

# Network Slicing-aware NFV Orchestration for 5G Service Platforms

Hamzeh Khalili\*, Apostolos Papageorgiou\*, Shuaib Siddiqui\*, Carlos Colman-Meixner<sup>x</sup>, Gino Carrozzo<sup>+</sup>,  
Reza Nejabati<sup>x</sup> and Dimitra Simeonidou<sup>x</sup>

\*i2CAT Foundation, Barcelona, Spain; <sup>x</sup>University of Bristol, Bristol, UK; <sup>+</sup>Nextworks, Pisa, Italy  
Corresponding author email: hamzeh.khalili@i2cat.net

**Abstract**—The advent of 5G promises to unleash highly pervasive network coverage and density, increased data rate and capacity, optimized instantiation of virtualized resources in a multi-tenant and multi-service network capable of fulfilling the stringent requirements of various heterogeneous applications. Network slicing is a key enabler of 5G to allow multiple customized and isolated virtual networks upon a single shared physical network infrastructure. In this work, we present a survey of how various network virtualization solutions address slicing, we review the management and orchestration tools available to implement it, and we describe the slicing-aware orchestration platform that we designed for our project (5GCity). Our solution embraces creation of slices of various network elements such as compute nodes, physical networks, radio parts and network edge resources by coordinating different underlying controllers. The platform is being evaluated in three live city pilots (Barcelona, Lucca, and Bristol), already achieving slice creation in a few seconds and control plane latency of a few milliseconds.

**Index Terms**—Network Slicing, Network Sharing, 5G, Management and Network Orchestration (MANO), Neutral Hosting

## I. INTRODUCTION

5G networks represent a great opportunity for network operators and the telecommunications ecosystem to satisfy the diverse requirements of various vertical industries. Operators aim at serving a plethora of heterogeneous services through their new 5G networks, each with very specific requirements broadly classified in the three categories of enhanced Mobile Broadband (eMBB), Ultra Reliable Low Latency Communications (URLLC), and massive Machine Type Communications (mMTC). Apart from the technological advancements in new radio systems and the new generation of core network as defined in 3GPP, 5G embraces also a deep softwarization of the various network functions, which greatly enhances the support of multi-tenancy in order to offer flexibility and improved service performance compared to its predecessors of the 3th and 4th Generation mobile networks [1]. Leveraging on softwarization technologies, i.e. Network Functions Virtualization (NFV) and Software Defined Networking (SDN), higher programmability, automation and significant cost/energy reduction are achievable. With the help of NFV and SDN, the IT resources embedded in the network infrastructure are employed to offer value-added services.

Such 5G services are expected to be often composed of a set of interconnected Virtual Network Functions (VNFs) and

Mobile Edge Computing Applications (MEC App), which will be dynamically instantiated over a highly distributed cloud and edge infrastructure. The transition from specialized hardware to VNFs executed on general-purpose processors is expected to affect not only the mobile core network but also aspects of the Radio Access Network (RAN) [2]. Hence, network-sharing models should adopt the trend towards virtualization, while network infrastructure sharing is another fundamental element for addressing business aspects of dense 5G mobile networks, as it would be unrealistic to expect the deployment of hundreds of vertically isolated access nodes in dense environments.

Network slicing in 5G networks is a promising solution for network sharing and QoS-related challenges faced by infrastructure providers and vertical sectors in 5G [3] [4]. In fact, slicing allows 5G networks to be separated into multiple isolated and customized virtual networks, each optimized to serve a specific vertical application. Network slicing relies on multi-tenancy atop a common physical infrastructure that allows different parties to share computing, storage, and network resources in order to instantiate and run their softwarized network services. Since the early steps of 5G specification, network slicing has gained a lot of attention among researchers, industries, and standardization bodies, such as Next Generation Mobile Network Alliance (NGMN), 3rd Generation Partnership Project (3GPP), European Telecommunications Standards Institute (ETSI), and Internet Engineering Task Force (IETF). Their efforts highlighted the role that network slicing is expected to play in meeting the wide range of requirements of various use cases.

