5G-Connected Virtualized Enterprise Infrastructure for Smart City

ELENA-MĂDĂLINA OPROIU[1,2], CATALIN COSTEA[1], MIHAI IDU[1],
CATALIN BREZEANU[1] , MARIUS IORDACHE[1] AND ION MARGHESCU[2]
elena-madalina.oproiu@sdettib.pub.ro
1Technology Department, Orange Romania, Bucharest, ROMANIA
2University Politehnica of Bucharest, Telecommunications Department, Bucharest, ROMANIA

Abstract. In 5G context, Mobile Service Providers should provide future mobile networks that can support at least the following four demands: mobile broadband, ultra-low latency, dense broadband and massive connectivity. This paper is proposing a “all-in-one” solution that could deliver all the services, to every device, everywhere in the network in a viable option, from the Enterprise Network Segment perspective. The main role of this Segment will be to identify the key vertical sectors’ requirements, anticipating relevant trends early and mapping them into the 5G architecture. From the enterprise networks perspective, the challenge regarding 5G is to provide end-to-end network and cloud infrastructure slices over the physical infrastructure in order to fulfil specific requirements for the 5G use cases. In our case, the Enterprise Segment for Smart City will connect sensors, machines, city administrations and citizens to cloud-based IoT applications and we will prototype it. Our IoT platform will be implemented using an open source application, as the platform to be deployed and prototyped also may be exposed to different scenarios of developments, as a simplified ecosystem for building and managing the app into a more automatic and programmable perspective.
Keywords: 5G, mobile telephony, enterprises

1 Introduction

From the enterprise networks perspective, the challenge regarding 5G is to provide end-to-end network and cloud infrastructure slices over the physical infrastructure in order to fulfil specific requirements for the 5G use case.
The main role of the Enterprise networks will be to identify the key vertical sectors’ requirements, anticipating relevant trends early and mapping them into the 5G design. A successful Enterprise Infrastructure must be shared, secure and scalable and it must enable a smarter, safer and more sustainable development for each of the use cases, by realizing multiple, highly, flexible, end-to-end dedicated network and cloud infrastructure slices over the same physical infrastructure, in order to fulfil specific requirements.
The IoT (Internet of Things) platform, open source project, is enabled by a rapid development, providing an IoT cloud on the customer premises, for various IoT applications, that must be scalable, resilient, efficient, customizable and friendly to be integrated. In fact the IoT open platform is the software middleware used for connecting, acquisition and data processing from different sensors. The prototyping of the IoT platform is the foundation of the next years IoT communications, as it is expected that tens of billions of devices will be connected. For this argument rising the need of platform deployments into a customer oriented manner.
The concept of the Enterprise IoT platform, software model, is based on a series of standards and engineering implementations, from technical perspective to business model adaptation, in our particular case the IoT platform for the Smart City Apps.
The Enterprise IoT platform within the 5G networks context is intending to enable the designing of a cost-efficient prototyping platform for sensors connections, easy to be adaptable to the needs and capable to be extended to any IoT scenario. The expected business evolution of the platform is related to the public sector applications, as a City Hall, due to the fact that today many municipalities are using an aging infrastructure. For sake of clarity, the IoT platform will be implemented using an open source application, as the platform to be deployed and prototyped also may be exposed to different scenarios of developments, as a simplified ecosystem for building and managing the app into a more automatic and programmable perspective, by using specific APIs (Application Program Interface).
Even the output of the prototyping will not be presented as a mature product, commercial ready, we will be able to access and use a PaaS (Platform as a Service) capable to be consumed by different 5G verticals and services.
The paper is organized as follows: Section 2 presents generalities about 5G Enterprise Segment; Section 3 presents Enterprise network infrastructure, we have detailed here each block involved in the diagram presented in Fig.2. and how we intend to prototype this segment. Finally, Section 4 presents the conclusions.

2 Generalities about Enterprise

The Enterprise Segment for Smart City will connect sensors, machines, city administrations and citizens to cloud-based IoT applications. We can observe from Fig.1 the integration of the Enterprise Segment in 5G architecture and also the concept of slicing at this level.

