Registration Interface Information Model and YANG Data Model
(draft-hyun-i2nsf-registration-interface-im-04, and
draft-hyun-i2nsf-registration-interface-dm-03)

IETF 101, London
March 21, 2018

Sangwon Hyun, Jaehoon (Paul) Jeong [Presenter],
Taekyun Roh, Sarang Wi and Jungsoo Park
Updates from the Previous Version

• In draft-hyun-i2nsf-registration-interface-im-04
  – We revised Section 4 to discuss about destructing an NSF instance no longer required via the registration interface.
  – We changed the term of NSF profile into NSF capability information.

• In draft-hyun-i2nsf-registration-interface-dm-03
  – We updated the YANG data model accordingly in order to align with the updates in draft-i2nsf-registration-interface-im-04.
Introduction

- Information Model (IM) & YANG Data Model (DM) for the Registration Interface are required for the following functions:
  - To Register
    • To register the capabilities of NSF created by Developer’s Management System (DMS)
  - To Query
    • To send a general request of NSF capabilities to DMS
  - To Interact with NFV MANO
    • To request DMS to instantiate/deinstantiate an NSF

- Secure the registration of distributed NSFs via Registration Interface in a centralized manner.
NSF Registration (1/2)

- Developer’s Management System (DMS) registers the NSF to Security Controller via Registration Interface

* The capabilities based on draft-hares-i2nsf-capability-data-model-06
NSF Registration (2/2)

1. Register NSF capabilities*
2. Store the mapping of NSF and its capabilities

*The capabilities based on draft-hares-i2nsf-capability-data-model-06
Additional Usage of Registration Interface

• Motivation
  – Query
    • Security Controller sends a query to DMS to search NSFs with required capabilities via Registration Interface.
  – Interaction
    • Security Controller requests the instantiation/deinstantiation of NSFs to DMS (or NFV MANO).

Note: The existing information model (draft-ietf-i2nsf-capability-00) & YANG data model (draft-hares-i2nsf-capability-data-model-06) are used to describe the security capability of an NSF.
Step 1

1. Request a high-level policy

I2NSF User → Security Controller → Registration Interface → Developer’s Mgmt System → NSF-n → NSF-1 → NSF-m → NSF-1
Capability-based NSF Search (2/5)

Step 2

1 Identify required capabilities

*The capabilities based on draft-hares-i2nsf-capability-data-model-06
Capability-based NSF Search (3/5)

Step 3

Send queries of required capabilities* to DMS(s)

General request of NSF capabilities

Security Controller

Registration Interface

Developer’s Mgmt System

I2NSF User

*The capabilities based on draft-hares-i2nsf-capability-data-model-06
Step 4

If DMS find an NSF with required capabilities

④ Register NSF capabilities

Security Controller

NSF capabilities

Registration Interface

Developer’s Mgmt System

I2NSF User

NSF-n

NSF-1

NSF-m

NSF-1
Step 5

If DMS find an NSF with required capabilities

NSF-n

NSF-1

NSF-m

NSF-1

Security Controller

I2NSF User

Store the mapping of NSF and its capabilities

NSF capabilities

Registration Interface

Developer’s Mgmt System
Instantiation Request of NSF (1/2)

1. Request the creation of a new NSF

Security Controller

I2NSF User

Registration Interface

NSF capabilities

Developer’s Mgmt System

NSF-1

NSF-n

NSF-m

NSF-1
Instantiation Request of NSF (2/2)

Security Controller

Registration Interface

I2NSF User

Developer’s Mgmt System

② Create a new NSF

NSF-n

NSF-1

NSF-m

NSF-1

NSF capabilities
Next Steps

• We will extend our I2NSF Hackathon implementation to demonstrate the feasibility of Registration Interface.

  – To construct I2NSF Framework in OpenStack environment along with OPNFV and Open Source MANO (OSM).

  – To implement the interaction with NFV MANO to instantiate and deinstantiate NFVs through Registration Interface in OpenStack environment.