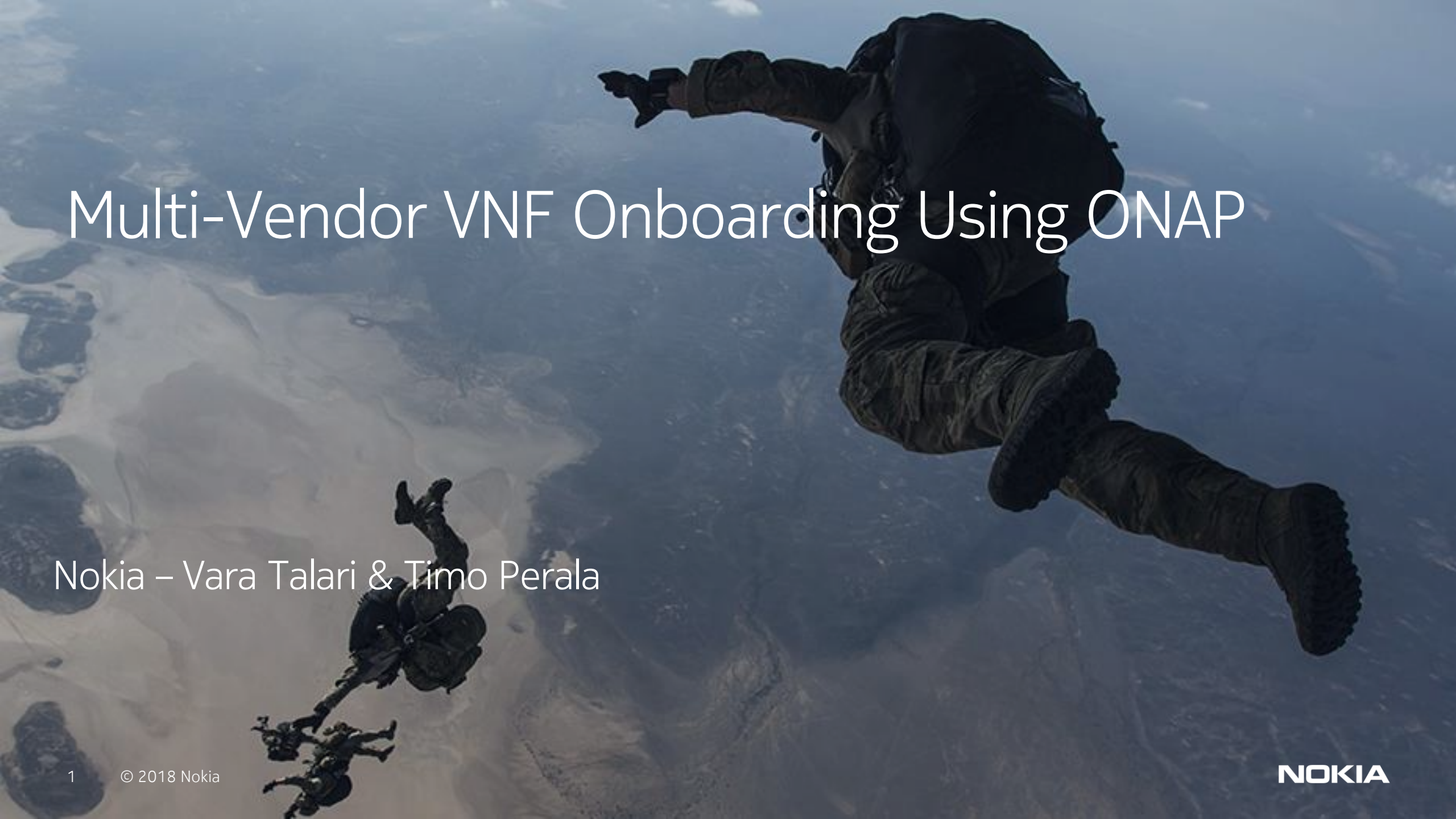
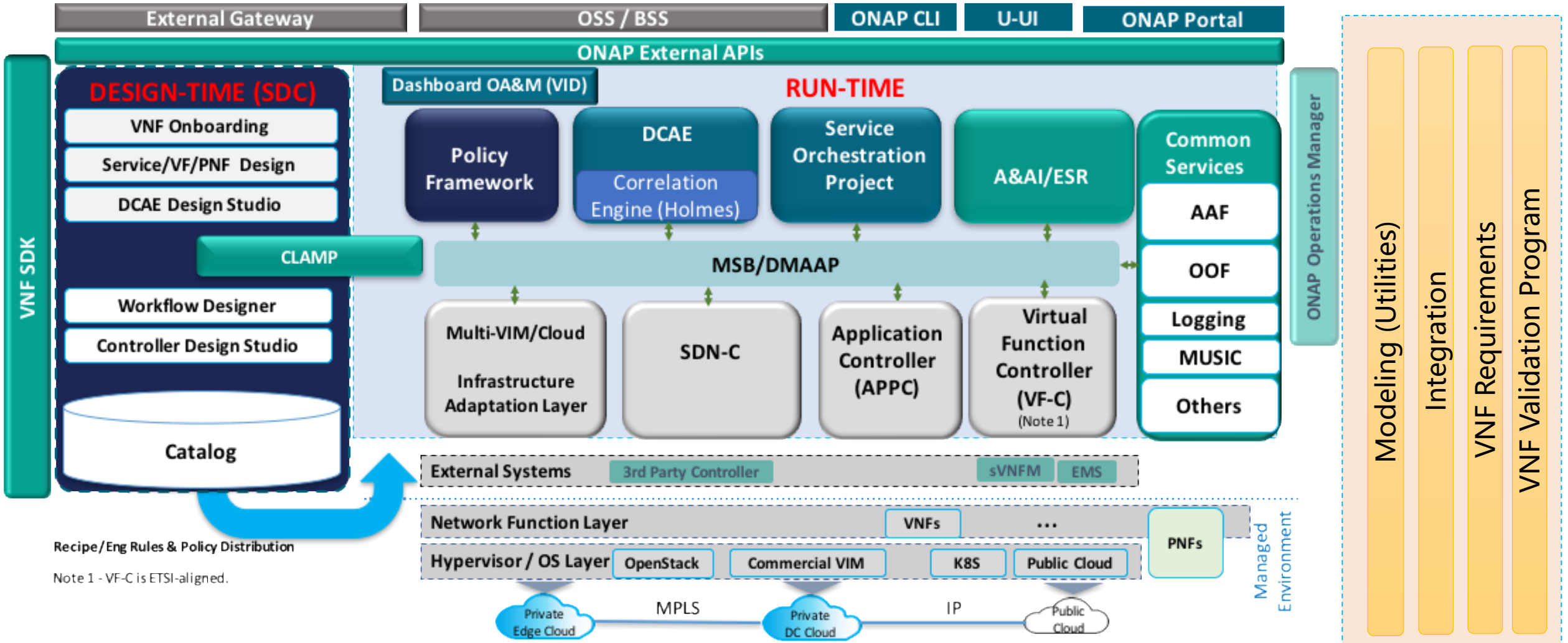