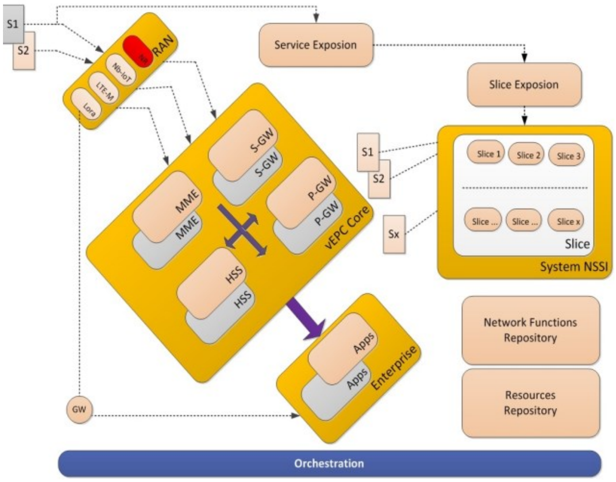

Fig. 1. End to end connectivity for Enterprise Network Segment

In order to fulfil the requirements listed above, the Enterprise infrastructure will be composed by several planes: Data Plane (composed by Enterprise Infrastructure and Enterprise Private Cloud), Enterprise Applications and Enterprise Services, Control Plane, Management Plane and Cross Plane Orchestration under the umbrella of One-Stop API. This plans are presented in the diagram from Fig.2. and detailed in section 3.

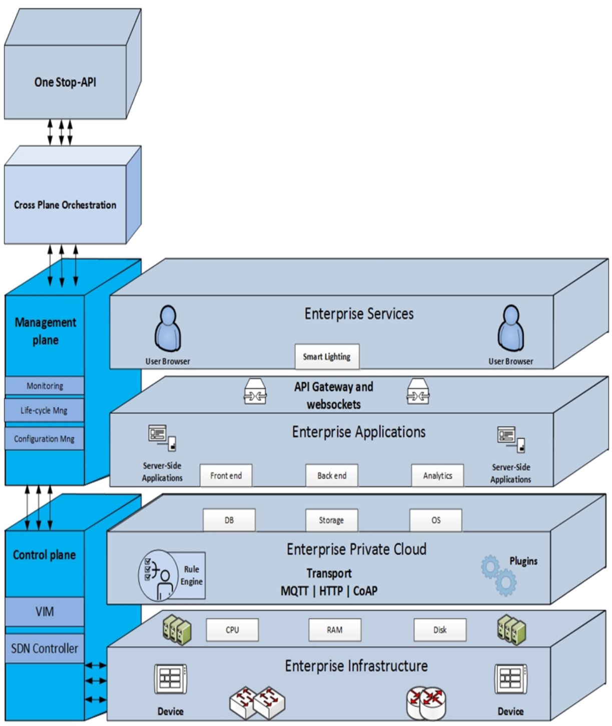

Fig. 2. Enterprise Segment overall diagram

With respect to use cases requirements, the Enterprise infrastructure should provide reliability and availability up to 99.95%. The Enterprise network must accommodate a high number of devices per unit area that are 5G capable, although they might not all be generating traffic simultaneously for the specified application. For each of the three use cases, the Enterprise network should provide a maximum positioning error tolerated by the application. The service deployment time is also a requirement and represents the duration required for setting up the network slices in order to provide services to end customer. The enterprise infrastructure must assure the protection of data, encompassing several level of security such as authentication, data confidentiality, data integrity and access control in a multi-tenant environment. Enterprise infrastructure should focus in minimizing the power consumption and provide scaling capabilities according to network load.
Bellow table summarizes the requirements for our use case, Smart City and their applicability to each enterprise infrastructure layer figured on Fig.2.

Table 1. Requirements applicability to infrastructure layers

Requirements / Layer	Reliability & Availability	Density	Position accuracy	Service deployment time	Security & Privacy	Low power consumption
Enterprise infrastructure	Yes	Yes	No	Yes	Yes	Yes
Enterprise private cloud	Yes	Yes	No	Yes	Yes	Yes
Enterprise applications	Yes	Yes	Yes	Yes	Yes	Yes
Enterprise services	Yes	No	Yes	No	Yes	No
Control plane	Yes	No	No	Yes	Yes	Yes
Management plane	Yes	No	No	Yes	Yes	Yes
Cross plane orchestration	Yes	No	No	Yes	Yes	Yes
One Stop API	Yes	No	No	Yes	Yes	No

From the connectivity perspective, the enterprise infrastructure can be summarized in three layers:

- physical networks – DC (datacenter) gateways, ToR (Top of the Rack) switches, blades NIC (Network Interface Card);
- underlay networks - based on VLAN (Virtual local area network) and created during the infrastructure provisioning;
- overlay networks - based in tunnelling mechanisms like VXLAN (Virtual Extensible LAN) and used for traffic isolation between tenants.

