

Network Working Group
Internet-Draft
Intended status: Informational
Expires: April 2018

Hyunsik Yang
Younghan Kim
Soongsil University
October 30, 2017

I2nsf on the NFV Reference Architecture
[draft-yang-i2nsf-nfv-architecture-00.txt](#)

Abstract

This document describes the adoption of i2nsf Framework onto the Network Functions Virtualization (NFV) Reference Model. In this document, we explain the i2nsf Framework adopted to NFV reference architecture with each corresponding component.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#). This document may not be modified, and derivative works of it may not be created, and it may not be published except as an Internet-Draft.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time.

It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at
<http://www.ietf.org/ietf/1id-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>

This Internet-Draft will expire on April 30 2018.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the [Trust Legal Provisions](#) and are provided without warranty as described in the Simplified BSD License.

Table of Contents

- [1. Introduction](#) [4](#)
- [1.1. Terminology](#) [4](#)
- [2. I2NSF framework onto the NFV Reference Model](#) [4](#)
- [2.1. NSF](#) [6](#)
- [2.2. Security Controller](#)..... [7](#)
- [2.3. Developer's Mgmt System](#)..... [7](#)
- [2.4. Interfaces](#) [8](#)
- [2.4.1. Consumer-Facing Interface](#) [8](#)
- [2.4.2. NSF-Facing Interface](#)..... [8](#)
- [2.4.3. Registration Interface](#)..... [8](#)
- [3. Use case - SFC Enabled I2NSF framework](#) [9](#)
- [3.1. SFC Policy Manager](#)..... [9](#)
- [3.2. SFC Catalog Manager](#)..... [10](#)
- [3.3. Developer's Mgmt System](#)..... [10](#)
- [4. Security Considerations](#)..... [11](#)
- [5. IANA Considerations](#) [11](#)
- [6. References](#) [11](#)
- [6.1. Normative References](#)..... [11](#)
- [6.2. Informative References](#)..... [12](#)

1. Introduction

The goal of I2NSF is to define a set of software interfaces and components for controlling and monitoring aspects of physical and virtual NSFs, enabling clients to specify rules set. To enable I2NSF environment, I2NSF framework not only considers physical infrastructure but also considers the NFV environment since NSF may be provided by virtualized infrastructure as a vnfs. Especially, I2NSF applicability document [[i2nsf-applicability](#)] describes the applicability of interface to Network Security Functions to network-based security services in NFV environment. Although it explains how i2nsf provides security service in NFV environment, it doesn't consider how i2nsf framework adopted onto the NFV reference architecture.

Therefore, we explain the i2nsf framework adopted to NFV reference architecture with each corresponding component.

1.1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC 2119](#) [[RFC2119](#)].

This document uses the terminology described in [[i2nsf-framework](#)], [[i2nsf-terminology](#)], [[i2nsf-applicability](#)], [[etsi-gs-nfv-003](#)] and [[nsf-triggered-steering](#)].

2. I2NSF framework onto the NFV Reference Model

The European Telecommunications Standards Institute (ETSI) defined the components for the basic NFV architecture including the NFV Infrastructure (NFVI), VNF Manager (VNFM), Virtualization Infrastructure Manager (VIM), and NFV Orchestrator (NFVO). [[etsi-gs-nfv-003](#)] NFVI provides the virtual resources, such as VM and virtual network, used to create, update, and delete VNFs running applications. VNFs are implemented through software virtualization techniques running over the NFVI.

Virtualized Infrastructure Manager (VIM) has a function for controlling and managing the NFVI compute, storage and network

resources, within one operator's infrastructure sub-domain. It also collects and forwards performance measurements and events.

VNFM manages the VNF lifecycle. When a VNF is created, the VNFM manages the VNF instance in the lifecycle, and the VNFM performs several actions such as software update/modification, monitoring data collection-a fault event in the VNF, and instance termination. According to definition of ETSI, the VNFM is divided into Generic VNFM and Specific VNFM. When the VNFs have their specific methods for provisioning and lifecycle management, a specific VNFM required.

In the I2NSF framework [[i2nsf-framework](#)], they defined several components such as NSF, Security controller and Developer's Mgmt System. To adopt these components to the NFV reference architecture, each component should be classified based on functionality. According to component functionality, it would correspond to NFV reference architecture components as Figure 1.

