



NSO NFVO

Raj Kamal
Software Consulting Engineer - CX
6-June-2020

*“The most powerful tool we have
as developers is automation.”*

Scott Hanselman

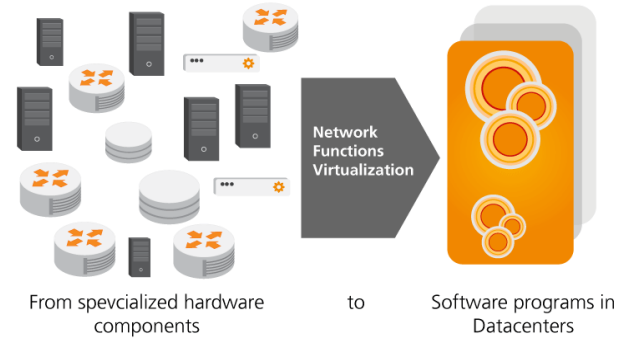
Agenda

- NFV ETSI MANO Architecture
- Cisco NSO NFVO Solution
- Use Cases

History

- The concept of NFV originated from service providers who wanted to make adding new network functions or applications easier and faster.
- A notable standards organization is the European Telecommunications Standards Institute (ETSI), was the first major organization to release an NFV standard in October 2013.

What is NFV?

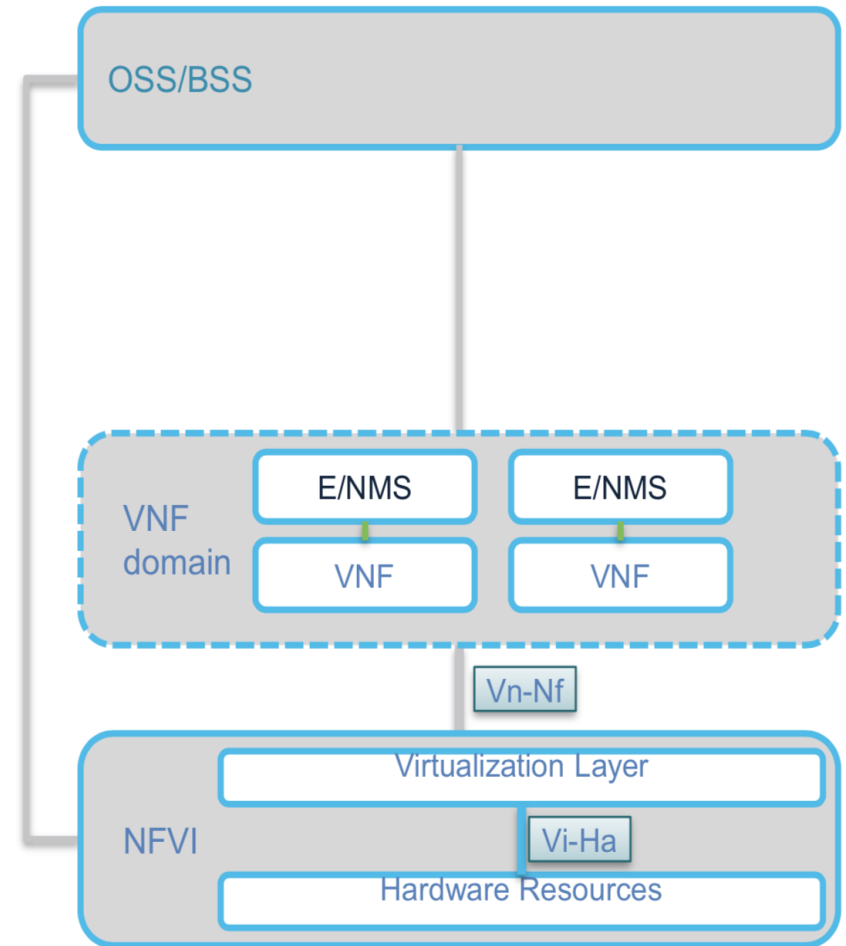


- NFV decouples network functions from the hardware, replacing dedicated network appliances (such as routers and firewalls) with software running on standard servers.
- Those network functions are called virtual network functions (VNFs)

Why NFV?

- With NFV, network functions are virtualized, multiple functions can be run on a single server.
- This means less physical hardware is needed, results in physical space, power, and overall cost reductions.
- Provides flexibility to run VNFs across different servers or move them around as needed when needed. This flexibility lets service providers deliver services and apps faster.

