NFV Information Model Extensions for Improved Reliability and Lifecycle Management

Giovanni Fausto Andreotti, Paolo Secondo Crosta, Emanuele Miucci, Giuseppe Monteleone
ITALTEL S.p.A.
Milan, Italy
e-mail: {fausto.andreotti, paolosecondo.crosta, emanuele.miucci, giuseppe.monteleone} @italtel.com

***Abstract*—This paper focuses on improvements in Management and Orchestration within the Network Function Virtualization (NFV) domain. The key benefits are related to automation in the Virtual Network Function (VNF) lifecycle, adaptation to different network traffic loads and new models for improving network resilience. These could be achieved by introducing some extensions of the NFV Information Model. Firstly, we propose the introduction in the VNF Descriptor (VNFD) of an Information Element providing the dependencies between Virtual Deployment Units (VDUs) that allows managing the VDUs' instantiation process in a more efficient way. Secondly, we suggest an extension related to the execution of script(s) - including the possibility to pass parameters - in response to particular events detected by the VNF Manager (VNFM). Finally, we propose a new Information Element for describing high availability features, thus defining possible redundancy schemes that allow the execution of specific operations tailored for each single instance of the VNF. In order to support the validity of the proposed approach, we provide some practical examples based on a real implementation of a VNF Session Border Controller.**

Keywords - Network Function Virtualization, Orchestration, Information Model, VNF Descriptor, Lifecycle Management.

I. INTRODUCTION

Network Function Virtualization (NFV), in addition to Software Defined Networking (SDN), is a rapidly emerging approach in the telecommunication field. By adopting NFV, Communication Services Providers (CSP) expect to achieve consistent cost reductions with respect to the current situation in which network equipment consists of proprietary black boxes, containing a bundle of proprietary Software (SW) and customized Hardware (HW) provided by a single Telecom Equipment Manufacturer. The adoption of the NFV concept is just the starting point to introduce in the Telco world the benefits that virtualization has brought in the Information Technology (IT) sector. Besides significant cost reductions, NFV also raises great expectations on the possibility (a) to achieve a never experienced network flexibility and service agility, (b) to introduce automation in all lifecycle of Network Functions (NFs) from deployment, installation and commissioning to operational phases, (c) to adapt the network to different traffic loads thanks to a novel cloud elasticity model, and (d) to develop new models for improving network resilience [10].

In fact, the objective of NFV is to allow Service Operators to achieve a high reduction in capital investments along with greater operational agility by implementing challenging architectural updates and deep changes in service models and operating procedures.

Virtualization is not a new technology. What is new is the way to use virtualization in Telco environments. Thanks to NFV, it is possible a paradigm shift moving from manual, complex and error prone configuration processes to deployment automation. Automation means flexibility, agility and the possibility to minimize complexity and errors. After the deployment, when the function is in operation it is possible to perform monitoring, scaling, healing, failover, continuous delivery and infrastructure upgrades.

In the NFV architecture, specified by the European Telecommunications Standards Institute (ETSI) NFV Industry Specification Group (ISG) [8], three main domains are identified [3]:

- Virtualized Network Function (VNF), as the software implementation of a network function which is capable of running over the Network Functions Virtualization Infrastructure (NFVI)
- NFVI, including hardware resources (Compute, Storage, Networking) and the virtualization layer that provides Virtual resources (Virtual Compute, Virtual Storage, Virtual Networking) supporting the execution of the VNFs
- NFV Management and Orchestration (MANO), which covers the lifecycle management of VNFs and Network Services (NS), managing the resources of NFVI and focusing on all virtualization-specific management tasks necessary in the NFV framework.

In this paper, we will focus on the Management and Orchestration domain. In particular, we propose some extensions of the NFV Information Model that can be used to increase efficiency in the lifecycle management of Network Functions and to improve the overall system reliability. Our experience as VNF provider conducted us to identify some flaws in the NFV Information Model and corresponding specific enhancements that future implementations could benefit from. In order to achieve this goal, we introduce some additional Information Elements (IEs) in the VNF Descriptors (VNFDs) that allow a deeper control of VNFs.

The paper is organized as follows: Section II gives an overview of the ETSI standard model for NFV, with regard

 ISBN: 978-1-61208-546-3

to the MANO architecture, including the lifecycle management of Virtual Network Functions and a general description of the NFV Information Model. Section III, the main part of the paper, provides a rationale for the extensions to VNF Descriptors, as well as some practical examples to support the validity of the proposed approach. Finally, Section IV will draw the conclusions.