Multi-Vendor VNF Onboarding Using ONAP

Nokia – Vara Talari & Timo Perala

NOKIA

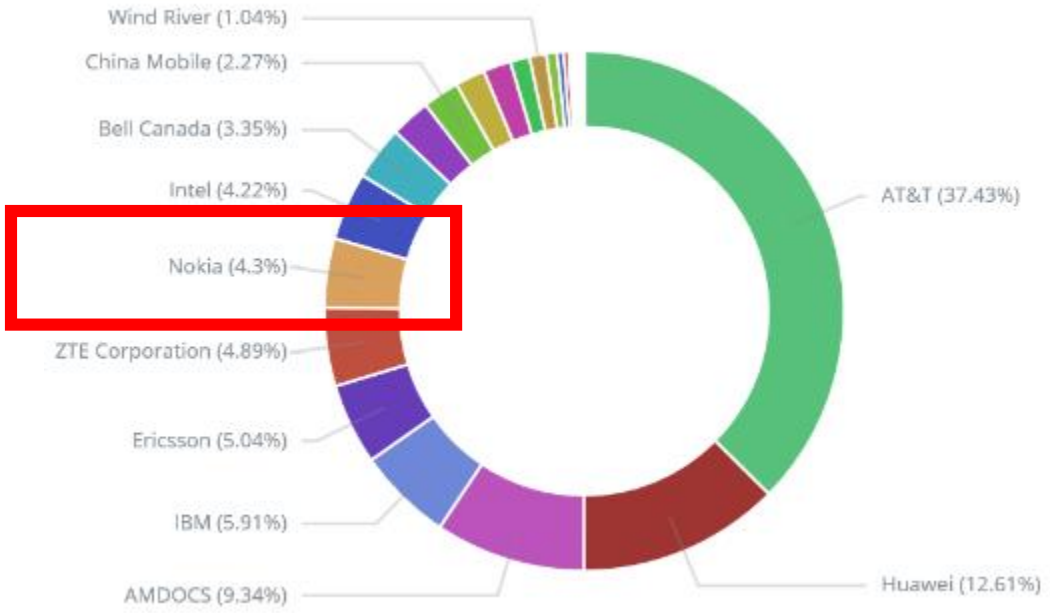
ONAP Casablanca Architecture
(High-Level View with Projects)



