

# 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

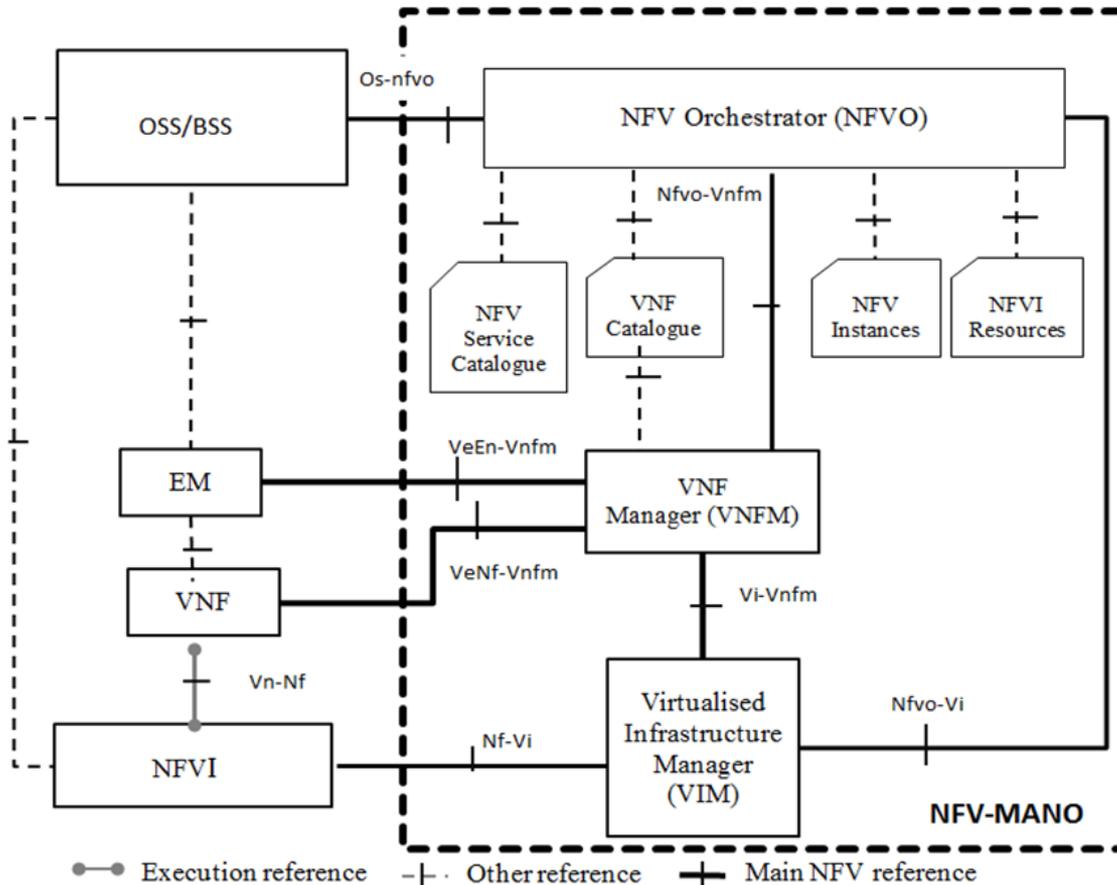


Fig. 1. ETSI NFV-MANO architectural framework with reference points [3]

- **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.

