

The Architecture of Generic VNFM using TOSCA

Byung Yun Lee*, Bhum Cheol Lee*

* ETRI(Electronics and Telecommunications Research Institute), Korea

bylee@etri.re.kr, bcleee@etri.re.kr

Abstract— Network function virtualization is quickly gaining acceptance as the new approach to delivering communication services. The promises of greater flexibility and dramatically reduced time to introduce new services coupled with the cost advantages, are driving communications service providers(CSP) around the world to begin deploying NFV-based services. An NFV operation system is divided into the NFV Orchestrator (NFVO), VNF Manager (VNFM), and VIM (Virtual Infrastructure Manager) subsystems in accordance with the ETSI standard, and the VNFM is defined as the system provided by the VNF manufacturer. In this paper, we present a generic VNF management system for administration and operation of the VNF under the following structure and to specify the Or-Vnfm, Ve-Vnfm and Vi-Vnfm interfaces, with the goal of developing a generic and scalable VNFM to support various VNFs.

Keywords— Preparation of papers, camera-ready, two-column format, IEEE format.

I. NFV MANAGEMENT FRAMEWORK

ETSI ISG has created the industry standards required in a short time to target specific technology areas that industry requires, for NFV management framework [1].

NFV management framework consists of three functional entities including NFV Orchestrator, VNF Manager and Virtualized Infrastructure Manager.

The NFVO(NFV Orchestrator) is responsible for the lifecycle management of Network Services across the entire Operator's domain (e.g. multiple VIMs: Virtualized Infrastructure Manager) and the orchestration of NFVI(Network Function Virtualization Infrastructure) resources across multiple VIMs fulfilling the Resource Orchestration functions.

The VNF Manager is responsible for the lifecycle management of VNF instances. Each VNF instance is assumed to have an associated VNF Manager. A VNF manager may be assigned the management of a single VNF instance, or the management of multiple VNF instances of the same type or of different types.

The Virtualized Infrastructure Manager is responsible for controlling and managing the NFVI compute, storage and network resources, usually within one operator's infrastructure domain. A VIM may be specialized in handling a certain type of infrastructure resource (e.g., compute-only, storage-only,

networking-only), or may be capable of managing multiple types of infrastructure resources.

NFV framework provides a virtual container interface in order to provide independent execution environment, and required resources (network, compute, storage, accelerator) rely on NFVI. Each VNF is given all the necessary resources through a virtual container interface.

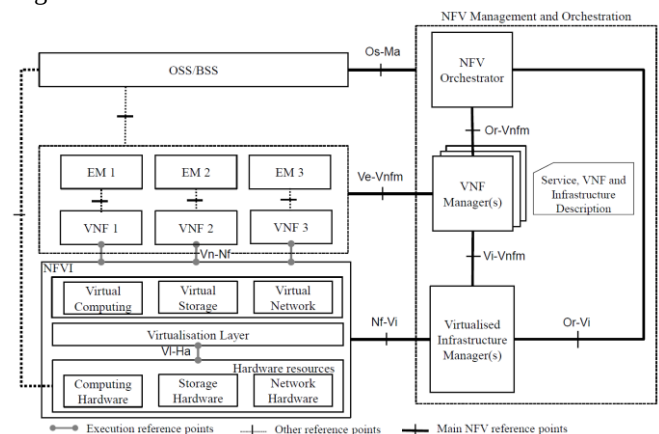
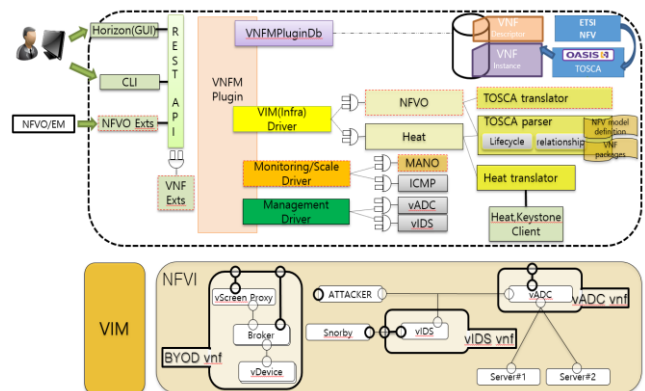


Figure.1 NFV MANAGEMENT FRAMEWORK (ETSI ISG)

II. ARCHITECTURE OF GENERIC VNFM

For a cloud NFV platform to provide the NS (Network Service) and the VNF included within it, an NFV management system must provide the means to install VNF using the NFVI resources and to control and manage the VNF functions needed by the OSS/BSS. An NFV operation system is divided into the NFV Orchestrator (NFVO), VNF Manager (VNFM), and VIM (Virtual Infrastructure Manager) subsystems in accordance with the ETSI standard, and the VNFM is defined as the system provided by the VNF manufacturer.



