourceMano), and then replace the respective containers in the docker stack file of OSM (/etc/osm/docker/docker-compose.yaml) with the ones that you just built.
2. Install the MEAO and the MEC-platform on VM6 by running the script: https://github.com/5GCity/5GCity-multi-tier-orchestration/blob/0.2/install_meao.sh
   - For a breakdown, further explanations, and troubleshooting of this script, please refer to https://github.com/5GCity/5GCity-multi-tier-orchestration/blob/0.2/MEAO.md.
3. Install the Slice Manager, the MTO, and the Resource Placement components in VM5 by following the installation instructions that can be found at https://github.com/5GCity/5GCity-slice-manager/blob/v1.3/doc/deployment/README.md, https://github.com/5GCity/5GCity-multi-tier-orchestration/blob/0.2/README.md, and https://github.com/5GCity/5GCity-resource-placement/blob/1.0/README.md, respectively. Note that these instructions include the configuration of these components to use the previously installed OpenStack, RAN Controller, MEAO, MEC-platform, and OSM.
4. Install monitoring in VM4 by following the instructions at https://github.com/5GCity/5GCity-monitoring/blob/v2.2/README.md
5. Install the Infrastructure Abstraction in VM3 by following the installation instructions at XXX. Note that this is not required if all RAN elements are controlled by the i2cat RAN controller.
6. Install the SDK in VM2 by following the instructions linked at https://github.com/5GCity/5GCity-SDK/blob/0.8/README.md.
7. Install the Dashboard (which includes the SDK-GUI) and the AAA in VM1 by following the instructions at https://github.com/5GCity/5GCity-Dashboard-new/blob/1.0/README.md and https://github.com/5GCity/5GCity-AAA/blob/1.0/README.md respectively.

Usage

A high-level understanding of the usage of the main user-facing components, namely the Dashboard and the SDK, is provided in the Appendices of this document.

Further, each 5GCity Orchestrator component provides its own (REST) API, which can be studied and used in order to exploit the features described in section 2, based on API documentation files (e.g. swagger), mini-tutorials, and README files, as indicated in the following list:

- Dashboard:
  - See Appendix A
- SDK:
  - See Appendix B
- Slice Manager:
  - Swagger documentation of the API is available at port 8989 of VM5 after having installed and started the Slice Manager on this VM.
  - Mini tutorial about how to use this API for slice creation and basic testing is included in https://github.com/5GCity/5GCity-slice-manager/blob/v1.3/README.md
- MTO:
  - Links to swagger documentation are provided inside https://github.com/5GCity/5GCity-multi-tier-orchestration/0.2/master/README.md
- MEAO:
  - Implements in a proprietary way the Mm1 interface of the ETSI MEC specification (NOTE: Mm1 interface in our system is the reference point between the MTO and the MEAO, it is used for triggering instantiation and termination of MEC applications and services in the MEC system as well as performing MEC platforms management).
- MEC-platform:
  - Implements in a proprietary way the Mp1 and Mm5 interfaces of the ETSI MEC specification (NOTE: Mm5 in our system is the reference point between the MEC platform and the MEAO. It is used for MEC Platform configuration, configuration of applications, DNS rules and services.)
- Monitoring:
  - See tutorial at https://github.com/5GCity/5GCity-monitoring/blob/v2.2/doc/5GCity-MonitoringSystem-Overview_v2.docx
- Resource Placemet:
  - See overview and API descrtiption at https://github.com/5GCity/5GCity-resource-placement/blob/1.0/README.md
  - Swagger documentation is available after having installed and started Resource placement on VM5 at http://localhost:3800/

4. Functional Validation tests

This section contains information about functional validation tests executed on core components of the 5GCity orchestrator. This involves namely the SDK and the Slice Manager, because the functional tests of the Slice Manager trigger almost all other modules (MTO, Resource Placement, OSM, Monitoring), while the correct functionality of the Dashboard and the AAA is validated mainly in the end-to-end functional tests of the platform, which will be presented in Deliverable D5.2.

SDK

The following tables show the time elapsed (expressed in milliseconds) to carry out various requests on both SDK Functions and SDK Services. We counted the time elapsed from the receipt of the request by the SDK until its completion. While *Create*, *Retrieve* and *Delete* requests only involve the SDK components, *Publish* and *Unpublish* actions involve also the 5G App & Service Catalogue (which is an orchestrator-external component) and OSM. For this reason, we consider a *Publish* request completed when the corresponding package is on-boarded on OSM (which is the last component of the chain) and we consider an *Un-publish* request completed when the corresponding package is deleted from OSM.

For each type of request, we repeated the test 10 times and then we calculated the average of the results.

Tests were carried out in the Nextworks lab with the SDK and the 5G App & Service Catalogue running on the same VM with 2 vCPU and 4 GB of vRAM, and OSM running on a virtual machine with 4 vCPU and 8 GB vRAM.

Table 2. SDK Function validation test results (ms)

Repetition	*Create*	*Retrieve*	*Delete*	*Publish*	*Un-publish*
#1	40	13	27	916	153
#2	31	9	23	700	131
#3	28	8	31	449	94
#4	34	10	25	637	117
#5	32	9	30	542	105
#6	41	14	32	815	138
#7	25	7	24	667	95
#8	27	7	28	513	101
#9	30	11	26	485	129
#10	33	12	30	789	136
AVG	**32,1**	**10**	**27,6**	**651,3**	**119,9**

Table 3. SDK Service validation test results (ms)

Repetition	*Create*	*Retrieve*	*Delete*	*Publish*	*Un-publish*
#1	51	16	27	656	156
#2	65	14	19	303	128
#3	70	14	35	517	131
#4	57	13	30	401	117
#5	63	18	22	388	144
#6	52	15	27	455	120
#7	55	12	32	364	115
#8	68	15	21	503	159
#9	52	16	20	327	139
#10	66	12	29	359	134
AVG	**59,22**	**14,5**	**26,2**	**427,3**	**134,3**

Slice Manager

To validate the different functionalities of the Slice Manager module in the 5GCity platform, the following Unit Tests were performed and successful assertions were obtained for all of them.

Table 4. Slice manager functional validation test results

Test	Unit Test description/scope	Result
Test #1	Test successful registration of a compute with and without user provided quotas	PASSED
Test #2	Test successful registration of a physical network with and without user provided quotas	PASSED
Test #3	Test successful registration and configuration of a RAN Infrastructure with and without user provided radio parameters	PASSED
Test #4	Test successful registration of a physical link	PASSED
Test #5	Test successful creation of a compute chunk considering feasible requirements in the user provided input and failed attempt when the provided requirements cannot be satisfied.	PASSED
Test #6	Test successful creation of a network chunk considering feasible requirements in the user provided input and failed attempt when the provided requirements cannot be satisfied.	PASSED
Test #7	Test successful creation of a radio chunk considering the selection of different combinations of radio access technologies (WiFi and LTE).	PASSED
Test #8	Test successful creation of a link chunk considering feasible requirements in the user provided input.	PASSED

Test #9	Test successful creation of a slice considering the inclusion of different combinations of chunk classes (compute, network, radio and link).	PASSED
Test #10	Test successful activation of the RAN components included in some slice and the successful online deployment of the related servers (secondary vEPC and DHCP).	PASSED
Test #11	Test successful instantiation of a network service considering different number of service components (VDUs and VLDs).	PASSED
Test #12	Test successful multi-site deployment considering the instantiation of a network service over two compute chunks with different availability zones.	PASSED
Test #13	Test successful exchange of monitoring data between the deployed VMs and the Prometheus server.	PASSED

Dashboard

To attest the implementation of the Dashboard component, the following tests covering its functionalities were performed and a summary of the results is provided below.