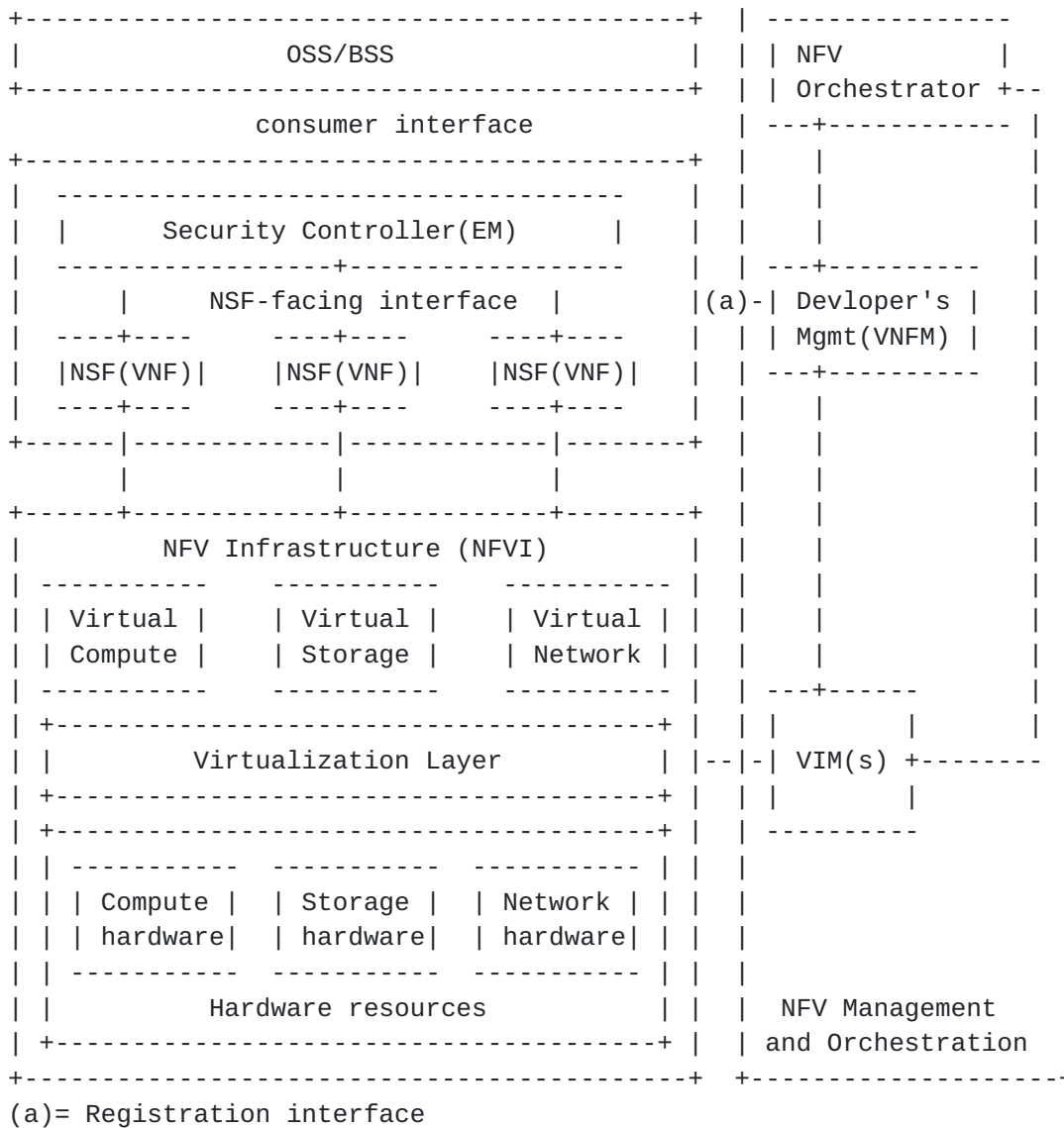


Figure 1. I2NSF architecture on NFV reference architecture

2.1. NSF

Network Security Function is one of the security service functions.