However, the objective of the “development of a generic VNF management system for administration and operation of the NFV” is to design the VNF Manager under the following structure and to specify the Or-Vnfm, Ve-Vnfm and Vi-Vnfm interfaces, with the goal of developing a generic and scalable VNFM to support various VNFs

VNFs Management technology for a variety of network services using standard interfaces and TOSCA-based information modeling technology to deploy a variety of NFVI.

- TOSCA Parser Development
- TOSCA Heat Translator Development
- ETSI ISG IFA007, IFA008 interworking interface development
- Life cycle, Scale Up/Down functionality of VNF
- Vendor Specific Extensions for the Function Plug-In Framework Development

(1) VNFM-Oriented Block Design

The generic VNFM has a VNFM-Oriented management structure under which the VNFM generates the VNF through the Ve-Vnfm and Vi-Vnfm interfaces[2].

The following diagram shows the VNFM- Oriented block diagram.

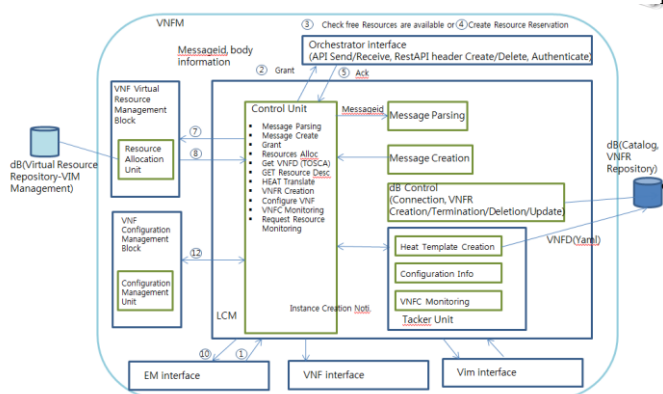


Figure 2. VNFM-oriented management structure

VNFM consists of LCM(Life Cycle Management block, VNF virtual resource management block, VNF configuration management block and Orchestrator interface block, EM interface block, VNF interface block, VIM interface block.

- Orchestrator interface Block

NFVO Interface block may be implemented by roughly classified into two types of the Work Flow.

- VNFM oriented request / response processing (Direct)
- NFVO oriented request / response processing (Indirect)

In this block, NFVO is asked to create/update/terminate an instance of VNF and returns the result to VNFM. In VNFM oriented case, VNFM has the functions about request Grant messages to NFVO and, Notify messages about VNF creation, and Request Grant Lifecycle Operation to the NFVO.

- LCM(Life Cycle Management Block)

VNF LCM block performs a control VNF management functions requested through the interface between the NFVO or VNF, EM. It provides the following features.

- VNF instance lifecycle management
- VNF performance and fault management
- VNF lifecycle change notification
- VNF lifecycle change granting
- VNF virtual resource allocation information request

- VNF virtual resource management Block

VNF virtual resource management Block manages the virtual resources associated with the VNFs, collecting the performance information of VNF and delivers the information to LCM.

- Management of virtual resource
- Receive alarm message
- Monitoring of virtual resource

- VNF configuration management Block

VNF configuration management Block handles VNF scaling interfaces, and it passes VNF instance, SW installation, and setup information to the VNF at the time of initial and recovery time

- Request VNF scaling
- VNF instance settings
- Performance and fault information management of VNF

VNF interface Block

VNF interface Block is responsible for the interface to the generated instance VNF, in association with the VNF Configuration Management block

- VNF communication interface
- VNF communication protocol processing
- VNF message communication

- EM interface Block

EM interface Block handles the interface with respect to EM and VNF operational settings.

- EM communication interface
- EM communication protocol processing
- EM message communication

- VIM interface Block

VIM interface block sends to the VNF virtual resource management block about a periodic, non-periodic data that is received from VIM, and convert the message from the VNF virtual resource management Block to VIM API.

