



Deployment Scenarios and Application Configuration Proposal

Fernando (Fred) Oliveira

Orchestration Scenarios

Confluence Page: <https://wiki.onap.org/display/DW/Orchestration+Scenarios>

- Assumptions

- Operator has already deployed ETSI SOL003 compliant Vendor VNF Manager(s)
- Operator has already deployed ETSI SOL005 compliant NFVO(s)
- Operator has existing Service Assurance tools that VNF(s) and PNF(s) interact with
- Operator is currently using proprietary mechanisms to configure [P/V]NFs
- Operator has used above mechanisms to deploy SOL001/SOL004 compliant VNFs
- Operator has multiple VIMs; with different HW capabilities and configurations
- Operator desires to integrate ONAP into the existing environment

- Objectives

- Examine orchestration scenarios in order to determine if new architectural requirements are necessary
- Develop recommendations for composition of the ONAP Service DM (How does the ETSI NFV SOL001 Network Service Descriptor supported?)
- Develop recommendations for interfaces between the orchestration elements (OSS, SO, VF-C, VNFM, NFVO)
- Develop recommendations for how ONAP controllers work with external orchestration elements (NFVO, VNFM, EMS)
- Develop recommendations for how external Service Assurance interacts with DCAE and Policy

Scenarios examined

- ONAP with Vendor VNFM, EMS, SA and Inventory #1a (VNF deployment)
- ONAP with Vendor VNFM, EMS, SA and Inventory #1b (VNF Scale Out)
- ONAP with Vendor VNFM, EMS, SA and Inventory #1c (VNF Scale In)
- ONAP with Vendor VNFM, EMS, SA and Inventory #1d (VNF deployment with Application Configuration via SOL003 ModifyVnflInfo interface)
- ONAP with Vendor VNFM, EMS, SA and Inventory #1e (VNF deployment with Application configuration via SOL003 ModifyVnflInfo interface and SOL002 ConfigureVNF interface)
- ONAP with Vendor VNFM, EMS, SA and Inventory #1f (VNF deployment with Application configuration via ONAP Netconf/Restconf)
- ONAP with Legacy Orchestration, EMS, SA and Inventory #2 (NS deployment)
- ONAP with Legacy Orchestration, EMS, SA and Inventory #3a&b (Service deployment with VNF with ONAP App config and NS deployment)

Derived Requirements

- External VNFM scenarios (1a, 1b, 1c):
 - ONAP needs to ingest and save (without modification) a SOL004 CSAR package for later consumption by a SOL003 compliant VNF Manager (VNFSKD, SDC)
 - ONAP needs to ingest and interpret a SOL001 compliant VNF Descriptor in order to design an ONAP Service (VNFSKD, SDC)
 - ONAP needs to understand resource requirements in the VNF-D for each deployment and scaling level (SO, A&AI, OOF)
 - ONAP needs to have a SOL003 compliant SBI (SO, VF-C, GNF-C)
 - ONAP needs a mechanism for specifying that a VNF instance should be runtime managed by a particular VNFM type (design time) and instance (run time) (SO, OOF, A&AI)
 - ONAP needs to have a way to inventory a VNF that was deployed using an external VNFM (A&AI)
- External NFVO scenarios (2, 3a, 3b):
 - ONAP needs to ingest and save (without modification) a SOL004 CSAR package for later consumption by a SOL003 compliant VNF Manager (VNFSKD, SDC)
 - ONAP needs to ingest and interpret a SOL001 compliant VNF Descriptor in order to design an ONAP Service (VNFSKD, SDC)
 - ONAP needs to be able to convert an ONAP Service into a SOL001 compliant Network Service Descriptor (NS-D)
 - ONAP needs to have a SOL005 compliant SBI (SO)
 - ONAP needs a mechanism for specifying that a service should be runtime managed by an external NFVO (SDC)
 - ONAP needs to have a way to inventory a Service that was deployed as a Network Service using and external NFVO (SO, A&AI)
 - ONAP needs a way to ingest and save (without modification) a SOL007 Network Service Package. (VNFSKD, SDC)
 - ONAP needs to ingest and interpret a SOL001 Network Service Descriptor (SDC)
 - ONAP needs to be able to design a Service that includes some VNFs and some hierarchy of NSs (SDC)
- VNF Application Configuration over SOL003 ModifyVnflInfo interface using an External vendor specific VNF Manager (1d)
 - ONAP needs to ingest and interpret a SOL001 compliant VNF Descriptor that includes ConfigurableProperties (VNFSKD, SDC, SO)
 - ONAP needs to present these ConfigurableProperties in the design of an ONAP Service (SDC)
 - Upon Service deployment, ONAP needs to pass the Application ConfigurableProperties with the values to the external VNFM that was used to deploy the VNF using the SOL003 ModifyVnflInfo interface (SO)
 - As part of a Change Management operation, ONAP needs to pass the updated ConfigurableProperties and new values to the external VNFM that was used to deploy the VNF using the SOL003 ModifyVnflInfo interface (SO)
 - As part of an ONAP recovery operation, ONAP needs to query the VNF to get the current state and values of the ConfigurableProperties using the external VNFM that was used to deploy the VNF using the SOL003 Query interface (SO)

