



**SAMSUNG**

# VNF Application Configuration in ONAP

2019-06-11/14 Kista ONAP DDF

Presenter:

Katsiaryna Kazak

# Agenda

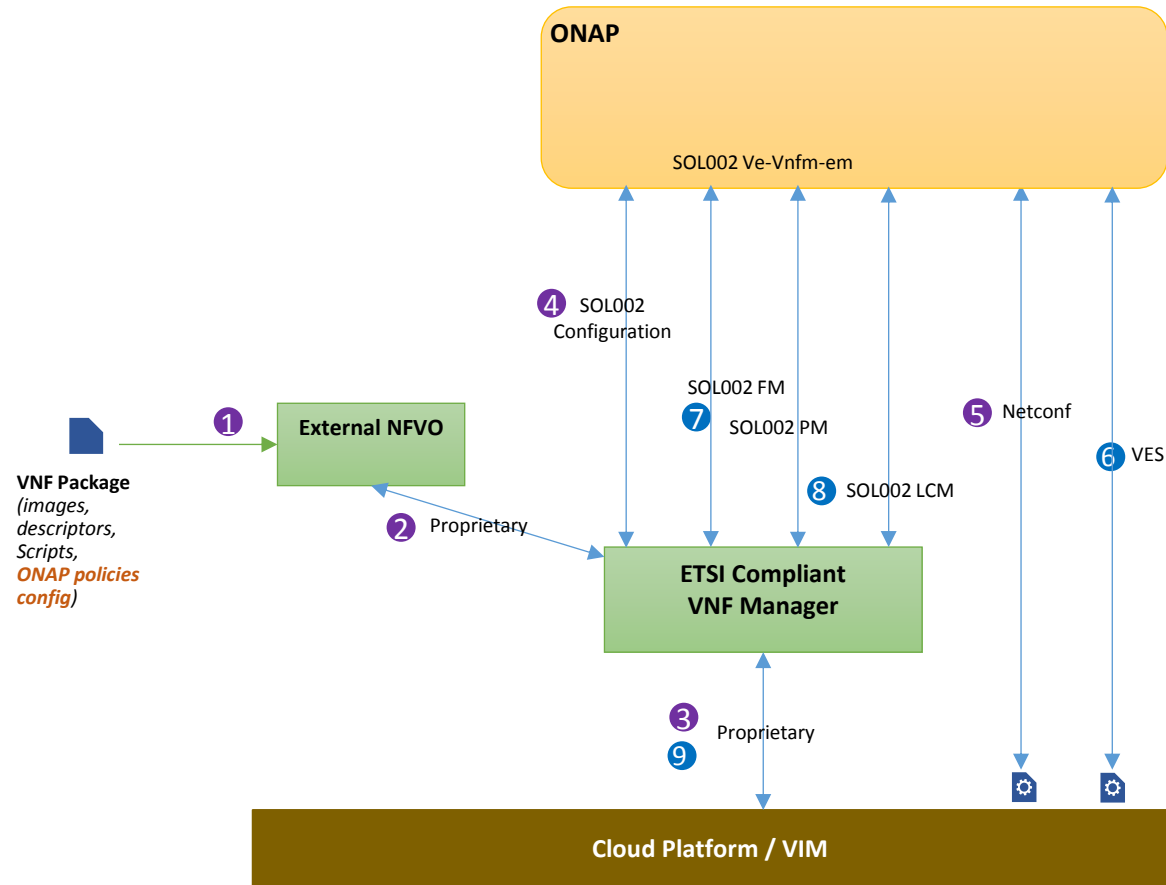
- Motivation
- Prototype work
- Proposed solution
- Proposed use-cases & interfaces
- Next steps
- Q&A

# Motivation

- Deployment scenarios

1. Operator is using ONAP to orchestrate ETSI compliant VNF, that expects SOL002 Application configuration
  - Service assurance is performed by the ONAP closedloop automation.
2. Operator is using ONAP to orchestrate a service while leveraging an ETSI compliant VNFM
  - There is no vendor specific (or generic) EMS, and ONAP performs the EMS roles.
  - VNFM orchestrates ETSI compliant VNF, that expects SOL002 Application configuration
  - Service assurance is performed by the ONAP closed loop automation. ONAP may call the VNFM to perform actions related to the virtual resource control (e.g. restart a VM, re-deploy the VNF, scale out, etc.)

# Samsung Prototype work



## Prerequisites:

- Register ONAP in External VNFM using SOL002 Authorization API
- Add ONAP-specific configuration to VNF package