- REST message creation
- Parsing the message
- Authentication capabilities for accessing the VIM

- ① EM send vnfDescriptionId information to VNFM .
- ② VNFM request Grant lifecycle operation to NFV Orchestrator
- ③ Orchestrator check available resources(e.g. Pool, resource type, CPU, memory)
- ④ Optionally make a reservation to create resource
- ⑤ VIM sent to Orchestrator about result of reservation
- ⑥ Orchestrator respond the request of the Grant operation from the VNFM
- ⑦ VNFM allocate the resources from the Orchestrator
- ⑧ VIM respond to the VNFM about the resource allocation
- ⑨ VNFM configure the VNF created in the VIM(deployment specific parameters)
- ⑩ VNFM respond the request of the initiate VNF request from the EM
- ⑪ EM add the initiated VNF as a managed device
- ⑫ EM configure the application specific parameters to operate the initiated VNF.
- ⑬ VNFM notify the Orchestrator about successful instantiation of VNF

Orchestrator map VNF to VIM and resource pool

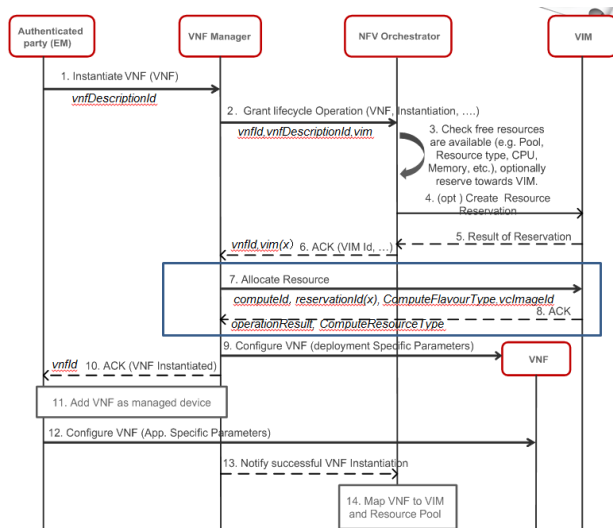


Fig 2 VNF instance creation procedure from EM

(2) Orchestration-Oriented Block Design

The generic VNFM must be able to provide a orchestration-centered VNF management structure through the Ve-Vnfm interface when it is interfaced to the Or-Vnfm. The following diagram shows the orchestration-centered block

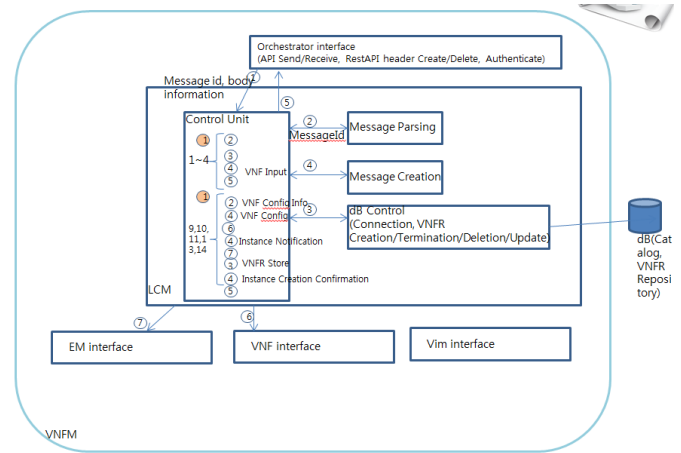


Figure 3. Orchestration-Centered management structure

III. INTERFACE DESIGN AND MODELLING

We defines the Or-Vnfm, Ve-Vnfm and Vi-Vnfm interfaces and the detailed TOSCA template design needed to support them for the development of a generic VNFM

① Or-Vnfm Interface

we provide an overview of interfaces and information elements associated to the Or-Vnfm reference point.

The Or-Vnfm reference point is used for exchanges between NFV Orchestrator and VNF Manager, and supports the following interfaces:

- VNF Package Management
- VNF Lifecycle Operation Granting
- Virtualised Resources Management
- VNF Lifecycle Management
- VNF Lifecycle Change Notification
- VNF Performance Management
- VNF Fault Management
- VNF Configuration Management
- VNF Indicator

② Ve-Vnfm Interface

- Block API for the EM interface
- Block API for the VNF interface

③ TOSCA Modelling

TOSCA is a technical specification of the topology and control procedures and a wide range of applications in a cloud environment intended to increase the portability of the application layer for a service in a language based on XML.

TOSCA defines a meta-model to define the IT services and to the meta-model also defines how to manage the service, as well as the structure of the service. Topology defines the structure of a service, the Topology Template is composed of a plurality of Node Templates and Relationship.