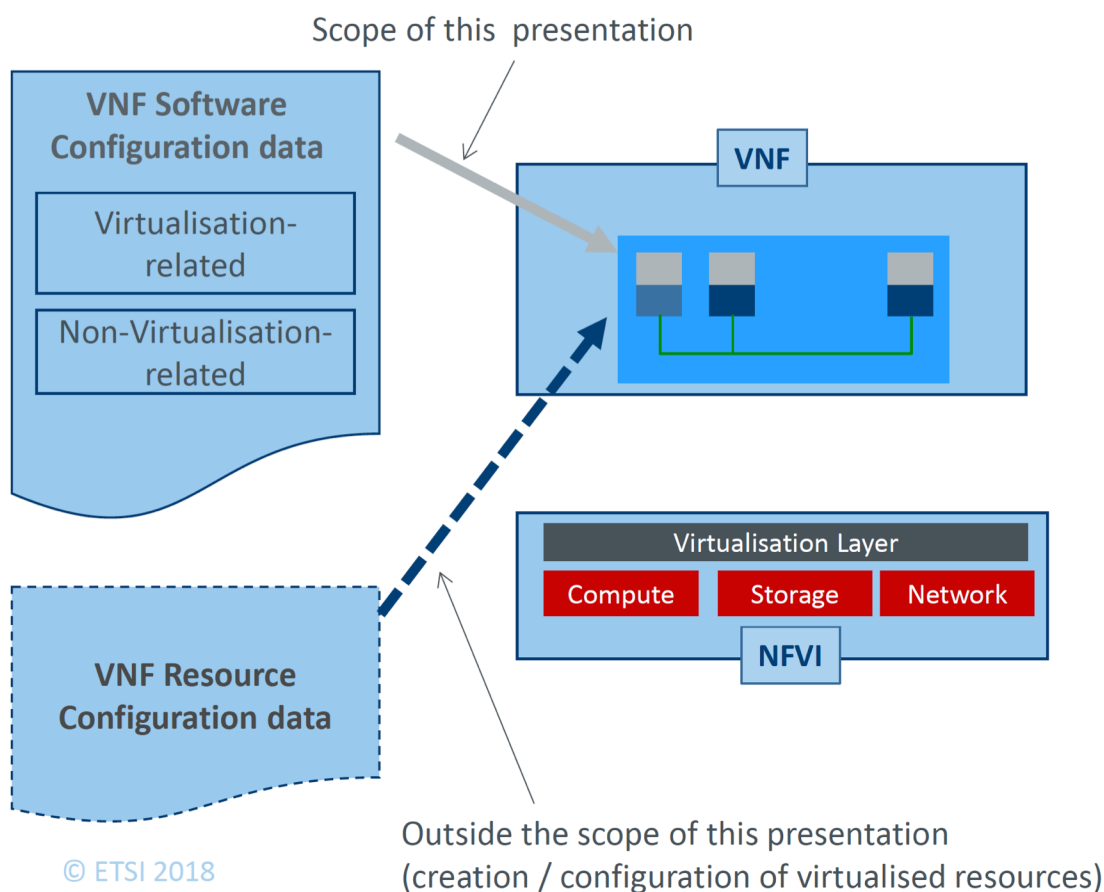
Proposal for Application Configuration using ETSI mechanisms; Treat SOL003 adapter as a GNF controller.