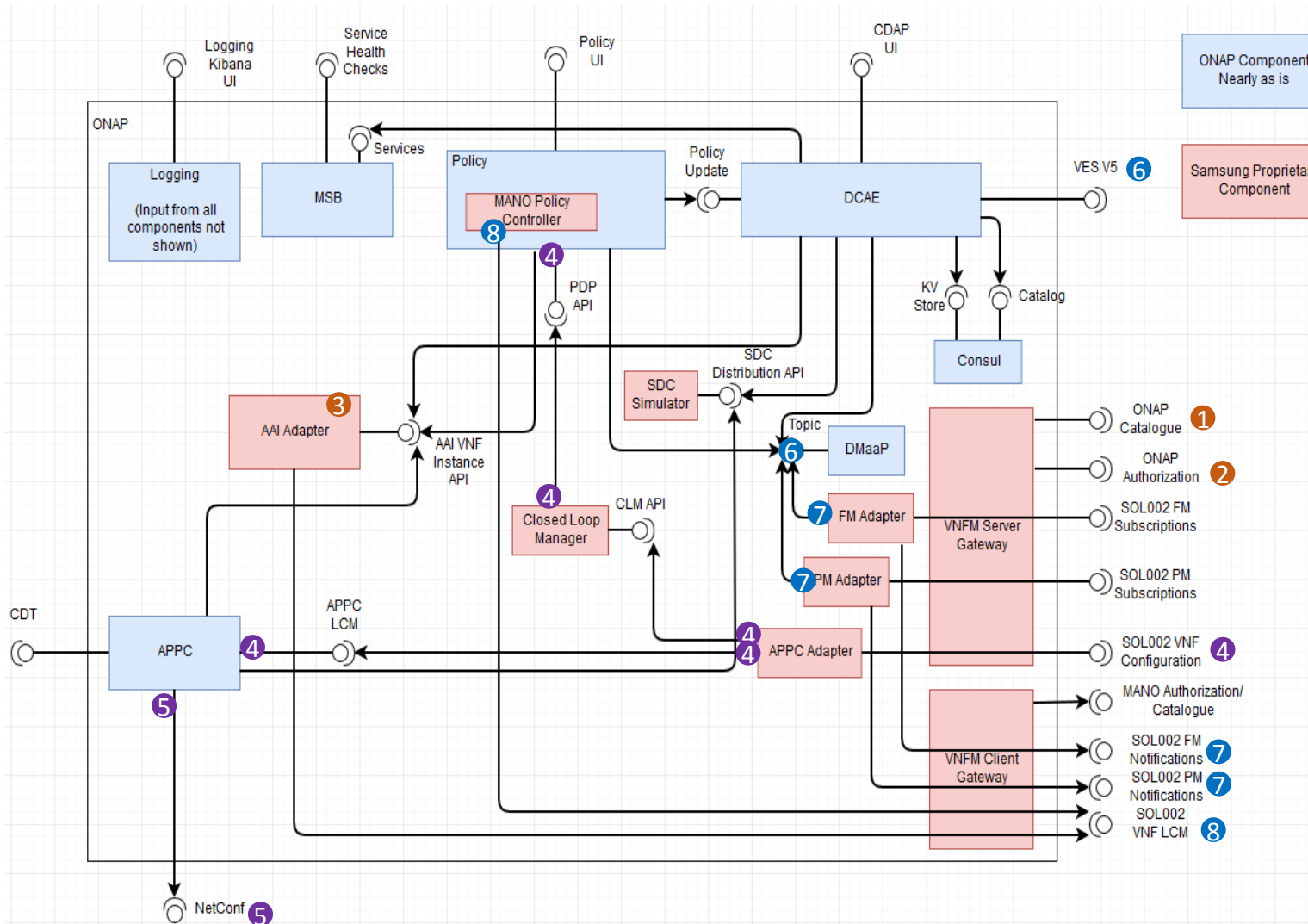
## Instantiation:

1. Designing and onboarding of VNF was happening in external NFVO
2. External NFVO was decomposing VNF and sending request to ETSI Compliant VNFM
3. VNFM performed resource assignments and instantiated VNF
4. VNFM calls ONAP to perform VNF Application configuration via SOL002
5. ONAP configures VNF using Netconf and send 200 OK response

## Heal/Scale out/in:

6. VNF sends VES event with high load data
7. In parallel metrics from DMaaP are sent to VNFM via SOL002 FM / PM interface
8. ONAP makes decision to scale out / heal / scale in and sends request via SOL002 LCM
9. VNFM scales out / heals / scales in VNF

# Samsung Prototype work



- Prototype based on Beijing release
- Subset of ONAP components was used

## Prerequisites:

1. SOL002 Catalogue API for registering ONAP in VNFM
2. SOL002 Authorization API for registering ONAP in VNFM
3. AAI Adapter returns VNF data via SOL002 LCM Resource API

