



# 5G CITY

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

## D4.2 5GCity Scalable Orchestrator Interim Release

Dissemination Level	
<input checked="" type="checkbox"/>	PU: Public
<input type="checkbox"/>	PP: Restricted to other programme participants (including the Commission Services)
<input type="checkbox"/>	RE: Restricted to a group specified by the consortium (including the Commission Services)
<input type="checkbox"/>	CO: Confidential, only for members of the consortium (including the Commission Services)

Grant Agreement no: <b>761508</b>	Project Acronym: <b>5G CITY</b>	Project title: <b>5G CITY</b>
---	---------------------------------------	----------------------------------

Lead Beneficiary: I2CAT Foundation	Document version: <b>V1.0</b>
---------------------------------------	-------------------------------

Work package: WP4
-------------------

Deliverable title: Scalable Orchestrator Interim Release

Start date of the project: 01/06/2017 (duration 30 months)	Contractual delivery date: M24	Actual delivery date: 31/05/2019
--	-----------------------------------	-------------------------------------

<b>Editor name:</b> 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	Gabrielle 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
University of Bristol	UNIVBRIS	Carlos Colman-Meixner

## Change History

Version	Date	Partners	Description/Comments
0.1	12-04-2019	I2CAT	First draft including initial ToC
0.2	22-05-2019	I2CAT	Draft for review including inputs from all partners, previously gathered in GitHub and Confluence, along with all necessary connecting sections
0.3	27-05-2019	I2CAT	Consolidated document for review including all contents and Appendices
0.4	29-05-2019	NXW, UNIVBRIS	Enhanced formulation and performed minor structural suggestions
1.0	30-05-2019	I2CAT	Addressed reviews, added Abbreviations etc, and consolidated document

---

# 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

Executive Summary .....	6
1. Introduction .....	7
2. Main features of the 5GCity Scalable Orchestrator Interim Release .....	8
3. Installation requirements and procedures .....	13
4. Conclusion .....	16
Abbreviations .....	17
Appendix A. Screenshots of the 5GCity Dashboard .....	18
Appendix B. Screenshots and business logic of the SDK .....	32

## Figures

Figure 1. Software modules implementing the 5GCity Scalable Orchestrator Interim Release .....	7
--	---

## Tables

Table 1. Profiling of the 5GCity Scalable Orchestrator Virtual Machines. ....	13
---	----

---

## Executive Summary

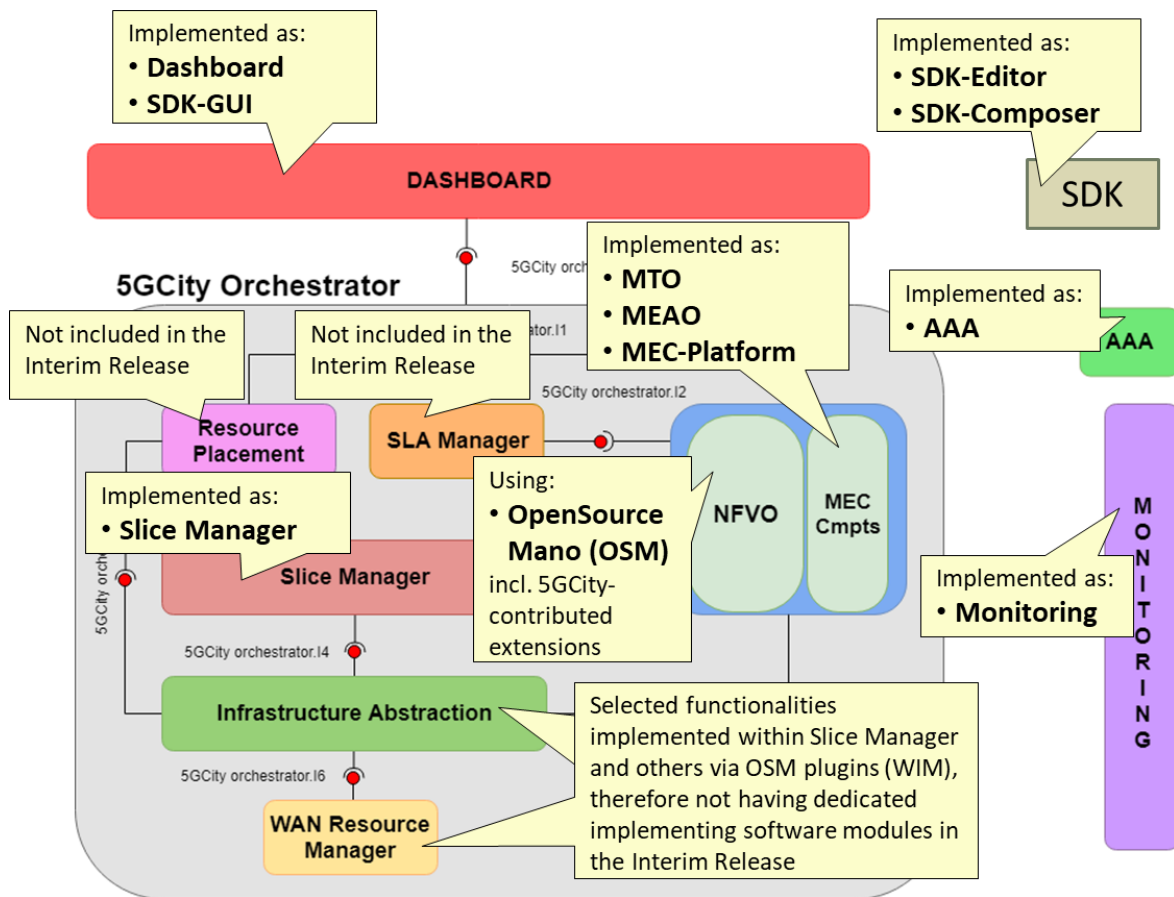
This Deliverable accompanies the software prototype of the 5GCity Scalable Orchestrator Platform by documenting the software components that have been included in the Interim Release, which are the main outcomes of work in tasks T4.1 and T4.2 until the end of the second Project Year. The overall 5GCity architecture has been described in Deliverable D2.2 while details of the logic of the 5GCity Orchestrator components have been elaborated in Deliverable D4.1. This document relates to the 5GCity Orchestrator architecture mainly in terms of software modules and provides information necessary to those who need to understand or use the developed prototype. Therefore, the deliverable is organized to describe:

- At first, which parts of the 5GCity Orchestrator architecture have been included in the Interim Release and how they map to actual software components.
- Then, the list of implemented features of each component, thus hinting to what can be achieved by using this prototype, which already serves as the basis for orchestrating the 5GCity Use Cases.
- Finally, we provide hardware requirements and installation instructions for the prototype, while we also point to additional documentation about installation and usage that is available online wherever necessary.

Additionally, we give a feeling about the usage and the capabilities of the prototype by going through some graphical views and business logic descriptions of the user-facing components of the Orchestrator, namely the 5GCity Dashboard and the 5GCity SDK.

# 1. Introduction

The development work executed for the 5GCity platform interim release described in this document represents a first important milestone in the implementation of the 5GCity Scalable Orchestrator architecture described in Deliverable D4.1. Starting from D4.1, the architectural components of the 5GCity Scalable Orchestrator have been implemented in terms of software modules resulting in a mapping between functional entities and actual software components depicted in Figure 1 below. In order to help the reader go through the rest of the document, the specific software modules of the release, which are referenced in the following sections, are listed and highlighted in the notes of Figure 1.



**Figure 1. Software modules implementing the 5GCity Scalable Orchestrator Interim Release**

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

*As already foreseen in previous Deliverables, the interfaces and their usage have been adjusted to the details of 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. Since this is only an Interim Release of a prototype implementation, such details are not analysed further. Updates in the architecture and the Interfaces are discussed also in Deliverables D2.3 and D2.4, and will be reflected in the final release of the 5GCity Scalable Orchestrator release deliverable.*

---

## 2. Main features of the 5GCity Scalable Orchestrator Interim Release

This section lists and explains the main features offered by the released software modules of the interim release. The goal is to list these capabilities which have been implemented based on the specifications provided in Deliverable D4.1.