II. ETSI NFV ARCHITECTURAL MODEL

The ETSI NFV architectural framework [1] [2] is shown in Figure 1.

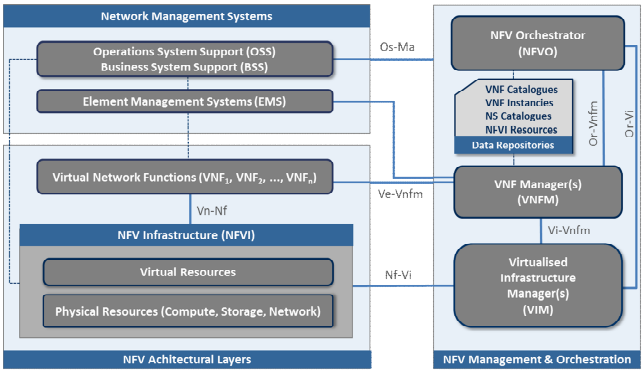

Figure 1. ETSI NFV architectural model.

The functional blocks in the framework can be grouped into three main entities: (1) NFV Architectural Layers, (2) NFV Management and Orchestration, and (3) Network Management Systems. These entities, as well their constituent functional blocks, are connected together using a set of defined reference points. The NFV architectural layers include the NFVI and VNFs. NFVI is the combination of both hardware and software resources, which make up the environment in which VNFs are deployed, while VNFs are implementations of NFs that are deployed on those virtual resources.

A. NFV-MANO Framework

The NFV MANO [3] consists of three functional blocks, the Virtualized Infrastructure Manager (VIM), the VNF Manager (VNFM) and the NFV Orchestrator (NFVO), and four data repositories (NS Catalogues, VNF Catalogues, VNF Instances and NFVI Resources).

1) VIM - It manages and controls NFVI physical and virtual resources in a single infrastructure domain. This implies that an NFV architecture may contain more than one VIM, with each of them managing or controlling NFVI resources from a given infrastructure provider. In principle, a VIM may be specialized in handling a certain type of NFVI resource (e.g., compute-only or storage only), or could manage multiple types of NFVI resources (e.g., nodes in the NFVI).

2) VNFM - Each VNF instance is assumed to have an associated VNFM. The VNFM is responsible for the management of the lifecycle of VNFs. A VNFM may manage a single or multiple VNF instances of the same or different types. It is also possible that a single VNFM handles all the active VNF instances for a certain domain.

3) NFVO - It is aimed at combining more than one function so as to create end-to-end services. To this end, the NFVO functionality can be divided into two broad categories: (a) resource orchestration, and (b) service orchestration. Resource orchestration is used to provide services that support accessing NFVI resources in an abstract manner regardless of the type of VIMs, as well as governance of VNF instances sharing resources of the NFVI infrastructure. Service orchestration deals with the creation of end-to-end services by composing different VNFs, and the topology management of the network services instances.

4) Data Repositories - These are databases that keep different types of information in the NFV MANO. Four types of repositories can be considered: (a) the NS Catalogue is a set of pre-defined templates, which define how network services may be created and deployed, as well as the functions needed for the service and their connectivity, (b) the VNF Catalogue is a set of templates which describe the deployment and operational characteristics of available VNFs, (c) the NFVI Resources repository holds information about available/allocated NFVI resources, and (d) the VNF Instances repository holds information about all function and service instances throughout their lifetime.

B. VNF Lifecycle Management

NFV is based on the principle of separating network functions from the hardware where they run on by using virtual hardware abstraction. The virtualization of network functions will change their lifecycle management by introducing automation and flexibility. Lifecycle management of VNFs is possible after a preliminary operation, the so-called VNF package on-boarding. After that, by accessing to a VNF catalogue it is possible to create one or more VNF instances of a VNF. A VNF instance corresponds to a run-time instance of the VNF software, i.e., all the VNF components are instantiated and the internal and external network connectivity configured.

During its lifecycle a VNF instance can be in one of the following states:

- instantiable, i.e., the on-boarded process for the VNF has been correctly performed
- instantiated, i.e., not configured, configured & not in service, configured & in service
- terminated.

The lifecycle is controlled by a set of operations, described in the following list:

- VNF Instantiation
- VNF instance Scaling (horizontal/vertical)
- VNF instance Update or Upgrade
- VNF instance Healing
- VNF instance Termination.