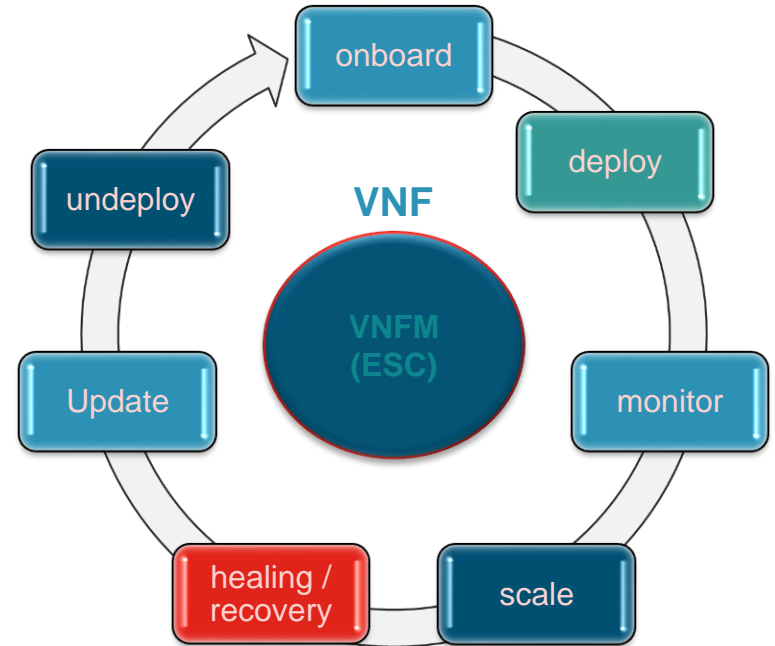
NFV Framework



VNF

- A VNF is a function of the virtualized network (NFV-Network function Virtualization).
- One or more Virtual Machines performing a single network function
- Collection of 1 or more VNFs providing a network service is service chain.
- It increases network scalability, helps in optimal use of network resources.

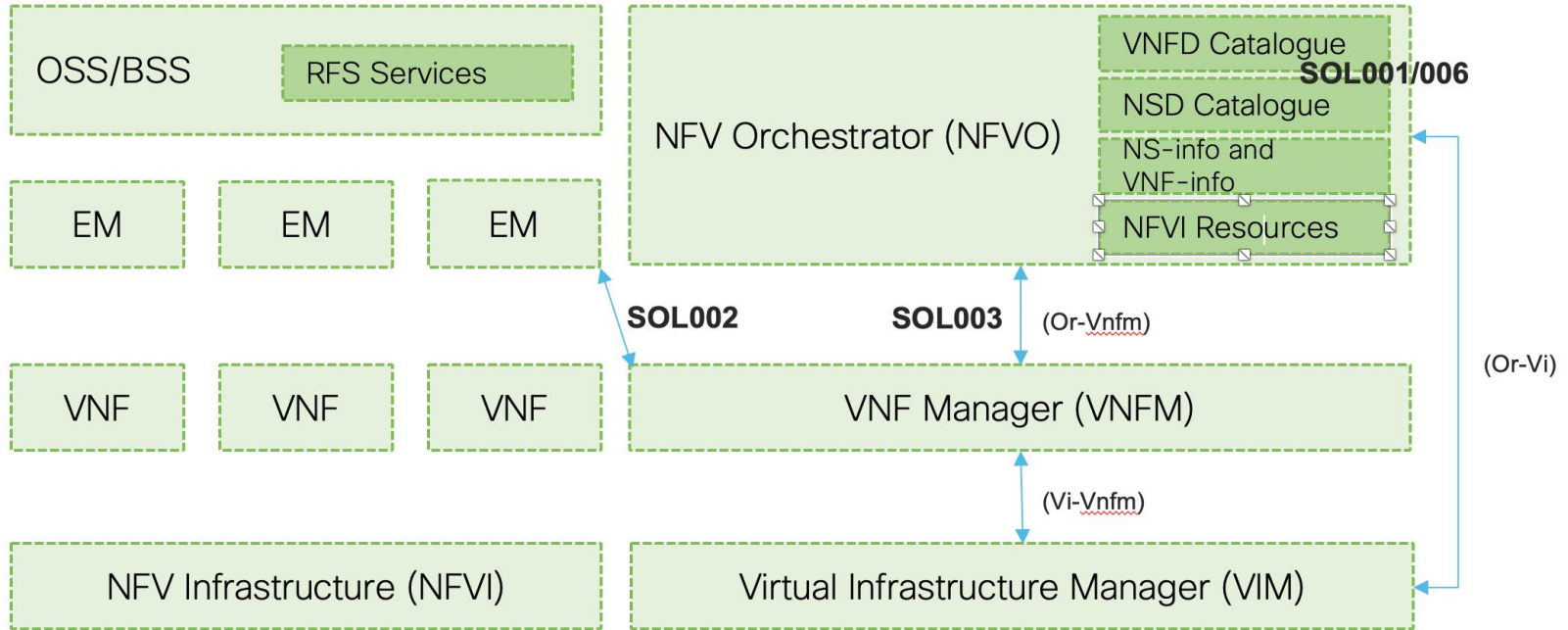
VNF Life Cycle



Network Function virtualization- Management and Orchestration (MANO)