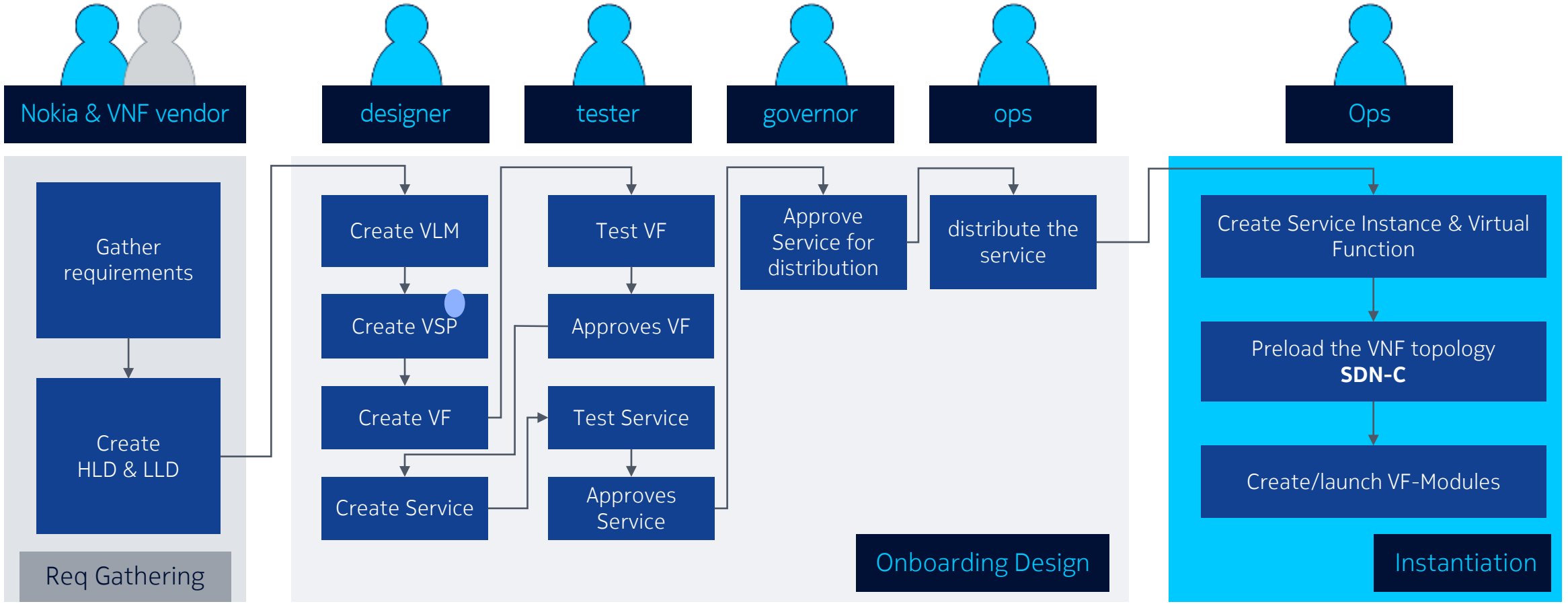
Nokia's ONAP Participation/Contribution

- Nokia among the founding platinum members of ONAP
- Nokia interest in ONAP is multifaceted
  - VNF and PNF vendor
  - Automation, Orchestration and Management Solution provider
  - Services business, e.g. integration, Multivendor VNF deployment (Radisys, Clear water, Nokia etc)
- Nokia main focus areas to date in ONAP
  - 5G use case/ PNF deployment
  - BBS use case
  - Alignment with established standards