In this paper, we review the basic network slicing concepts, and discuss how slicing is addressed in state-of-the-art orchestration technologies. Further, we introduce the slicing platform developed for the 5GCity project, which allows to orchestrate the creation of slices that include chunks of various layers of the 5G network (Cloud and edge IT resources, core and access network of diverse standards). To this end, the rest of the paper is organized as follows: Section II discusses some slicing concepts and standards, Section III provides a gap analysis of the slicing-related aspects of existing NFV orchestrators, and Section IV presents our proposed slicing-aware orchestration framework and its current deployment and evaluation status in three major European cities.

## II. NETWORK SLICING STANDARDS

Infrastructure sharing through slicing relies on virtualization and isolation of compute, storage, and network resources. Virtual Machines (VMs) allow slicing of compute and storage resources (e.g. CPUs and RAM), while Virtual Networks (VNs) allow the slicing on network links and nodes (e.g. wavelength, frequency, bandwidth, transponders) [5].

While cloud computing focuses on sharing compute and storage of multiple servers hosted in multiple datacentres, 5G networks also involve intensive slicing at the edge in order to provide multi-tenancy and radio-related capabilities. This combination makes slicing of 5G networks challenging from a control and data plane perspective [6]. As a result, NGMN, 3GPP, ETSI, IETF, and various research projects funded in the context of 5G, are focusing on defining and solving the end-to-end (E2E) slicing in 5G Networks and its fundamental challenges.

NGMN introduced the 5G slicing concept as a flexible software network approach of multiple VNs over a shared infrastructure, in which the resulting E2E network slice is a combination of VNs across the Core Network (CN) and the Radio Access Networks (RANs). In addition to this, NGMN defined a three-layer slicing architecture including: i) a Service Layer that represents the needs to be supported by the network operators or a third party, ii) a Network Slice Instance Layer that is created based on a network slice blueprint to describe the requirements of network instances, e.g., eMBB (note that the network slice instance can be represented by multiple sub-network instances, e.g., IP Multimedia Subsystem-IMS, while a sub-network instance consists of several Network Function instances with their corresponding resources), and iii) a Network Resources Layer that represents the network resources assigned to a slice [7].

Based on the layered model proposed by NGMN, 3GPP released the standard 3GPP TR 28.801 [8], which also defines a Network Slice Instance (NSI) as a set of Network Slice Subnet Instances (NSSI), composed by at least one VNF and/or Physical Network Functions (PNFs). For 3GPP, NFV Network Services (NFV-NS) represent the view of network slices in terms of provided network functionality.

Recently, the ETSI NFV group introduced a standard named ETSI GS NFV-IFA 014 [9], which is in line with the 3GPP view and defines the concept of virtualized resources per slice subnets or NSSI as nested NFV-NS.

The current standards show a significant convergence between approaches defined by NGMN, 3GPP, and ETSI in that an E2E slice is formed by a set of nested services which can be formed by one or more VNFs/PNFs, together with an assignment of specific network and compute resources that can be used by the slice elements. However, the implementation of this concept within network service orchestrators is still challenging, as discussed in the next section.

## III. NETWORK SLICING AND NFV ORCHESTRATION

To make network slicing models a reality, a deep integration with NFV orchestration systems is needed in order to

implement all the mechanisms for the lifecycle management of slices and the virtualized functions and resources that they include.

Most of the solutions discussed in this paper are in line with the ETSI NFV standards and are trying to integrate slicing lifecycle management workflows across Cloud, edge, and 5G radio access elements. Among others, we have analyzed platforms like Tacker [10], OpenBaton [11], TeNoR [12], ONAP [13], SONATA [14], 5Gtango [15], and OSM [16]. Summaries of the solutions, together with comments about their slicing-related aspects, are provided in following subsections.