### **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.

#### 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 (3<sup>rd</sup> party external software), directly interfaced to OSM NFVO and catalogue service

### **SDK-Editor**

- Retrieve a list of available SDK functions.
- Retrieve a specific SDK function.
- Delete a specific SDK function.
- Create new SDK functions from VNFDs contained in 5G app & service catalogue.
  - Define a list of common monitoring parameters for each SDK function.
- Import SDK functions from VNFDs in 5G App & Service Catalogue at start-up:
  - 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).

### **SDK-Composer**

- 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 SDK services to 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.
- Retrieve the list of available SDK functions.
- Retrieve specific SDK functions.

### **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.

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)

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

## **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.

## **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 DynDNS 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 DynDNS 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**

We use OSM release 5 by referring the official documentation in <https://osm.etsi.org/> to describe its features.

- Note 1: The 5GCity extensions to ETSI OSM (which can be found in the modified OSM version of the 5GCity GitHub repository) enable 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 are under integration in order to be included in future official OSM releases.
- Note 2: 5GCity partners have contributed the OSM WIM plugin, which is already included also in the official OSM Release 5 and is 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.

---

## 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 Scalable Orchestrator Interim Release version 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/0.2/RAN-Controller.md>.

### Requirements

The installation of the Interim Release of the 5GCity Scalable 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.

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

<i>Host</i>	<i>5GCity modules hosted</i>	<i>Requirements (recommended)</i>
VM1	Dashboard SDK-GUI	Ubuntu 16.04 LTE, 2 vCPU, 4GB RAM, 20GB vHDD
VM2	SDK-Editor SDK-Composer	Ubuntu 16.04 LTE, 2 vCPU, 4GB RAM, 20GB vHDD, with the following software installed: <ul style="list-style-type: none"><li>• <i>systemd, git, postgresql-server, and java 8 JDK</i></li><li>• Maven 3.3.9</li></ul>
VM3	AAA	Ubuntu 16.04 LTE, 2 vCPU, 4GB RAM, 20GB vHDD
VM4	Monitoring	
VM5	Slice Manager MTO	
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\\_FIVE#Install\\_OSM\\_Release\\_FIVE](https://osm.etsi.org/wikipub/index.php/OSM_Release_FIVE#Install_OSM_Release_FIVE)
  - NOTE: 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/0.1/install\\_meao.sh](https://github.com/5GCity/5GCity-multi-tier-orchestration/blob/0.1/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/0.1/MEAO.md>.
3. Install the MTO and the Slice Manager in VM5 by following the installation instructions included in <https://github.com/5GCity/5GCity-slice-manager/blob/v1.1/doc/deployment/README.md> and <https://github.com/5GCity/5GCity-multi-tier-orchestration/blob/0.1/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/v2.0/README.md>
5. Install AAA in VM3 by following the installation instructions at <https://github.com/5GCity/5GCity-AAA/blob/0.1/README.md>
6. Install the SDK-Editor and the SDK-Composer in VM2 by following the instructions linked at <https://github.com/5GCity/5GCity-SDK-Composer/blob/0.7/README.md>
7. Install the Dashboard and the SDK-GUI in VM1 by following the instructions at <https://github.com/5GCity/5GCity-Dashboard-new/blob/0.1/README.md>

## **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 Scalable 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/v1.1/README.md>
  - MTO:
    - Links to swagger documentation are provided inside <https://github.com/5GCity/5GCity-multi-tier-orchestration/0.1/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/v2.0/doc/5GCity-MonitoringSystem-Overview\\_v2.docx](https://github.com/5GCity/5GCity-monitoring/blob/v2.0/doc/5GCity-MonitoringSystem-Overview_v2.docx)

---

## 4. Conclusion

This deliverable accompanies the software prototype of the 5GCity Scalable Orchestrator Platform (Interim Release) distributed as open source in GitHub. 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 previously released.

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

This Deliverable is an intermediate release which will be superseded by the Final Release (Deliverable D4.4).



---

## Abbreviations

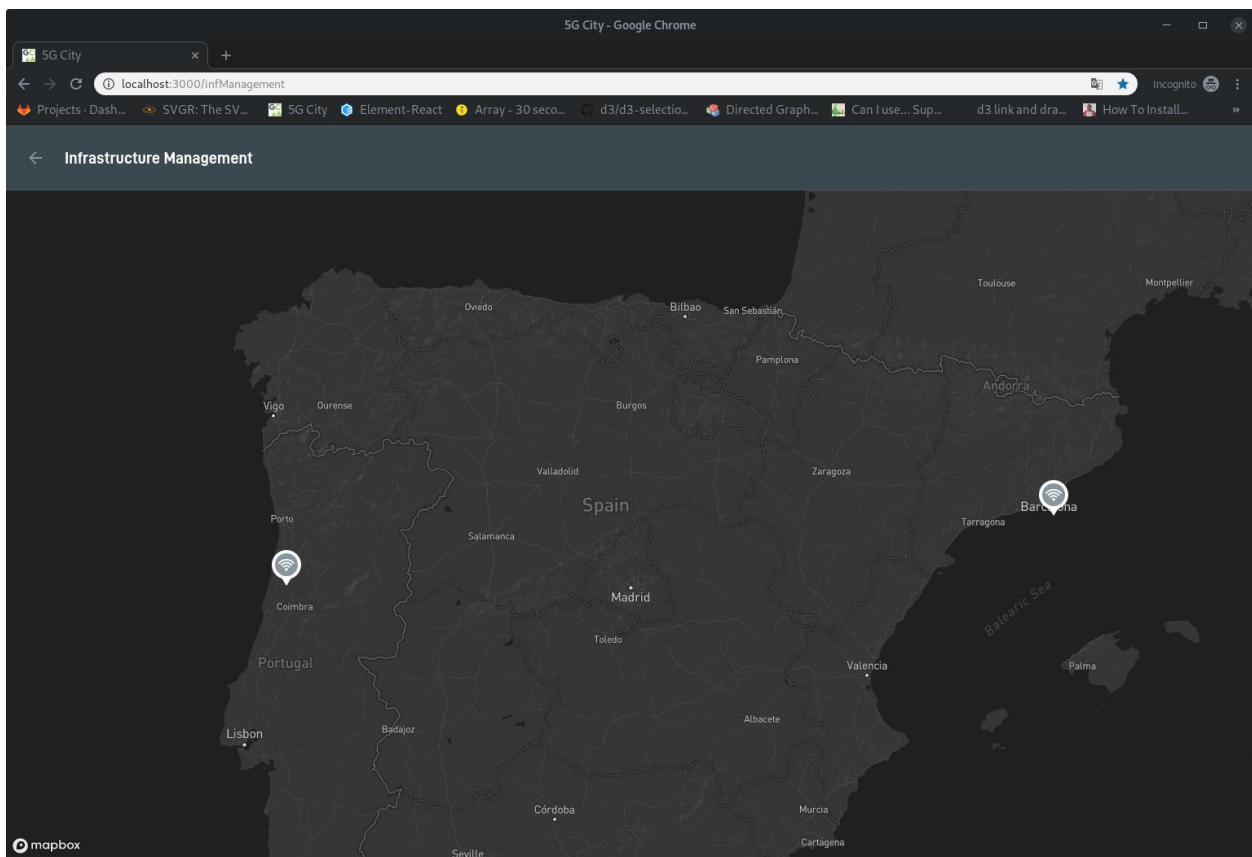
AAA	Authentication, Authorization and Accounting
AP	Access Point
API	Application Programming Interface
CRUD	Create, Read, Update, Delete
DNS	Domain Name Service
ETSI	European Telecommunications Standards Institute
GB	Gigabyte
GUI	Graphical User Interface
JDK	Java Development Kit
ME	Mobile Edge
MEAO	Mobile 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
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