2 Enterprise Network Infrastructure

The first plane of the Enterprise architecture is the Data plane (infrastructure) which contains an end-to-end heterogeneous network- **Enterprise Infrastructure** or Physical Architecture and a distributed cloud platform- **Enterprise Private Cloud**.

The Enterprise Infrastructure for Smart City IoT segment is composed by:

- Two routers which act as gateways to provide demarcation between WAN networks and Enterprise infrastructure;
- Two aggregation switches in order to allow seamless expansion of the infrastructure in the future;
- Two ToR switches used for servers' connectivity;
- Three servers to host the infrastructure depicted above.

To fulfil our requirements detailed above, each server utilized in our prototype has the following hardware capabilities:

- 2 processors, 12 core/processor @ 2.4 GHz;
- 128 GB RAM;
- 2 TB Hard Disk;
- 2 Network adapters 1Gbps.

Having the target to build a resilient infrastructure, Fig.3. depicts the physical components and their connectivity.

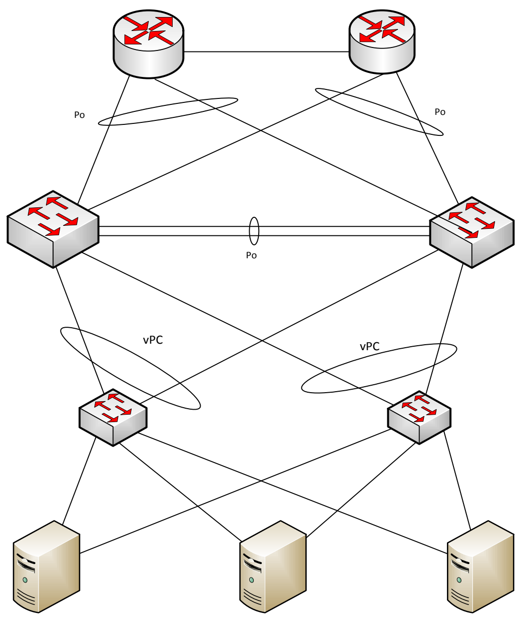

Fig. 3. Enterprise physical architecture

The underlay networks are used for transport of the remote management, control and date planes traffic and relay on VLANs for segmentation. Following the Fig.4., the administrative subnet is used for remote management access to all nodes of the infrastructure. The internal subnet is used for communication between nodes, being accessed by the REST APIs (Representational State Transfer Application Programming Interface) of the Openstack framework and being reachable only within the infrastructure nodes. Traffic subnet is used to communicate with external networks and will allow sensors data to reach the IoT platform.

The scope of the overlay subnet is to provide communication between internal IoT platform components and is dynamically created based on Neutron inputs.

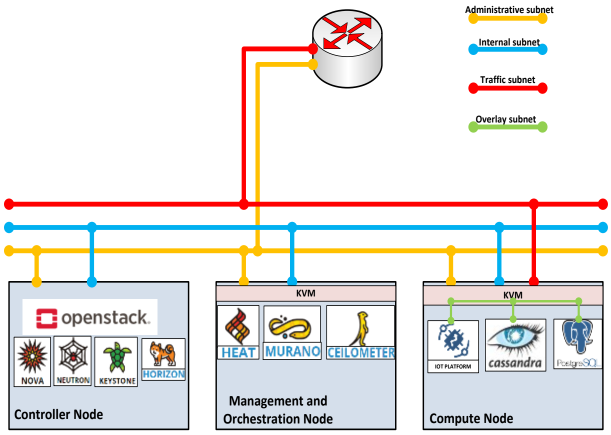

Fig. 4. Connectivity in Enterprise infrastructure

Enterprise Private Cloud Infrastructure is a type of cloud computing infrastructure that has the same characteristics and advantages as a public cloud including even more benefits but at a specific cost. The main difference is that this type of cloud infrastructure is a private one, it is developed, installed and administrated for dedicated needs of one and specific private organization. This offers connectivity and all additional services only for one company/institution being able to share it in special cases.