- Functional block in NFV architecture stands for Management and Orchestration.
- Consist of three functional components
 - Virtualized Network Function Manager
 - Orchestrator
 - Virtualized Infrastructure Manager
- Manage of virtualized resources
 - Compute: Machines with CPU and Memory
 - Storage: Volumes of storage
 - Networks: subnets, ports, links and rules.

ETSI VNF MANO High Level Architecture

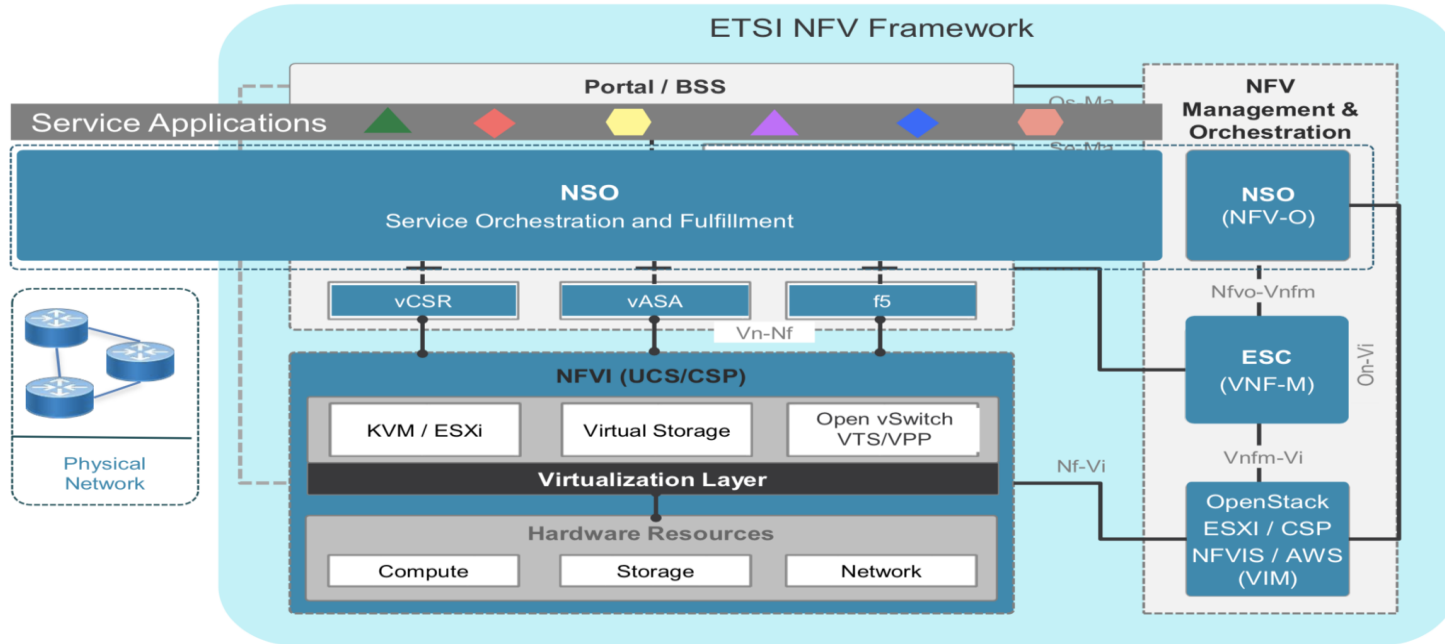


Cisco Solution for ETSI NFV Architecture



NSO-NFVO

One Solution Architecture



NSO-NFVO

- Cisco's NFVO solution is built on Cisco NSO.
- NSO manages the end-customer specific service configuration and resource orchestration on top of the running virtual machines. This covers both the NFVO and the OSS part.