1) **Tacker**: [10] is an official OpenStack project that builds a generic VNF Manager (VNFM) and a NFV Orchestrator (NFVO) to deploy and operate NSs and VNFs on an infrastructure managed by OpenStack. **With regard to slicing**, it can be assumed that Tacker would inherently use all OpenStack features that are meant to support slicing. For example, [17] interestingly mentions the OpenStack mechanism of assigning various QoS policies to different OpenStack projects (which would belong to the same slice) as a way to consolidate homogeneous and connected slices. **However**, the Tacker itself provides no explicit additional mechanisms for slicing.

2) **OpenBaton**: [11] is an open source NFVO which supports -among others- dynamic registration of NFV PoPs (Points of Presence) and parallel deployment of multiple services (each containing multiple VNFs), which might belong to different tenants. **With regard to slicing**, OpenBaton provides an external module called "Network Slicing Engine", which can enforce certain QoS configurations across resources that shall belong to the same slice. Similarly to what we mentioned for Tacker, it uses OpenStack QoS policies across projects to achieve its "slice configurations". **However**, it seems to be currently limited to this QoS enforcement functionality and it does not deal with comprehensive slicing data models or slice lifecycle operations.

3) **T-Nova (TeNoR)**: The T-NOVA project orchestrator, called TeNoR [12], is another ETSI NFV compliant NFVO. **With regard to slicing**, it offers some supporting functionalities by enabling multi-tenancy via a user management system that enables to assign services and resources to specific tenants, with the granularity required in a slicing framework. **However**, TeNoR does not explicitly handle slice management, and it is partly the same community that identified the requirement to implement a more sophisticated slice manager in the context of our current project.

4) **ONAP**: Open Network Automation Platform (ONAP) [13] provides a platform for real-time policy-driven orchestration and automation of physical and virtual network functions, which allow providers and developers to automate new services and support service lifecycle management. **With regard to slicing**, working documents of ONAP provide some instructions (i.e. required steps) in order to define, create, and manage network slices. **However**, slicing is not (yet) a main concern for the platform, while the definition of slice is quite different to slice definitions that stem from the ETSI NFV

principles, making it difficult to talk about “slicing-aware NFV orchestration” when it comes to ONAP network slices.

5) **SONATA and 5Gtango**: SONATA [14] (and its evolution in 5Gtango [15]) provide a development toolchain for virtualized services, including NFVO functionalities. SONATA provides a modular and flexible MANO framework where a service- or function-specific manager can be added, thus modifying the provided default managers to a specific service or function needs. **With regard to slicing**, these works follow the guidelines provided by a new ETSI document for slicing support [9]. These guidelines simply map the slice concept to the NS concept, practically defining a slice “simply” as a set of -potentially nested NSs, and “assuming” the NS lifecycle management to be sufficient for performing slice management functionalities. **However**, this solution tightly couples the slice lifecycle operations to NSs, so that it is hard to imagine how slices would be managed if other kinds of services/applications would run on the slices, e.g., edge/MEC applications, or Cloud-native applications, instead (or in addition to) NSs.

6) **OSM**: ETSI Open Source MANO (OSM) [16] is an operator-led ETSI community that is delivering a production-quality open source Management and Orchestration (MANO) stack aligned with ETSI NFV information models. **With regard to slicing**, OSM seems to follow exactly the same approach like SONATA/5Gtango (see above). **However**, this could lead, as already mentioned, to a very NS-concept-dependent definition and lifecycle of slices.

#### IV. 5GCITY SLICING-AWARE ORCHESTRATION SOLUTION

The main goal of the 5GCity project is to define how to turn a city into a distributed and multi-tenant edge infrastructure, which uses orchestration and service programming to integrate 5G services administered by a neutral host. These 5G services will be used by municipalities to host Smart City services, by Virtual Network Operators (VNO) to extend their networks, and by additional third party providers such as media or automotive verticals to offer innovative services to their customers. Therefore, the 5GCity orchestration and control platform includes a scalable orchestrator that is based on the ETSI NFV MANO specification, extending it to support network slicing, intelligent Service-Level Agreement (SLA) management, efficient resource allocation, and a broader palette of underlying infrastructure technologies [18].

