Realizing Network Function Virtualization Management and Orchestration with Model Based Open Architecture

YuLing Chen, Yinghua Qin, Mark Lamb, Wenjing Chu
Nov., 2015
NFV creates new requirements for Network Management and Orchestration

**Management and Orchestration**

- Isolation between physical and virtualized network functions introduced new requirements.
  - New hierarchies
    - Virtualized infrastructure (NFVI)
    - Virtualized Network Function (VNF)
    - Network Service
  - New entities and relationships
    - VNF Forwarding Graphs
    - Virtual Links
    - Links between VNF and PNF
    - Relationship across different layers
  - New NFV specific technologies

**Monitoring**

- To monitor new hierarchies, entities, relationships, and in the context of new technologies.
  - Fault and Performance Management
    - Cross Layer/System event correlation and root cause analysis
    - Real time monitoring to support carrier-grade high availability
  - Policy Management
    - VNF or NS Scaling policy
      - To support Auto-Scaling
    - Access Control
    - Resource Management
    - Fault Management
      - To support Self-Healing
ETSI NFV-MANO Architectural Framework

- **NFV Orchestrator (NFVO)**
  - Network Service lifecycle management
  - security validation and authorization
  - global resource management
  - policy management.
- **VNF Manager (VNFM)**
  - VNF Lifecycle management
  - Adaptation, configuration, and coordination for event reporting among NFVI and EMS.
- **Virtualized Infrastructure Manager (VIM)**
  - NFVI resource management
  - Performance and event collection and forwarding

---

![Diagram](image)

*Fig. 1.* ETSI NFV-MANO architectural framework with reference points [3]
Dell Research NFV MANO POC using Foglight

• **ETSI functional blocks Refinement**
  – Further develop the functional blocks into software components to realize the MANO functionality.

• **VNFM Integration Interface specification**
  – Specification of VNFM Integration Interface to integrate with VIMs, VNFs, and EMSs.

• **Experimentation using Foglight for NFV monitoring**
  – Using Foglight for NFV monitoring and integration with orchestration tools for VNF auto scaling.
Dell Research NFV-MANO Architecture with key components and interfaces

- Further development of ETSI NFV-MANO architecture.
  - Identify key components in ETSI NFV-MANO functional blocks.
  - To bridge the specification and the pragmatic NFV-MANO system implementation.

- Three major open integration interfaces for integration between functional blocks
  - NFVO Northbound Interface to integrate with OSS/BSS
  - VNF Manager Interface for integration between NFVO and VNFM
  - VNF Manager Integration interface to integrate between VNFM and VIMs/VNFs/EMSs
VNF Life Cycle Management Task Break down

- Broken down into three layers
- Orange squares are top level VNF LCM tasks
- Blue ovals are platform independent tasks
- Green squares are platform dependent tasks

VNFM Integration Interface is derived from blue ovals.

11/20/15
VNF Manager Integration Interface

This set of interface captures the operations that VNF Manager needs to complete for VNF life cycle management.

The interface follows REST API design to expose system resources and apply CRUD operations on them.

Notify API opens the notification channel between VNFM and the external systems to allow VNFM to update VNF state based on the notification sent from the cloud adaptors.

<table>
<thead>
<tr>
<th>Resource</th>
<th>Operations</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VNFC Image</td>
<td>C,D</td>
<td>Create or delete the image associated with the VNFC in or from the VIM.</td>
</tr>
<tr>
<td>VNFC</td>
<td>C,R,D</td>
<td>Create, delete, or query the virtual instance associated with VNFC.</td>
</tr>
<tr>
<td>VNFC Link</td>
<td>C,D</td>
<td>Create, delete, query, or test the network connectivity between the two VNFCs.</td>
</tr>
<tr>
<td>VNFC Public Access</td>
<td>C,R,D,T</td>
<td>Create, delete, query, or test the external public access of the VNFC instance.</td>
</tr>
<tr>
<td>VNFC Admin</td>
<td></td>
<td>With different parameters, start, stop, suspend, resume, or reboot the VNFC instance.</td>
</tr>
<tr>
<td>VNFC Saling</td>
<td></td>
<td>With different parameters, scales up, down, in, and out the VNFCs.</td>
</tr>
<tr>
<td>VNFC Healing</td>
<td></td>
<td>With different parameters, applies different approaches for VNFC Healing. For example, with failover parameter, it fails-over VNFC instance to the hot standby instance.</td>
</tr>
<tr>
<td>VNFC Config</td>
<td></td>
<td>With different parameters, connects to either EM or VNF control instance for VNF configuration.</td>
</tr>
<tr>
<td>Notify</td>
<td></td>
<td>Notify the subscriber with an object that contains the low level messages related to VNF lifecycle state change.</td>
</tr>
</tbody>
</table>
• Using Foglight to collect the data from NFV architecture.
• Using Foglight Model based solution to unite the data models across different layers in NFV architecture for correlation.
• Using Foglight Rule Engine for VNF Scaling policy definition.
• Trigger auto scaling by integrating with NFV orchestration tools.
The example VNF is a virtualized network traffic generation tool called TeraVM.

At deployment time, NFV-MANO POC auto-deploys the VNF on OpenStack through VNFM Integration API.

The VNF is also configured and started by the NFV-MANO POC through VNFM Integration API.

An VNF auto-scaling rule is defined using Foglight Rule Engine.
VNF Auto Scaling using Dell Foglight and VNFM Integration Interface

- At runtime, Foglight keeps monitoring the performance of the VNF.
- When the VNF is under heavy load and exceeds the defined performance threshold, Foglight receives an SNMP trap.
- An action is triggered to auto-scale the VNF from the defined rule in Foglight rule engine.
- The auto-scaling action calls VNFM Integration API to bring up a new TeraVM Test Module instance to join the traffic generation.
Demo
Thank you!