NSO-NFVO

- The NFVO Core Function Pack enables use of MANO compliant VNFDs and NSDs according to the SOL006 specification in NSO.
- Enables onboarding and instantiation of these through VNFM.
- Resources like cpu, memory and disk-space in a virtual compute managed with the Resource Orchestration function.
- Further, VNF managed by NSO, in the NSO device tree.

NSO-NFVO

- The Cisco NFVO solution has been proven to onboard simple and complex (multi VM, multi-virtual deployment units) VNFs from over 20 vendors, supporting a broad range of VNF functions, including routing, load balancers, vEPC and other security functions.

NSO-NFVO Components

- Cisco-ETSI-NFVO
 - contains the YANG models according to the MANO specification (SOL006)
 - implements the instantiation logic of MANO descriptors on VNFMs and OpenStack, e.g VNF-Info and NS-Info.
 - Resource Orchestration functionality
- ESC NED SOL006 (YANG)
 - The NED for the ESC device.
- SOL003 NED Multi-VNFM
 - The NED for ETSI SOL3 compliant devices.
- OpenStack NED
 - The NED for OpenStack, used for the resource orchestration features

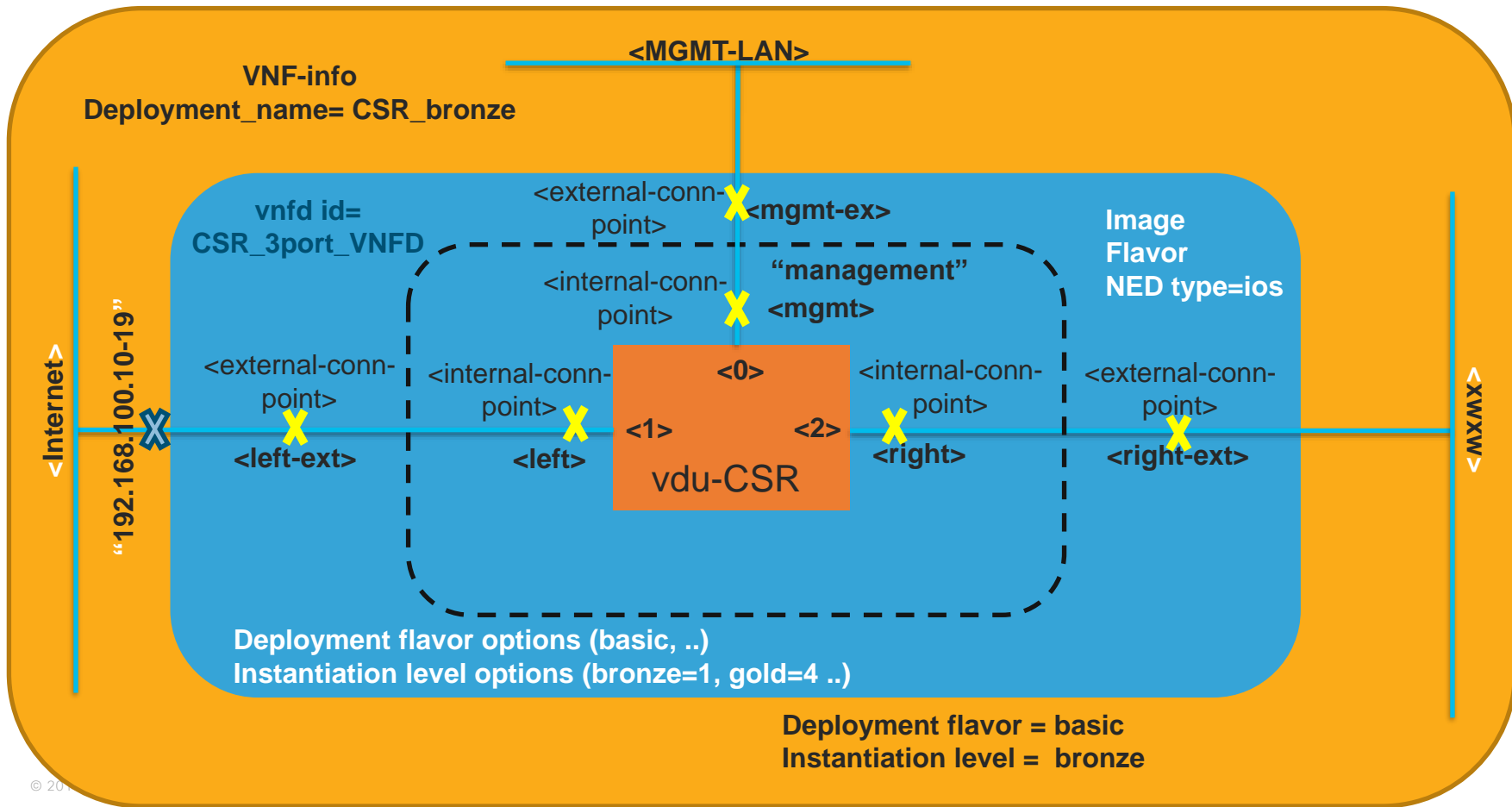
NFVO Alignment

- SOL003
- SOL004 (VNF Packaging)
- SOL006 (YANG)
- SOL001 (TOSCA)
- Multi-VNFM
- Multi-VIM
- ESC and NFVIS (NETCONF)

NFVO Terms

- A **VDU** references a VM image and runtime requirements
- A **VNFD** is a deployment template describing the VNF in terms of deployment and operational behavioural requirements.
- An **NSD** is static information elements used by the NFVO to instantiate a Network Service
- A **VNFC** is a deployable data referred to VNF under VNF Info.
- A **VNF Info** is a runtime record created by an NSD instantiating a VNFD
- A **NS Info** is a runtime record created by an NSD

VNFD



Working with VNFDs

- A VNF is described using a descriptor format VNFD.
 - The VNFD contains all information needed to boot virtual machines and network connection points.
 - VNFD consists of one or more Virtual Deployment Units.
- A VDU corresponds to one virtual machine.
 - VDU connection point connect to an internal virtual link or and external connection point.
 - Deployment flavour and instantiation levels specify which and how many of each VDU to instantiate.