- Leverage SOL001 VNF-D ConfigurableProperties to describe all of the Application Configuration that the VNF supports
- Use SOL003 ModifyVnfInfo interface to program the application configuration
 - Initial Deployment related data (DNS, EMS, NTP, ...)
 - Initial Application Configuration (Cluster partner(s), profile, users, ..)
 - Ongoing configuration updates (new profile, new users, ...)
- Agnostic to VNFM → VNF configuration mechanism
 - Restconf/Netconf
 - SOL002 VNF Configuration interface
 - Proprietary

ETSI defined configuration data (SOL001 ConfigurableProperties)



Classification of VNF configuration data



Virtualisation-related configuration parameters

- ✓ Parameters whose value is or can be influenced by processing functions in the NFVI and/or NFV-MANO
- ✓ e.g. IP address of a Connection Point of a VNFC to be configured on another VNFC

Non-virtualisation-related configuration parameters

- ✓ Parameters whose value cannot be influenced by processing functions in the NFVI and/or MANO
- ✓ e.g. APN-to-GGSN mappings in a SGSN VNF

ETSI defined configuration data (SOL001 ConfigurableProperties)

CONFIGURATION PATH B: (OSS -> NFVO or EM ->) VNFM -> VNF software instance



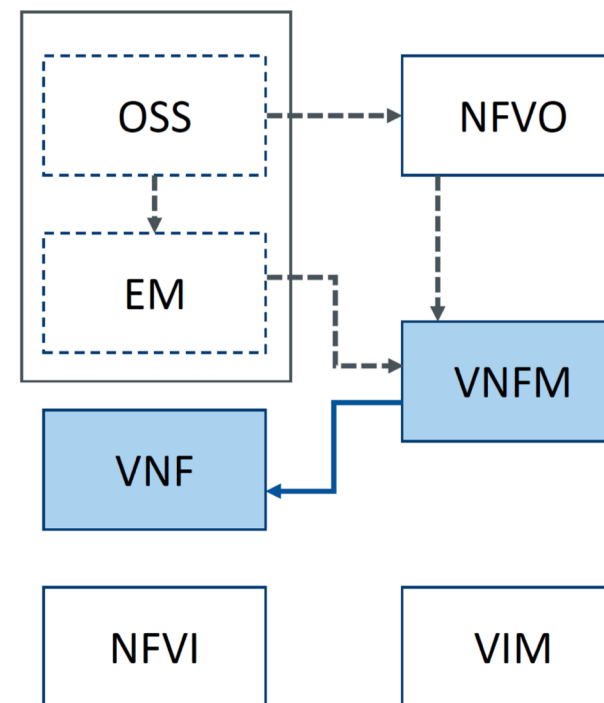
Relies on the support of the optional **VNF Configuration** interface (push mode) or on VNF LCM notifications followed by a **Query VNF** operation (Pull Mode) defined in ETSI GS NFV-IFA 008.