Private Cloud Infrastructure (layer 2 from the Fig. 2.) is a type of environment with a single tenant in which case all hardware resources (compute, storage) and network are dedicated (and in many cases owned by) to a single client or company and all data is protected behind a firewall. In general, this type of approach is dedicated and chosen by middle and large size company/enterprise businesses.

An important benefit of private cloud is that single organization that has access to resources, can allocate however it is necessarily and this operation could lead to an efficiency of cost for energy and for operation entire cloud. In addition, flexibility is a big plus. Every private organization can choose own technologies, architecture and software such that can secure and isolate applications and use cases.

To start development and implementation of entire Enterprise Cloud Infrastructure dedicated to our Smart City use case and all network components, it is necessarily to describe the needs and all blocks that make up entire service. All elements and components from private cloud network could be observed from the bottom layer of Fig.4. and from Fig.5.

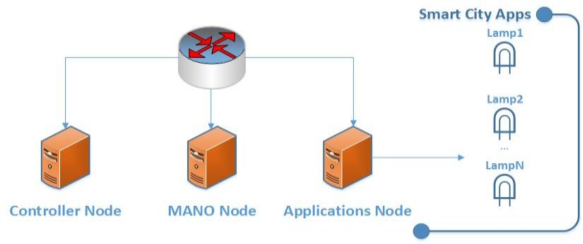

Fig. 5. High level view of Smart City applications

This environment will be developed in Orange Romania laboratory on physical infrastructure (servers and connectivity) described above. Using this entire hardware infrastructure, we will instantiate a Private Cloud Infrastructure based on OpenStack open source software collections.

OpenStack is a cloud operating system that controls large pools of compute, storage and networking resources throughout a datacentre, all managed through a dashboard that gives administrators control while empowering their users to provision resources through a web interface [1].

On the third physical server, we will instantiate 3 VMs (Virtual Machine) that will take the role of applications and will be the core of the data processing and storage for all information that can be obtained from Smart City infrastructure of sensors and actuators. There are many OpenStack components, some of them that will be used in the project are listed below.

Nova (compute) includes the controller and compute nodes. These get VM images from OpenStack's image service and after that it will create a VM on the specific server.

Neutron (networking) creates virtual networks and network interfaces, and attaches to many proprietary vendor networking products.

Keystone (identity storage) grants users and processes access to different OpenStack tools based on an authentication token that Keystone generates. [2]

Besides controller node, MANO (Management and Orchestration) and Applications servers will use over bare metal, KVM as hypervisor. **KVM** (Kernel-based Virtual Machine) is an open source software that can enable full virtualization solution. Using it, we can deploy and manage multiple VMs with different OSS (Operations Support System images (Linux-based, Windows, etc).

KVM converts Linux into a bare-metal hypervisor. All hypervisors need some operating system-level components—such as a memory manager, process scheduler, input/output (I/O) stack, drivers, security manager, a network interface to run VMs. KVM has all these components due the fact that it’s part of the Linux kernel. Every VM is implemented as a regular Linux process, scheduled by the standard Linux scheduler, with dedicated virtual hardware like a network card, graphics adapter, CPU(s) (Central Processing Unit), memory, and disks. [3]

Enterprise applications (layer 3 from the the Fig. 2.) contains all applications from our use case that are using to export data, different APIs an SDK (Software Development Kit) and other applications to process and gather all information in one place. This plane consists in databases, subscriber portal, back-end applications and scripts to fulfil the upper layer services requirements.

On the third physical server from Orange Romania laboratory, we will instantiate 3 VMs that will take the role of applications and will be the core of the data processing and storage for all information that can be obtained from Smart City infrastructure of sensors and actuators. Besides this 3 virtual servers, we can, on demand, instantiate new VMs for another third party applications or data provider.

On the first VM, will be deployed all the components that cover IoT platform functionalities and on the other two will be deployed one relational database and one no-SQL (Structured Query Language) database where can be stored all kind of data gathered from city solutions. Both databases represent key points in entire architecture and help the IoT platform to be more efficient and to process and stored data in correct manner for every type of client or service.