## Instantiation:

4. VNFM uses SOL002 VNF Application configuration
5. ONAP configures VNF using Netconf and send 200 OK response

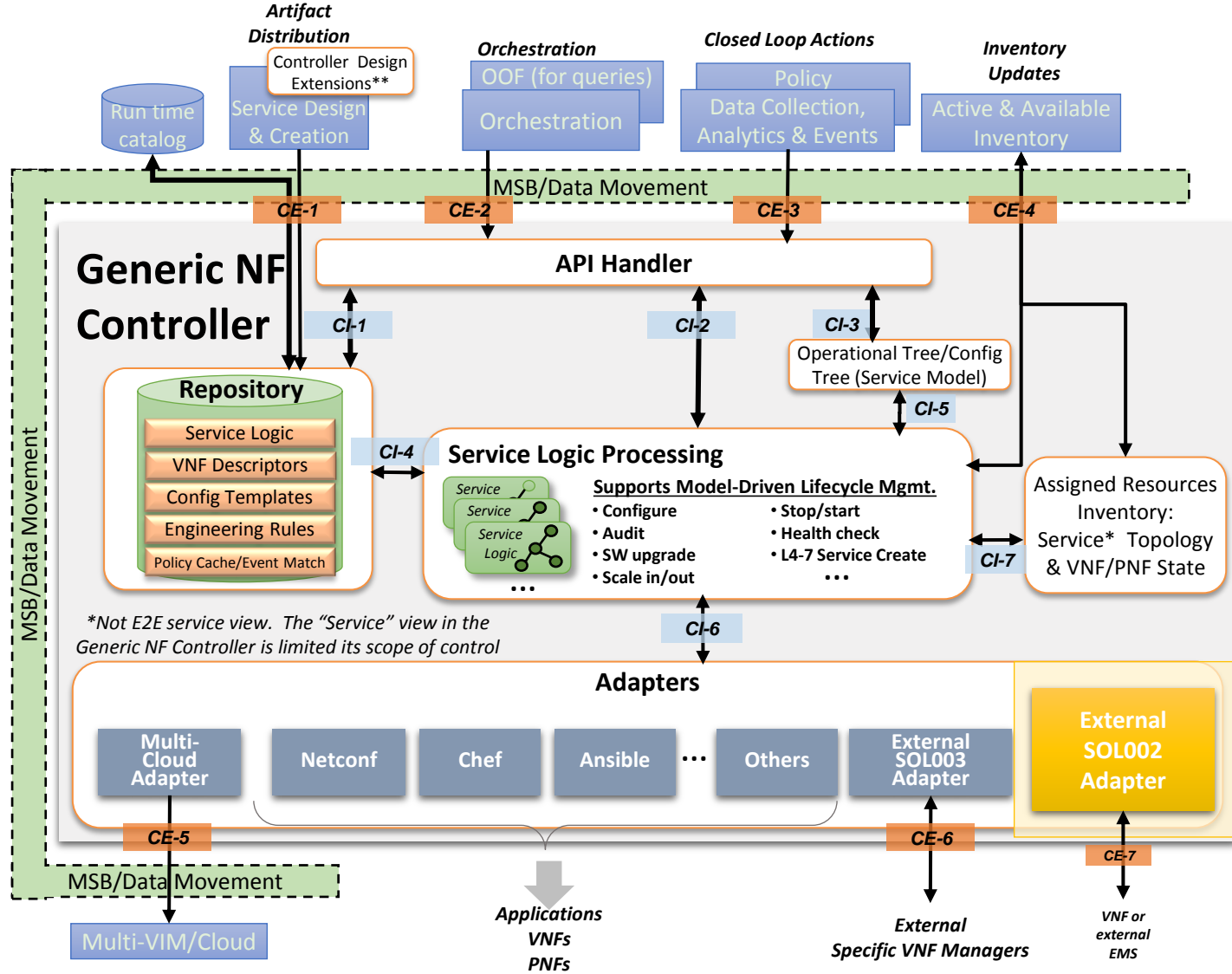
## Heal/Scale out/in:

6. VNF sends VES event with high load data
7. In parallel metrics from DMaaP are sent to VNFM via SOL002 FM / PM interface
8. ONAP makes decision to scale out / heal / scale in and sends request via SOL002 LCM

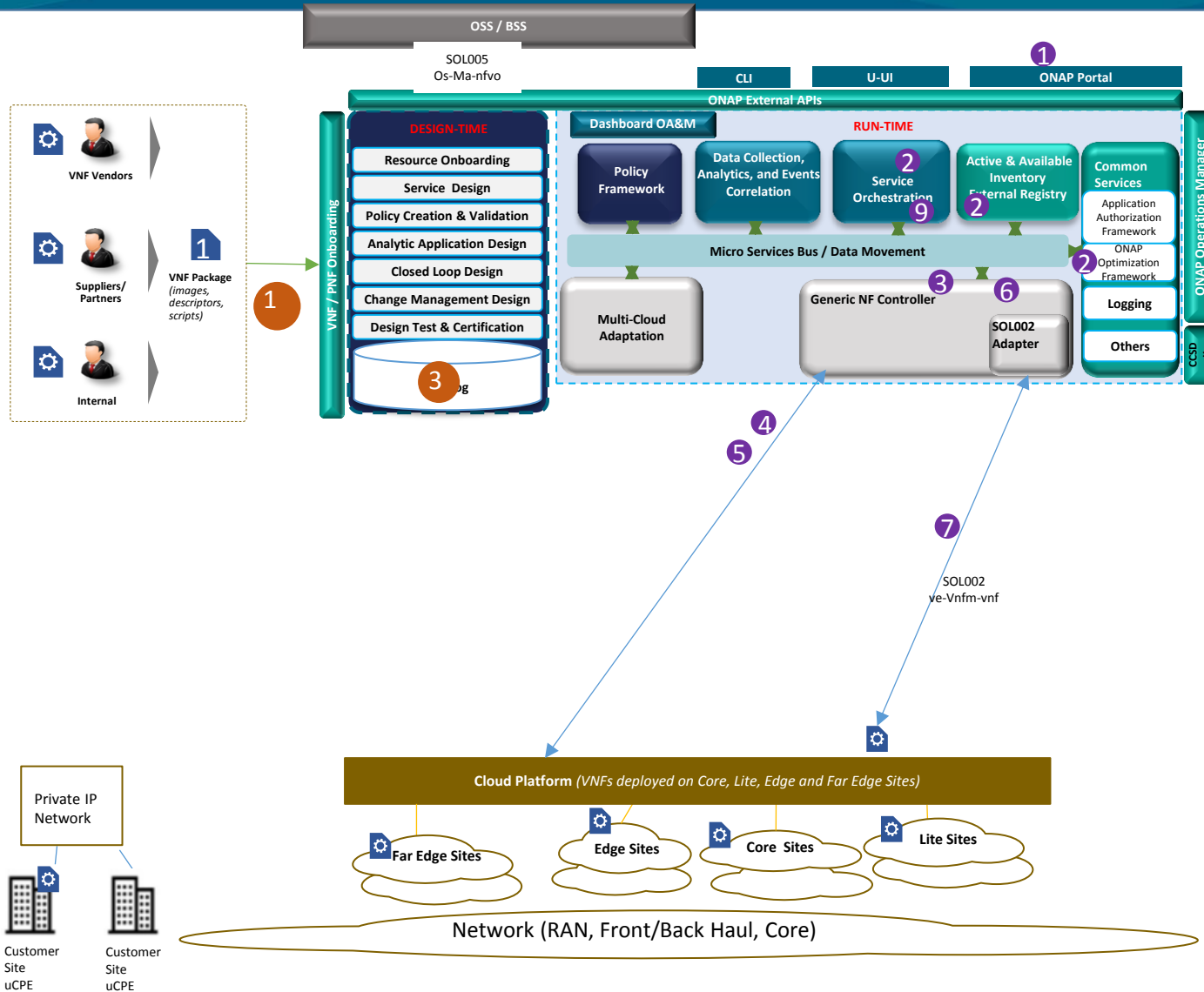
# Introducing ETSI Application Configuration in ONAP