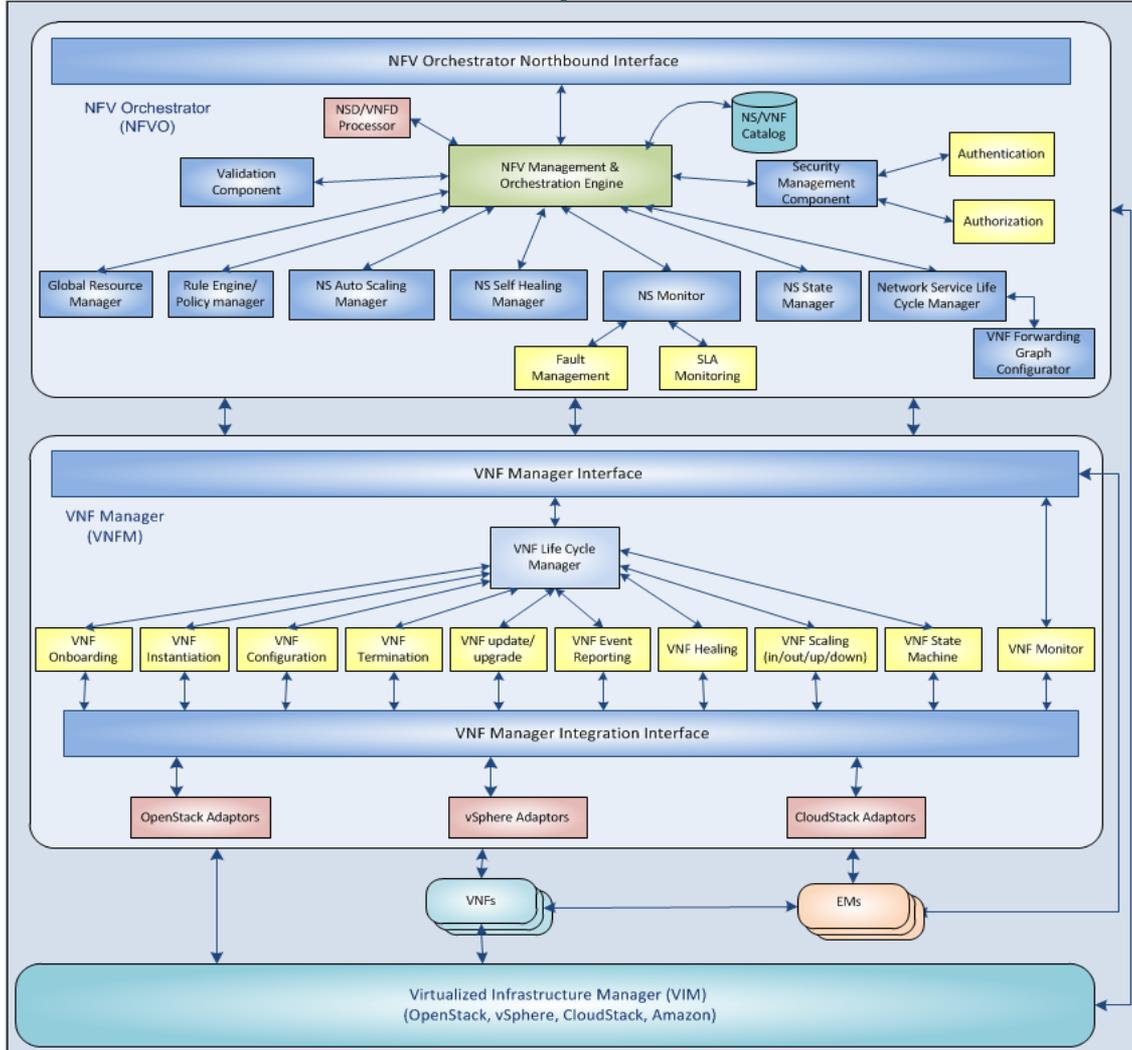
- **Virtualized Infrastructure Manager (VIM)**