Non-virtualisation-related configuration data

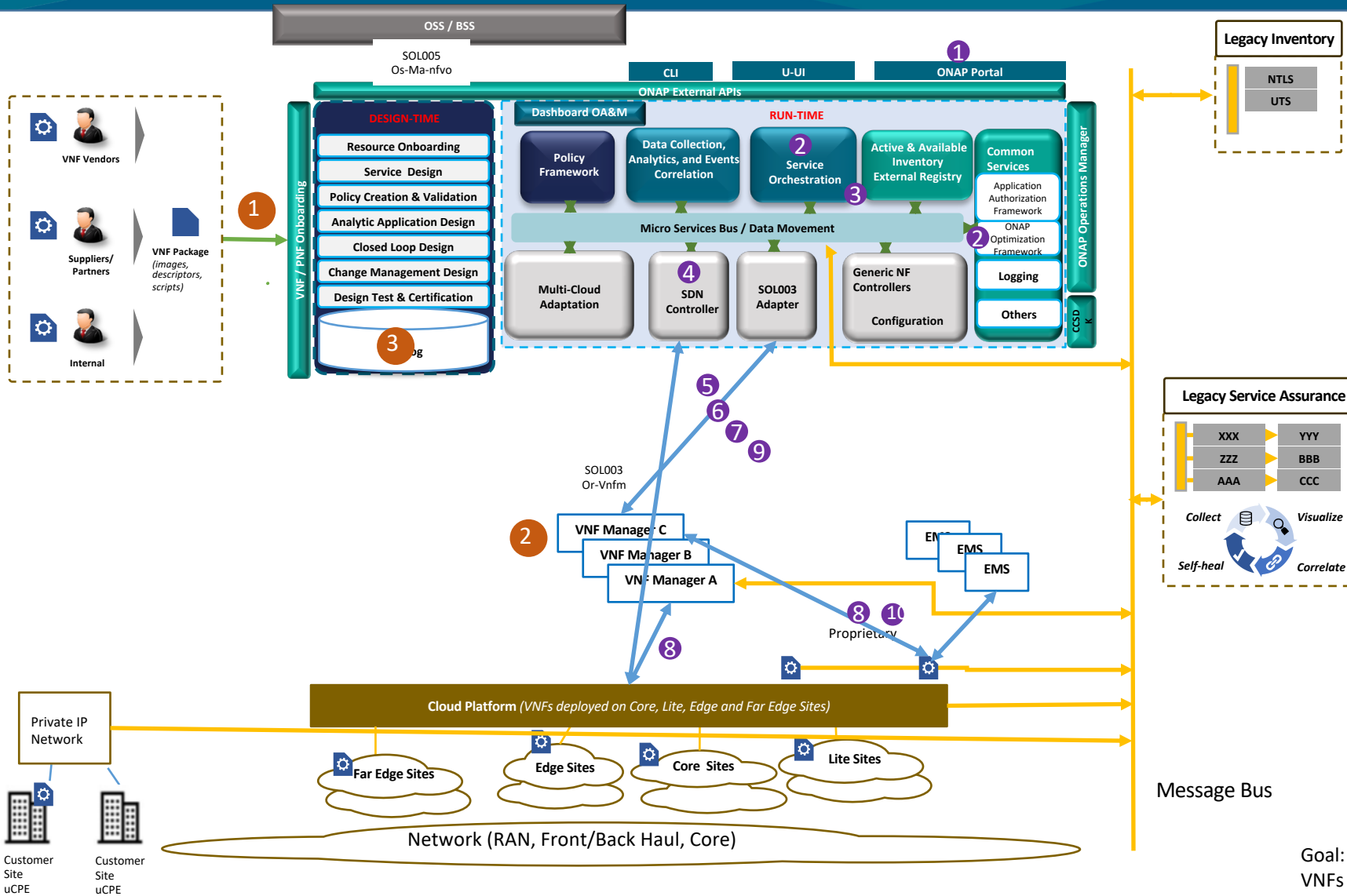
- Relies on **the configurable properties** declared in the VNFD
- NFVO and VNFM are used as a “tunnelling” mechanism between the OSS and the VNF Application