---

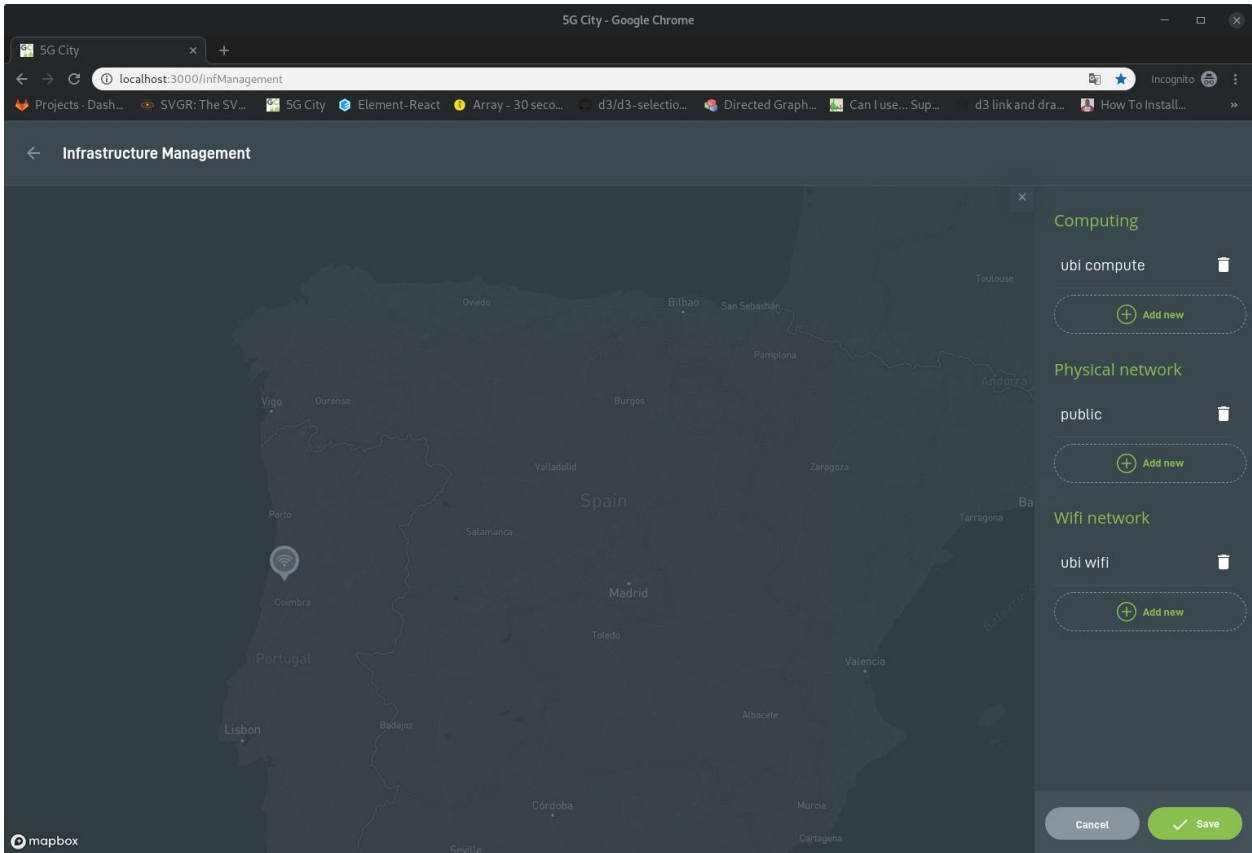
## Appendix A. Screenshots of the 5GCity Dashboard

### Infrastructure Management

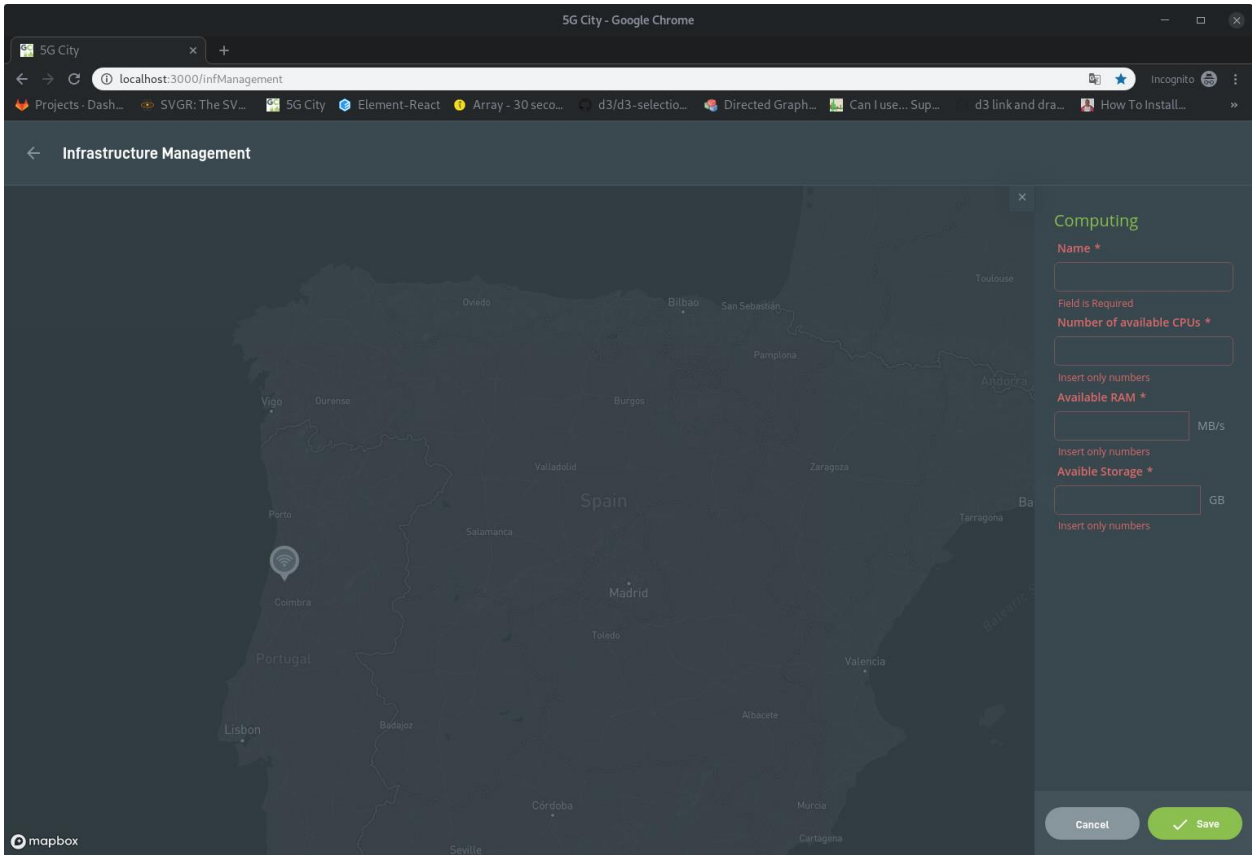
Users with Infrastructure Owner role, by using the web solution are able to manage the physical resources currently provisioned in the 5GCity platform. By selecting the "Infrastructure Management" feature these users are able to see the locations in which resources are currently provisioned as can be seen in the figure below:



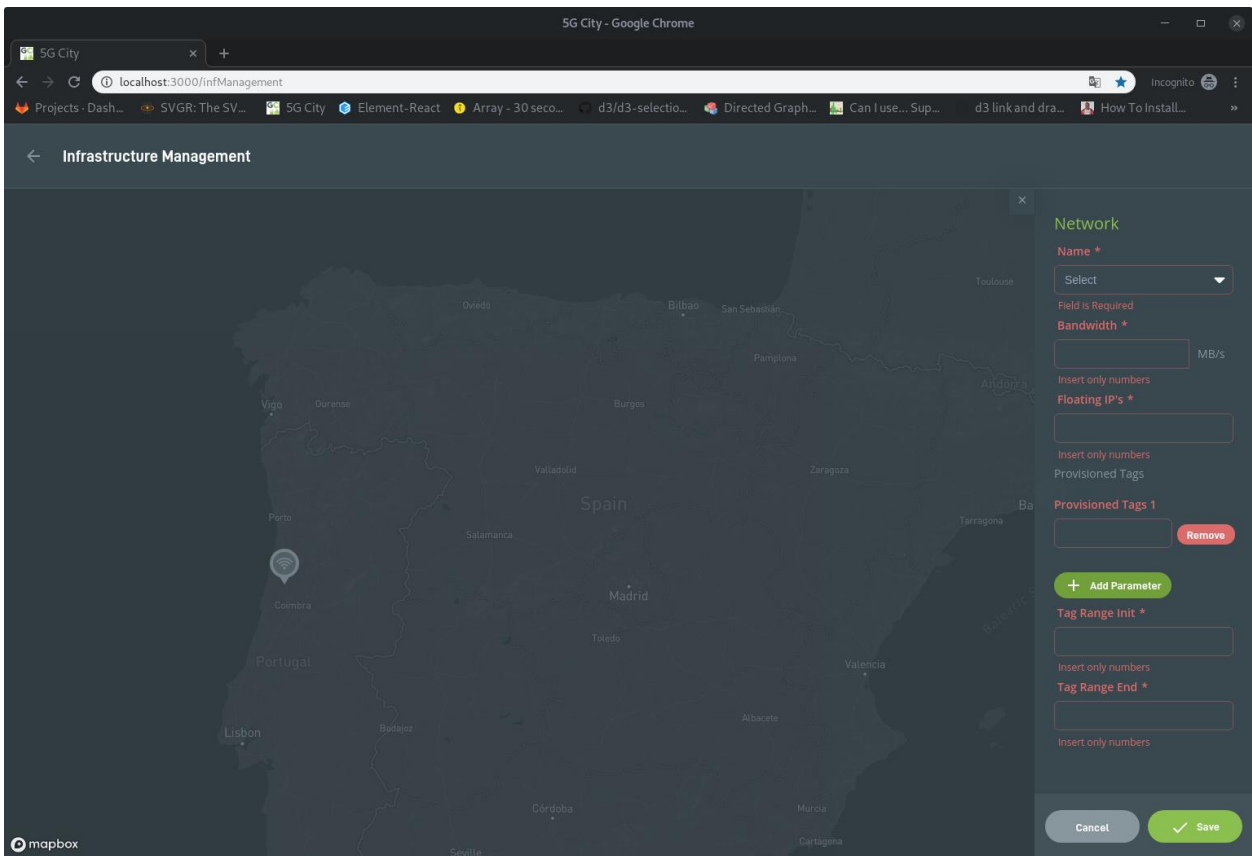
By clicking in one of the marked locations, users are not only able to see which resources are currently configured but also to add new ones or remove one of the previously configured. At this moment, users can manage physical computes, physical networks and WiFi APs.



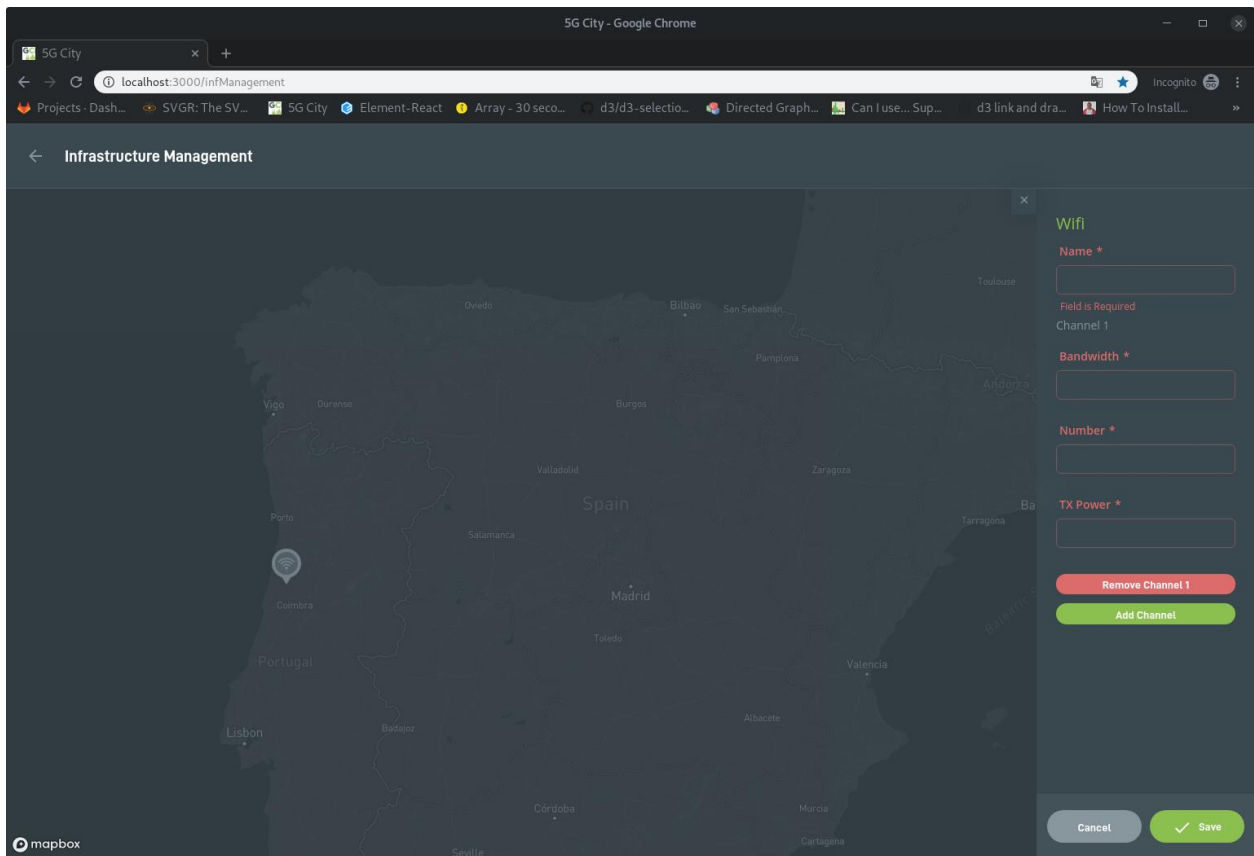
Focusing on compute edition, when the user selects to add a new compute he visualizes the following screen.



When adding physical networks, users are able to configure properties as exemplified below.



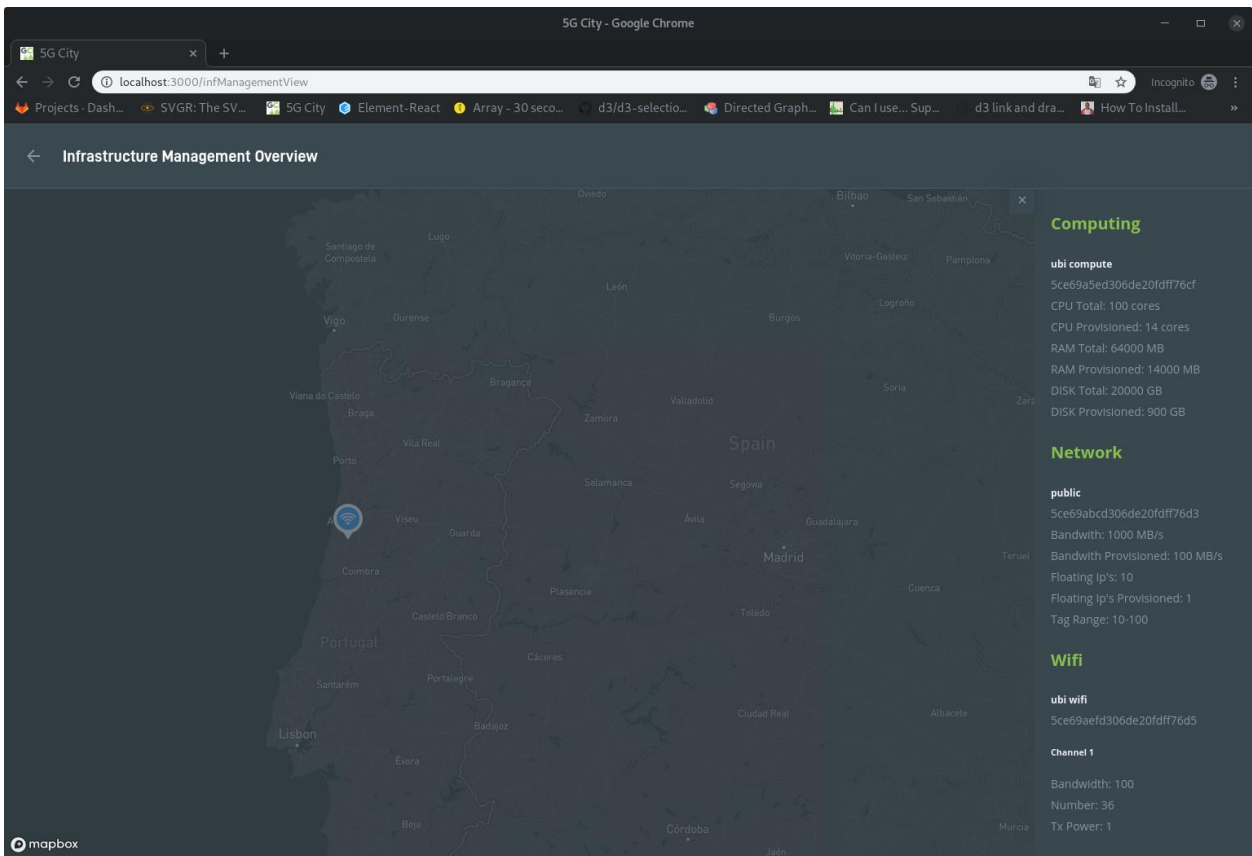
Lastly, to configure a WiFi AP, authorized users are able to define the variables displayed below.