- NFVI resource management
- Performance and event collection and forwarding

# 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

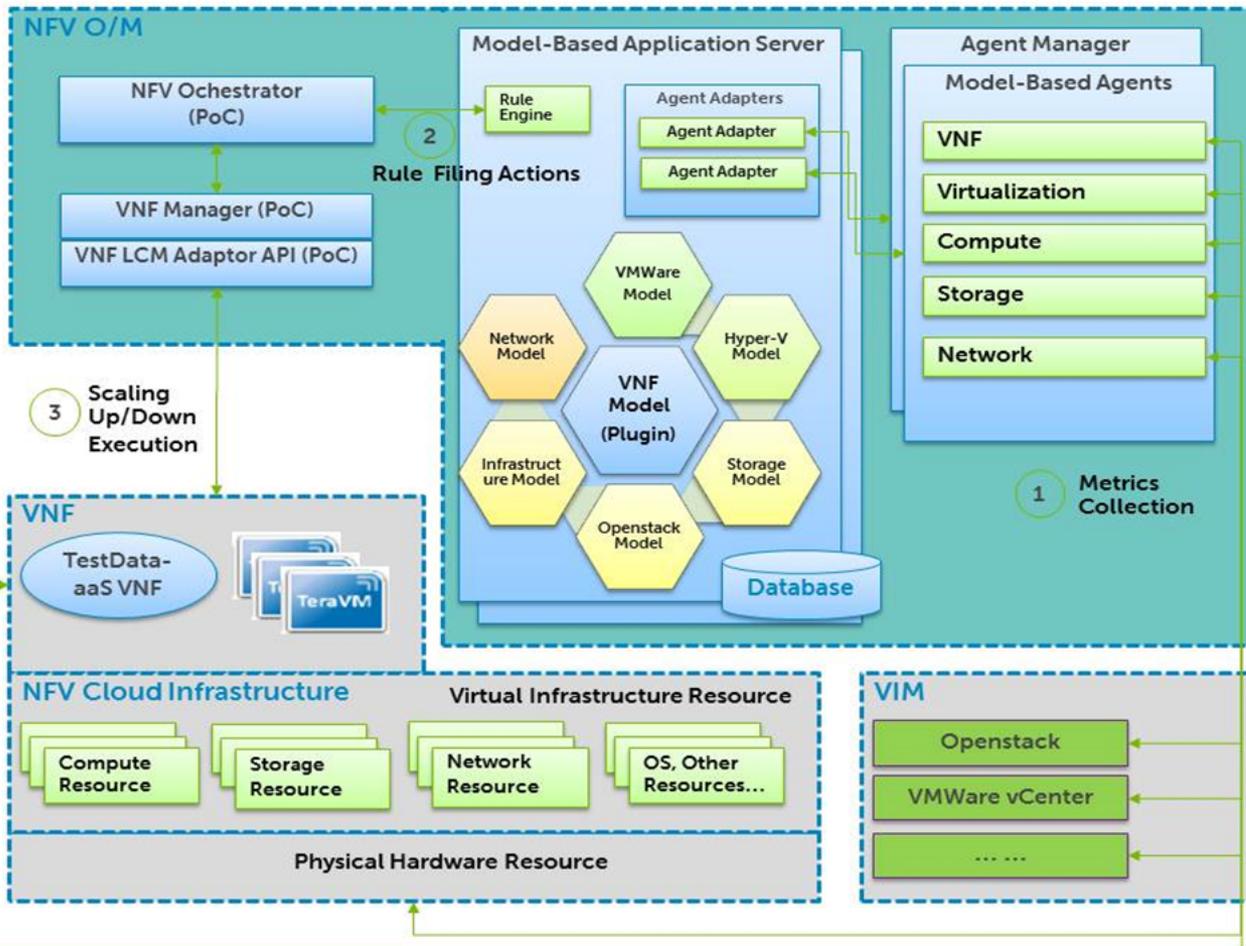


# VNF Manager Integration Interface

Resource	Operations	Description
<i>VNFC Image</i>	C,D	Create or delete the image associated with the VNFC in or from the VIM.
<i>VNFC</i>	C,R,D	Create, delete, or query the virtual instance associated with VNFC.
<i>VNFC Link</i>	C,D	Create, delete, query, or test the network connectivity between the two VNFCs.
<i>VNFCPublicAccess</i>	C,R,D,T	Create, delete, query, or test the external public access of the VNFC instance.
<i>VNFCAdmin</i>		With different parameters, start, stop, suspend, resume, or reboot the VNFC instance.
<i>VNFCScaling</i>		With different parameters, scales up, down, in, and out the VNFCs.
<i>VNFCHealing</i>		With different parameters, applies different approaches for VNFCHealing. For example, with fail-over parameter, it fails-over VNFC instance to the hot standby instance.
<i>VNFCConfig</i>		With different parameters, connects to either EM or VNF control instance for VNF configuration.
<i>Notify</i>		Notify the subscriber with an object that contains the low level messages related to VNF lifecycle state change.