Types of applications from enterprise segment:

1. Thingsboard.IO – open source IoT platform
2. PostreSQL – open source relational SQL database
3. Cassandra NoSQL – open source non-relational database

The entire concept of Smart City can be understood as all devices deployed in city areas that provides data and communication technologies used to transport all data to cloud or to any platform that can analyse it, process it and integrate in visualizations or any form that can offer key information in running the cities towards an economical, secure, efficient and green environment for all citizens. In this picture, the Smart City Applications use all components of enterprise infrastructure to serve all kind of use cases. For Orange Romania, the Smart City use case is Smart Lighting, that forms application layer of city and gather all information to offer intelligence such that a city can have an economical and efficiency lighting system. An endpoint as a web portal is offered to end user to can monitor and control entire system.

IoT platform needs to be compliant, for Smart City use cases, with some technical capabilities listed below. Besides that, will have to connect IoT devices via telecom networks (with/without SIM cards – depends on use case) with reduced power and act like an IaaS that provides hosting space and processing power for all services.

IoT Platform should integrate a set of tools to facilitate the interconnection between devices and business applications, with the following components:

- Connectivity Interfaces (public and private) to collect data, send command or notification from/to IoT/M2M (Machine to Machine) devices.
- Device Management (supervision, configuration, resources, firmware, etc.).
- A specific and configurable policy for message and commands routing between devices and services.
- Data management system and data storage with different storage periods and variable access, processing speeds.
- Web portal for users management and visualization tools.
- IoT Platform architecture is presented in Fig.6. and it is composed of following complementary levels:
- Connectivity: manages the communications with the client devices and applications;
- Bus: a set of message-oriented programs allowing exchanges between all client software modules;
- A set of rules or a bus that can allow messages/commands exchange in asynchronous manner;
- Service: various modules supporting the high level functions (device management, data processing and storage, etc.).

In Fig.6. we proposed an architecture of an IoT platform, taking into account all the aspects listed above.

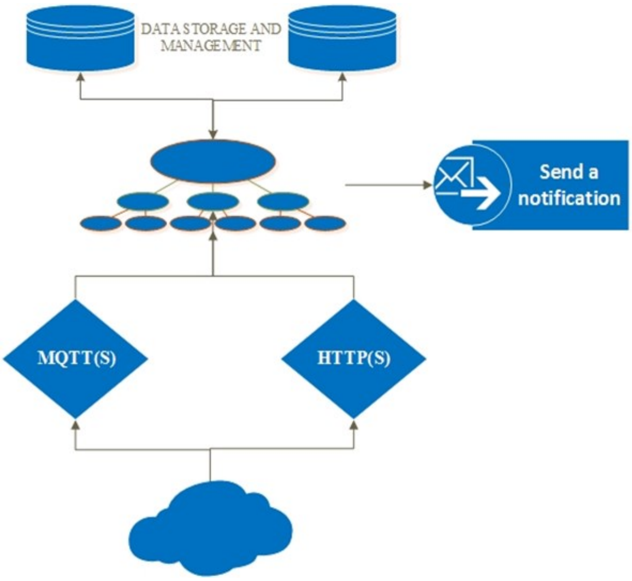

Fig. 6. Proposed architecture of IoT platform

We intend to utilize the applications depicted above to run the services for our Smart City (Enterprise Services - layer 4 on Fig.2.).

The **Enterprise Services** plane defines and implements the services of the use case. For beginning, we intend to implement the Intelligent Lighting use case. After implementing this services we want to extend our implementations to other services that could be optimum offered by a 5G Smart City, like: environmental monitoring, (pollution, temperature, humidity, noise), connected parking, real-time traffic information and control, connected buildings, smart home, connected household appliances and public safety alerts for improved emergency response times.

Control Plane for Enterprise is composed by two components as we can observe from Fig.2.: **VIM-Virtualized Infrastructure Manager** and **SDN-Software Defined Networks Controller** and it implements the abstractions provided by Software Networks technologies (essentially as SDN and NFV-Network Function Virtualisation) to support an abstracted model for the 5G networks functions.