Our platform aims at operating with a large number of devices in dynamic conditions (e.g. network, device overload, multi-tenancy, etc.) in an efficient manner. Among others, the 5GCity orchestration platform is responsible for the lifecycle management and orchestration of all 5G-based edge services and for the control and management of the available city edge infrastructure. It also includes 5G edge service programming models, as well as a northbound API to enable access to the different edge services and the orchestrator functionalities.

##### A. 5GCity Orchestrator Architecture

The 5GCity orchestrator platform provides dynamic provisioning of network slices and unified management of the

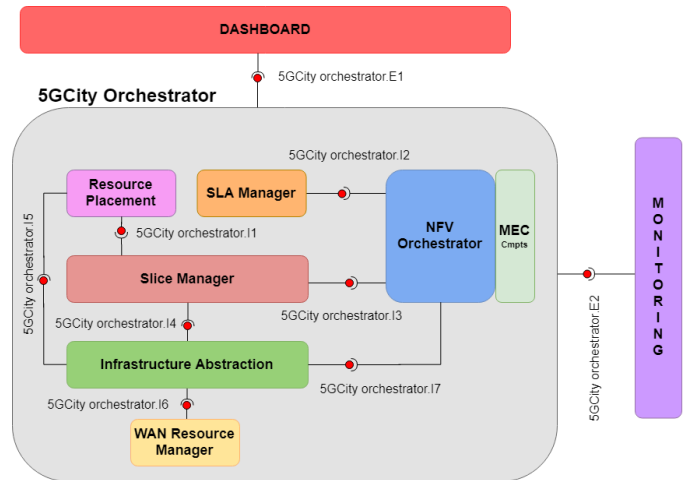


Fig. 1. 5GCity orchestration platform architecture

connectivity of end-to-end services running upon these slices. The 5GCity orchestrator architecture is illustrated in Fig.1. It is defined in accordance with the ETSI NFV model and includes extra functionalities which allow to better address the need to implement slices and orchestration workflows across the heterogeneous virtualized domains spanning from the core datacenter to the edge PoP down to the radio far-edge (see [19] for more details). More specifically:

- **NFV Orchestrator**: NFVO is responsible for NS onboarding and VNF management, as described in the respective ETSI NFV specifications [20]. For this we adopt the ETSI OSM solution, and integrate it with MEC orchestrators and other controllers used in our platform.
- **MEC components**: NFV Orchestration is extended with MEC components that to orchestrate and configure special types of VNFs, namely VNFs that are ME (Multi-access Edge) apps [21], following the ETSI MEC specification [3]. MEC support is added to the NFVO to include the MEC descriptions and interact with special edge services (e.g. location services).
- **Resource Placement**: This component is responsible for computing an optimal allocation of VNFs over different physical resource domains.
- **SLA Manager**: Service Level Agreement (SLA) represent the relationship between slice/service users and slice/service providers to ensure and maintain an acceptable Quality of Service (QoS) level.
- **Infrastructure Abstraction**: It enables communication between the 5GCity orchestration platform and the multiples underlying infrastructure elements such as VIMs and access/metro networks parts (i.e. SDN-controlled metro network, radio fronthaul and backhaul).
- **WAN Resource Manager**: It is responsible for managing resources of the lower layer, especially configuring the traffic flows among them, in cases that are not covered by the functionality of the NFVO.
- **Slice Manager**: It enables the virtualized and non-

virtualized network elements and functions to be easily logically segmented, configured and reused (isolated from one another) in order to meet various demands. Each slice has its own specific compute and network resource chunks that suit the network services offered by it.