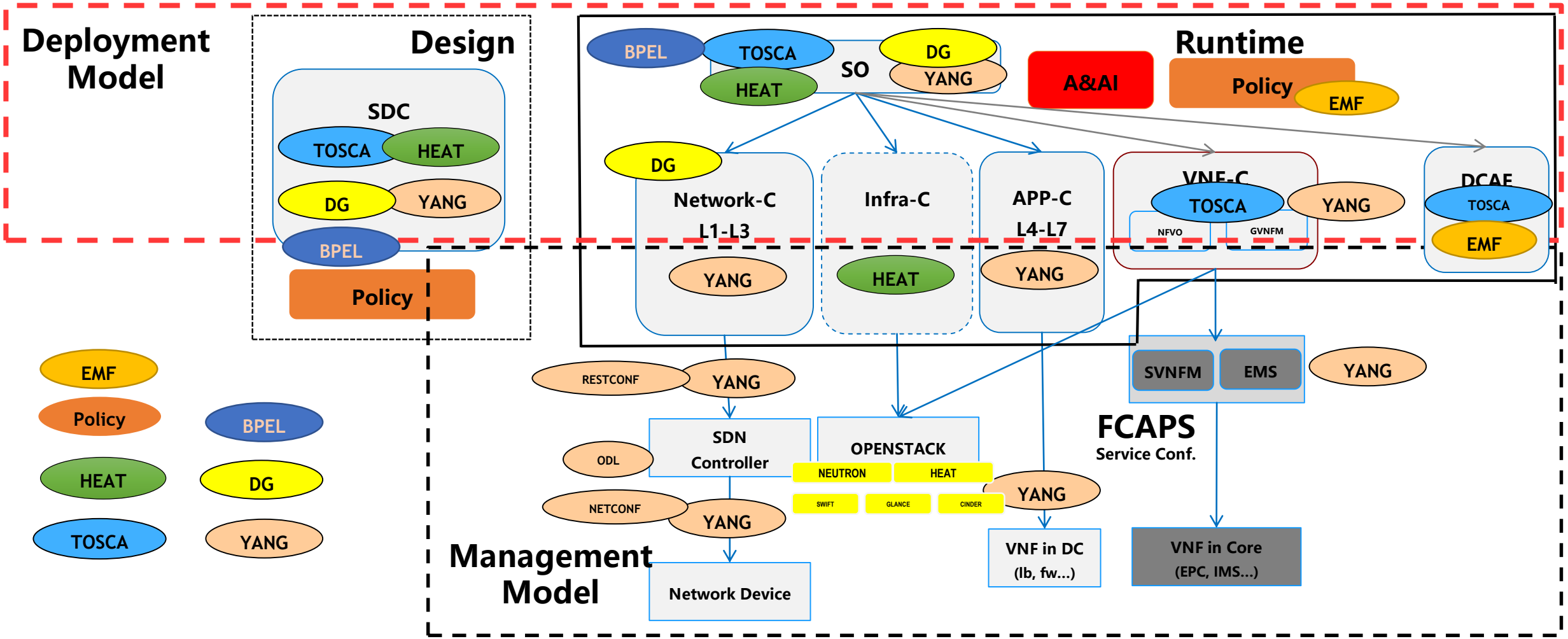
ONAP contributors, as of April 3 2019



VNF Onboarding on ONAP
Process overview

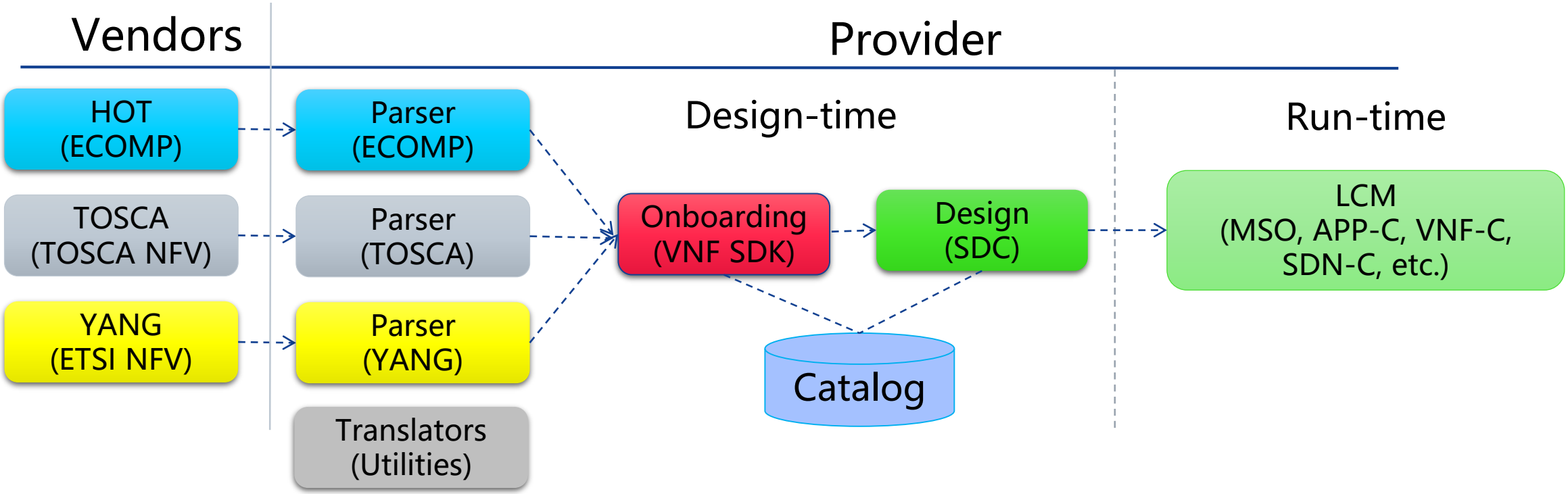


Landscape of modeling in ONAP



NOKIA

VNF Modelling

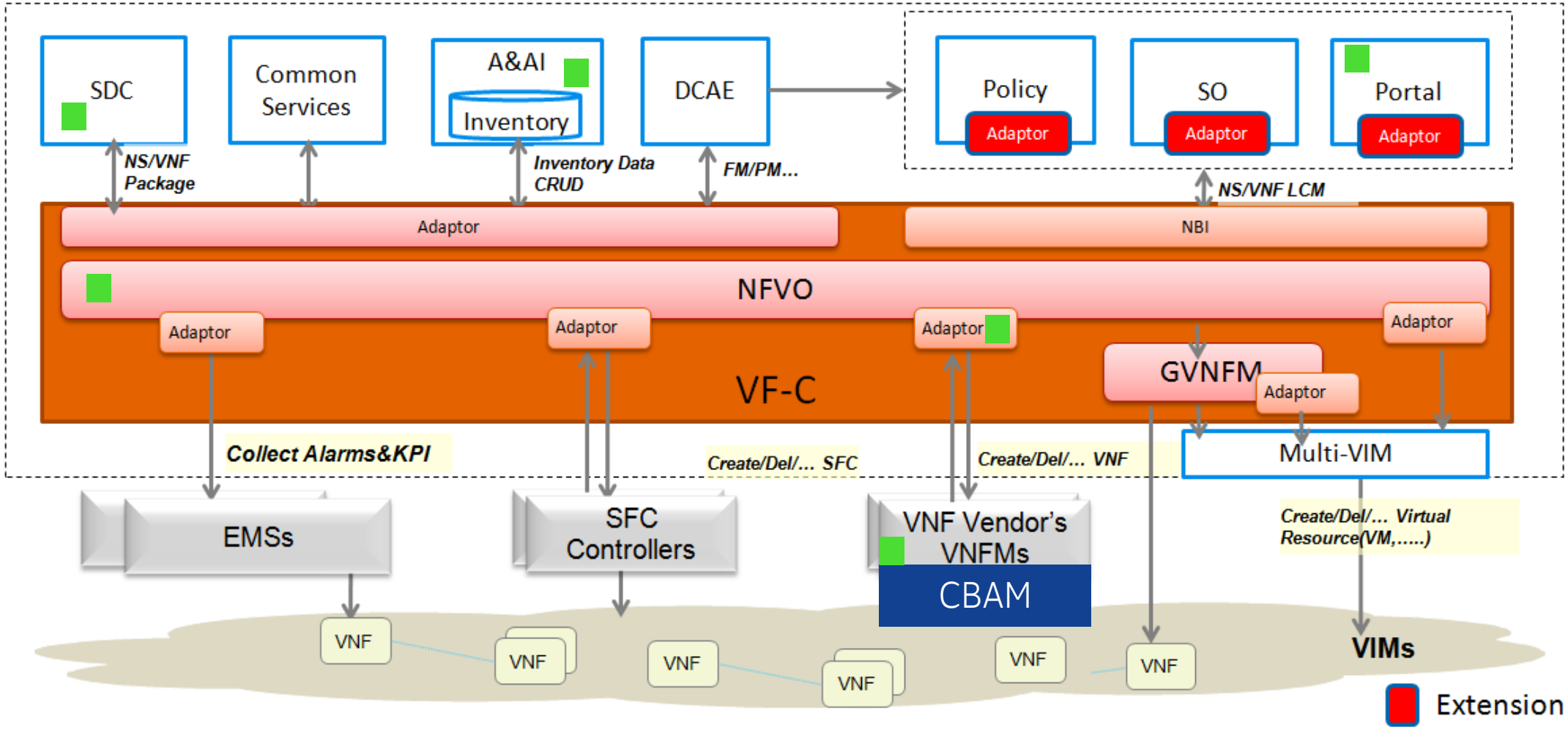


Different DMs are used for VNF templates. ECOMP uses HEAT, OPEN-O uses TOSCA. We will be working together on supporting HEAT, TOSCA and YANG VNF modeling.

The core implementation engine for LCM and operation in run time should be data model agnostic.