VIM is managing the network function virtualization infrastructure and serves as a conduit for control path interaction between virtual network functions and NFVI (NFV Infrastructure). The main task for VIM is to inventory, provision, de-provision and manage virtual compute, storage and networking while also communicating with the underlying physical resources. The VIM is responsible for operational aspects such as logs, metric, alerts, root cause analysis, policy enforcement, service assurance etc. In Fig.7. is shown the interaction responsibilities of the VIM with the management and orchestration functional block, and SDN controller.

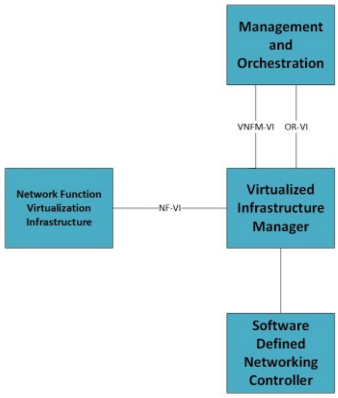

Fig.7. Interaction between VIM and SDN Controller

VIM comes in the form of complete software stacks. On the market right now there are only two major software VIM stacks prevalent in Network Function Virtualization: OpenStack and VMware.

To achieve our goal in building the Enterprise platform, we choose a more open source approach, and we get to handle on OpenStack, which is growing in popularity among the NFV deployment at carriers worldwide.

The most important aspect of OpenStack pertaining to its usage as a private cloud platform is the tenant model. Every virtual or physical object governed by the OpenStack system exists within a private space referred to as a tenant or project [4].

Our purpose is to use one of the computing nodes as a Management and Orchestration where we intend to deploy alongside Cross Plane Orchestrator, VNFM (Virtual Network Function Manager) and VIM also a Tacker server. The main purpose of the Tacker server deployment in our infrastructure is to perform the scaling procedures and optimize all the physical pool resources of the server, with future perspective in case of managing multiple OpenStack sites without having the need to deploy Tacker server on each of these sites.

After installation of OpenStack it needs to initialized the flavours that the platform will support. Most Tacker sample TOSCA (Topology and Orchestration Specification for Cloud Applications) templates will ask Tacker to create flavours on demand [5]. If not, the specified flavour in templates must exist in OpenStack. Tacker repository's sample TOSCA templates are referring to cirros image named "cirros-0.3.5-x86_64-disk", so this image should be uploaded into OpenStack before Tacker uses it.

TOSCA represents a specification that aims to standardize how we describe software applications and everything that is required for them to run inside a cloud based application. TOSCA provides a way to describe not only an application, but also its dependencies and supporting cloud infrastructure. TOSCA contains two building blocks references: nodes and relationships.

As a conclusion, VIM is the software resource block responsible for ensuring that physical and virtual resources work smoothly in any condition. Compared with the more traditional Operating System, VIM comes with extra features of collecting resources logs from many other machines at the same time.

SDN controller represents the control point of the SDN network and its scope is to control the flow rules of the overlay forwarding path entities (vSwitches-Virtual Switches or vRouters-visrtual Routers) and might also have the capability to manage the physical network equipment.

Under the perimeter of Enterprise cloud, the SDN controller is responsible for performing network functions and to provide on-demand connectivity for applications and services while keeping an isolation between different tenants.

In the Enterprise cloud implementation, Neutron Openstack service is responsible for enabling networking and controlling the vSwitches embedded in the compute node as depicted in Fig.8.

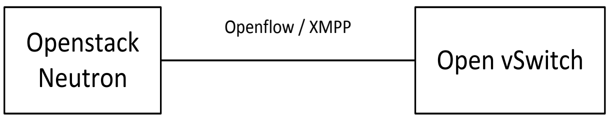

Fig.8. Neutron control to Open vSwitch

In order to provide network dynamicity for discovery and allocation of virtual networks and compute resources, the SDN controller is bound to the VIM. This interaction is based on ML2 (Modular Layer 2) plugin and RPC (Remote Procedure Calls) service for bidirectional agent communication.

As depicted in Fig.9., Neutron agents which run in Enterprise infrastructure are split as follows:

On Controller Node will run neutron-server, neutron-dhcp-agent, neutron-l3-agent and neutron-plugin-agent;

On Compute Node will run neutron-plugin-agent (neutron-openswitch-agent).