### B. 5GCity Slice Manager Way of Operation

In 5GCity, network slicing covers the end-to-end service provisioning, operation, management and termination workflows on a per-slice basis. The slices are dynamically allocated in the network and in the multi-access edge, extending through the radio infrastructure. The Slice Manager enables a unified system for dynamic deployment and provisioning of new services, and allows the efficient and secure control and orchestration of services for different verticals. Each slice fulfills specific requirements related to quality policies, security functions, radio resource management, cloud and network resource management capabilities, etc. The key requirements fulfilled by our solution can be summarised in the following:

- Support the creation of customizable and user-specific slices with flexible and dynamically deployable resources, based on explicit user requests, existing slice templates, or implicit info about the planned services.
- Allow for dynamic provisioning and instantiation of end-to-end network services.
- Support multi-tenancy over different underlying technologies and resources used to compose the slices.
- Execute various run-time control operations during the lifetime of slices.

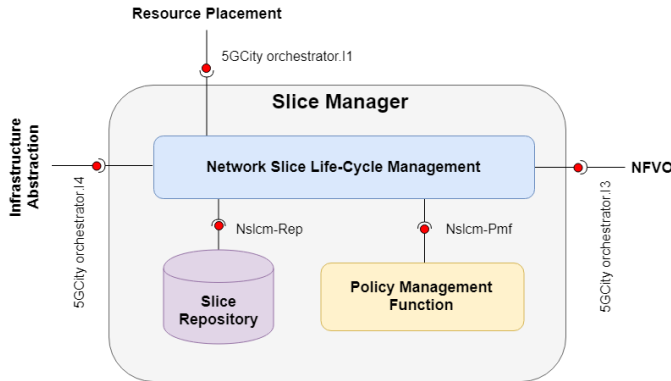


Fig. 2. Internal architecture of 5GCity slice manager

As shown in Fig.2, the internal architecture of the Slice Manager consists of three main functional blocks: 1) Network Slice Life-Cycle Management, 2) Slice Repository and 3) Policy Management Function. The Slice Manager is responsible for interacting with the Infrastructure Abstraction to control and manage resource chunks over the physical and virtual infrastructures.

The Network Slice Life-cycle Management is an entry point to the Slice Manager of the 5GCity orchestration platform that can be accessed by the slice user via a dashboard or an API. It is responsible for coordinating and allocating required

slices and services upon request. Network Slice Life-cycle Management can request a slice with certain properties, deploy services, provision a network function, or modify the resources allocated to the existing slice. A slice user can also request to terminate the slice completely and release the assigned resources.

The Slice Repository is responsible for storing information related to the network slices, as well as information related to the virtual resources and physical resources of the slices, such as which virtual resources belong to the slice, which physical resources they are mapped to, and to whom each slice belongs. The required resources are concertized and provisioned by the Infrastructure Abstraction by a series of commands that will eventually directed to VIMs, SDN controllers, and access network controllers that manage compute, storage, network and RAN/WiFi elements.

The parameters stored in our Slice Repository can be elaborated to a qualified and more complete Information Model (aka "slice descriptor"), which is an important gap in the NFV specifications though critical for the management of 5G services within our neutral host scenario.

The Policy Management Function is responsible for run-time policy enforcement over the deployed slice such as creation and expiration dates.

As shown in Fig.2, the Slice Manager module is designed as a set of components, interconnected via external and internal interfaces. The external interfaces connect the Slice Manager with the external components of 5GCity orchestration platform, while the internal interfaces are consumed by the different entities composing the Slice Manager.

The *5GCity orchestrator.I1* is an external interface offered by the Resource Placement and used by the Slice Manager to trigger the decision process for finding an optimal placement of all the resources involved in a network service. As soon as this has been computed, the Slice Manager performs the required configuration actions upon a Network Service through the NFVO interface, for example it can dictate OSM to perform NS deployment over the resources specified in the response of the Resource Placement module.