It is worth mentioning that a subset of lifecycle management (LCM) operations, as VNF Instantiation and Termination, are always available for every single VNF instance.

Some other operations, such as VNF Scaling, are performed if required by the deployment flavor of the VNF

 ISBN: 978-1-61208-546-3

instance. Some procedures related to VNF lifecycle management provide both manual and automatic mechanisms. Scaling, for instance, can be manually requested or automatically performed when triggered upon the occurrence of specific events defined as criteria for matching rules and actions for scaling. All the operations during the lifecycle are handled by specific workflows that are built on the basis of the tasks to perform and the associated parameters (i.e., in case of instantiation, the VNF ID, the deployment flavor, etc.). A workflow has a starting point and different tasks that can be performed sequentially or in parallel, in order to perform all the necessary activities.

A VNFD is used for defining workflows for the automation of specific phases. For instance, by modelling the VNFD using a description language it is possible to describe the VNF with a service template in terms of components (e.g., Virtual Deployment Units or VDUs), relationships (dependencies, connections) and management processes. The management processes can be defined as plans describing how a VNF instance is instantiated and/or terminated considering that the VNF is a complex application composed by different nodes.

C. NFV Information Model

In this subsection, we provide a brief description of the IEs used to carry the necessary information about a VNF. In fact, one of the ways the IEs can be used is as part of descriptors in a catalogue or template context.

The IEs to be handled by the NFV MANO, including the ones contained in the VNFD, need to guarantee the flexible deployment and portability of VNF instances on multi-vendor and diverse NFVI environments, e.g., with diverse computing resource generations, diverse virtual network technologies, etc. To achieve this goal, hardware resources need to be properly abstracted and VNF requirements must be described in terms of such abstractions.

With reference to Figure 1, the Vi-Vnfm interface [4] enables the interaction between the VNFM and the VIM, providing the methods to operate cloud resources on the NFVI, in particular computing, storage and networking resources. After the upload of the VNF package, the VNFM under operator's request or by a request coming from the Or-Vnfm interface [5] can start to perform the lifecycle management of a VNF via the Ve-Vnfm interface [6].

The VNF package contains all artifacts needed to perform the lifecycle management for the associated VNF:

- descriptors (VNFDs)
- metadata, scripts and other proprietary artifacts
- optionally, SW images of the VNF Components (VNFCs).

The VNFD is a template which describes a VNF in terms of its deployment and operational behaviour requirements. It is primarily used by the VNFM in the process of VNF instantiation and lifecycle management of a VNF instance. The information provided in the VNFD is also used by the NFVO to manage and orchestrate network services and virtualised resources on the NFVI. The VNFD also contains connectivity, interface and KPIs requirements that may be used by NFV MANO functional blocks to establish appropriate virtual links within the NFVI between its VNFC instances, or between a VNF instance and the endpoint interface to the other NFs. The VNFD contains all the information needed for the lifecycle management, such as:

- basic information for VNF identification
- internal networks description
- VDUs description: for each VNFC (corresponding to a VM type) it is defined the flavor of the VM, the SW image for the VM and the number of VMs to activate. Configuration scripts are also provided for the lifecycle management phases and triggered during the instantiation or by specific events
- meters or measurements associated to VNF scaling. In fact, when the VNF is in operation, measurements can be collected from the VNF itself and/or from the infrastructure, e.g., the number of session attempts per second; the contemporary active sessions; CPU, RAM, disk usage; etc.
- alarms and associated actions. Criteria may be defined in terms of rules to check on the measurements, e.g., the value of a meter is greater than a specific threshold for a specified period of time; etc. When a rule is matched, specific actions can be performed, such as the activation of a scaling policy, etc.
- scaling policies, e.g., to add an instance of a VNFC when specific conditions are matched.

III. ETSI NFV INFORMATION MODEL EXTENSIONS

The aim of this section is to provide a detailed description, the rationales, the relationships and the benefits of each proposed extension of the Information Model.

The ETSI GS NFV-IFA 011 [7] is the reference specification that provides requirements for the structure and the format of a VNF Package to describe the VNF properties and associated resource requirements in an interoperable template.

In the following, we describe the extensions – in terms of newly added IEs – of this specification, and related use cases to show the benefits introduced in the lifecycle management of VNFs.