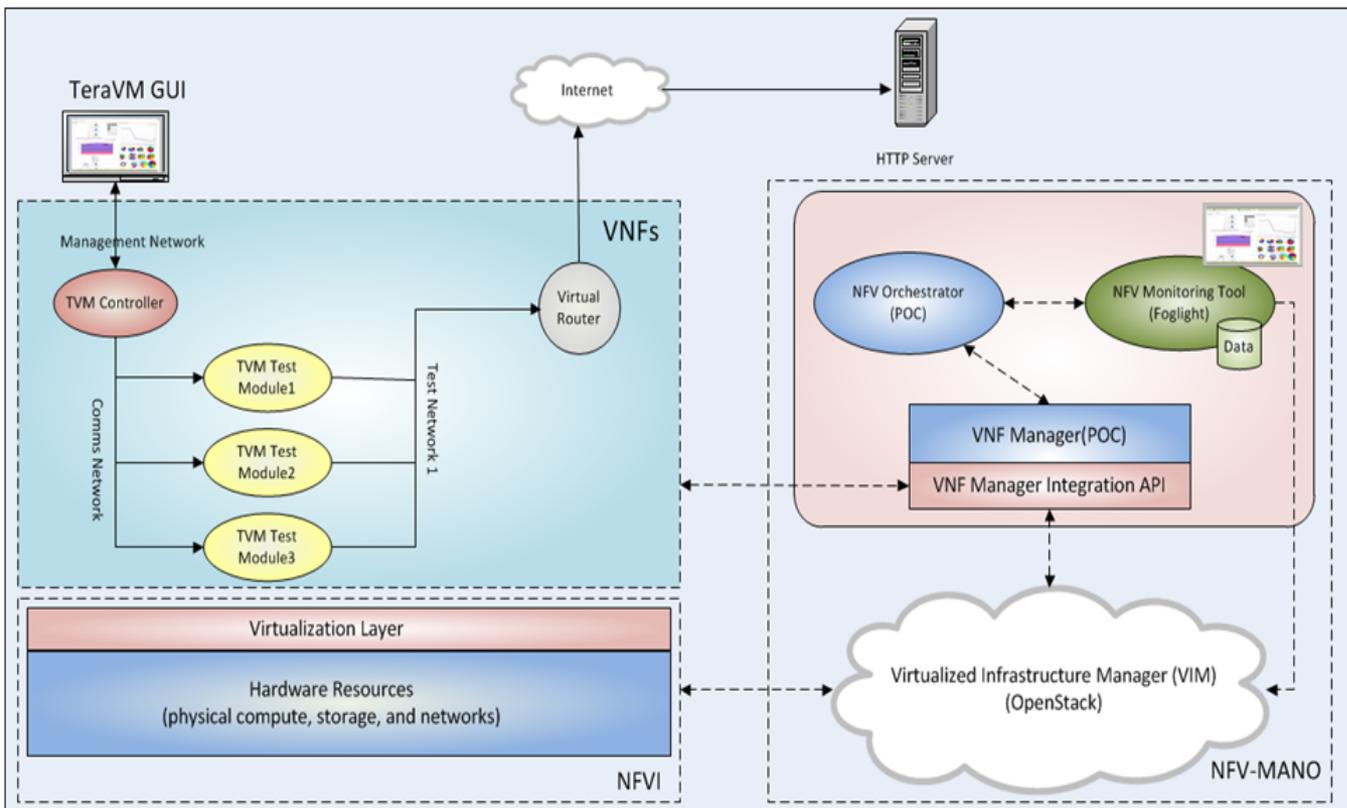
- This set of interface captures the operations that VNF Manager needs to complete for VNF life cycle management.
- The interface follows REST API design to expose system resources and apply CRUD operations on them.
- Notify API opens the notification channel between VNFM and the external systems to allow VNFM to update VNF state based on the notification sent from the cloud adaptors.