The *5GCity orchestrator.I3* is another external interface that offered by the Slice Manager and used by the the NFVO in order to check the ownership and other requirements of a slice before initiating NS- and VNF-related actions that contradict these requirements. Note that the NFVO part which can use this interface is a "thin layer" that sits on top of OSM, the MEC orchestrators, and potentially other orchestrators in the future (cf. also [21]).

The last external interface is the *5GCity orchestrator.I4*, which is consumed by the Slice Manager module in order to trigger an action for preparing and setting up the slice components in the infrastructure through the Infrastructure Abstraction module (i.e. VIM technology-independent).

The internal interfaces connect internal elements of the Slice Manager. The Nslcm-Rep internal interface is used by the Network Slice Life-Cycle Management and terminates at the Slice Repository. This interface is used to perform all

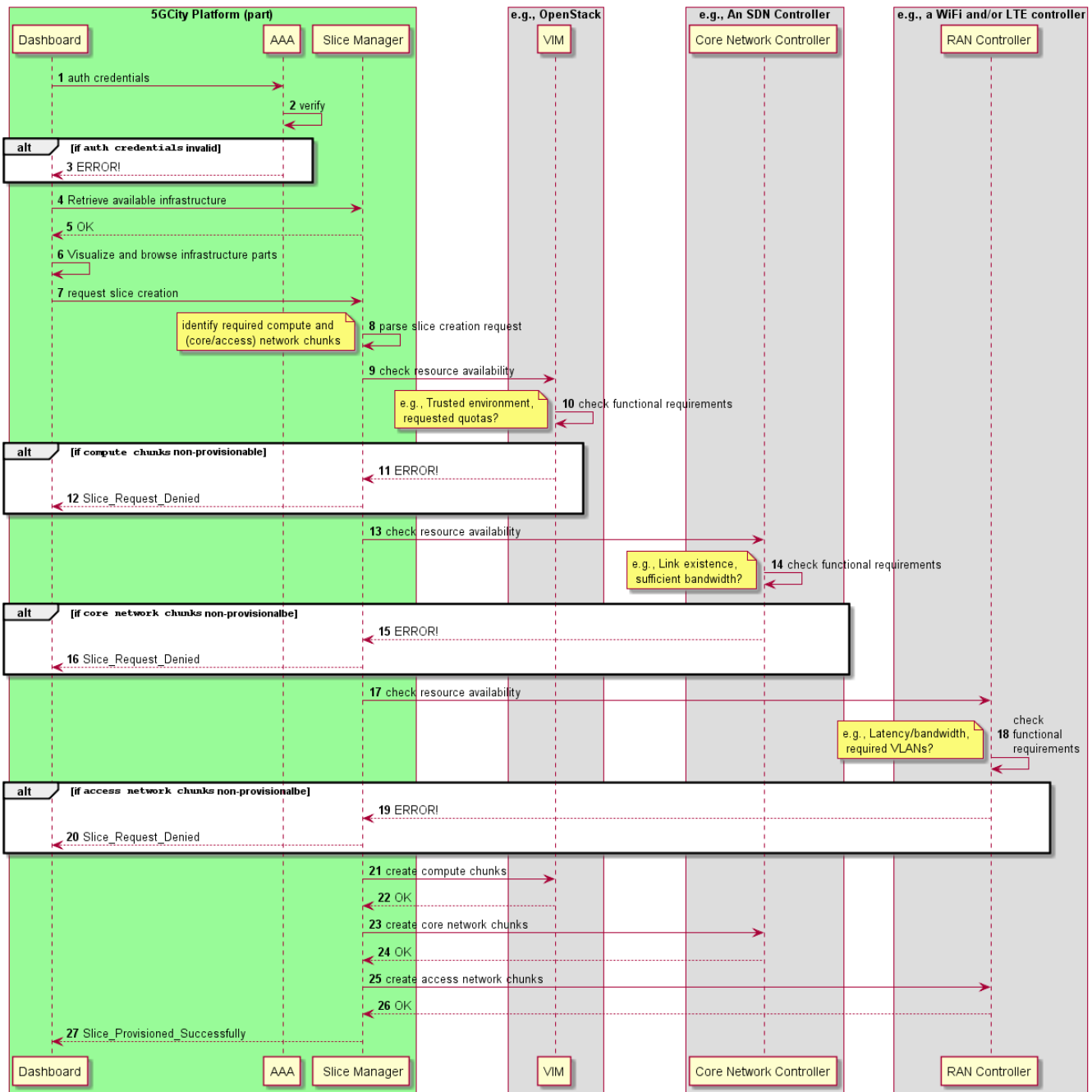


Fig. 3. 5GCity slice provisioning procedure

slice model storage and access operations for the slice model instances of slices that are used in the system. The Nslcm-Pmf internal interface in the 5GCity is used by the Network Slice Life-Cycle Management and terminates at the Policy Management Function. It is used for checking the slice policy prior to the creation of the slice.

### C. 5GCity Slicing-related workflows

Fig.3 exemplarily shows the slice provisioning workflow in the 5GCity orchestration platform. The provisioning includes

i) compute nodes, which are managed by (potentially different) VIMs, ii) physical network links, which are managed by SDN controllers and iii) access nodes, which are managed by RAN controllers. To create an end-to-end network slice, a slice user needs to login to the system through a unified AAA-protected Dashboard in order to send the request to the Slice Manager component in the orchestration layer (namely to its Network Slice Life-Cycle Management component). To do so, a slice user should browse the available infrastructure (with the help of the Infrastructure Abstraction), create chunks of