An overview of the Information Model extensions is summarized in Table I. Generally, a VNF is composed by one or more VNFCs that are described by means of deployment templates, i.e., VNFD and related VDUs, respectively. In the table we have listed the proposed attribute extensions - *dependencies*, *MetadataScript* and *highAvailability* IEs - and their relationships with descriptors at the VNFD (vnfd) or VDU (vnfd:vdu) level.

TABLE I. OVERVIEW OF INFORMATION MODEL EXTENSIONS

ETSI GS NFV-IFA 011 Specification Extensions		
Entity relationship	VNF	VNFC1, ..., VNFCn
Deployment template	VNFD	VDU
Descriptor	vnfd	vnfd:vdu
proposed attribute extensions (new IEs)	dependencies	MetadataScript, highAvailability

All the use cases are based on a real implementation of a Session Border Controller (SBC). This VNF provides its functionalities thanks to the interworking of 5 VNFCs, as

depicted in Figure 2: a front-end load balancer (FELB), an Operation and Maintenance module (OAM), a component managing the SIP signalling traffic (SIG) working with a database (DB) for storing the information of the active calls and a Border Gateway (BGW) function engaged when audio or media transcoding is required for the incoming traffic.

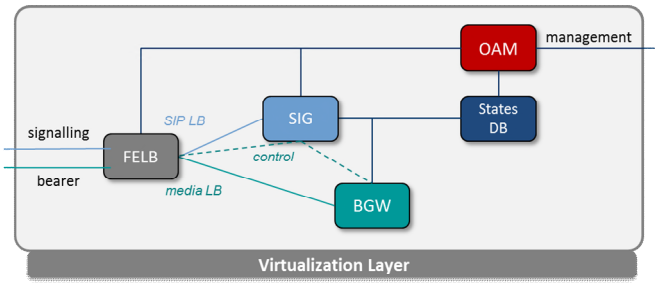

Figure 2. SBC logical components.

These components can be organized and managed to implement protection mechanisms in order to guarantee redundancy and to support high availability requirements.

A. Dependencies

In this subsection, we provide the description of the *dependencies* and *VduDependencies* attributes that could be added to indicate the dependencies among the VDUs during the instantiation process. In fact, sometimes it is necessary to coordinate the process of instantiation with information that is available - at platform level (e.g., IP addresses) or application level - at specific times. As originally proposed in the ETSI MANO specification [3], we believe it is necessary to include in the VNFD an IE providing the dependencies between VDUs since it describes constraints that affect the structure of a VNF.

Table II shows the structure of the *dependencies* IE that has to be added to the VNFD standard description [7].

TABLE II. *DEPENDENCIES* INFORMATION ELEMENT

Attribute(s) of the *dependencies* VNFD IE	
Attribute	dependencies
Qualifier	M
Cardinality	0..N
Content	VduDependencies
Description	Describes dependencies between VDUs. Defined in terms of source and target VDU, i.e., target VDU "depends on" source VDU. In other words, sources VDU shall exist before target VDU can be instantiated/deployed.

The *VduDependencies* IE provides indications on the order in which VDUs associated to the same VNFD have to be instantiated. The contents of a *VduDependencies* type shall comply with the format provided in Table III.

TABLE III. *VDUDEPENDENCIES* INFORMATION ELEMENT

Attribute(s) of the *VduDependencies* IE		
Attribute	source	target
Qualifier	M	M
Cardinality	1..N	1..N
Content	Identifier	Identifier
Description	The listed VDUs shall be instantiated before the VDUs listed in the target parameter.	The listed VDUs shall be instantiated after the VDUs listed in the source parameter have been instantiated completely.

In Figure 3, it is shown a sequence diagram based on a real implementation of a VNF SBC. It is worth to mention that in this case, all the VNFCs should be instantiated after the OAM component, since this component has a central role coordinating the communications with the VNF Manager on behalf of all the other VNFCs.

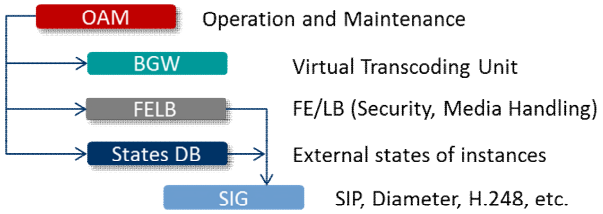

Figure 3. SBC components sequence diagram.