Virtualisation-related configuration data

- Relies on the **configurable properties** declared in the VNFD and/or on pre-defined parameters available in **VnfInfo** (DHCP server address to use, Addresses and ports assigned to the Connection Points)



ONAP with Vendor VNFM, EMS, SA and Inventory #1d (VNF deployment with Application configuration via SOL003 ModifyVnflInfo interface)



Design/Develop Time:

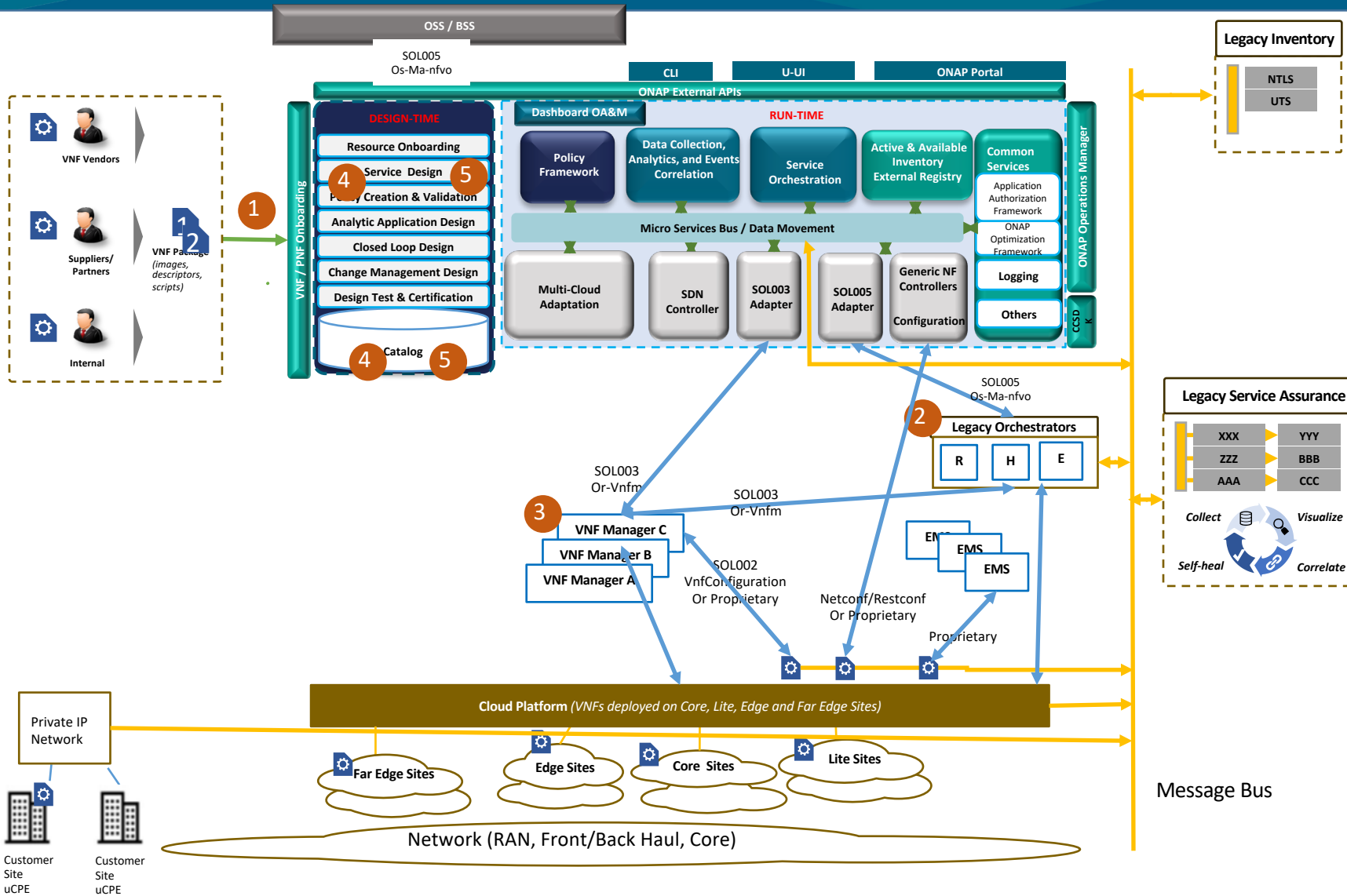
- Onboard and catalog a SOL004 VNF package with a SOL001 VNFD.
 - Application level *ConfigurableProperties* attributes included in the VNF-D from the vendor
 - Could also be provide in a separate artifact in the SOL004 package
- VNFM C registers as a SOL003 compliant VNFM
 - Support "ModifyVnflInfo" 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&AI
- ONAP makes assignments (IP Address) for the VNF
- 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.
- ONAP decomposes service config data for this VNF and updates the application configuration data over SOL003 ModifyVnflInfo interface with VNFM C
- VNFM C updates the VNF via a proprietary mechanism