the components that he desires in their slice, and compose a slice as a collection and interconnection of those chunks. The slice users will be able to check all the infrastructure currently registered in the platform and available to be rented. The Slice Manager is responsible for checking the resource availability and requirements. If all chunks can be set up and configured accordingly, the Slice Manager will be able to create and provision the entire slice successfully. Note that the slice user will only be able to check the status and topology of the virtual resources that they have previously rented being unaware of the resources rented by users with a similar role, even if they share resources. Now the slices are ready to be associated with various network services and the request for service deployment in an infrastructure is dispatched from the Slice Manager through the NFVO as follows:

- The logged-in Slice User browses for services in a catalogue (potentially a catalogue of the underlying NFVO, e.g., an OSM NS Catalogue).
- It sends (through the Slice Manager) a request for service instantiation to the NFVO.
- The NFVO deploys the service over the infrastructure using a multitude of underlying controllers, e.g. OpenStack and OpenDayLight.

#### D. Deployment and Evaluation

To deploy and validate our slicing platform we are installing the 5GCity testbed in the cities of Bristol, Barcelona, and Lucca, executing video broadcasting, video co-production, and other Use Cases upon the generated slices. Among the KPIs (Key Performance Indicators) set in the context of this deployment, the following two are most related for the slicing-aware NFVO lifecycle:

- Avg. Control Plane Delay  $\leq 20$  ms, e.g., for the time between the triggering of the API of the Slice Manager and the triggering of the respective configuration action in the lower layers (infrastructure).
- Slice Creation + Service Instantiation time  $\leq 1$  min.

The initial tests already achieve these KPIs, which is vital for not impeding our 5G Use Cases. Although we expect our platform to perform control plane operations faster than the more complex slicing frameworks discussed in related work, we considered detailed measurements and comparisons of such metrics to be meaningless for the moment, because the results heavily depend on the exact (city) deployments, used infrastructure and hardware, Use Cases etc. Designing scenarios and metrics that could enable such an evaluation is however a subject of future work.