As soon as physical resources are added to 5GCity platform, they become available to be used in the context of slice creation.

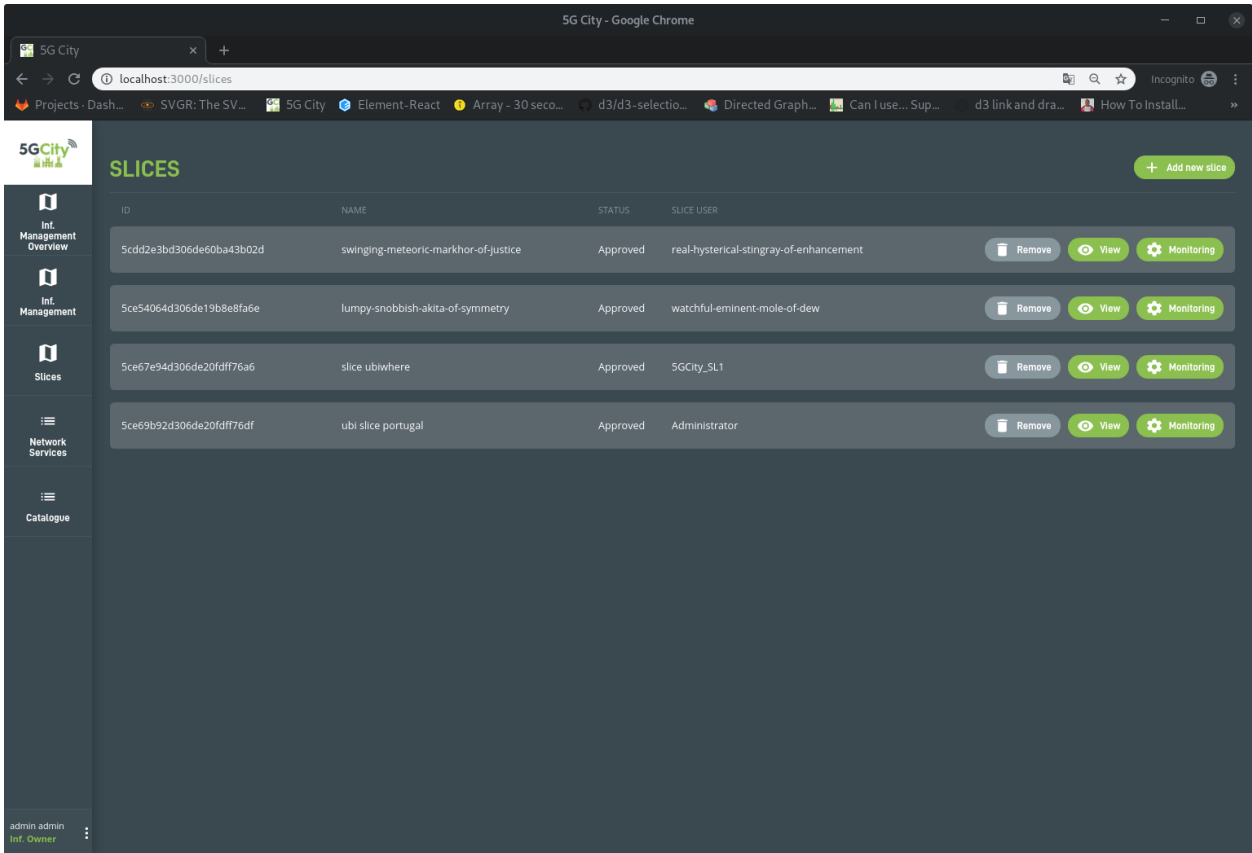
## Infrastructure Overview

Infrastructure overview allows infrastructure owner to visualize the currently configured physical infrastructure location as well as the usage of each physical resource.