Goal: Design and deploy a composite Service consisting of multiple vendor VNFs in an automated and repeatable way leveraging vendor provided SOL003 compliant VNFM(s) and ONAP Application configuration.

ONAP with Legacy Orchestration, EMS, SA and Inventory #3a (Service deployment with VNF (with ONAP App config) and NS deployment) (Design Time)

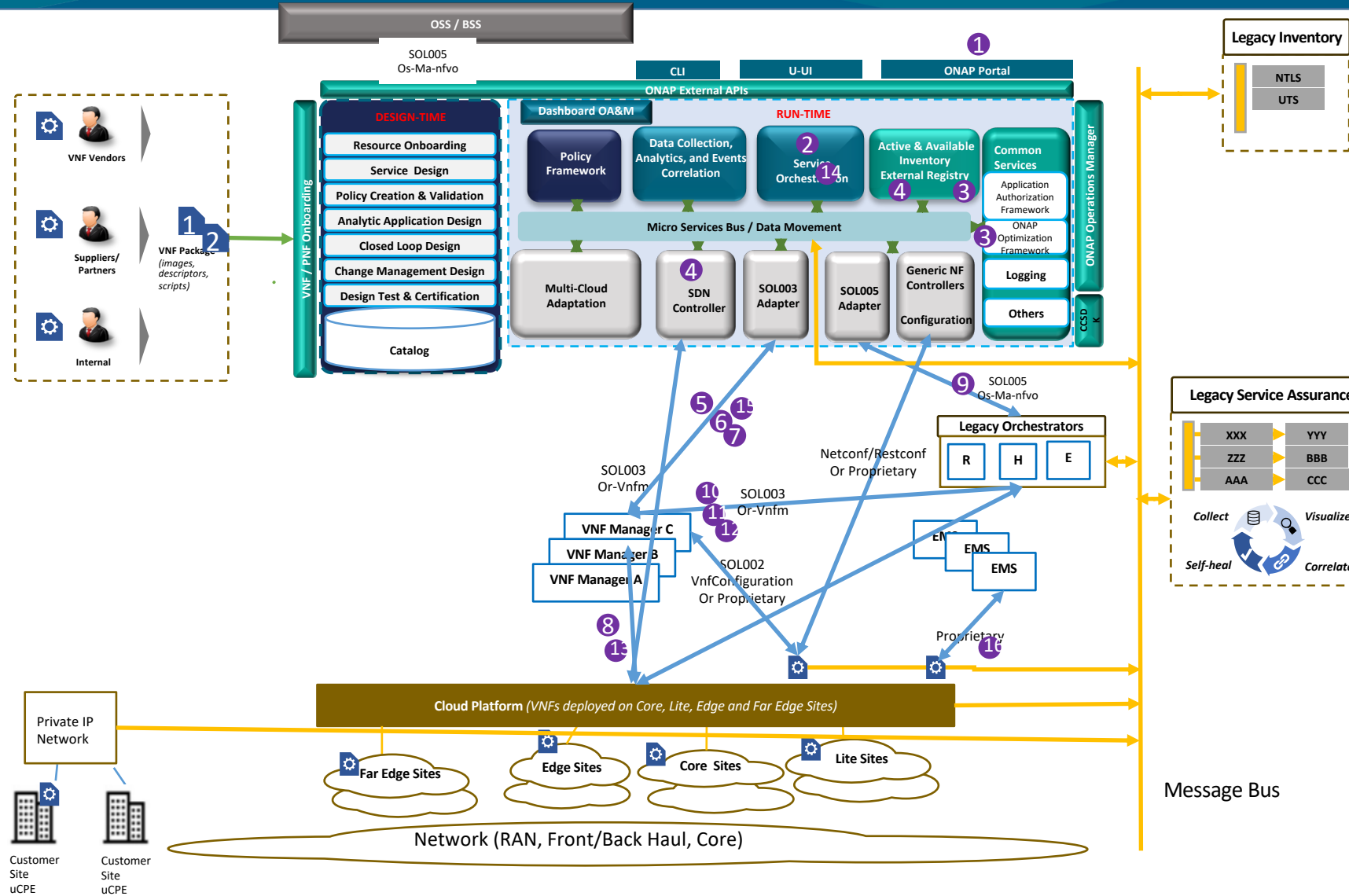


Design/Develop Time:

1. Onboard SOL004 VNF packages with a SOL001 VNFD.
 - a) Application level ConfigurableProperties included in the VNFD from the vendor
 - b) Could also be provide in a separate artifact in the SOL004 package
2. NFVO R registers as a SOL005 compliant NFVO
3. VNF M C registers as a SOL003 compliant VNF M
 - a) Supporting "ModifyVnfInfo" operation
4. Design an ONAP Service A referencing VNF 1 and and NFVO R.
 - a) ConfigurableProperties extracted from the VNFD or alternate artifact and composed into Service level configuration
5. Design an ONAP Service B with deployment and application configuration referencing VNF 2 and Service A.
 - a) ConfigurableProperties extracted from the VNFD or alternate artifact along with Service A configuration properties and composed into Service B configuration

Goal: Leverage ONAP VNF onboarding and Service Design capabilities to develop a hierarchical composite service parts of which can be directly instantiated by a VNF M along with a subservice that can be instantiated and managed by an existing SOL005 compliant orchestrator.