Table 5. Dashboard functional validation test results

Test	Unit Test description/scope	Result
Test #1	Test successful login in the platform using Infrastructure Owner role	PASSED
Test #2	Test successful login in the platform using Slice Requester role	PASSED
Test #3	Test successful login in the platform using Developer role	PASSED
Test #4	Test successful logout of the platform	PASSED
Test #5	Test forbidden access when accessing with wrong credentials	PASSED
Test #6	Test successful infrastructure visualisation with Infrastructure Owner role	PASSED
Test #7	Test successful compute resource addition with Infrastructure Owner role	PASSED
Test #8	Test successful compute resource edition with Infrastructure Owner role	PASSED
Test #9	Test successful network resource addition with Infrastructure Owner role	PASSED
Test #10	Test successful network resource configuration with Infrastructure Owner role	PASSED
Test #11	Test successful infrastructure visualization with Slice Requester role	PASSED
Test #12	Test successful slice visualisation with Slice Requester role	PASSED
Test #13	Test successful slice creation with Slice Requester role	PASSED
Test #14	Test successful slice removal with Slice Requester role	PASSED
Test #15	Test successful network service catalogue visualisation with Slice Requester role	PASSED
Test #16	Test successful network service instantiation with Slice Requester role	PASSED
Test #17	Test successful network service removal with Slice Requester role	PASSED
Test #18	Test successful SDK tenant creation with Infrastructure Owner role	PASSED

Test #19	Test successful user addition to tenant with Infrastructure Owner role	PASSED
Test #20	Test successful network service visualisation with Slice Requester role	PASSED
Test #21	Test successful function visualisation with Slice Requester role	PASSED
Test #22	Test successful function creation with Slice Requester role	PASSED
Test #23	Test successful function edition with Slice Requester role	PASSED
Test #24	Test successful function removal with Slice Requester role	PASSED
Test #25	Test successful network service creation with Slice Requester role	PASSED
Test #26	Test successful network service edition with Slice Requester role	PASSED
Test #27	Test successful network service removal with Slice Requester role	PASSED
Test #28	Test successful network service publish to 5GCity Network Service catalogue with Slice Requester role	PASSED

5. Conclusions

This deliverable complements the software prototype of the 5GCity Orchestrator (Final Release). The document provides high-level release notes of the software produced and links to live documentation repositories related to the various components made available on GitHub.

Key design references for this document are the Deliverables D2.2, D4.1, D2.3, D4.2, which have been previously published.

This document serves the purpose to summarize high-level mappings of software modules to conceptual architectural components, list supported features, list installation requirements, functional tests, and usage instructions.

Subsequent fixes on the various modules, will be directly reported in our software documentation material included in our GitHub repositories.

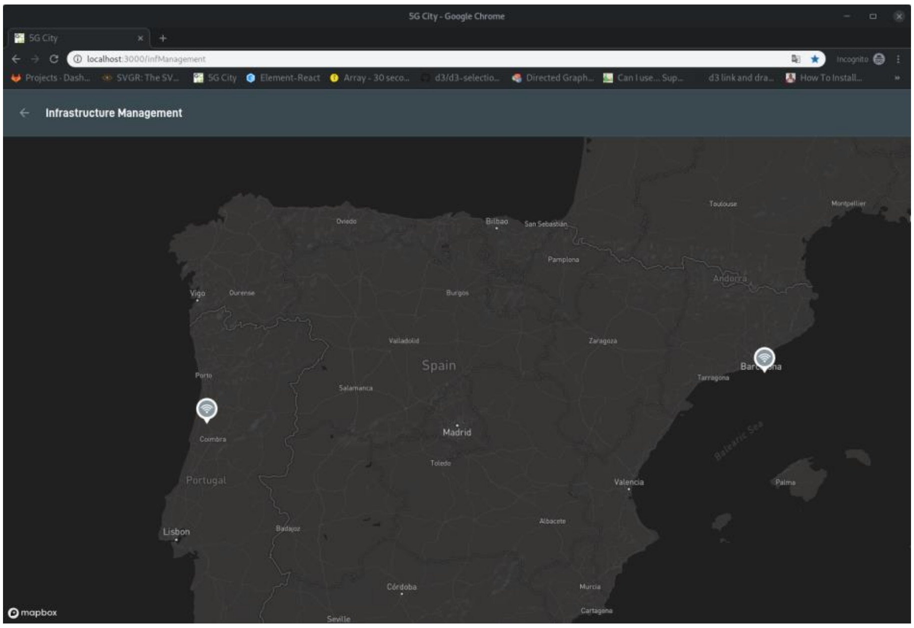
Abbreviations

AAA	Authentication, Authorization and Accounting
AP	Access Point
API	Application Programming Interface
CRUD	Create, Read, Update, Delete
DNS	Domain Name Service
ETSI	European Telecommunications Standards Institute
GB	Gigabyte
GUI	Graphical User Interface
JDK	Java Development Kit
ME	Mobile Edge
MEAO	Mobile Edge Application Orchestrator
MEC	Mobile Edge Computing
MTO	Multi-Tier Orchestrator
NBI	NorthBound Interface
NFVO	Network Function Virtualization Orchestrator
OSM	OpenSourceMano
RAM	Random Access Memory
RAN	Radio Access Network
REST	REpresentational State Transfer
RO	Resource Orchestrator
SDK	Service Development Kit
SDN	Software-Defined Networking
SLA	Service-Level Agreement
vCPU	Virtual Central Processing Unit
VDU	Virtual Deployment Unit
vEPC	Virtual Evolved Packet Core
vHDD	Virtual Hard Disk Drive
VIM	Virtual Infrastructure Manager
VM	Virtual Machine
VNFD	Virtual Network Function Descriptor
WAN	Wide Area Network
XML	Extended Markup Language

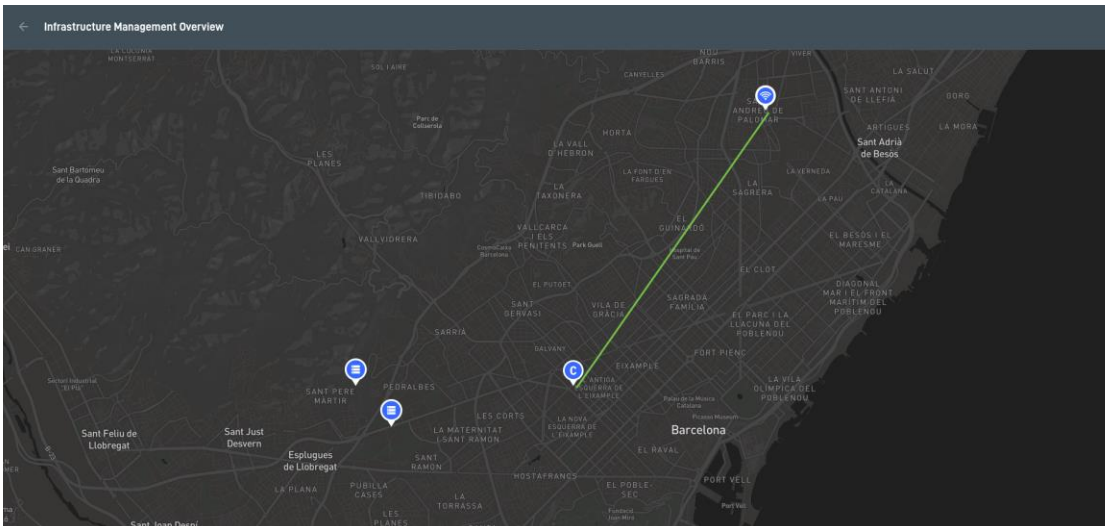
Appendix A. *Screenshots of the 5GCity Dashboard*

Infrastructure Management/Overview

Users with Infrastructure Owner role, by using the web solution can manage and visualise the physical resources currently provisioned in the 5GCity platform. By selecting the "Infrastructure Management" feature these users can see the locations in which resources (both compute and network) are currently provisioned.

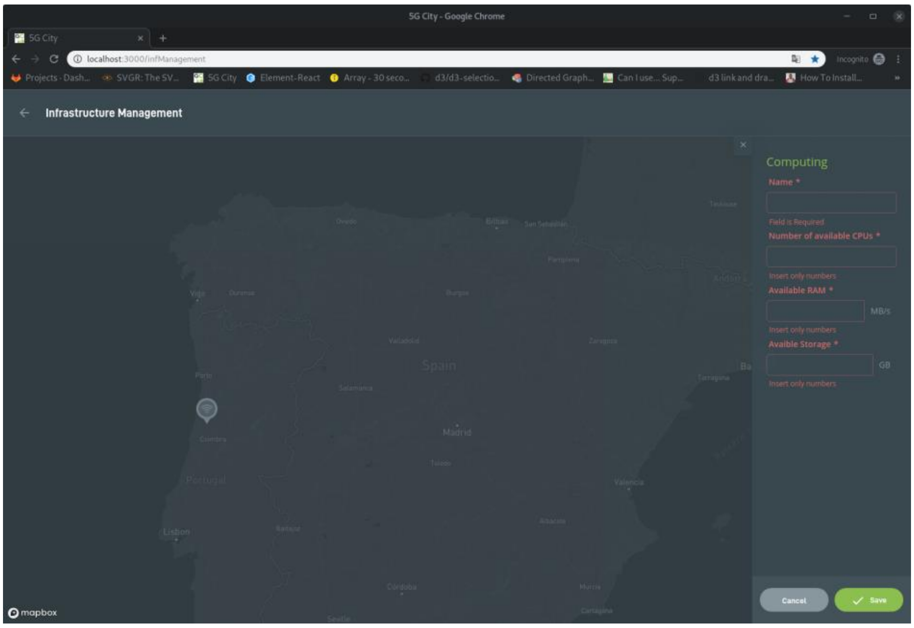


Different resources are represented by different icons allowing the user to quickly grasp the resource deployments available under a certain area. In the figure below we can see a city deployment with two compute resources, a RAN Controller and a access network resource.