As shown in Figure 4, the dependencies explained above have been expressed by using the JavaScript Object Notation (JSON) [9].

```
"dependency": [{
        "vdu-id": "felb",
        "depends-on": "oam"
        }, {
        "vdu-id": "sig",
        "depends-on": "oam"
        }, {
        "vdu-id": "states",
        "depends-on": "oam"
        }, {
        "vdu-id": "bgw",
        "depends-on": "oam"
        }, {
        "vdu-id": "sig",
        "depends-on": "felb"
        }, {
        "vdu-id": "sig",
        "depends-on": "states"
    }].
```

Figure 4. *Dependencies* script example.

Alternatively, ETSI envisages the use of a scripting language to express dependencies on virtual resources, but at the time of this writing, no consensus has been reached yet about the format to be used and the standardization process of a Domain Specific Language is still underway.

B. MetadataScript

In this subsection, we provide the description of the *MetadataScript* and *LifeCycleMetadataScript* attributes. These extensions are related to the execution of script(s) in response to particular events detected on a VNFM reference point. The ETSI GS NFV-IFA 011 specification [7] already supports the execution of scripts – but only at the VNF level

- with the *LifeCycleManagementScript* IE that can be launched in response to lifecycle events or external stimulus detected by the VNFM. These LCM scripts should be embedded in the VNF Package and used in the LCM execution environments provided by generic VNF Managers. In par.6.2.6, the specification provides a list of requirements (VNF_PACK.LCM.001) for the scripting Domain Specific Language (DSL).

Table IV shows the structure of the *MetadataScript* IE that has to be added to the VDU standard description [7].

TABLE IV. *METADATASCRIPT* INFORMATION ELEMENT

Attribute(s) of the *MetadataScript* VDU IE	
Attribute	MetadataScript
Qualifier	M
Cardinality	0..N
Content	LifeCycleMetadataScript
Description	Includes a list of events and corresponding scripts producing metadata required during the VDU instantiation.

A *LifeCycleMetadataScript* IE, instead of the *LifeCycleManagementScript* formerly defined in the original specification, has been defined and extended to comply with specific needs originated from practical use cases.

The attributes of the *LifeCycleMetadataScript* IE shall follow the indications provided in Table V. The advantages of this extension, compared with the existing standard specification, are (a) the possibility to execute script(s) at the VDU level, and (b) the possibility to pass parameter(s) to the script(s).

TABLE V. LIFECYCLEMETADATASCRIPT INFORMATION ELEMENT

Attribute(s) of the *LifeCycleMetadataScript* IE				
Attribute	event	script	role	parameter
Qualifier	M	M	M	M
Cardinality	1	1	1	0..N
Content	String	Not specified	String	Not specified
Description	Describes a VNF lifecycle event or an external stimulus detected on a VNFM reference point.	Includes metadata template.	Describes the role of the VDU in redundancy scheme(s). Possible values are “Active” or “Passive”.	VDU specific parameters passed to the script. Each of them represents the run-time value of a NFVI resource (e.g., IP address, VNFC instance name, etc.).

In Figure 5, we provide an example based on a real implementation of the Session Border Controller VNF.

It is worth noting that this IE allows the VNFM a complete flexibility in the lifecycle management process of different VNFs/VNFCs: in this example, the script for the instantiation of the OAM component needs information from the infrastructure (i.e., the IP address of the connection point cp_oam_int, the hostname of the VNFC and the related domain_name) which will be available only at runtime.

```
"lifeCycleMetadataScript": [{
            "event": "CREATION",
            "script": "instantiate_oam",
            “role”:”active”,
            "parameters": [
            "$$param.cp_oam_int.ipaddress",
            "$$param.hostname",
            "$$param.domain_name"
            ]
    }].
```

Figure 5. *LifeCycleMetadataScript* script example.

According to our experience, the proposed syntax is general and can be easily adapted in order to suit different VNFM providers.

C. HighAvailability

In this subsection, we provide the description of the *highAvailability* attribute. Availability is defined as the state to perform a required function at a given instant of time or at any instant of time within a given time interval, assuming that the external resources, if required, are provided. This attribute is important for telecom operators that want to offer their customers services that perform as expected whenever the service is requested.

Comparing the VDU IE originally proposed in the ETSI MANO specification [3] with the one described in ETSI GS NFV-IFA 011 [7], the *high_availability* IE is no longer specified. The reason provided by ETSI is based on the assumption that the VNFM alone can hardly manage the multitude of redundancy schemes: high availability policies should be performed by each single VNFC at the application level.