- Proposed solution:
  - Introduction of SOL002 Adapter component OR possibly adding this functionality into SOL003 Adapter
  - The main integration point to ONAP will be GNFC module
  - Additional integration points may include the Policy engine, A&AI, APP-C, SO, SDC etc.

# Generic NF Controller Architecture



# ONAP orchestrates ETSI compliant VNF



## Design/Develop Time:

- Onboard and catalog a SOL004 VNF package with a SOL001 VNF-D.
  - Application level ConfigurableProperties attributes included in the VNF-D from the vendor
  - Could also be provide in a separate artifact in the SOL004 package
- Design an ONAP Service A with deployment and application configuration referencing the onboarded VNF and VNF type.
  - ConfigurableProperties extracted from the VNF-D(s) or alternate artifact and composed into Service level configuration

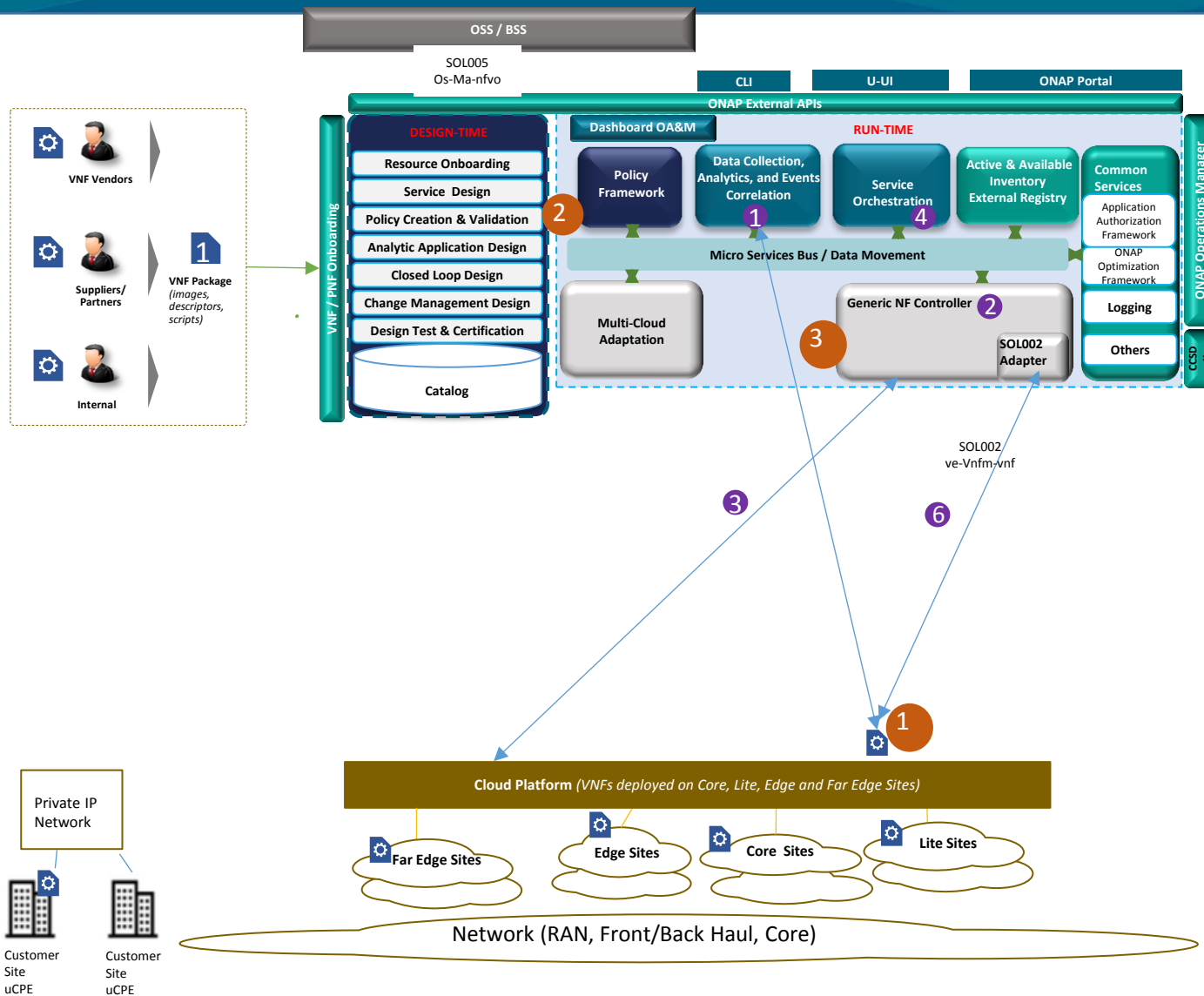
## Run Time:

- ONAP receives request to create an instance of Service A with appropriate configuration data.
- ONAP "decomposes" request into VNF 1 and homes it.
- ONAP makes resource assignments for deployment of VNF 1 based on information in the VNF-D
  - #vCPU, RAM, Network, Storage, CPU pinning, SR-IOV, EPA
  - Issue on if/how resources are "reserved" by OOF/A&I
- SO makes assignments (IP Address) for the VNF via GNF-C
- SO calls VIM to instantiate VNF 1
- ONAP SO decomposes service config data for this VNF and asks GNF-C to configure VNF 1
- GNF-C updates VNF application configuration via SOL002 request „VnfConfiguration“

It's a very simple extension to existing controllers. We just add a new configuration management SBI in addition to current ones



# ONAP scales out ETSI compliant VNF



## Precondition:

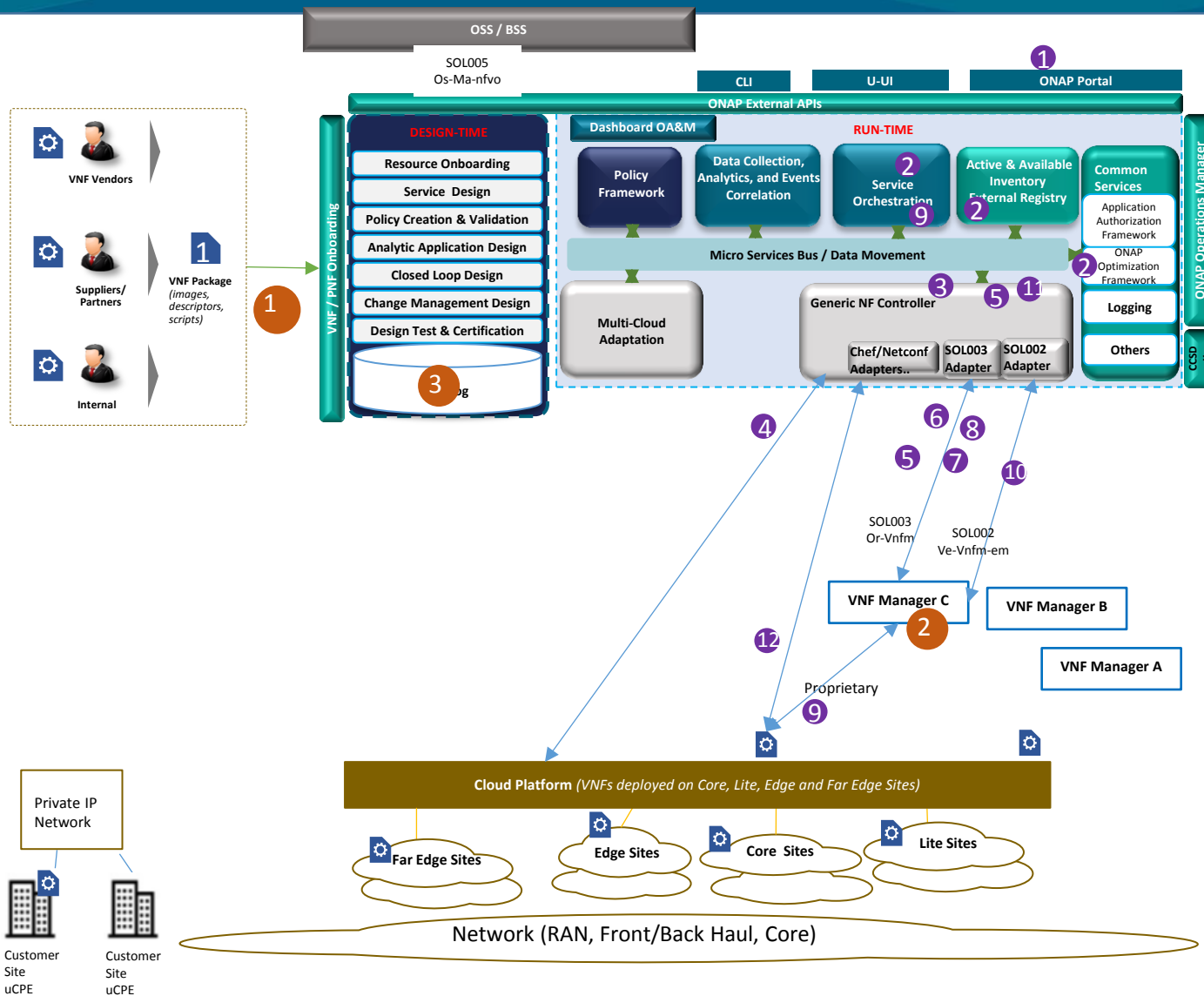
1. VNF 1 was previously instantiated by ONAP via GNF-C using SOL002 Application Configuration. **VNF supports sending events directly to DCAE via VES.**
2. Policy was created such that VNF scaling would be triggered upon VNF 1 busy trigger.
3. GNF-C has registered to listen on VNF 1 related topics (DCAE)