Sample VNF Model and Catalogue

```
grouping vnf {
  leaf id {
    type string;
    description
      "Identifier of this VNF information element. This attribute
      shall be globally unique. The format will be defined in the
      data model specification phase.";
    reference
      "GS NFV-IFA011: Section 7.1.2, VNF information element";
  }

  leaf provider {
    type string;
    mandatory true;
    description
      "Provider of the VNF and of the WNFD";
    reference
      "GS NFV-IFA011: Section 7.1.2, VNF information element";
  }

  leaf product-name {
    type string;
    mandatory true;
    description
      "Name to identify the VNF Product. Invariant for the VNF
      Product lifetime.";
    reference
      "GS NFV-IFA011: Section 7.1.2, VNF information element";
  }

  leaf software-version {
    type string;
    mandatory true;
    description
      "Software version of the VNF. This is changed when there is
      any change to the software that is included in the VNF
      Package";
    reference
      "GS NFV-IFA011: Section 7.1.2, VNF information element";
  }

  leaf version {
    type string;
    mandatory true;
    description
      "Identifies the version of the WNFD";
    reference
      "GS NFV-IFA011: Section 7.1.2, VNF information element";
  }

  leaf product-info-name {
    type string;
    description
      "Human readable name of the WNF. Can change
      during the VNF Product Lifetime.";
    reference
      "GS NFV-IFA011: Section 7.1.2, VNF information element";
  }
}
```

```
list vdu {
  key "id";
  min-elements 1;
  description
    "The Virtualisation Deployment Unit (VDU) is a construct supporting
    the description of the deployment and operational behaviour of a
    VNF component, or the entire VNF if it was not componentized in
    components.";
  reference
    "GS NFV IFA011: Section 7.1.2, VNF information element";

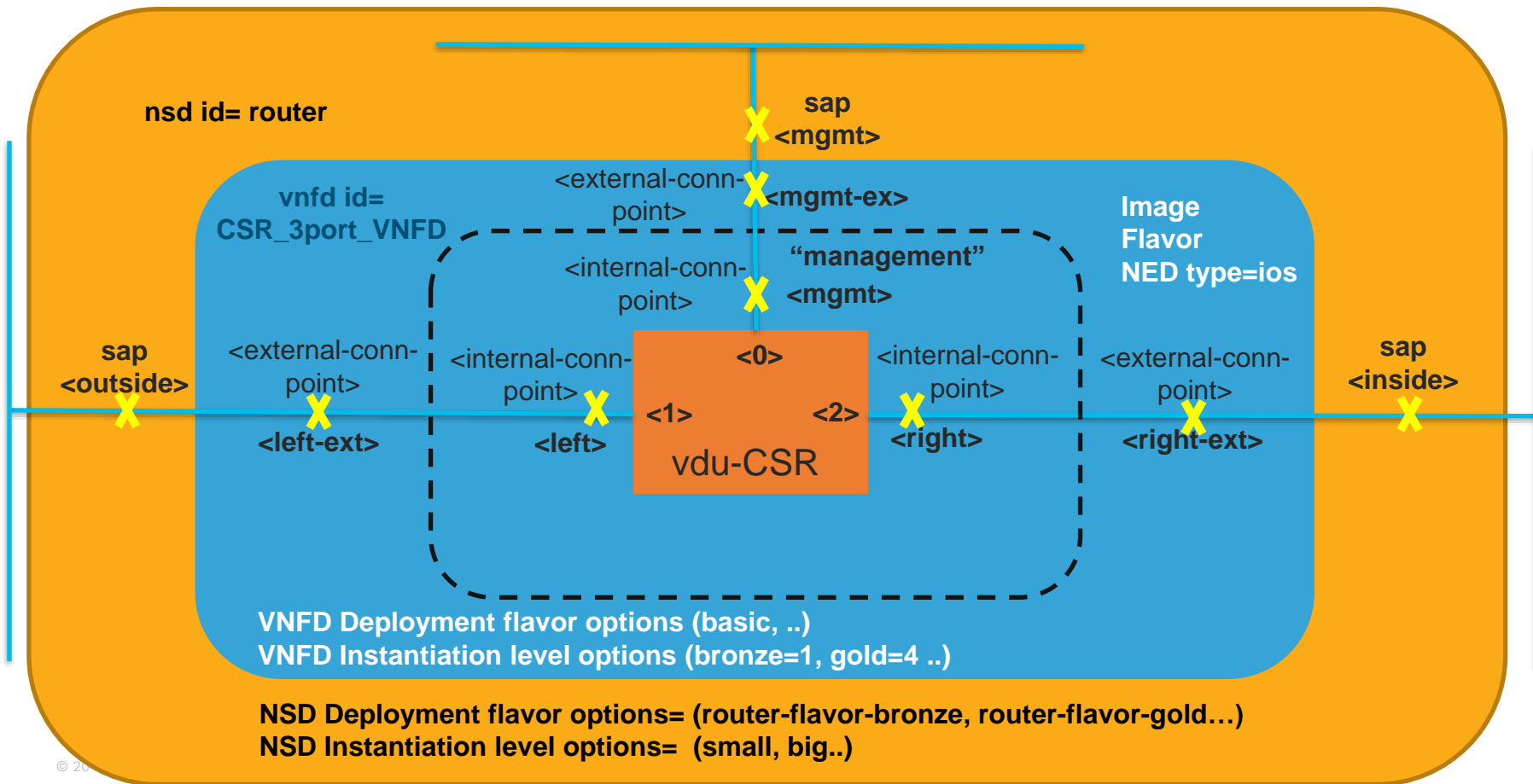
  leaf id {
    type string;
    description
      "Unique identifier of this VDU in WNF.";
    reference
      "GS NFV IFA011: Section 7.1.6.2, Vdu information element";
  }

  leaf name {
    type string;
    mandatory true;
    description