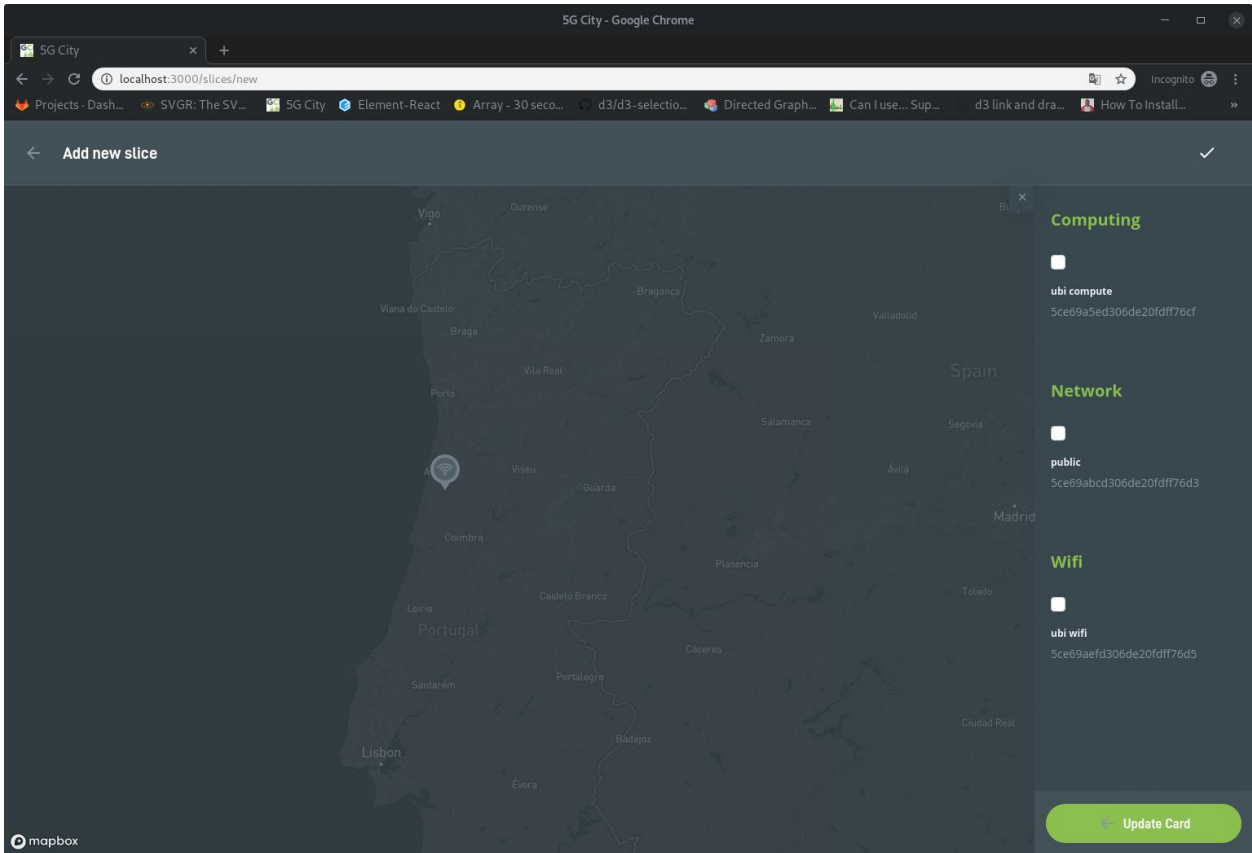
## Slice Management

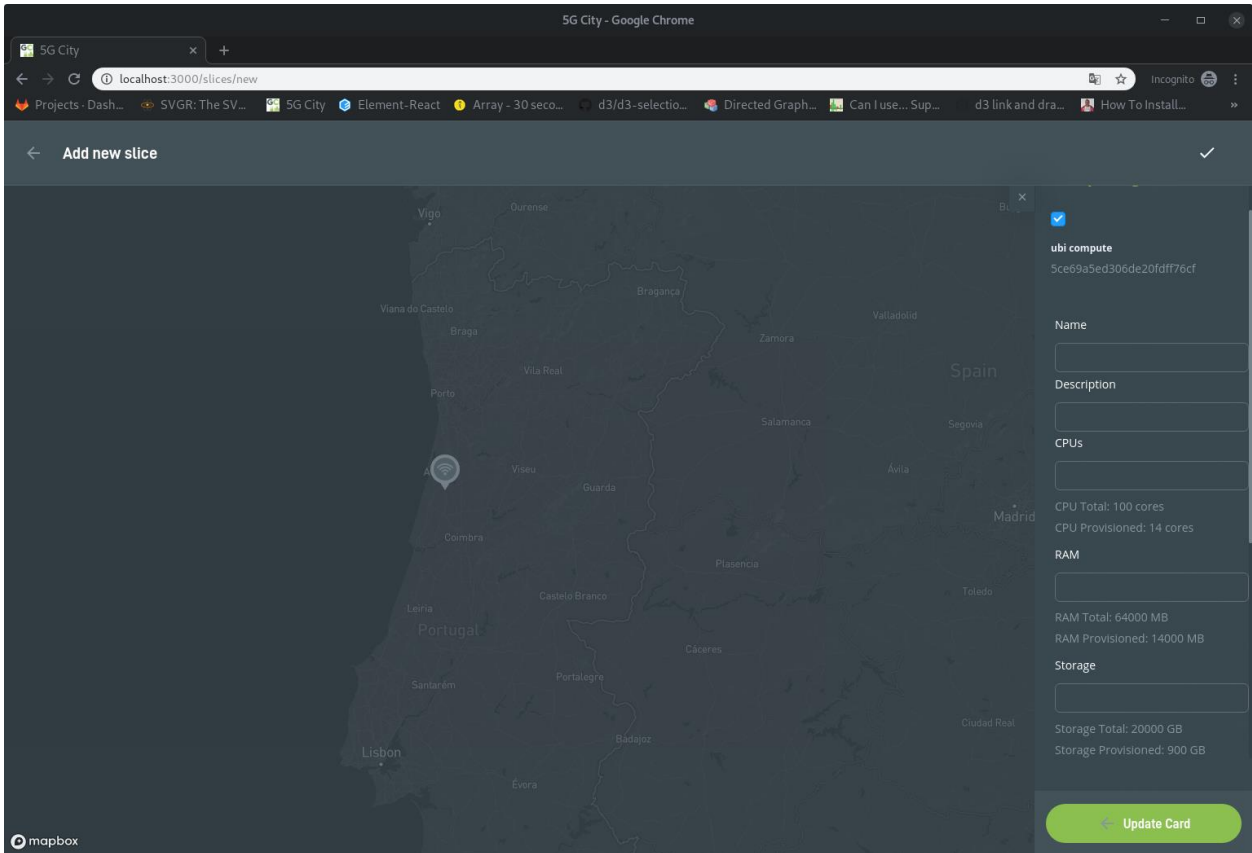
Users with roles infrastructure owner and slice requester can list, create, edit, and delete slices directly in the 5GCity platform. It should be noted that while infrastructure owners are able to list and manage all slices created in the context of 5GCity, users with slice requester role are only able 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.