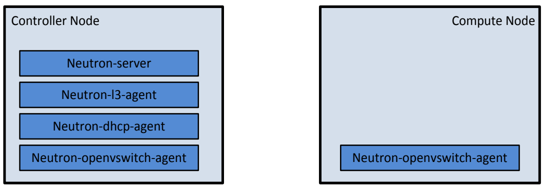

Fig.8. Neutron agents

Table 2 specifies the scope of each Neutron agent in order to allow the provisioning of virtual network interfaces and resources.

Table 2 Neutron agents usage

Agent	Usage
Neutron-server	Provides REST API exposure
Neutron-l3-agent	Provides L3 routing
Neutron-dhcp-agent	Provides DHCP services
Neutron-openvswitch-agent	Provision network resources

Neutron will help on VM booting process by wiring the VM port and providing IP address to it using DHCP (Dynamic Host Configuration Protocol) agent. Neutron will also help when communication between different VMs or to external physical world is needed.

Management Plane for Enterprise is composed by three level as we can observe from Fig.2.: Monitoring, Life-Cycle Management and Configuration Management. The overall purpose of the Management Plane is to enable the monitoring, life-cycle management and configuration mechanisms required to assemble the supported virtual resources running network functions.

Regarding Management plane of a Smart City Enterprise architecture, 3GPP defines a Network Slice Management Function (NSMF) as responsible for the life-cycle management of network slice instances, linked and interconnected to Network Slice Subnet Management Function (NSSMF) for lifecycle management of subnets instances [6].

All management and orchestration blocks from a network slice (in this case Smart Lighting slice) should include and operate, such that can facilitate the proper functioning of the service, the following functions: provisioning, optimization and performance, monitoring and maintain entire Network Slice Instance in parameters in which it was designed.

Monitoring Plane is the functional block who is responsible to perform the architecture sensing functionalities, collection, filter and enrichment of counters, events and alarms retrieved from the network resources that compute the slice delivered to each vertical. It is also responsible of collecting, filtering and enriching network information that will be use to understand the performance and usage of the network slice and services. It gathers information, such as counters, events and/or alarms, form the running physical/ virtual network resources, slices and services.

The Enterprise service **Life-Cycle-Management** is related to the automation working processes of services instantiation, activation and deactivation over a deployed infrastructure enabled for virtualized and orchestrated functions.

The entire network system is based on virtualized network elements that permit through programming the dynamicity of service's Enterprise, based on templates and descriptors selection, resources reservation, scaling, configuration and instantiation.

Configuration Management is the part of the management plane that includes mechanism for: design, deployment, provisioning, configuration (input for control plane), monitoring, de-provisioning of resources, including the slice and the service.

Cross-plane orchestrator must be able to automate the different functionalities (FCAPS- Fault, Configuration, Accounting, Performance, and Security) associated to the automation of the slice management. The orchestrator will coordinate the operations of the data plane, control plane, management plane and service plane for optimised network slicing, and deploy NFV actions for operational requirements on demand related to QoE (Quality of Experience) sensors/actuators to monitor/optimise the QoE of a use case service. The other task is to prototype the demanding vertical business use cases.

The idea of Cross-Plane Orchestration for Enterprise is firstly to provide a set of enabling automated mechanisms which can perform the cross-plane configuration of all the architectural components involved in the layers from diagram presented in Fig.2. (control plane, management plane, services plane and applications plane) in order to do an efficient slice management. Another purpose of the Cross-Plane Orchestration is to provide a set of coordination functions across several logical layers and constructs (i.e. service, slice, resource, and infrastructure) with the aim orchestrating the provisioning of end-to-end slices.

There are two tips of orchestration: horizontal orchestration and vertical orchestration [7]. It is called horizontal orchestration when it refers to multiple domains, either administrative or technologic, and the orchestration is performed within the same layer or for the purpose of the same logical construct. Additionally, it is called vertical orchestration when it refers to multiple layers, e.g. a single network provider performs vertical orchestration when deploying a service instance where the service is composed by one or more slices and each slice is composed by multiple resources [7]. So tacking into account the aspects presented above we can identify in our Virtualized Enterprise Segment the both models of orchestration, vertical and horizontal. The vertical orchestration is referring to our multiple layers (service, slice, resource, and infrastructure) and the horizontal is composed by: one slice orchestrator, one service orchestrator for our Intelligent Lighting service and one resource orchestrator who is represented in our case by VIM. The end to end functionality is the follow: the service orchestrator asks slice orchestrator (VNF manager in our case) who asks further resource orchestrator.