6

VF-C architecture overview

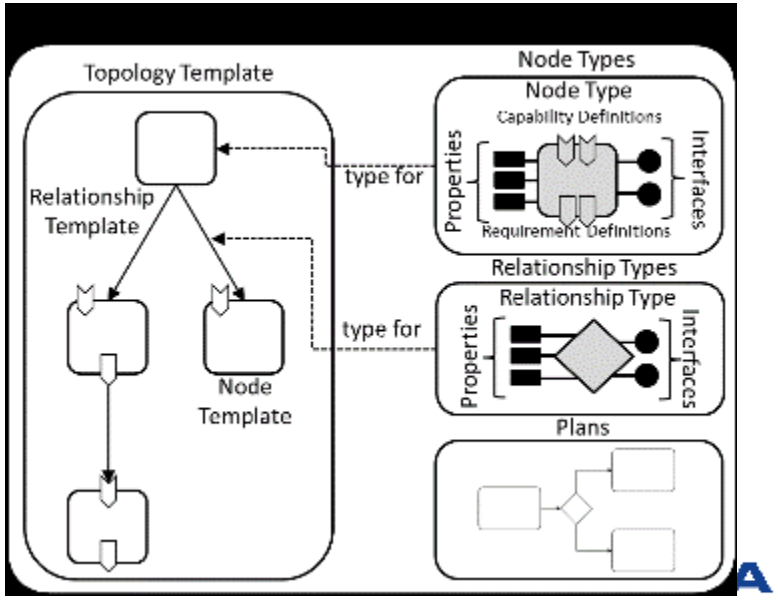


VNF TOSCA Template Requirements for ONAP

- VNF TOSCA template is aligned to the newest TOSCA protocol, “Working Draft 04-Revision 06”.
- VNF TOSCA template supports HPA features, such as NUMA, Hyper Threading, SRIOV, etc.

- In the ONAP, VNF Package and VNFD template can be designed by manually or via model designer tools.
- The VNF package structure is align to the NFV TOSCA protocol, and supports CSAR
- The VNFD and VNF package are all align to the NFV TOSCA protocol, which supports multiple TOSCA template yaml files, and also supports self-defined node or other extensions.

A Topology Template consists of a set of Node Templates and Relationship Templates that together define the topology model of a service as a (not necessarily connected) directed graph.
A node in this graph is represented by a Node Template.



Reference:
https://onap.readthedocs.io/en/beijing/submodules/vnfrqts/requirements.git/docs/Chapter5.html

VNF TOSCA Template Requirements for ONAP

- VNF topology: it is modeled in a cloud agnostic way using virtualized containers and their connectivity. Virtual Deployment Units (VDU) describe the capabilities of the virtualized containers, such as virtual CPU, RAM, disks; their connectivity is modeled with VDU Connection Point Descriptors (VduCpd), Virtual Link Descriptors (Vld) and VNF External Connection Point Descriptors (VnfExternalCpd);

- VNF deployment aspects: they are described in one or more deployment flavours, including instantiation levels, supported LCM operations, VNF LCM operation configuration parameters, placement constraints (affinity / antiaffinity), minimum and maximum VDU instance numbers, and scaling aspect for horizontal scaling.
- SR-IOV Passthrought :Definitions of SRIOV_Port are necessary if VDU supports SR-IOV.
- Hugepages: Definitions of mem_page_size as one property shall be added to Properties and set the value to large if one VDU node supports huagepages.
- NUMA (CPU/Mem): We add definitions of numa to requested_additional_capabilities if we wand VUD nodes to support NUMA.
- Hyper-Theading: Definitions of Hyper-Theading are necessary as one of requested_additional_capabilities of one VUD node if that node supports Hyper-Theading.
- OVS+DPDK: Definitions of ovs_dpdk are necessary as one of requested_additional_capabilities of one VUD node if that node supports dpdk.

tosca.nodes.nfv.VDU.Compute

The NFV Virtualization Deployment Unit (VDU) compute node type represents a VDU entity which it describes the deployment and operational behavior of a VNF component (VNFC)

tosca.nodes.nfv.VDU.VirtualStorage

The NFV VirtualStorage node type represents a virtual storage entity which it describes the deployment and operational behavior of a virtual storage resources

Reference : https://onap.readthedocs.io/en/beijing/submodules/vnfrqts/requirements.git/docs/Chapter5.html

TOSCA Template updates for VNF deployment via VF-C /CBAM (VNFM)

1. The VNF must declare the externalVnfmId and onapCsarId as modifyable attribute in CBAM package. Each should have a default value. (The concrete value will be filled out by CBAM)
2. Each operation must declare a jobId additional parameter in CBAM package (value will be filled out by CBAM)
3. The heal operation must declare the jobId, vmName, vnfcId and action parameters in CBAM package (values will be filled out by CBAM)
4. Each operation (including built-in) must include the following section as the last pre_action (all JS are provided by CBAM)

javascript: javascript/cbam.pre.collectConnectionPoints.js include: - javascript/cbam.collectConnectionPoints.js output: operation_result

