Management and Orchestration in Complex and Dynamic Environment
Management and Orchestration Challenges
in Complex and Dynamic Environment

- Long-Lived Transaction
- Scalability and Performance
Network-Wide Transaction in Complex and Dynamic Environment
Network-Wide Transaction
NFV Management and Orchestration

Service Catalog

OSS

Services Orchestrator

NFV Orchestrator

VNF Manager

third-party VNFM

Virtual Infrastructure Managers

OpenStack

VMware

SDN Controller

VNF Library

Service, VNF, and Infrastructure Description

Network Function Virtualization (NFV) management and orchestration involves the integration of services, virtual networks, and infrastructure elements. The diagram illustrates the flow of information and interactions between different components:

- **Services Orchestrator** receives requests and manages service creation.
- **OSS (Operations Support System)** interacts with OSS-B (Business Support System) for service management.
- **NFV Orchestrator** controls the lifecycle of VNFs (Virtual Network Functions).
- **VNF Manager** handles the deployment, management, and removal of VNFs.
- **third-party VNFM** represents third-party VNF management systems.
- **Virtual Infrastructure Managers** (OpenStack, VMware) provide the virtual infrastructure for VNFs.
- **SDN Controller** manages the network's software-defined features.
- **VNF Library** contains a variety of VNFs, including vCPE (Virtual Customer Premises Equipment), vFW (Virtual Firewall), vR (Virtual Router), and third-party VNFs.

These components work together to ensure seamless service delivery and efficient resource utilization in a virtualized network environment.
We don‘t know how much time is needed to accomplish a task!
Multi-Stage Transaction
Why Multi-Stage Transaction?

- VIM resource allocation.
- External data resource allocation.
- Devices are not reachable until parts of the service have been set up.
- Respond to changes in the environment and the service parameters.

Leave the transaction as soon as possible!
Why Multi-Stage Transaction?

(S1) Service Instance Creation

(S2) Device Configuration Change

Service Order Request

1

Service Model

Device Model

2

CDB

3

Cloud Infrastructure

4

Cloud Manager

5

Resource Allocation

Service Instance Redeploy

6

6

6
NFVO Architecture

Simple Service Management Interface for Dynamic End-to-End Service Management of Complex Services
NFVO and VNFM Implementation

Service Order Request

Service Model

Device Model

CDB

1

2

3

4

5

6

6

6

Network Infrastructure

Subscriber App

Request for new VNF

vm-manager

VNFM NED

vnfm-specific service

Reactive FASTMAP

Vm-manager

L start

C devices

L device

vnfm_datamodel
Spinning up VMs first and configuring them when available.

When some devices are not reachable until parts of the service have been set up.

Service automatically responds to changes in the environment (e.g. VM mobility, network connectivity).

Service automatically responds to changes in service parameters (e.g. service catalog, policy).

Service takes a long time to activate and you do not want to lock the database.
Scalability and Performance
Addressing Performance Limitations

Vertical Scaling

- **Increase capacity:** reached the hard limits
- **Optimize code:** ran out of options
- **Change NFVO behavior:** nothing more to be done

There is only so much you can do!
Design for Scalability and Performance

**Per-App NFVO**
- Service Model
- Device Model
- 20k devices

**NFVO Cluster**
- Service Model
- Device Model
- Device Node 20k devices
- Static Device Mapping
- Device Node 20k devices

**NFVO LSA**
- Top Node
- RFS Model
- Device Node 20k devices
- Dispatching

**VPN Services**
- 20k devices

**NFV**
- Service Model
- Device Model
- 20k devices

**Service Model**
- Device Model
- 20k devices

**Device Model**
- 20k devices
Horizontal Scaling

NFVO Cluster

- Device mappings need to be maintained.
- Performance can be severely impacted by cluster functionality:
  - Every call to device data on the service node results in a NETCONF RPC.
  - Cluster caching should be enabled to improve performance but it will increase memory utilization.
Horizontal Scaling
NFVO LSA (Layered Service Architecture)

- The top node only sees a small number of devices (RFS nodes).
- Total number of devices has no impact on performance of the top node.
Use LSA for virtually limitless scalability.

Make your top-level service model agnostic to device, platform, interface, and technology.

Devise the simplest dispatch method possible or at least one that is easy to maintain.

Implement integration with external systems at the top layer.

Implement resource allocation at the appropriate layer.

Use LSA-ready design today even if you run just one node.
Q&A