## V. CONCLUSION

Network slicing is a comprehensive solution for 5G networks, which allows multiple logical networks to run over a shared infrastructure. In this paper, we have analyzed and evaluated several relevant orchestration frameworks capable of supporting a slicing solution. These solutions show some limitations which we address with the 5GCity slicing-aware

solution described in this paper. In fact, the 5GCity orchestration platform can provide a comprehensive network slicing solution for 5G networks, in which end-to-end slices are provisioned across various heterogeneous network elements, from the core network to the edge, involving the use of various VIMs and SDN/RAN controllers. Finally, it allows the integration of NFV and MEC orchestration, thus exploiting the existence of edge-related runtime VNF parameters that are part of the MEC information models. Deployments in Barcelona, Bristol, and Lucca are currently confirming the achievements of important platform KPIs related to control plane latency and slice/service instantiation time.

## ACKNOWLEDGMENT

This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 761508.

## REFERENCES

- [1] NGMN Alliance, "5G white paper," *Next generation mobile networks, white paper*, pp. 1–125, 2015.
- [2] 5G PPP Architecture Working Group, "View on 5G architecture," *White Paper*, July, 2016.
- [3] ETSI MEC, "Mobile Edge Computing (MEC) - Deployment of Mobile Edge Computing in an NFV Environment, ETSI GR MEC 017 V1.1.1," 2018.
- [4] I. Neokosmidis, et al., "Assessment of Socio-techno-economic Factors Affecting the Market Adoption and Evolution of 5G Networks: Evidence from the 5G-PPP CHARISMA project," *Telematics and Informatics*, vol. 34, no. 5, pp. 572–589, 2017.
- [5] K. Samdanis, et al., "from network sharing to multi-tenancy: The 5G network slice broker," *IEEE Communications Magazine*, vol. 54, no. 7, pp. 32–39, July 2016.
- [6] X. Li, et al., "Network Slicing for 5G: Challenges and Opportunities," *IEEE Internet Computing*, vol. 21, no. 5, pp. 20–27, 2017.
- [7] NGMN, "Description of Network Slicing Concept," Next Generation Mobile Networks Alliance, Tech. Rep., 01 2016.
- [8] 3GPP, "Telecommunication management; Study on management and orchestration of network slicing for next generation network - TR 28 801," 3rd. Generation Partnership Project, Tech. Rep., 2016.
- [9] ETSI, "Network Functions Virtualisation (NFV) - Report on Network Slicing Support with ETSI NFV Architecture Framework, GR NFV-EVE 012 v3.1.1," European Telecommunications Standards Institute, Tech. Rep., 12 2017.
- [10] Tacker, "Tacker - OpenStack NFV Orchestration platform," <https://wiki.openstack.org/wiki/Tacker>.
- [11] OpenBaton, "OpenBaton - An extensible and customizable NFV MANO-compliant framework," <https://openbaton.github.io/>.
- [12] TeNoR project, "TeNoR T-nova Orchestration platform," <http://www.tnova.eu/open-source/>.
- [13] ONAP, "Open Network Automation Platform," <https://www.onap.org/>.
- [14] SONATA project, "SONATA NFV: Agile Service Development and Orchestration in 5G Virtualized Networks," <http://www.sonata-nfv.eu/>.
- [15] 5Gtango, "5Gtango project," <https://www.5gtango.eu/>.
- [16] OSM, "Open Source Mano," <https://osm.etsi.org/>.
- [17] G. Lavado, "Whitestack: Achieving end-to-end NFV with OpenStack and Open Source MANO," *openstack summit presentation*, 5 2018.
- [18] C. Colman-Meixner, et al., "Deploying a Novel 5G-enabled Architecture on City Infrastructure for Ultra-High Definition and Immersive Media Production and Broadcasting," *accepted at IEEE Transactions on Broadcasting*, 2019.
- [19] 5GCity project, "Deliverable 4.1, 5GCity Orchestration Design, Service Programming and Machine Learning Models," Tech. Rep., 2018.
- [20] ETSI, "Network Functions Virtualisation (NFV) - Terminology for Main Concepts in NFV, GS ETSI NFV 003 v1.4.1," 2018.
- [21] G. Baldoni, et al., "Edge Computing Enhancements in an NFV-based Ecosystem for 5G Neutral Hosts," *IEEE NFV-SDN*, 2018.