Network access resources such as wifi or small cells are managed by a RAN Controller. To allow the user to quickly identify which RAN Controller is reponsible for managing which access network resources, 5Gcity dashboard links these elements as can be seen in the previous figure. From 5Gcity dashboard authorised users (Infrastructure Owners) are able to manage all these resources therefore being able to view/add/edit/remove computes, add/edit/remove RAN Controllers as well as configure network elements (Wifi or SmallCells) currently onboarded on 5Gcity.

In the image below, we can see an example focusing on compute edition, where the user selects to add a new compute visualizing the following screen:



Computing

Name *

Field is Required

Number of available CPUs *

Insert only numbers

Available RAM * MB/s

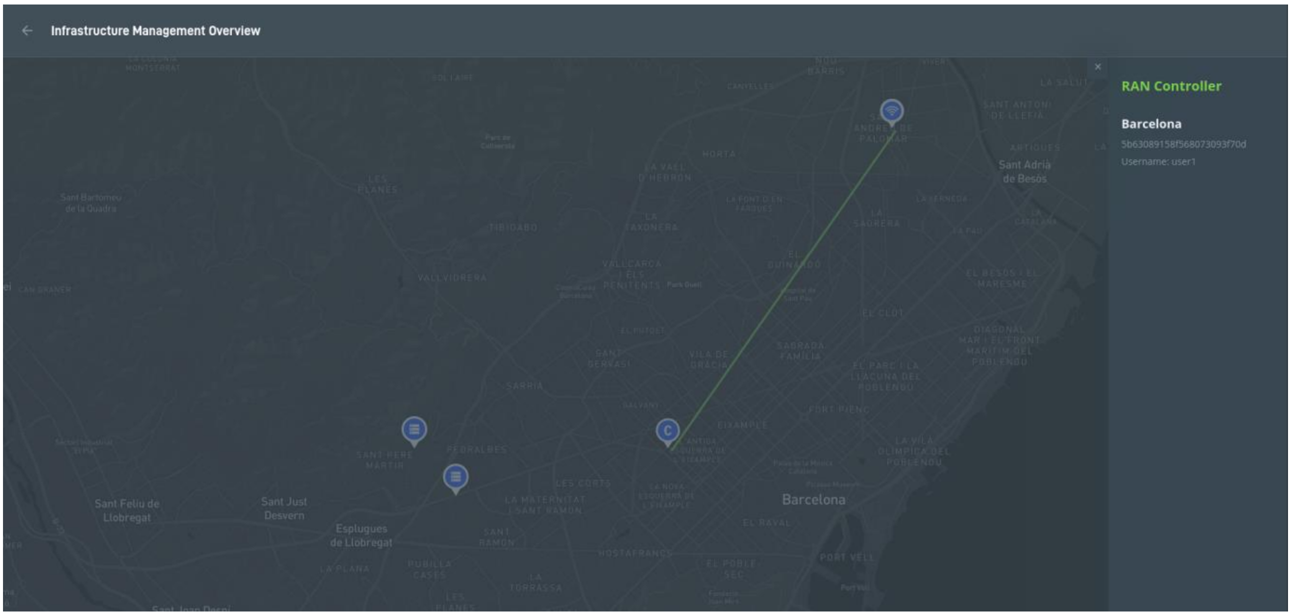
Insert only numbers

Avaible Storage * GB

Insert only numbers

Cancel ✓ Save

Through this interface, the user can add one compute resource providing a name as well as its hardware specifications. In the two figures that follow, we can see represented a RAN Controller as well as a Small Cell network element.