      "Human readable name of the VDU.";
    reference
      "GS NFV IFA011: Section 7.1.6.2, Vdu information element";
  }

  leaf description {
    type string;
    description
      "Human readable description of the VDU.";
    reference
      "GS NFV IFA011: Section 7.1.6.2, Vdu information element";
  }
}
```

```
{
  "data": {
    "etsi-nfv-descriptors:nfv": {
      "vnfd": [
        {
          "id": "ASA",
          "provider": "Cisco",
          "product-name": "virtual ASA",
          "software-version": "9.4.3",
          "version": "9.4.3",
          "vnfm-info": ["netconf"],
          "vdu": [
            {
              "id": "firewall",
              "name": "firewall",
              "int-cpd": [
                {
                  "id": "inside",
                  "layer-protocol": ["etsi-nfv-descriptors:ipv4"],
                  "cisco-etsi-nfv:interface-id": 1
                },
                {
                  "id": "mgmt",
                  "layer-protocol": ["etsi-nfv-descriptors:ipv4"],
                  "cisco-etsi-nfv:interface-id": 0,
                  "cisco-etsi-nfv:management": [null]
                },
                {
                  "id": "outside",
                  "layer-protocol": ["etsi-nfv-descriptors:ipv4"],
                  "cisco-etsi-nfv:interface-id": 2
                }
              ],
              "virtual-compute-desc": "vcd",
              "virtual-storage-desc": ["root"],
              "sw-image-desc": "firewall",
              "cisco-etsi-nfv:device-type": {
                "cli": {
                  "ned-id": "asa-id:cisco-asa"
                }
              }
            }
          ]
        }
      ]
    }
  }
}
```

NSD



Working with NSDs

- A network service is described using a descriptor format NSD.
- Describe a MANO network service consisting of several VNFDs and network service chains between these VNFDs.
- Describes how each VNFD's connection point connects to a NSD service access point (SAP) or a NSD virtual link
- The SAP defines the connection points the network service exposes to applications.
- NSD can contain one or more virtual links.
SAP or VNFD connection points can connect to these virtual links.