CBAM supplied JavaScripts

cbam.post.collectConnectionPoints.js

cbam.pre.collectConnectionPoints.js

cbam.collectConnectionPoints.js

Reference:
https://docs.onap.org/en/latest/submodules/vfc/nfvo/driver/vnfm/svnfm.git/nokiav2/docs/vnfintegration.html

TOSCA Template updates for VNF deployment via VFC

The ONAP package must be written so that the VDU.Compute, VDU.VirtualStorage, VnfVirtualLinkDesc, VduCpd has exactly the same name as in CBAM package
the metadata section of the ONAP package must be the following:

- the vendor must be the same as in Nokia package vendor field
- the vnfdVersion must be the same as in Nokia package the descriptor_version field
- the name must be the same as in Nokia package the product_info_name field
- the version must be the same as in Nokia package the software_version field
- the vnfmType must be NokiaSVNFM

The complete CBAM package must be placed in the in Artifacts/OTHER/cbam.package.zip file

Convertor:

Visit http://msb.api.simpledemo.onap.org/api/NokiaSVNFM/v1/convert

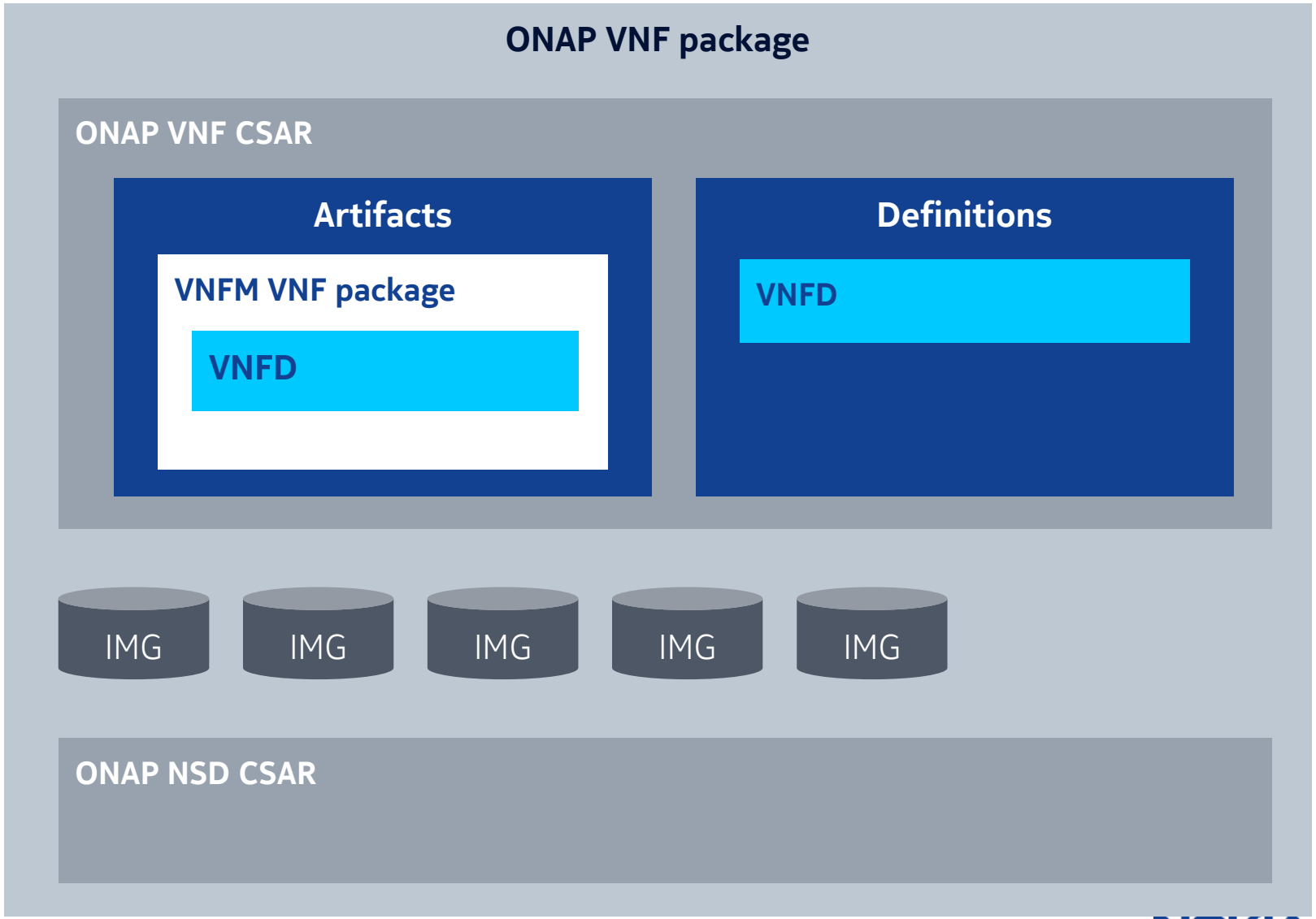
Select the CBAM package to be converted into an ONAP package