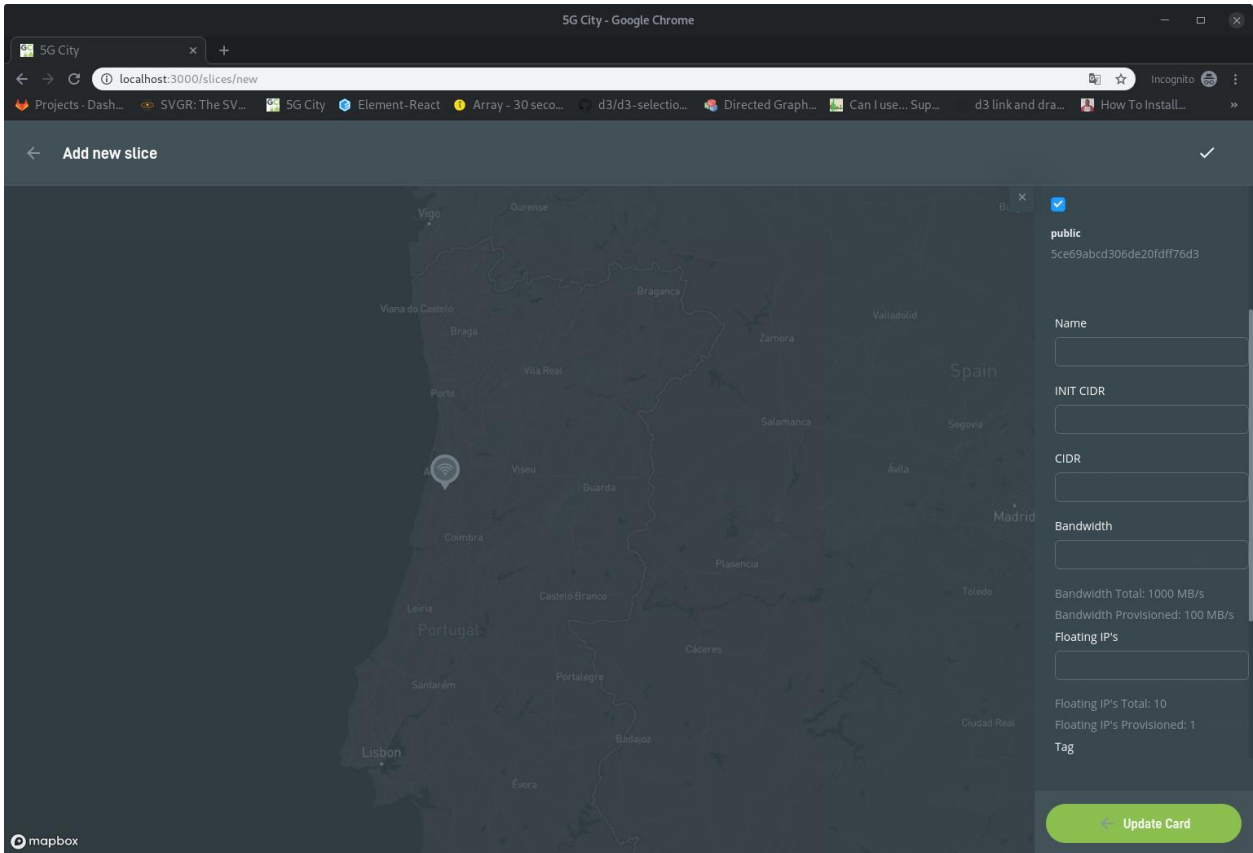


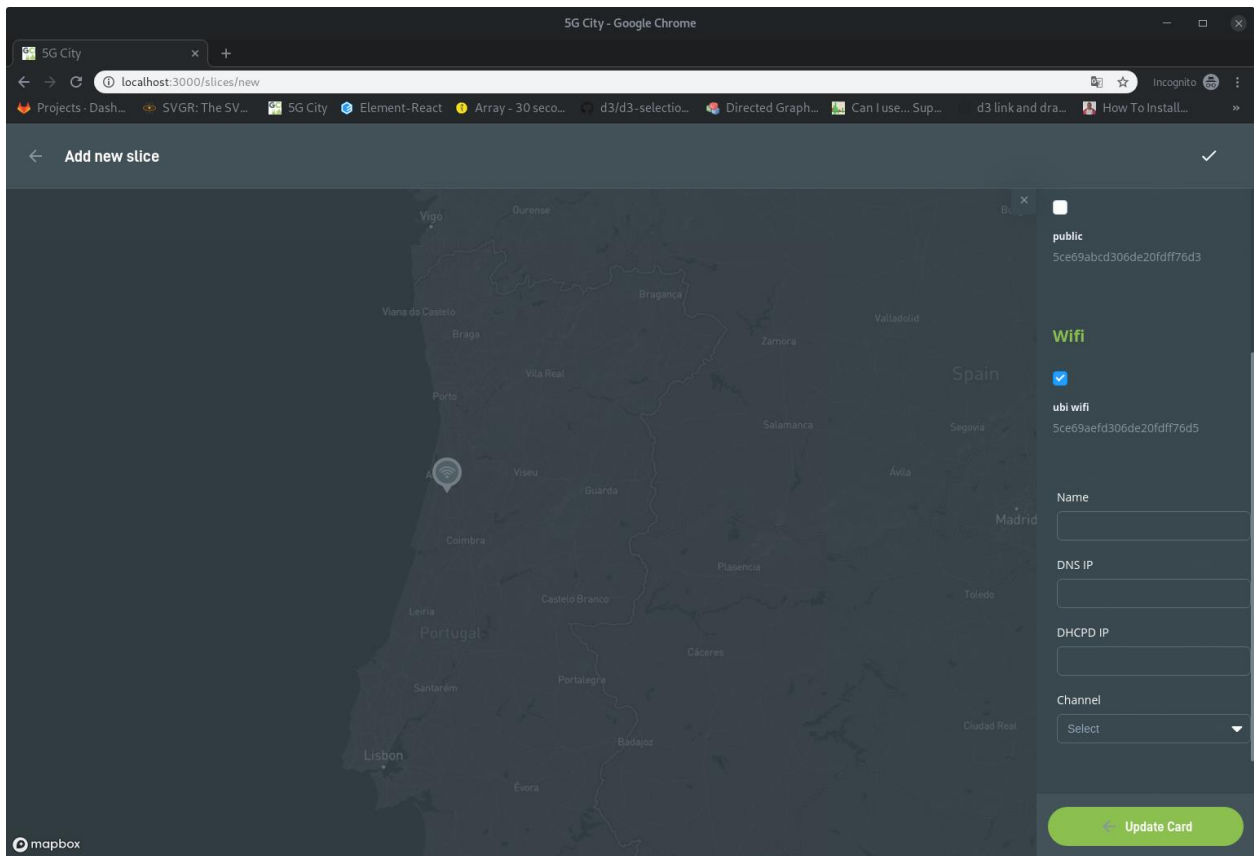
To create or edit a slice, users have to select the resources by location that they intend to add to the targeted slice. By clicking in a location, users are able to add, edit or remove resources. By each one of the resource types users can add to a slice, a set of parameters is mandatory to configure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



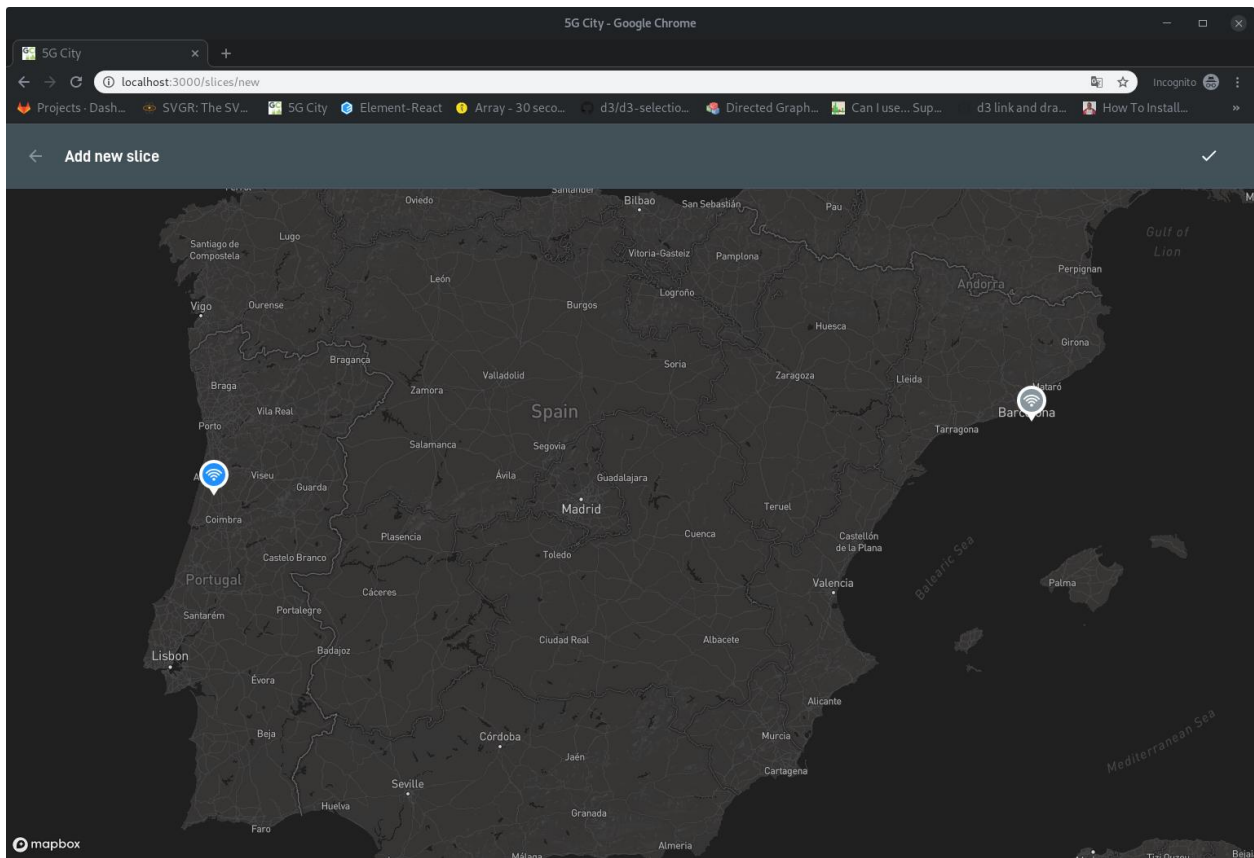








In order to allow users to quickly visualize in which location a slice has resources configured, 5GCity platform marks with a different colour the locations that have at least one resource configured in the context of the current slice.