Sample NSD Model and Catalogue

```
grouping nsd {
  leaf id {
    type string;
    description
      "Identifier of this NSD information element. It Globally
      uniquely identifies an instance of the NSD.";
    reference
      "GS NFV IFA014: Section 6.2.2.2 Network Service Descriptor
      information element";
  }

  leaf designer {
    type string;
    description
      "Identifies the designer of the NSD.";
    reference
      "GS NFV IFA014: Section 6.2.2.2 Network Service Descriptor
      information element";
  }

  leaf version {
    type string;
    description
      "Identifies the version of the NSD.";
    reference
      "GS NFV IFA014: Section 6.2.2.2 Network Service Descriptor
      information element";
  }

  leaf name {
    type string;
    description
      "Provides the human readable name of the NSD.";
    reference
      "GS NFV IFA014: Section 6.2.2.2 Network Service Descriptor
      information element";
  }
}

leaf-list nested-nsd-id {
  must ". != ../id";
  type leafref {
    path "../..nsd/id";
  }
  description
    "References the NSD of a constituent nested NS.";
  reference
    "GS NFV IFA014: Section 6.2.2.2 Network Service Descriptor
    information element";
}

leaf-list vnfd-id {
  type leafref {
    path "../..vnfd/id";
  }
  description
    "References the VNFD of a constituent VNF.";
  reference
    "GS NFV IFA014: Section 6.2.2.2 Network Service Descriptor
    information element";
}
```

```
{
  "data": {
    "tailf-nfvo:nfvo": {
      "nsd": [
        {
          "id": "fw-router",
          "version": "1.0",
          "vnfd": [
            {
              "vnfd": "ASA",
              "connection-point": [
                {
                  "id": "cp-inside",
                  "virtual-link": "inside-net"
                },
                {
                  "id": "cp-mgmt",
                  "service-access-point": "mgmt"
                },
                {
                  "id": "cp-outside",
                  "service-access-point": "outside"
                }
              ]
            }
          ],
          "vnfd": "CSR1kv",
          "connection-point": [
            {
              "id": "left",
              "virtual-link": "inside-net"
            },
            {
              "id": "mgmt",
              "service-access-point": "mgmt"
            },
            {
              "id": "right",
              "service-access-point": "inside"
            }
          ]
        }
      ]
    }
  }
}
```

Workflow Hierarchy

Onboard VNF To Catalogue

- Virtual Network Function Descriptors (VNFD)
- Virtual link Descriptors (VLD) + Connection Points
- Images + Virtual Compute Resources loaded to VIM

Network Service Descriptors (NSD)

- Define Network Services Flavors
- VNFDs + VLD + SAP

Instantiate via NS-INFO

- Apply Service Logic to Collect appropriate resources for Network Service and pass VNF-Info to VNFM

Instantiate VNFs using VNF-Info

- passing it to appropriate VIM
- Manage LifeCycle of VNF
- Inform NFVO of Live Cycle Events

Instantiation of the VNF on the NFVI

- compute, storage and network

Day1 Configuration and NFVO/VNFM

- NSO informed of "service alive" – configure DAY1
- VNFM/NSO maintains LCM

N
F
V
O

V
N
F
M

V
I
M

E
M

NFVO service

- NETCONF NFVO service
 - NFVO Core work using the NETCONF interface towards an ESC VNFM.
- SOL003 NFVO service
 - The NFVO compliant with the ETSI SOL003 specification defining RESTful communication between the NFVO and VNFM.
- Resource Orchestration
 - To monitor and allocate VIM resources.

NSO-NFVO Features

VNF Lifecycle Management

- Create VNF Identifier
- Instantiate VNF
- Modify VNF Information
- Terminate VNF
- Delete VNF Identifier
- Heal VNF
- Operate VNF
- Lifecycle Notification
 - Subscription
 - Consumption
- VNF Operation Task
 - Rollback
 - Cancel
 - Retry
 - Fail

VNF Lifecycle Operation Granting

VNF Fault Management

VNF Performance Management

VNF Package Management

- Query VNF Packages
- Read VNFD
- Read Package Content
- Fetch Package Artifacts

NSO-NFVO Benefit

- The complete service (VNF service and MANO service) is managed in one YANG model and one transaction.
- Lifecycle management
- Fast Deployment of services
- Automation using Restful.
- Scale-out architecture.
- High Performance.