## Run Time:

1. **VNF 1 sends VES load event with high value to DCAE which Policy associates with overload.**
  - a) **DCAE distributes overload on VNF 1 topic**
  - b) **GNF-C would indicate to VNFM C that VNF 1 is overloaded**
  - c) **DCAE/Policy would trigger SO to do a VNF scale-out**
2. ONAP makes resource assignments for scale-out of VNF 1 based on info in the VNF-D
  - a) Resources : #vCPU, Ram, Network, Storage, CPU pinning, SR-IOV, EPA
  - b) Reservation of resources?
3. SO calls VIM to scale-out VNF 1
4. GNF-C notifies SO that VNF 1 has finished scale
5. **ONAP SO decomposes service config data for this VNF and asks GNF-C to configure new VNFC**
6. **GNF-C updates VNFC application configuration via SOL002 request „VnfConfiguration” with updated VnfConfigurationData**

It's a very simple extension to existing controllers. We just add a new configuration management SBI in addition to current ones

# ONAP orchestrates a service with ETSI compliant VNFM



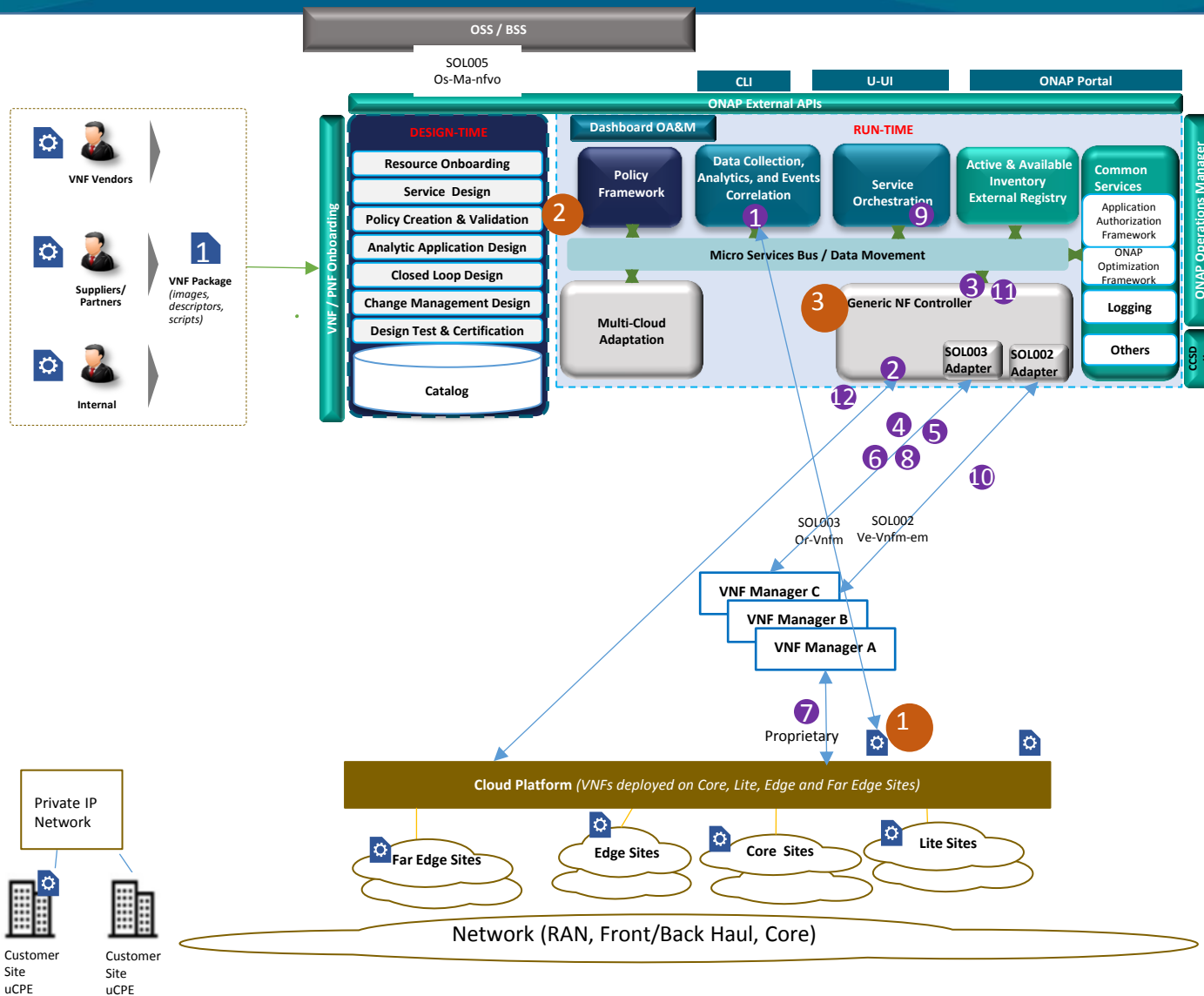
## Design/Develop Time:

- Onboard and catalog a SOL004 VNF package with a SOL001 VNFD.
  - Application level ConfigurableProperties attributes included in the VNF-D from the vendor
  - Could also be provide in a separate artifact in the SOL004 package
- VNFM C registers as a SOL003/SOL002 compliant VNFM
  - Support SOL003 "ModifyVnfInfo" operation
  - Support SOL002 „VnfConfiguration" operation
- Design an ONAP Service A with deployment and application configuration referencing the onboarded VNF and VNFM type.
  - ConfigurableProperties extracted from the VNF-D(s) or alternate artifact and composed into Service level configuration

## Run Time:

- ONAP receives request to create an instance of Service A with appropriate configuration data.
- ONAP "decomposes" request into VNF 1 and homes it.
- ONAP makes resource assignments for deployment of VNF 1 based on information in the VNF-D
  - #vCPU, RAM, Network, Storage, CPU pinning, SR-IOV, EPA
  - Issue on if/how resources are "reserved" by OOF/A&I
- SO makes assignments (IP Address) for the VNF via GNF-C
- SO asks GNF-C to instantiate VNF 1, passing "deployment data" values.
- ONAP selects VNFM C (match type and homing) and calls via SOL003 API, requesting creation of VNF 1, passing "deployment data" values.
- VNFM C upcalls ONAP asking for a "grant" of resources based on information in the VNF-D.
- ONAP responds with resources allocated in #3 along with VIM credentials.
- VNFM C calls VIM to create VNF 1, and applies any needed deployment data via proprietary API.
- VNFM C sends SOL002 request „VnfConfiguration" to ONAP GNF-C (playing role of EM here)
- ONAP SO decomposes service config data for this VNF and asks GNF-C to configure VNF 1**
- GNF-C updates the VNF via Chef/Netconf/Ansible**

# ONAP scales out service with ETSI compliant VNFM via VES



## Precondition:

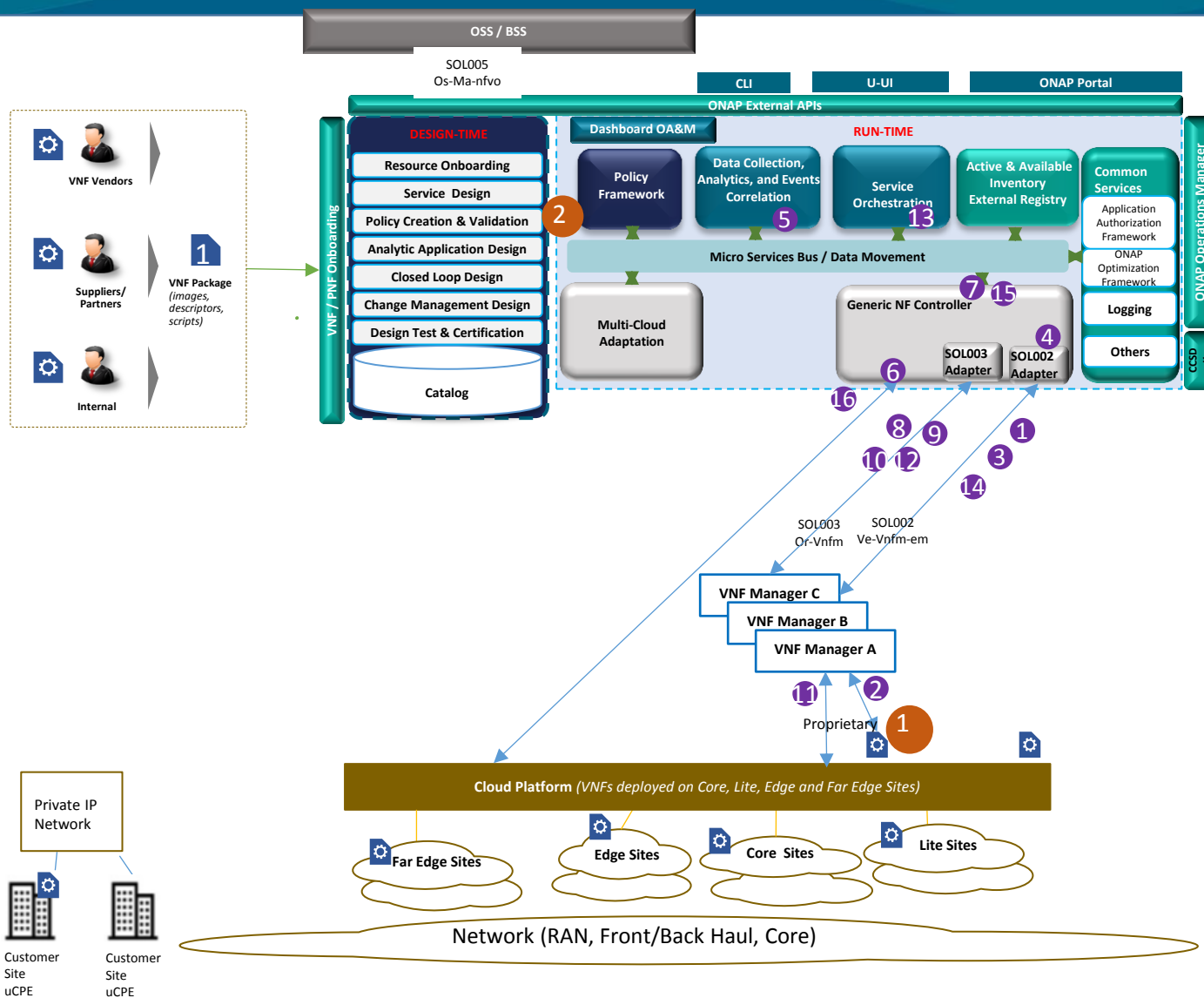
- VNF 1 was previously instantiated by ONAP using VNFM C via GNF-C using SOL003 and SOL002 Configuration. VNF supports sending events directly to DCAE via VES
- Policy was created such that VNF scaling would be triggered upon VNF 1 busy trigger.
- GNF-C has registered to listen on VNF 1 related topics (DCAE)