Relationship Templates defines the topology of the service represented by graph models Lode having directionality. Node in the graph is expressed as a Node Template, Node Template is represented in form of a Node Type components that make up the service. Node Type defines the interface for handling operations through the properties of this component and that component[3].

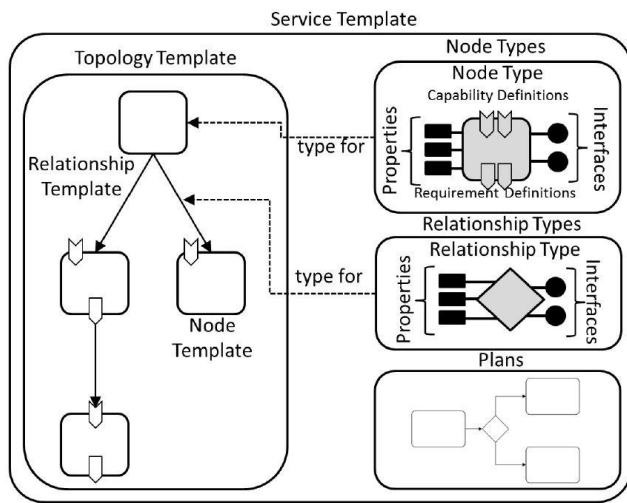


Figure 4. TOSCA Concept

TOSCA.node.etri.nfv

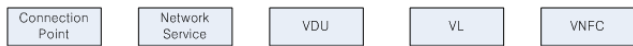
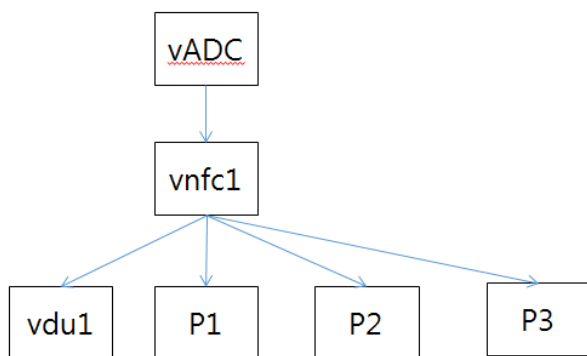


Figure 5. TOSCA Concept

TOSCA.node.etri.nfv.resource



Figure 6. TOSCA Concept



We define the main elements of Connection Point, Network Service, VDU, VL, the VNFC Node Template for virtualizing the network. It also refers to the components necessary for each instance of Node Template is actually composed of

virtualization called Resource Descriptor defined as Compute, Network, Port, Router, Security Group, Subnet, as shown in Figure .5 and expressed as a Node Template

vADC shows an example of an information model VNF with TOSCA. vADC is composed of one vnfc the vdu vnfc has one and three connection points (Connection Point).

VNF called vADC shows an example of an information model VNF with TOSCA. vADC is composed of one vnfc the vdu vnfc has one and three connection points (Connection Point).

IV. CONCLUSION

In this paper, we propose the structure of VNFM (Virtual Network Manager) to manage virtual network feature to configure the network services configured in a number of VNFs, using TOSCA (Topology and Orchestration Specification for Cloud Application) to deploy effectively in a variety of virtual infrastructures based reflecting the model information described in a standards-based environment MANO proposed by ETSI, and proposes a mechanism to handle them effectively. It also presents the characteristics of the information models that are currently carried out in the generic VNFM.

ACKNOWLEDGMENT

This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIP) (B0101-15-233, Smart Networking Core Technology Development)

REFERENCES

- [1] ETSI GS NFV-MAN 001, NFV Management & Orchestration
- [2] ETSI GS NFV 002, NFV Architectural Framework, 2014-12
- [3] https://www.oasisopen.org/committees/tc_home.php?wg_abbrev=tosca, TOSCA



Byung Yun Lee is currently a Principal Member of Telecommunication Internet Research Division at Electronics and Telecommunication Research Institute (ETRI), Korea. He received the PhD degree in computer engineering from Chungnam National University, Korea, in 2003. Since joining ETRI in 1992, his work has focused on SDN/NFV technology, and network management.



Bhum Cheol Lee received M.S. and Ph.D. degree in Electric Engineering from Yonsei University, Korea in 1983 and 1997, respectively. He is currently Manager of Networking Computing Convergence Lab. in Electronics and Telecommunications Research Institute (ETRI), Korea. His research interests are Smart Network, Parallel Flow Processing and Network Virtualization