Instead, in our opinion, an attribute specified at the VDU level allows the VNFM to execute specific operations tailored for each single instance, thus simplifying the implementation of the VNF itself.

Table VI shows the structure of the *highAvailability* IE that has to be added to the VDU standard description [7].

TABLE VI. *HIGHAVAILABILITY* INFORMATION ELEMENT

Attribute(s) of the *highAvailability* VDU IE	
Attribute	highAvailability
Qualifier	M
Cardinality	0..1
Content	Enum
Description	Defines redundancy model to ensure high availability. Possible values are “ActiveActive” or “ActivePassive”. • ActiveActive: implies that two instance of the same VDU will co-exists with continuous data synchronization. • ActivePassive: implies that two instance of the same VDU will co-exists without any data synchronization.

For example, the statement *"highAvailability":"ActivePassive"* implies the active part to request a set of parameters which can be different from the configuration set which is needed by the passive part.

Furthermore, the active and passive counterparts would require a different set of instantiation/configuration scripts. As shown in Figure 5, this condition could be easily enforced by using an additional attribute defined in the *LifeCycleMetadataScript* IE, i.e., the *role* attribute, which can assume "*active*" (or "*passive*") values, as described in Table V.

IV. CONCLUSIONS

In this paper, some improvements have been proposed in Management and Orchestration of Virtual Network Functions, based on extensions of the Information Model specified by ETSI. The need for these additional IEs in the VNFD/VDU descriptor(s) has been originated from a real implementation of a novel NFV-compliant Session Border Controller solution. The key points addressed have been more flexibility in the management of network functions and increased reliability of virtualized systems. As a result of this work, we provided a detailed description, rationales, relationships and possible benefits coming from the new attributes, as well as practical examples to support the validity of the proposed approach. The extensions have been applied and successfully validated on a SBC solution, thus demonstrating very useful and easy to fit into the ETSI specification framework.

REFERENCES

[1] ETSI Industry Specification Group (ISG) NFV, "ETSI GS NFV 002 V1.1.1: Network Functions Virtualization (NFV); Architectural Framework," http://www.etsi.org/deliver/etsi_gs/nfv/001_099/002/01.01.01_60/gs_nfv002v010101p.pdf, [retrieved: March, 2017].

[2] ETSI Industry Specification Group (ISG) NFV, "ETSI GS NFV 003 V1.1.1: Network Functions Virtualization (NFV); Terminology for Main Concepts in NFV," http://www.etsi.org/deliver/etsi_gs/NFV/001_099/003/01.01.01_60/gs_nfv003v010101p.pdf, [retrieved: March, 2017].

[3] ETSI Industry Specification Group (ISG) NFV, "ETSI GS NFV-MAN 001 V1.1.1: Network Functions Virtualization (NFV); Network Functions Virtualization Management and Orchestration", December 2014.

[4] ETSI Industry Specification Group (ISG) NFV, "ETSI GS NFV-IFA 006 V2.1.1: Network Functions Virtualization (NFV); Management and Orchestration; Vi-Vnfm reference point – Interface and Information Model Specification", April 2016.

[5] ETSI Industry Specification Group (ISG) NFV, "ETSI GS NFV-IFA 007 V2.1.1: "Network Functions Virtualization (NFV); Management and Orchestration; Or-Vnfm reference point – Interface and Information Model Specification", October 2016.

[6] ETSI Industry Specification Group (ISG) NFV, "ETSI GS NFV-IFA 008 V2.1.1: "Network Functions Virtualization (NFV); Management and Orchestration; Ve-Vnfm reference point – Interface and Information Model Specification", October 2016.

[7] ETSI Industry Specification Group (ISG) NFV, "ETSI GS NFV-IFA 011 V2.1.1: "Network Functions Virtualization (NFV); Management and Orchestration; VNF Packaging Specification", October 2016.

[8] ETSI - NETWORK FUNCTIONS VIRTUALISATION http://www.etsi.org/technologies-clusters/technologies/nfv.

[9] The JSON Data Interchange Format. http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf.

[10] V. Eramo, E. Miucci, M. Ammar; and F. G. Lavacca, "An Approach for Service Function Chain Routing and Virtual Function Network Instance Migration in Network Function Virtualization Architectures," in IEEE/ACM Transactions on Networking, vol.PP, no.99, pp.1-18.

 ISBN: 978-1-61208-546-3