## 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 to.

5G City - Google Chrome

localhost:3000/network

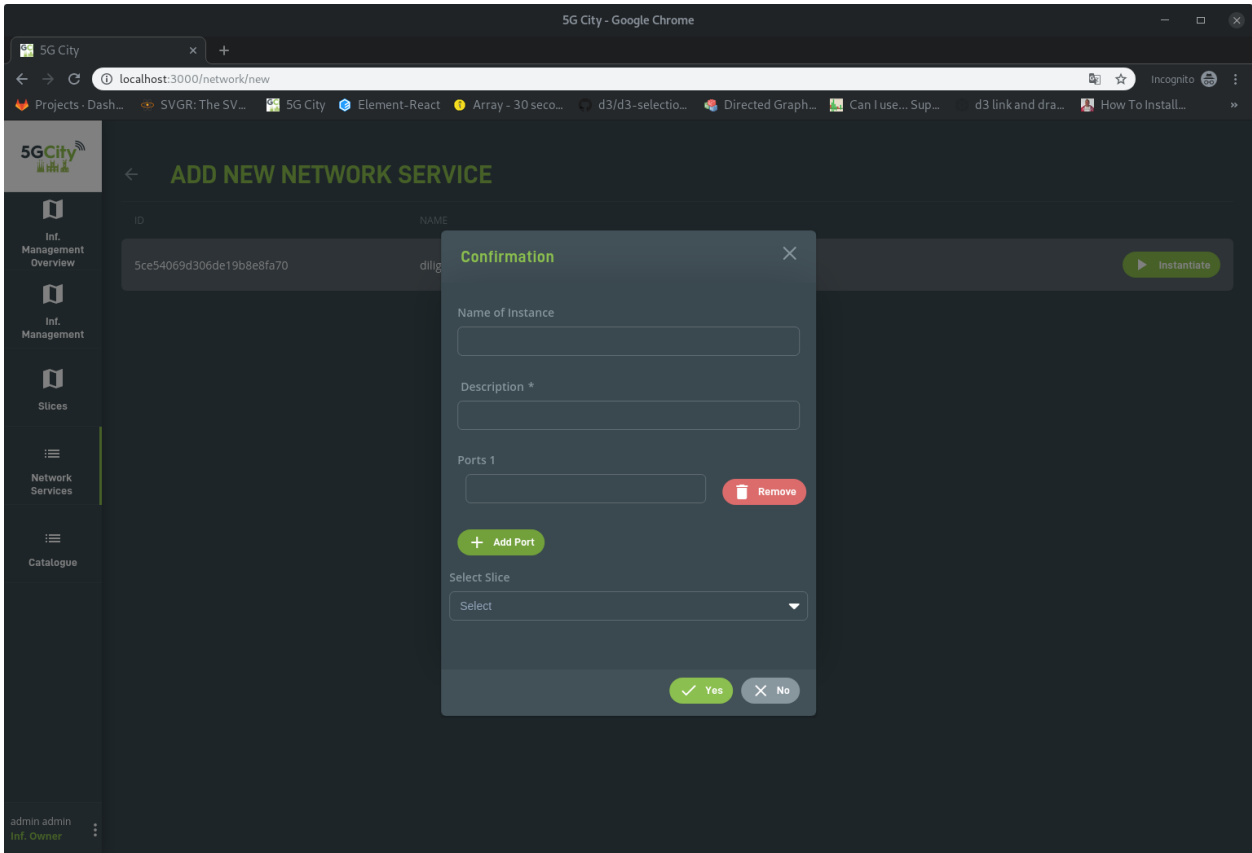
5G City

## RUNNING NETWORK SERVICES

+ Add new network service

ID	SHORT NAME	SLICE NAME	Remove	View	Monitoring
5ce5442bd306de20dff7611	ubi Instance 2	lumpy-snobbish-akita-of-symmetry			
5ce543bed306de20dff7610	ubi Instance	lumpy-snobbish-akita-of-symmetry			
5ce5406bd306de19b8e8fa71	hysterical-fuzzy-eel-of-art	lumpy-snobbish-akita-of-symmetry			
5ce69bd0d306de20dff76e0	ubi Instance	ubi slice portugal			

admin admin  
Inf. Owner



---

## 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

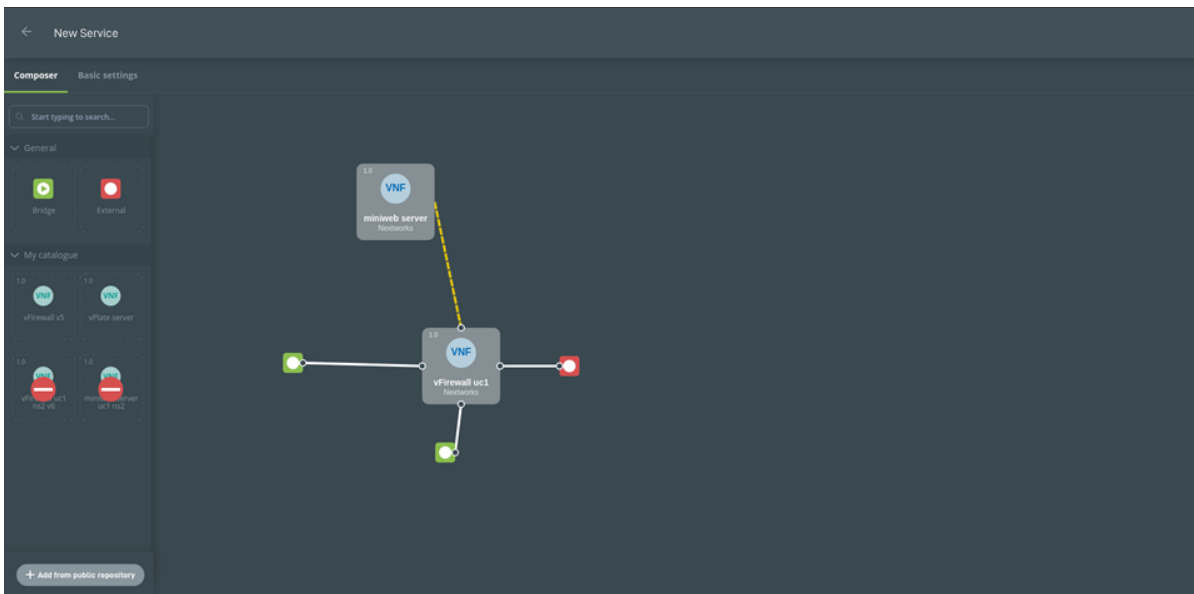
Furthermore, the SDK business logic allows creation of new SDK Services based on visibility level through which it is possible to filter access to specific functions and services based on the visibility level granted to a user/group. For example, "Bronze users" (visibility level 3) cannot use Silver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

Visibility level	Symbolic Name	Description
0	Platinum	These users can see all the functions and services tagged with visibility level $\geq 0$
1	Gold	These users can see all the functions and services tagged with visibility level $\geq 1$
2	Silver	These users can see all the functions and services tagged with visibility level $\geq 2$
3	Bronze	These users can see all the functions and services tagged with visibility level $\geq 3$

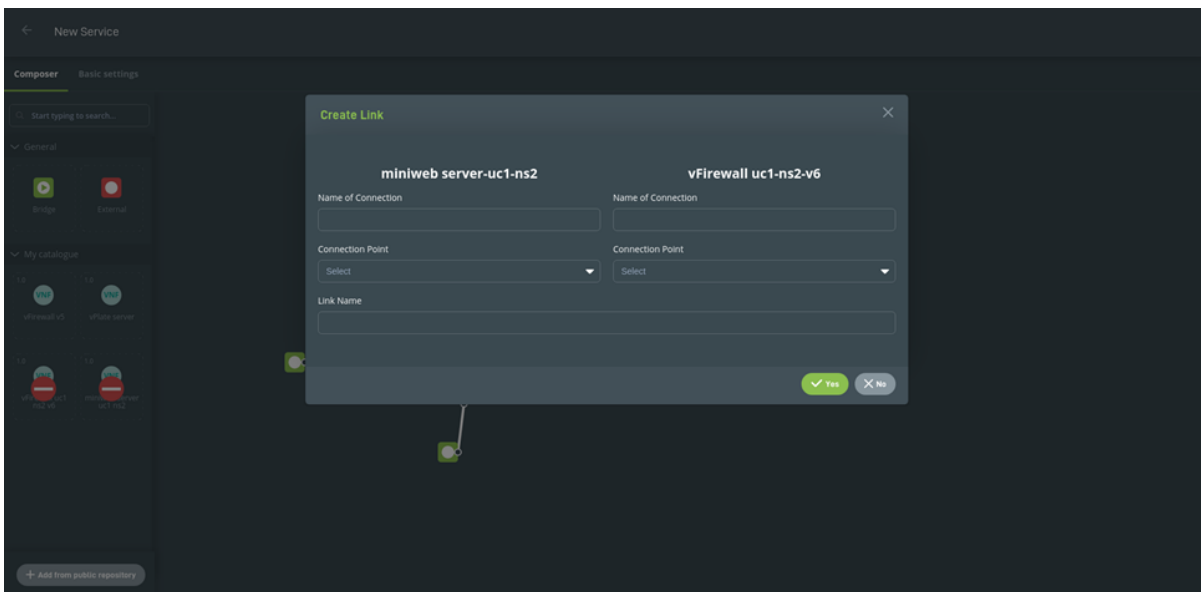
The 5GCity SDK has on top the GUI, as part of the 5GCity Dashboard. The SDK User will be able through GUI to run all the operations described in SDK Composer section.

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:



When finished, the new SDK Service is listed to the Users repository and ready to be published to the 5G App & Service Catalogue:

5GCity

### MY REPOSITORY - SERVICES

+ Add new service

ID	SHORT NAME	DESIGN	VERSION	REPOSITORY	LAST UPDATE	Actions
17	5GCITY-NS2-test	NXW	0.2	N.A	N.A	Clone Delete Edit
40	5GCITY-NS2-test-2	NXW	0.2	N.A	N.A	Clone Delete Edit
59	5GCITY-NS2-test-2	NXW	0.2	N.A	N.A	Clone Delete Edit
75	5GCITY-NS2-test-2	NXW	0.2	N.A	N.A	Clone Delete Edit

SL1 Admin  
Slice Requester

<END OF DOCUMENT>