In the ETSI reference architecture, VNF(Virtual Network Function)is the network functions which provide specific service.

Therefore, NSF corresponds to the VNF in NFV reference architecture.

2.2. Security Controller

According to I2NSF framework, the security controller has a role which translate policy according to user's request and delivers low level policy to NSFs(manages NSF). It also collects NSF capability from developer's Mgmt System. Based on this information, the security controller forwards policy to NSF.

In the NFV reference architecture, EM has a role that it may be aware of virtualization and collaborate with the VNF Manager to perform those functions that require exchanges of information regarding the NFVI Resources associated with the VNF. EM performs typical management functionality for one or several VNFs.

Therefore, the Security controller corresponds to Element management since it should provide the function which controls NSF and policy. In the case of a distributed security controller model, an interface which is used to communicate between controllers should also be considered.

2.3. Developer's Mgmt System

According to the definition of I2NSF Registration Interface, Developer's Mgmt system registers NSF which can be provided by specific vender. Developer's Mgmt system also can be one of the vendors too.

In the NFV reference architecture, VNFM manages the VNF lifecycle. It also performs several actions such as software update, monitoring and fault management.

Generally, generic VNFM means that only one VNFM handle all of the VNF in the NFV environment. However, if additional VNFMs are required for management of specific VNFs, additional VNFMs can be defined as specific VNFMs.

Therefore, if Developer's Mgmt System manages the NSF lifecycle, it can logically correspond to a specific VNFM.

2.4. Interfaces

2.4.1. Consumer-Facing Interface

The Consumer-Facing Interface is an interface for communication between the User and the Security Controller. It is used to enable different users of a given I2NSF system to define, manage, and monitor security policies for specific flows within an administrative domain.

In the NFV reference architecture, OSS is Operational Support Systems and BSS stands for Business Support Systems. OSS/BSS support the system which relates to infra management such as billing, order and metering.

Although an interface is not defined between User and EM in the NFV reference architecture, Consumer-Facing interface can be deployed between user and EM.

2.4.2. NSF-Facing Interface

The NSF-Facing Interface is an interface for communication between Security Controller and NSF. It is used to specify and monitor flow-based security policies enforced by one or more NSFs.

In the NFV reference architecture, Software Architecture (SWA)-4 Interface is defined. The interface SWA-4 is used by the EM to communicate with a VNF. This management interface is used for the runtime management of the VNF according to the Fulfillment, Assurance, and Billing and FCAPS (Fault, Configuration, Accounting, Performance, Security) network management models and frameworks.

Therefore, NSF-Facing Interface corresponds to the SWA-4 interface.

2.4.3. Registration Interface

Registration Interface is used to register NSF from Developer's Mgmt System to the security controller. An NSF's capabilities can either be pre-configured or retrieved dynamically through the I Registration Interface.

Above, this document mentioned that, the Developer's Mgmt System handles the NSF life cycle and this interface corresponds to Ve-Vnfm which is defined in the NFV reference architecture. Ve-Vnfm is defined as IFA008 in ETSI document.

IFA008 composed of two interfaces. One is Ve-Vnfm-em, another is Ve-Vnfm-VNF. If security controller is deployed as an EM, then the registration interface corresponds to Ve-Vnfm-em.

3. Use case - SFC Enabled I2NSF framework

In the I2NSF WG, some documents mentioned use cases for cloud based security with forwarding mechanism. Especially SFC enabled I2NSF document [[nsf-triggered-steering](#)] showed the use case which used SFC as a forwarding mechanism. In addition, it defined additional components and extended functionality of components. Therefore, in the following section, we explain the details of each component and consider how it corresponds to the NFV reference architecture.

3.1. SFC Policy Manager

SFC policy manager is a part of the security controller. It is responsible for interpreting a high level policy into a low-level SFC policy, which is given by I2NSF client. It also handles delivery of the interpreted policy to classifiers for security function chaining. Moreover, it also generates an SF forwarding table and distributes the forwarding information to SFF(s).

In the NFV reference architecture, MANO performs similar functions as the SFC policy manager.