## Run Time:

- VNF 1 sends load VES event with high value to DCAE which Policy associates with overload.
  - DCAE distributes overload on VNF 1 topic
  - GNF-C would indicate to VNFM C that VNF 1 is overloaded
  - DCAE/Policy would trigger SO to do a VNF scale-out
- ONAP makes resource assignments for scale-out of VNF 1 based on info in the VNF-D.
  - Resources : #vCPU, Ram, Network, Storage, CPU pinning, SR-IOV, EPA
  - Reservation of resources?
- SO calls GNF-C to scale-out VNF 1
- GNF-C contacts VNFM C via SOL003 API, requesting a scale-out of VNF 1.
- VNFM C upcalls GNF-C asking for a "grant" of resources based on info in the VNF-D.
  - Should be same resources as were allocated in 2a
- GNF-C responds with resource information allocated in #2a along with VIM credentials.
- VNFM C calls VIM to scale VNF 1, applies any needed deployment data for a new VM via proprietary API
- VNFM C sends VNF 1 scale completed notification to GNF-C
- GNF-C notifies SO that VNF 1 has finished scale
- VNFM C sends SOL002 request „VnfConfiguration” with VnfConfigurationData for new VNFC to ONAP GNF-C (playing role of EM here)**
- ONAP SO decomposes service config data for this VNF and asks GNF-C to configure new VNFC**
- GNF-C updates the VNFC via Chef/Netconf/Ansible**

More use-cases are possible with different combinations: VES event can be sent by VNFM or other components

# ONAP scales out service with ETSI compliant VNFM via PM/FM



## Precondition:

1. VNF 1 was previously instantiated by ONAP using VNFM C via GNF-C using SOL003.  
VNF doesn't support sending VES events. VNF is monitored by VNFM C via proprietary protocols.
2. Policy was created such that VNF scaling would be triggered upon VNF 1 busy trigger.

## Run Time:

1. GNF-C (Playing EM Role) SOL002 Adapter calls VNFM C using SOL002 in order to:
  - a) Create PM Job
  - b) Subscribe to PM messages
2. VNF 1 sends load event with high value to VNFM C via proprietary protocol.  
Or, VNFM C collects performance metrics using proprietary mechanism from VIM.
3. VNFM C sends notify event to GNF-C with performance metrics .
4. GNF-C SOL002 Adapter translates event into ONAP format and sends to DCAE via Dmaap.
5. DCAE distributes event on VNF 1 topic
  - a) DCAE/Policy would trigger SO to do a VNF scale-out
6. ONAP makes resource assignments for scale-out of VNF 1 based on info in the VNF-D
  - a) Resources : #vCPU, Ram, Network, Storage, CPU pinning, SR-IOV, EPA
  - b) Reservation of resources?
7. SO calls GNF-C to scale-out VNF 1
8. GNF-C contacts VNFM C via SOL003 API, requesting a scale-out of VNF 1.
9. VNFM C upcalls GNF-C asking for a "grant" of resources based on info in the VNF-D.
  - a) Should be same resources as were allocated in 2a
10. GNF-C responds with resource information allocated in #2a along with VIM credentials.
11. VNFM C calls VIM to scale VNF 1, applies any needed deployment data for a new VM via proprietary API
12. VNFM C sends VNF 1 scale completed notification to GNF-C
13. GNF-C notifies SO that VNF 1 has finished scale
14. VNFM C sends SOL002 request „VnfConfiguration" with VnfConfigurationData for new VNFC to ONAP GNF-C (playing role of EM here)
15. ONAP SO decomposes service config data for this VNF and asks GNF-C to configure new

More use-cases are possible with different combinations: FM/PM may come from different source

# Interfaces

- **ONAP Authorization**
  - Authorization API according to SOL002 Chapter 4.5. This means OAuth 2.0 on the next level of specs
- **ONAP Catalogue**
  - ONAP Catalogue API according to MANO convention. This contains endpoints for FM, PM and VNF Configuration
- **SVNFM Authorization/Catalogue**
  - SVNFM Authorization API and Catalogue API. Authorization is OAuth 2.0. The catalogue contains all the endpoints used by ONAP
- **SOL002 VNF Configuration**
  - VNF Configuration API according to SOL002 Chapter 9
- **SOL002 FM**
  - VNF Performance Management API according to SOL002 Chapter 7
- **SOL002 PM**
  - VNF Performance Management API according to SOL002 Chapter 6

# Next steps

- Further discussion in the Orchestration Scenarios Task force
  - Agree on mechanism(s) to pass values of the application parameters to VNFs through APP-C/GNF-C
  - Agree on communication between SOL002 Adapter and APP-C
- Design and develop a PoC of using both SOL003 & SOL002 Plugins to APP-C/GNF-C so that configuration and other LCM can follow the current path
- Extend Existing ONAP VNFM simulator to SOL002 interfaces

Questions?



# ONAP

OPEN NETWORK AUTOMATION PLATFORM