ONAP with Legacy Orchestration, EMS, SA and Inventory #3b (Service deployment with VNF with ONAP App config and NS deployment) (Run Time)



Run Time:

- ONAP receives request to create an instance of Service B.
- ONAP "decomposes" the request into an NS and a VNF.
- 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
- ONAP makes assignments (IP Address) for the VNF and the NS
- ONAP selects VNFM C (match type and homing) and calls via SOL003 API, requesting creation of VNF 1, passes "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 creds.
- VNFM C calls VIM to create VNF 1, and applies any needed deployment data via proprietary API.
- ONAP calls NFVO R via SOL005 API, requesting creation of NS, passing configuration values.
- NFVO R calls VNFM C requesting an instance of VNF 2.
- VNFM C upcalls NFVO R asking for a "grant" of resources.
- NFVO R responds with resources and VIM creds.
- VNFM C calls VIM to create VNF 2, and applies any needed deployment data via proprietary API.
- ONAP decomposes service config data for VNF 1
- ONAP updates VNF 1 application configuration data over SOL003 ModifyVnfInfo interface with VNFM C
- NFVO R updates VNF 2 application configuration data over SOL003 ModifyVnfInfo interface with VNFM C.

Goal: Leverage ONAP VNF onboarding and Service Design capabilities to develop a hierarchical composite service, parts of which can be directly instantiated by a VNFM along with a subservice that can be instantiated and managed by an existing SOL005 compliant orchestrator.