More specifically the NFV orchestrator(NFVO) performs on-boarding of new Network Service (NS), VNF-FG(forwarding graph) and VNF Packages. In addition, it manages NS lifecycle (including instantiation, scale-out/in, performance measurements, event correlation and termination).

Therefore, SFC policy manager corresponds to NFVO. In addition, if SFC policy manager is a part of Security controller, this function should be separated from security controller.

3.2. SFC Catalog Manager

SFC catalog manger is a part of the security controller. It is responsible for maintaining the information of every available SF instance such as IP address, supported transport protocol, service name, and load status. Moreover, it should respond to the queries for available SF instances from SFC Policy Manager so as to help to generate a forwarding table entry relevant to a given SFP. It also request Developer's Management System to dynamically instantiate supplementary SF instances to avoid service congestion or the elimination of an existing SF instance to avoid resource waste.

In the NFV reference architecture, SFC catalog manager corresponds to Element management since information which is related to VNF capability is managed by EM. Moreover, this function is similar to security controller as we explained earlier.

3.3. Developer's Mgmt System

In the SFC enabled document, the function of Developer's Mgmt system is extended. Following the request message from SFC catalog manager, it creates additional SF instances and eliminates some of the SF instances.

As mentioned above, if Developer's Mgmt system manages the NSF's lifecycle, it corresponds to a specific VNF Manager. VNF life cycle management includes instantiating, creating, provisioning, scaling, monitoring, and termination of VMs in a VNF instance. Therefore, the Developer's Mgmt system corresponds to a specific VNF Manager.

However, for scaling performed at a network service level, the role of Developer's Mgmt system should extend to the MANOManage and orchestrator).

SFC catalog manger is a part of the security controller. It is responsible for maintaining the information of every available SF instance such as IP address, supported transport protocol, service name, and load status. Moreover, it should respond to the queries for available SF instances from SFC Policy Manager so as to help to generate a forwarding table entry relevant to a given SFP. It also request Developer's Management System to dynamically instantiate supplementary SF instances to avoid service congestion or the elimination of an existing SF instance to avoid resource waste.

In the NFV reference architecture, SFC catalog manager corresponds to Element management since information which is related to VNF capability is managed by EM. Moreover, this function is similar to security controller as we explained earlier.

4. Security Considerations

N/A

5. IANA Considerations

This document has no IANA actions.

6. References

6.1. Normative References

[RFC2119]

Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.

[i2nsf-framework]

Lopez, D., Lopez, E., Dunbar, L., Strassner, J., and R. Kumar, "Framework for Interface to Network Security Functions", [draft-ietf-i2nsf-framework-07](#) (work in progress), August 2017.

[i2nsf-terminology]

Hares, S., Strassner, J., Lopez, D., Xia, L., and H. Birkholz, "Interface to Network Security Functions (i2nsf) Terminology", [draft-ietf-i2nsf-terminology-04](#) (work in

progress), July 2017.

Yang, et al.

Expires April 30, 2018

[Page 11]

[etsi-gs-nfv-003]

ETSI NFV ISG, "Network Functions Virtualisation (NFV); Terminology for Main Concepts in NFV", ETSI GS NFV 002 V1.1.1 NFV 002, October 2013, <http://www.etsi.org/deliver/etsi_gs/nfv/001_099/002/01.01_01_60/gs_nfv002v010101p.pdf>

6.2. Informative References

[i2nsf-applicability]

J. Jeong., S. Hyun., T. Ahn., S. Hares., D. Lopez., "Applicability of Interfaces to Network Security Functions to Network-Based Security Services", [draft-ietf-i2nsf-applicability-00](#)(work in progress), October, 2017.

[nsf-triggered-steering]

Hyun, S., Jeong, J., Park, J., and S.Hares, "Service Function Chaining-Enabled i2nsf Architecture", [draft-hyun-i2nsf-nsf-triggered-steering-03](#)(work in progress), July 2017.

Authors' Addresses

Hyunsik Yang

Soongsil University
369, Sangdo-ro, Dongjak-gu,
Seoul 156-743, Korea
Email: yangun@dcn.ssu.ac.kr

Younghan Kim

Soongsil University
369, Sangdo-ro, Dongjak-gu,
Seoul 156-743, Korea
Email: younghak@ssu.ac.kr