# Using Foglight for NFV Monitoring and Auto-Scaling



- 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.

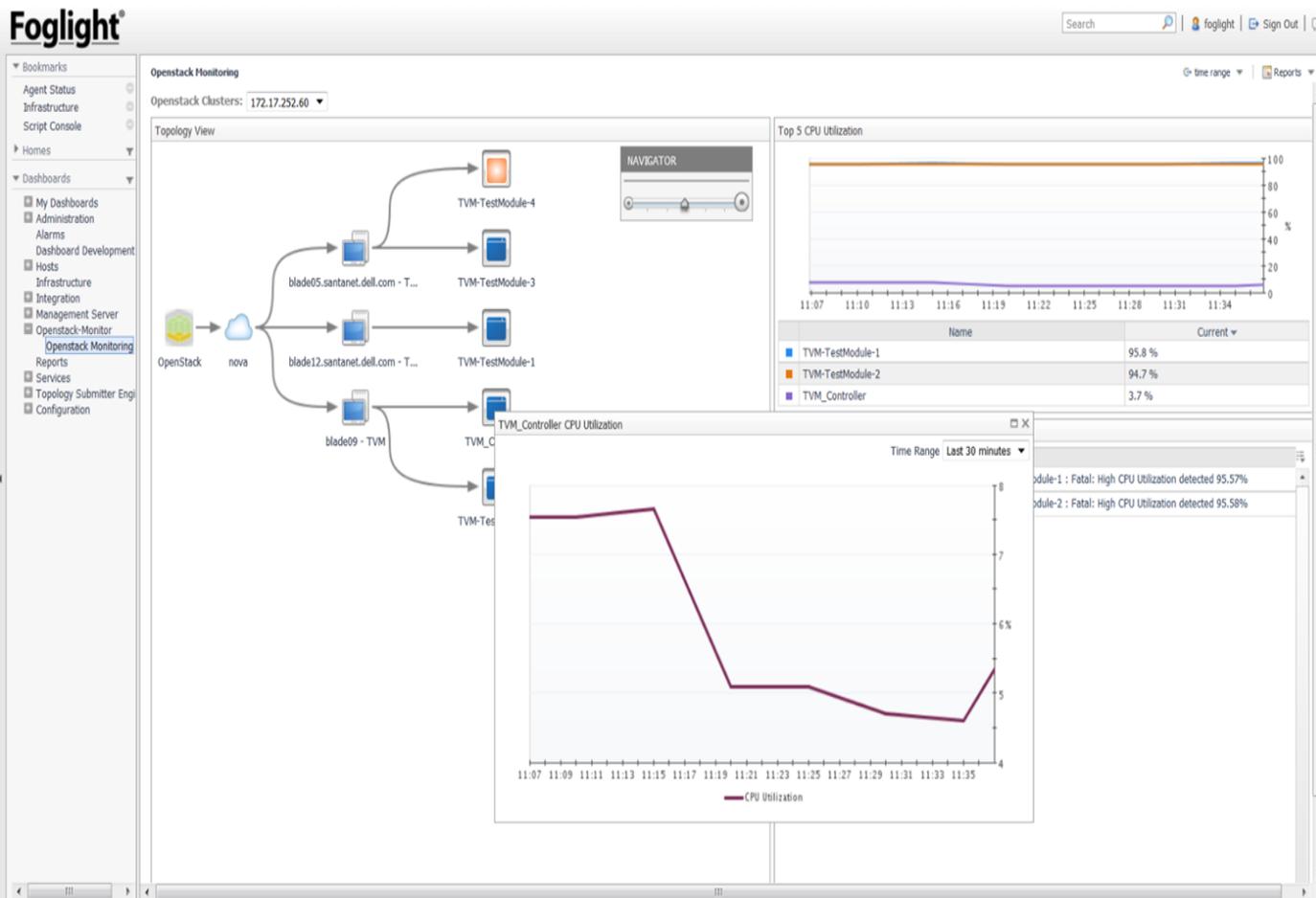
# POC to automatically deploy and scale Tera VM in OpenStack



- 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!*