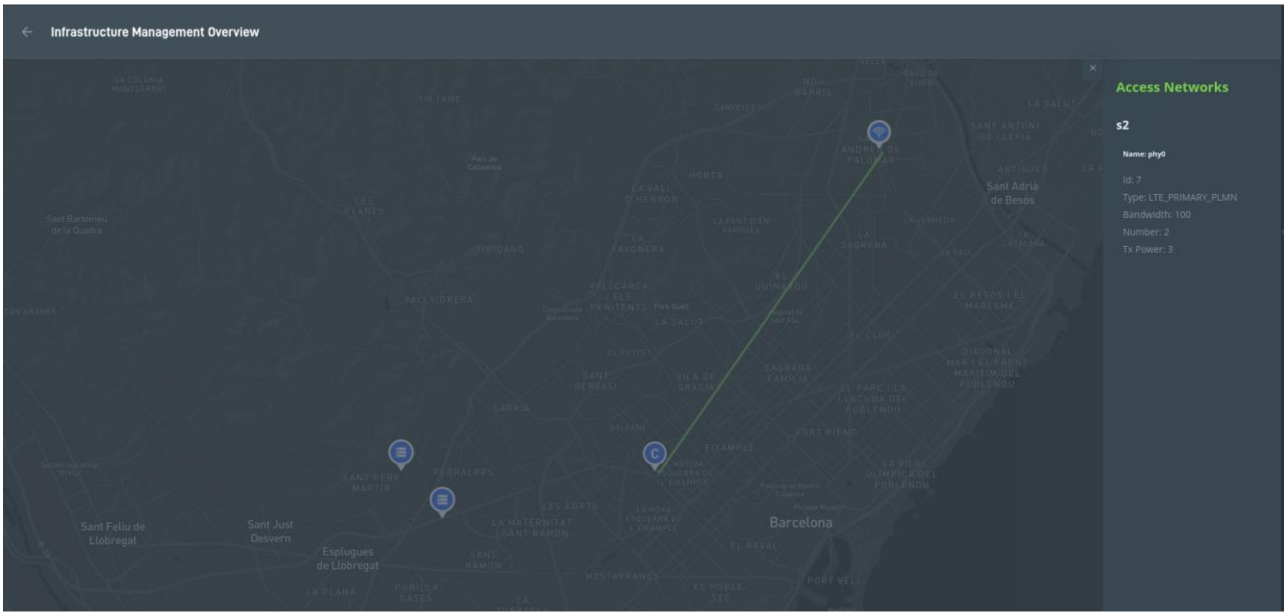
Infrastructure Management Overview

RAN Controller

Barcelona

5b63089158f568073093f70d

Username: user1



Infrastructure Management Overview

Access Networks

s2

Name: phy0

Id: 7

Type: LTE_PRIMARY_PLMN

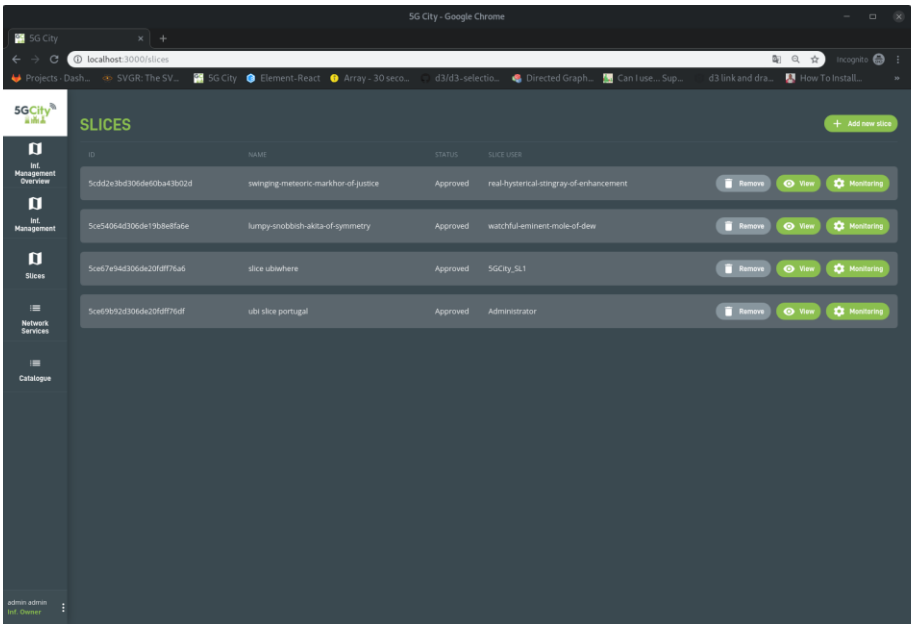
Bandwidth: 100

Number: 2

Tx Power: 3

Slice Management

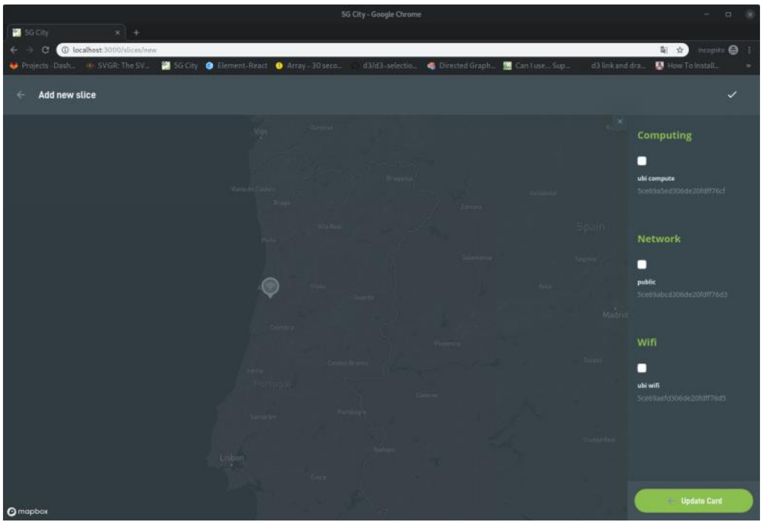
Users with access to infrastructure owner or slice requester role can list, create, edit, and delete slices directly in the 5GCity platform. It should be noted that while infrastructure owners are able to list and manage all slices created in the context of 5GCity, users with slice requester role are only able to list or manage slices created in the context of the tenant that they belong to. In the figure that follows, it is represented the slice list visualization page.



SLICES

ID	NAME	STATUS	SLICE USER	
5cdd2e3bd306de60ba43b02d	swinging-meteoric-markhor-of-justice	Approved	real-hysterical-stingray-of-enhancement	Remove View Monitoring
5ce54064d306de19b8e8fa6e	lumpy-snobbish-akita-of-symmetry	Approved	watchful-eminent-mole-of-dew	Remove View Monitoring
5ce67e94d306de20fdff76a6	slice ubiwhere	Approved	5GCity_SL1	Remove View Monitoring
5ce69b92d306de20fdff76df	ubi slice portugal	Approved	Administrator	Remove View Monitoring

To create or edit a slice, users have to select the resources by location that they intend to add to the targeted slice. By clicking in a location, users are able to add, edit or remove resources. By each one of the resource types users can add to a slice, a set of parameters is mandatory to configure as represented below:



Computing

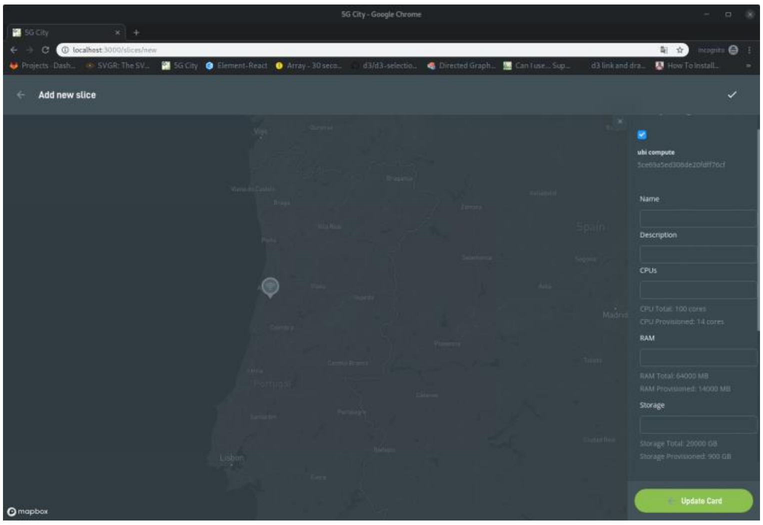
□ ubi compute
5ce69a5ed306de20fdff76cf

Network

□ public
5ce69abcd306de20fdff76d3

Wifi

□ ubi wifi
5ce69aefd306de20fdff76d5



Add new slice

☑ ubi compute
5ce69a5ed306de20fdff76cf

Name

Description

CPUs

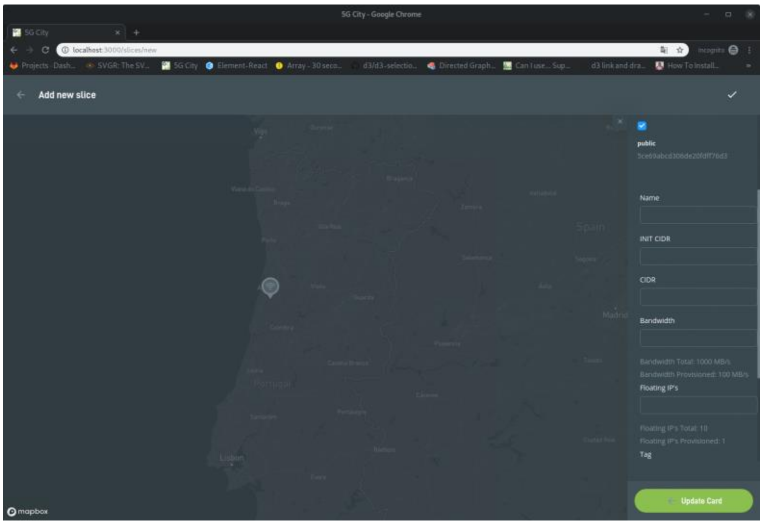
CPU Total: 100 cores
CPU Provisioned: 14 cores