NSO-NFVO

Use case



VPN Automation

Resource Gathering

- Get Management IP
- Get VLAN
- Day-0 files for VNF

Day-0 Spin VNF

- Associate VNF Image
- Stich VNF with NICs and VLANs
- Day-0 base configs
- Spin up VNF
- Add VNF to NSO

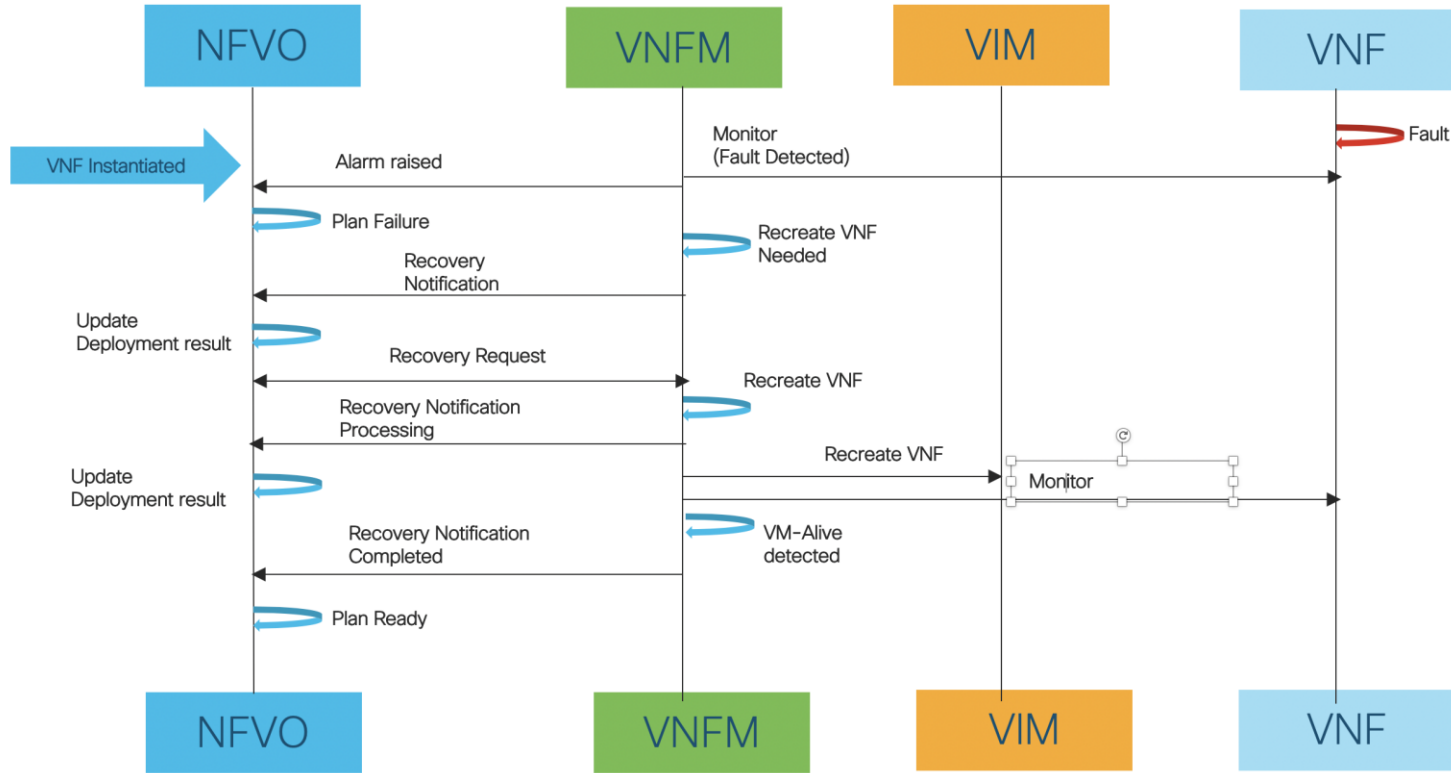
Day-1 Apply Config

- Get IP and VLAN from Resource Manager
- Apply Day-1 config to VNF

Day-2 Activity

- Start Day-2 Activity

VNF Recovery



NSO-NFVO

One Solution Mobility 5G Stack