One Stop API – **OSA** is an key point designed for Enterprise architecture who serve as an entry point of the slicing framework, enabler for the customer or vertical such that can design, provision, and instantiate an entire (or part of a) service instance or slice instance.

Taking into account all the aspects depicted above, the enterprise infrastructure is using the Openstack framework with all relevant services in order to provide IaaS (Infrastructure as a Service) type of solution for hosting the IoT platform.

The provisioning flow of Smart City service would be based on following steps:

1. A request from One Stop API is sent to Heat orchestration engine containing the NSD (Network Service Descriptor) identifier;
2. Heat is analysing the request and the NSD;
3. Heat sends a request to VNFM Murano for IoT platform instantiation;
4. Murano analyses the IoT platform requirements based on associated VNFD;
5. Murano requests the granting of life cycle management to Heat;
6. Heat authorizes the request and transmits to Murano the associated VIM details;
7. Murano is sending a request to VIM in order to allocate network;
8. VIM creates the network using the Neutron service and confirms the allocation to Murano;
9. Murano is sending a request to VIM in order to allocate a compute host;
10. VIM creates a VM according to the compute requirements of IoT platform and confirm the instantiation to Muranos.

These steps could be observed from Fig.9.

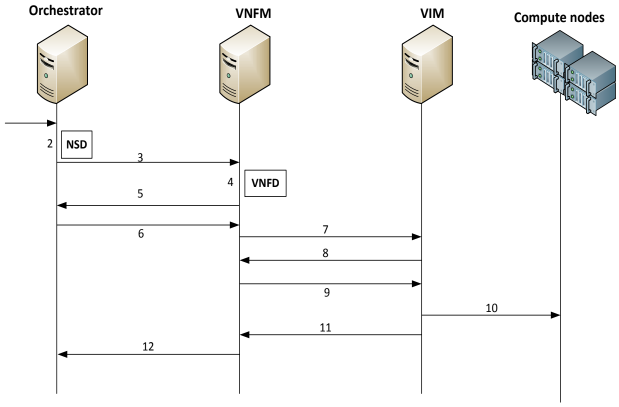

Fig. 9. IoT platform instantiation

4 Conclusions

As we have presented in this paper, the main role for 5G-Connected Virtualized Enterprise Infrastructure is to identify the key vertical sectors' requirements, anticipating relevant trends early and mapping them into the 5G architecture.

In our case, from the enterprise networks perspective, the challenge regarding 5G is to provide end-to-end network and cloud infrastructure slices over the physical infrastructure in order to fulfil the specific requirements detailed in Table 1, for the Smart City use case. This paper is proposing a "all-in-one" solution like it exists today that delivers all the services to every device everywhere in the network in a viable option.

As we said above, the Virtualized Enterprise Segment for our 5G use case, Smart City, will connect sensors, machines, city administrations and citizens to cloud-based IoT applications. In few months we will prototype it. Our IoT platform will be implemented using an open source application, as the platform to be deployed and prototyped also may be exposed to different scenarios of developments, as a simplified ecosystem for building and managing the app into a more automatic and programmable perspective.

In conclusion, we think that the Virtualized Enterprise IoT platform in the 5G networks could enable the design of a cost-efficient prototyping platform for sensors connections, easy to be adaptable and extended, in order to fulfil the requirements for any IoT scenario.

This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 761913; project SLICENET.

References

1. *https://www.openstack.org/software/*
2. *http://cdn.ttgtmedia.com/searchCloudComputing/downloads/OpenStack+Guide.pdf*

3. https://www.redhat.com/en/topics/virtualization/what-is-KVM
4. *Michael Solberg , Ben Silverman , Openstack for Architects, Packt Publishing (February 6, 2017), ISBN-13: 978-1784395100*
5. https://docs.openstack.org/tacker/latest/install/openstack_vim_installation.html
6. *Management Plane System Definition, APIs and Interfaces*- deliverable D2.4 on Slicenet Project, (2018)
7. *Overall Architecture and Interfaces Definition*, deliverable D2.2 on Slicenet Project, (2017)