RAM

RAM Total: 64000 MB
RAM Provisioned: 14000 MB

Storage

Storage Total: 20000 GB
Storage Provisioned: 900 GB



Add new slice

☑ public
5ce69abcd306de20fdff76d3

Name

INIT CIDR

CIDR

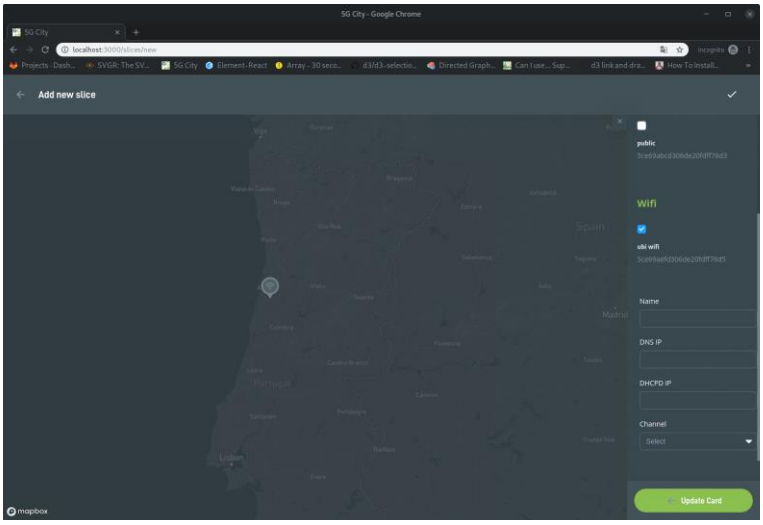
Bandwidth

Bandwidth Total: 1000 MB/s
Bandwidth Provisioned: 100 MB/s

Floating IP's

Floating IP's Total: 10
Floating IP's Provisioned: 1

Tag



Add new slice

□ public
5ce69abcd306de20fdff76d3

Wifi

☑ ubi wifi
5ce69aefd306de20fdff76d5

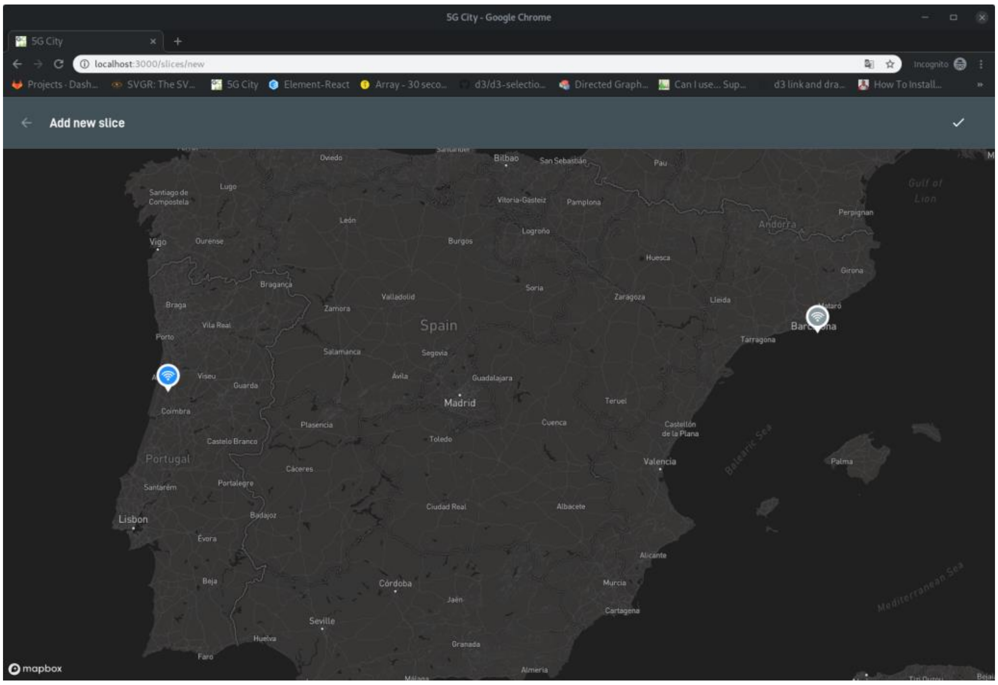
Name

DNS IP

DHCPD IP

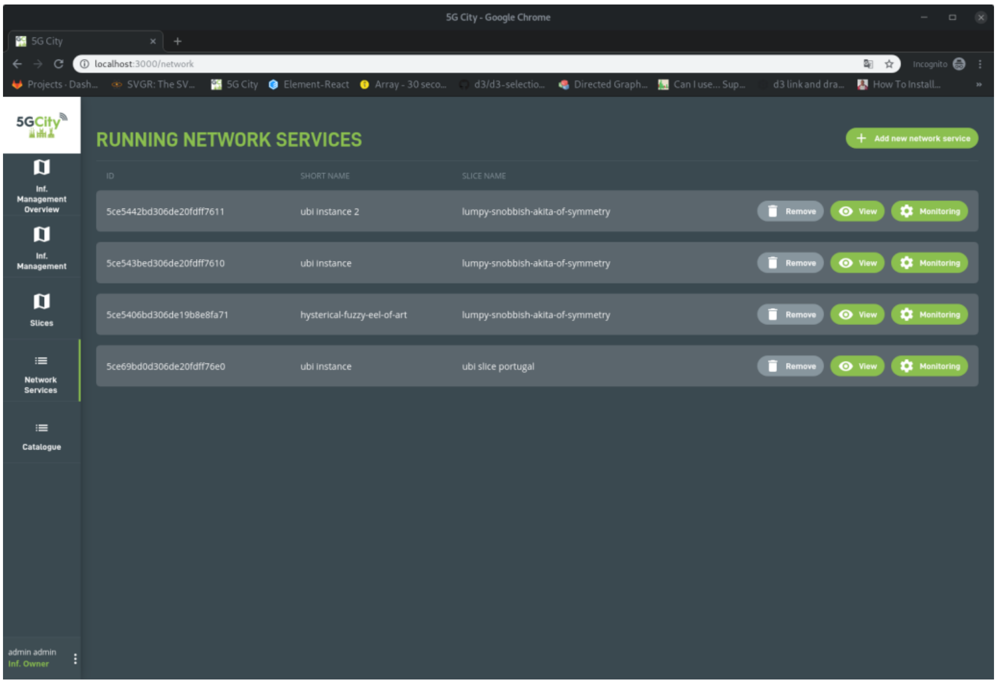
Channel
Select

In order to allow users to quickly visualize in which location a slice has resources configured, 5GCity platform marks with a different colour the locations that have at least one resource configured in the context of the current slice.



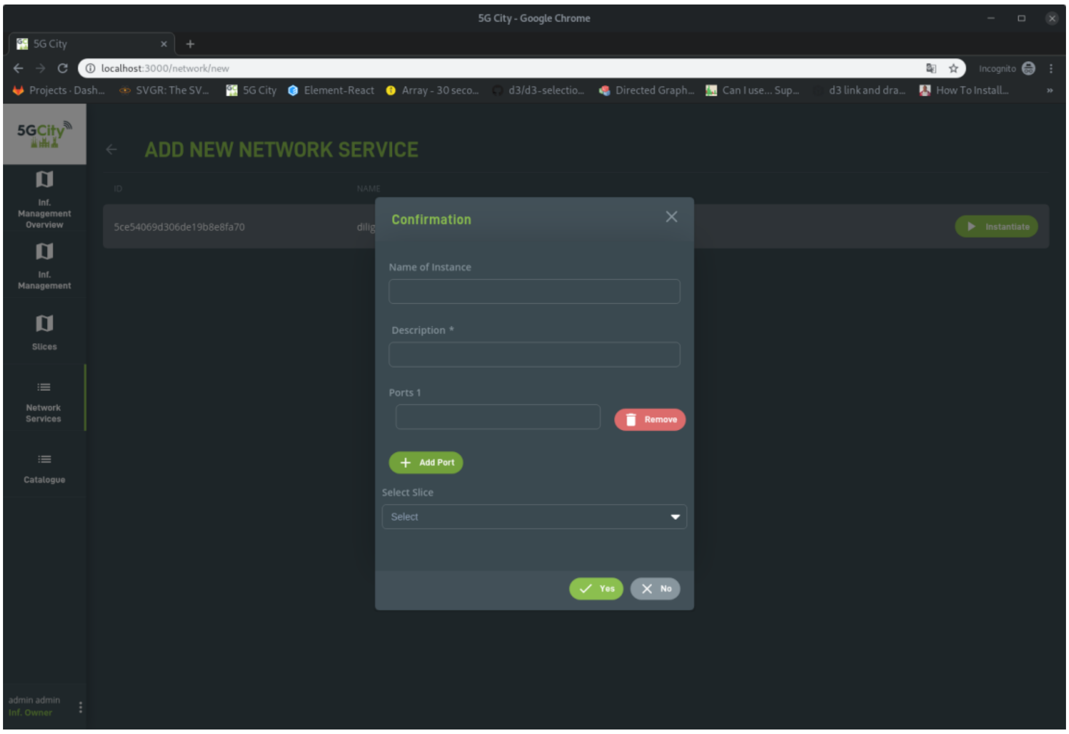
Service Management

User roles Infrastructure owner and slice requester besides being able to manage slices, are also able to manage services currently running in the context of each slice (based on the permissions they have in each slice). When instantiating a service, users are required to define in which slice the service is to be instantiated.



RUNNING NETWORK SERVICES

ID	SHORT NAME	SLICE NAME	
5ce5442bd306de20fdff7611	ubi instance 2	lumpy-snobbish-akita-of-symmetry	Remove View Monitoring
5ce543bed306de20fdff7610	ubi instance	lumpy-snobbish-akita-of-symmetry	Remove View Monitoring
5ce5406bd306de19b8e8fa71	hysterical-fuzzy-eel-of-art	lumpy-snobbish-akita-of-symmetry	Remove View Monitoring
5ce69bd0d306de20fdff76e0	ubi instance	ubi slice portugal	Remove View Monitoring


5G City - Google Chrome
5G City
localhost:3000/network/new
Projects - Dash...
SVGR: The SV...
5G City
Element-React
Array - 30 seco...
d3/d3-selectio...
Directed Graph...
Can I use... Sup...
d3 link and dra...
How To Install...
Incognito
5GCity
Inf. Management Overview
Inf. Management
Slices
Network Services
Catalogue
admin admin
Inf. Owner
ADD NEW NETWORK SERVICE
ID
NAME
5ce54069d306de19b8e8fa70
dilig
Instantiate
Confirmation
Name of Instance
Description *
Ports 1
Remove
Add Port
Select Slice
Select
Yes
No

Appendix B. *Screenshots and business logic of the SDK*

The 5GCity SDK is integrated with the 5GCity AAA in order to:

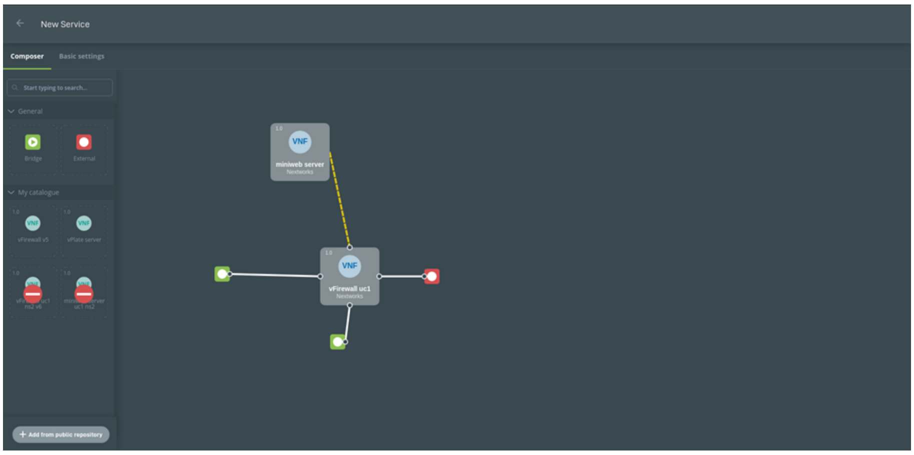
- Allow user to view and modify only owned and public SDK Services
- Allow user to view and modify only owned and public SDK Functions

Furthermore, the SDK business logic allows creation of new SDK Services based on access level through which it is possible to filter access to specific functions and services based on the access level granted to a user/group. For example, “Bronze users” (access level 3) cannot use Silver or gold SDK Functions and/or SDK Services to compose their service

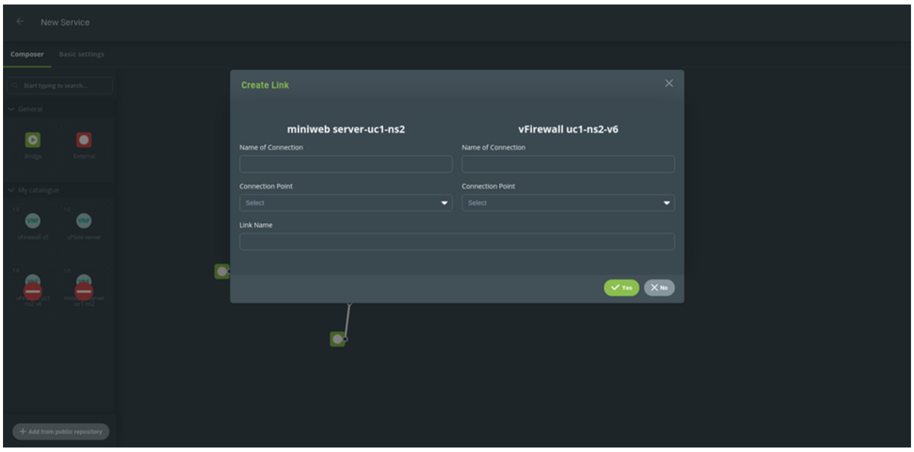
Access level	Symbolic Name	Description
0	Platinum	These users can see all the functions and services tagged with access level ≥ 0
1	Gold	These users can see all the functions and services tagged with access level ≥ 1
2	Silver	These users can see all the functions and services tagged with access level ≥ 2
3	Bronze	These users can see all the functions and services tagged with access level ≥ 3

The 5GCity SDK has on top the GUI, as part of the 5GCity Dashboard. The SDK User will be able through GUI to run all the operations described in SDK section.

The creation of a new SDK Service from the GUI is done by the user by dragging and dropping SDK functions and SDK services:



The SDK Components are then interconnected among them with links:



Create Link

miniweb server-uc1-ns2

Name of Connection

Connection Point: Select

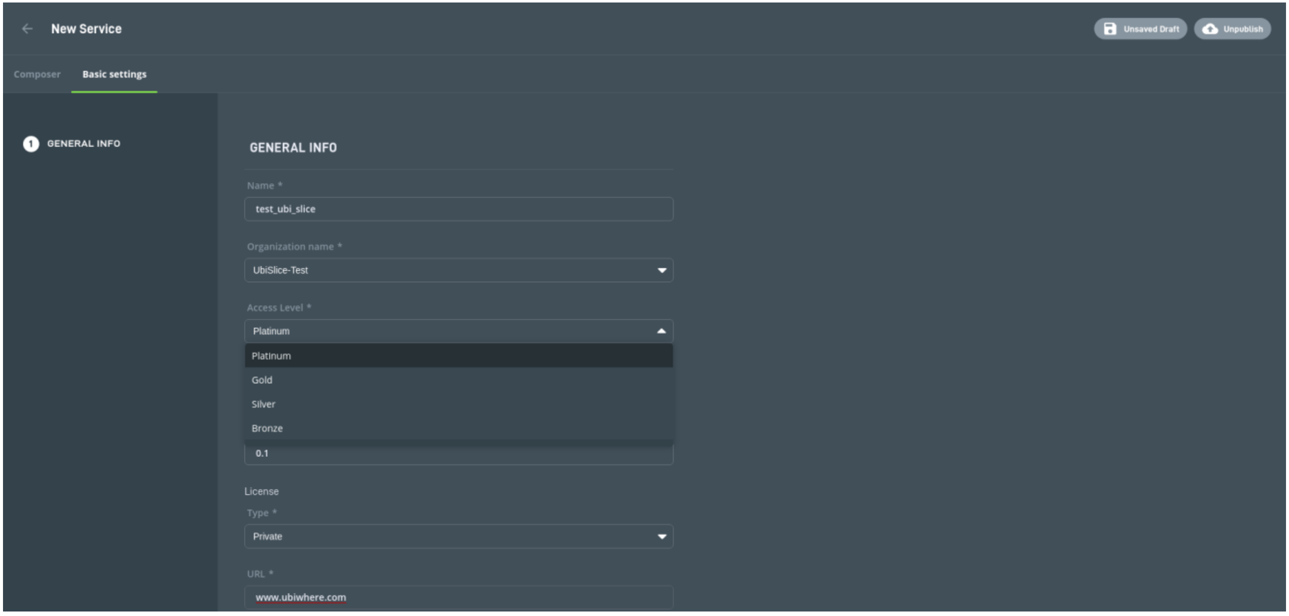
vFirewall uc1-ns2-v6

Name of Connection

Connection Point: Select

Link Name

Besides defining the service's topology, the user also has available a panel to define the service's basic properties:



GENERAL INFO

Name *: test_ubi_slice

Organization name *: UbiSlice-Test

Access Level *: Platinum

- Platinum
- Gold
- Silver
- Bronze

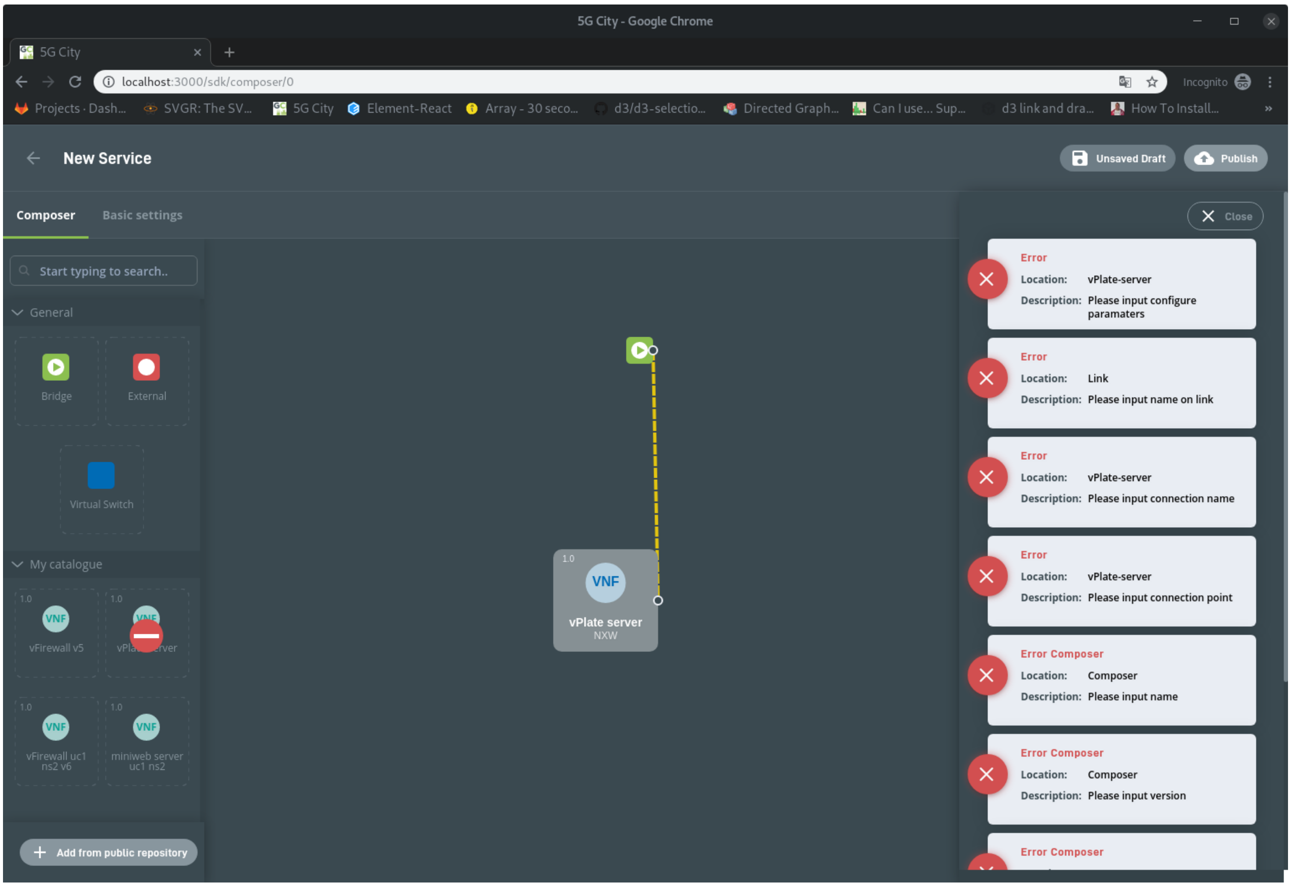
0.1

License

Type *: Private

URL *: www.ubiwhere.com

Before publishing the service to the catalogue, a validation routine is performed and in case of a missing or invalid input the user is informed and required to act before being able to proceed:



Error
Location: vPlate-server
Description: Please input configure paramaters

Error
Location: Link
Description: Please input name on link

Error
Location: vPlate-server
Description: Please input connection name

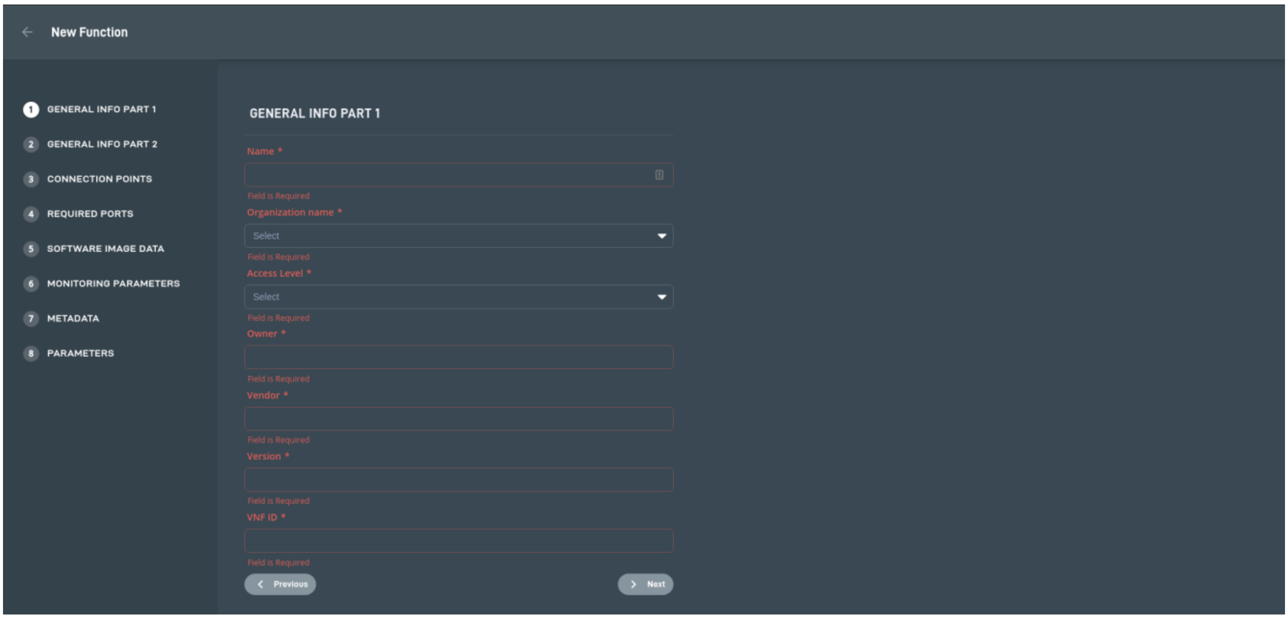
Error
Location: vPlate-server
Description: Please input connection point

Error Composer
Location: Composer
Description: Please input name

Error Composer
Location: Composer
Description: Please input version

Error Composer

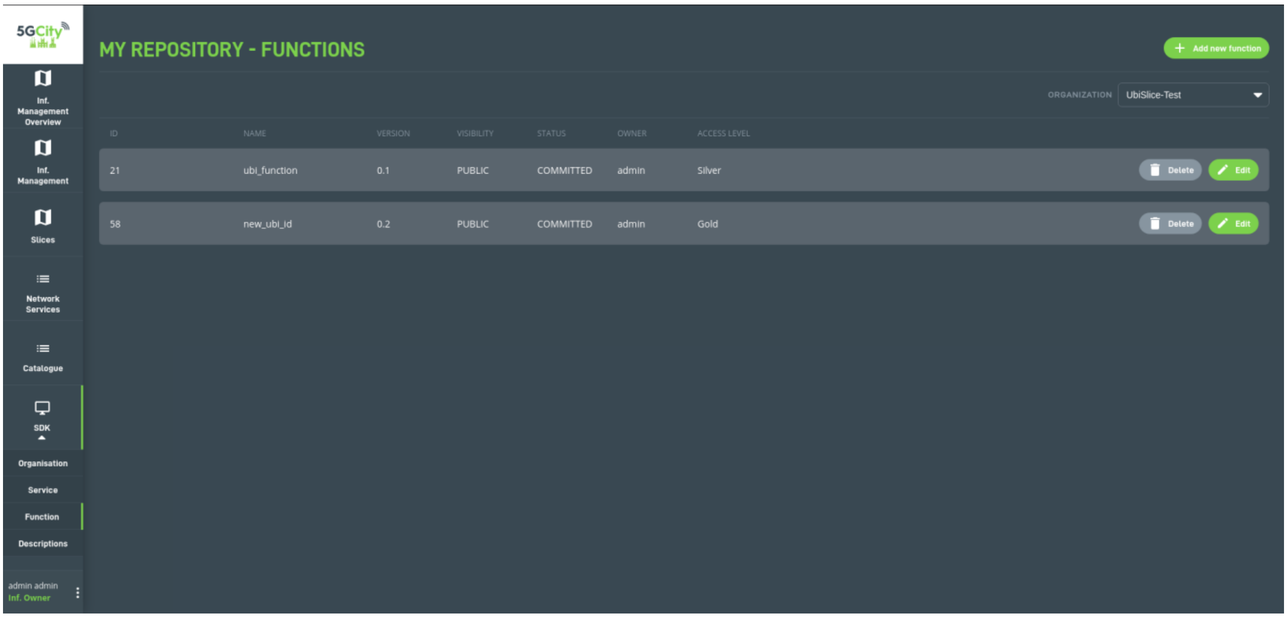
Function creation and edition is also supported by the graphical user interface as exemplified below:



GENERAL INFO PART 1

Name *
Field is Required
Organization name *
Select
Field is Required
Access Level *
Select
Field is Required
Owner *
Field is Required
Vendor *
Field is Required
Version *
Field is Required
VNF ID *
Field is Required

Through this interface, the user is able to list both services and functions which he has access to as exemplified in the figure below:

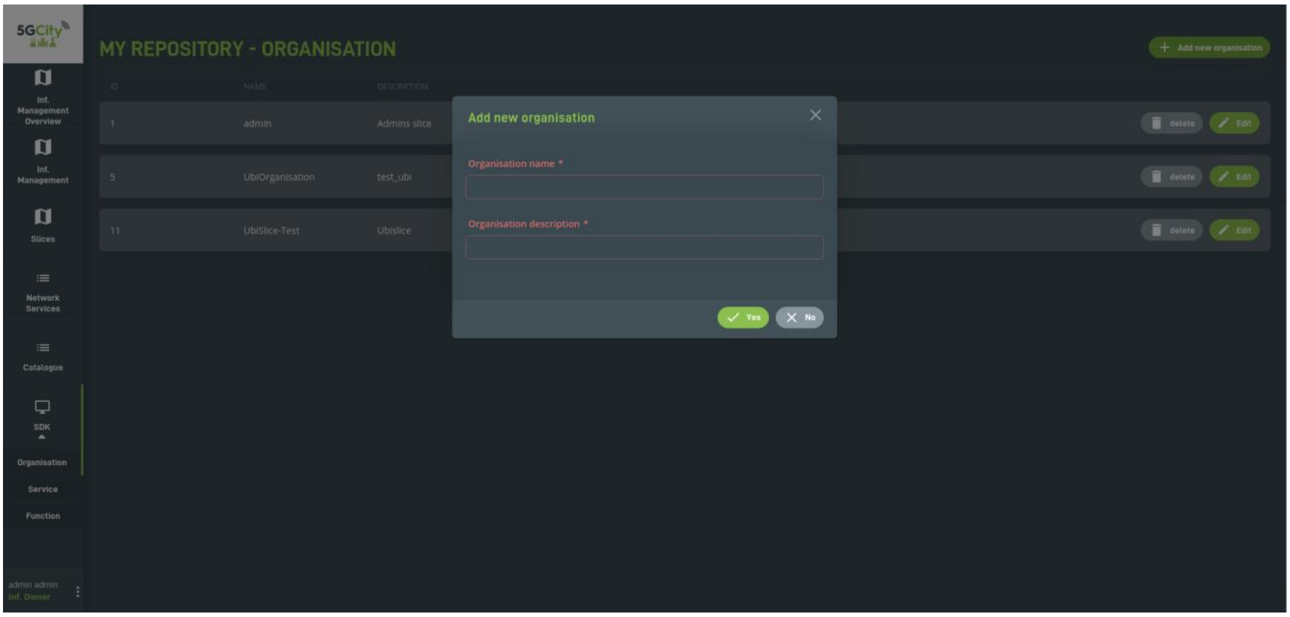


MY REPOSITORY - FUNCTIONS

ORGANIZATION: UbiSlice-Test

ID	NAME	VERSION	VISIBILITY	STATUS	OWNER	ACCESS LEVEL
21	ubi_function	0.1	PUBLIC	COMMITTED	admin	Silver
58	new_ubi_id	0.2	PUBLIC	COMMITTED	admin	Gold

Focusing on the multi-tenancy feature added to the SDK API, the graphical user interface as also accommodate this feature in the form of organizations. In the figures below, are represented the main screens associated with this feature:



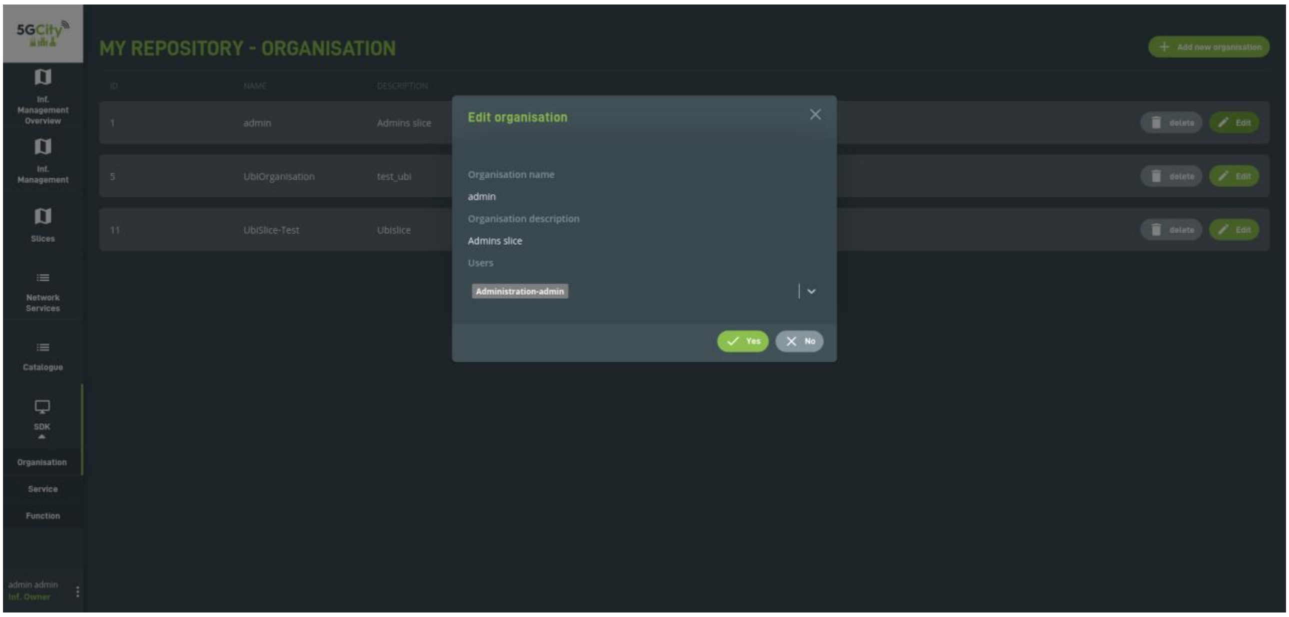
MY REPOSITORY - ORGANISATION

ID	NAME	DESCRIPTION
1	admin	Admins slice
5	UbiOrganisation	test_ubi
11	UbiSlice-Test	Ubislice

Add new organisation

Organisation name *

Organisation description *



MY REPOSITORY - ORGANISATION

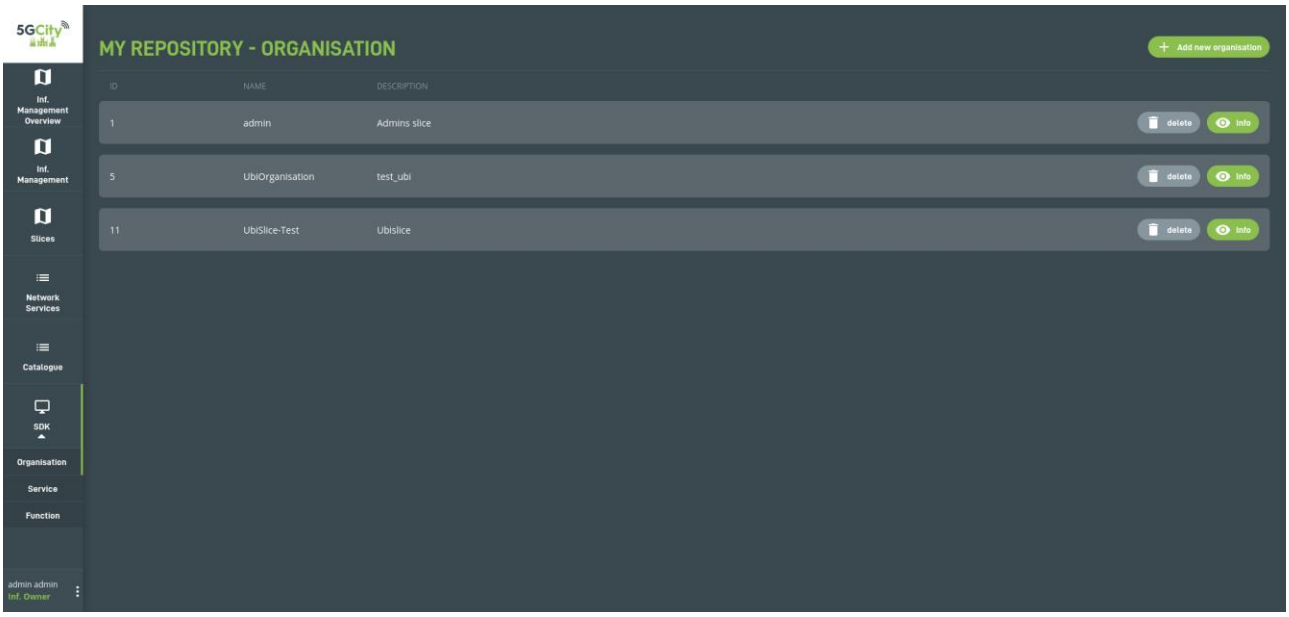
ID	NAME	DESCRIPTION
1	admin	Admins slice
5	UbiOrganisation	test_ubi
11	UbiSlice-Test	Ubislice

Edit organisation

Organisation name: admin

Organisation description: Admins slice

Users: Administration-admin



MY REPOSITORY - ORGANISATION

ID	NAME	DESCRIPTION
1	admin	Admins slice
5	UbiOrganisation	test_ubi
11	UbiSlice-Test	Ubislice

<END OF DOCUMENT>