Click on upload button and the ONAP package will be downloaded

Reference: https://docs.onap.org/en/latest/submodules/vfc/nfvo/driver/vnfm/svnfm.git/nokiav2/docs/vnfintegration.html

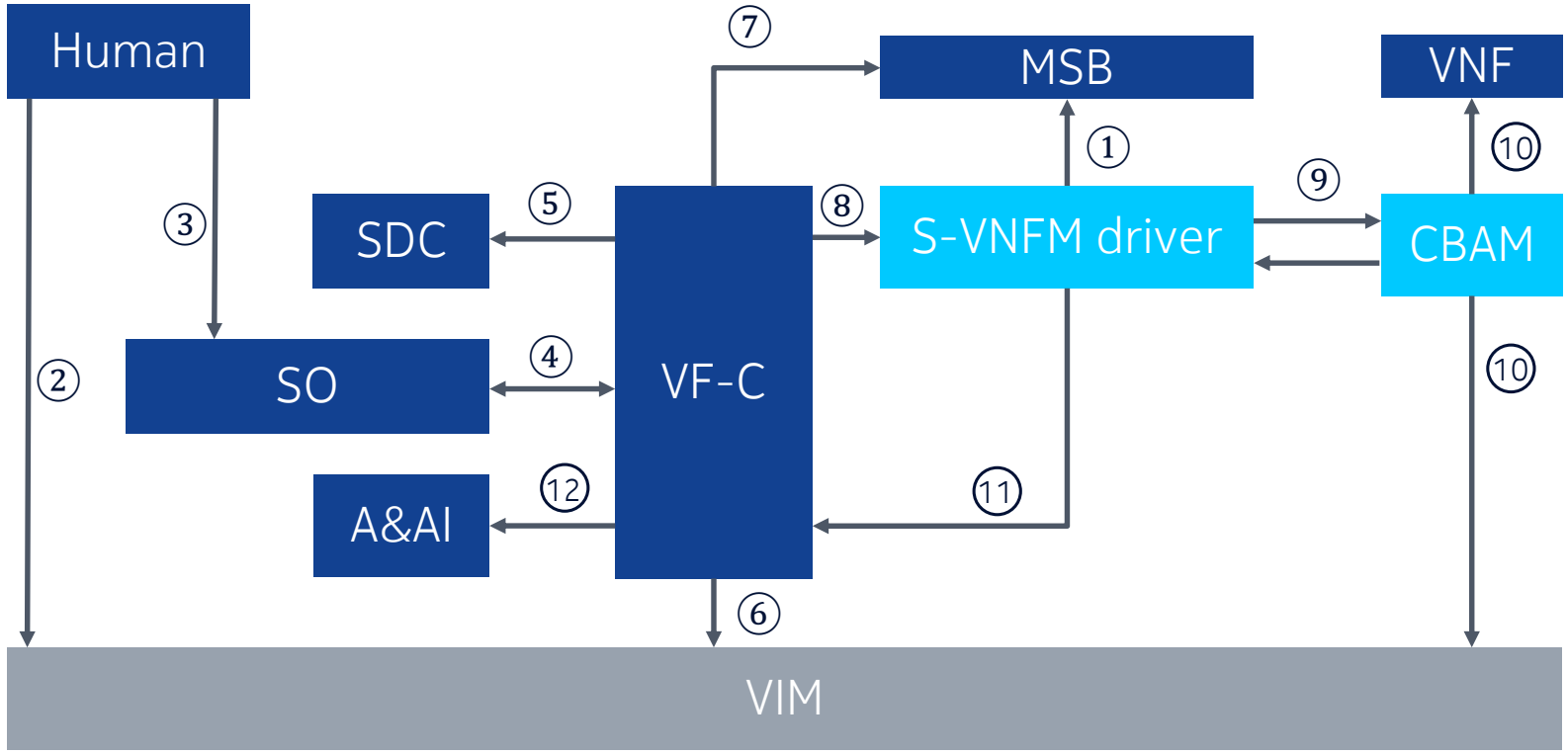
ONAP VNF package

- Images are not handled by ONAP
  - Manually uploaded to the cloud
  - Linked manually to VNFC
- VNFD embedded in ONAP CSAR
  - Certain elements are duplicated



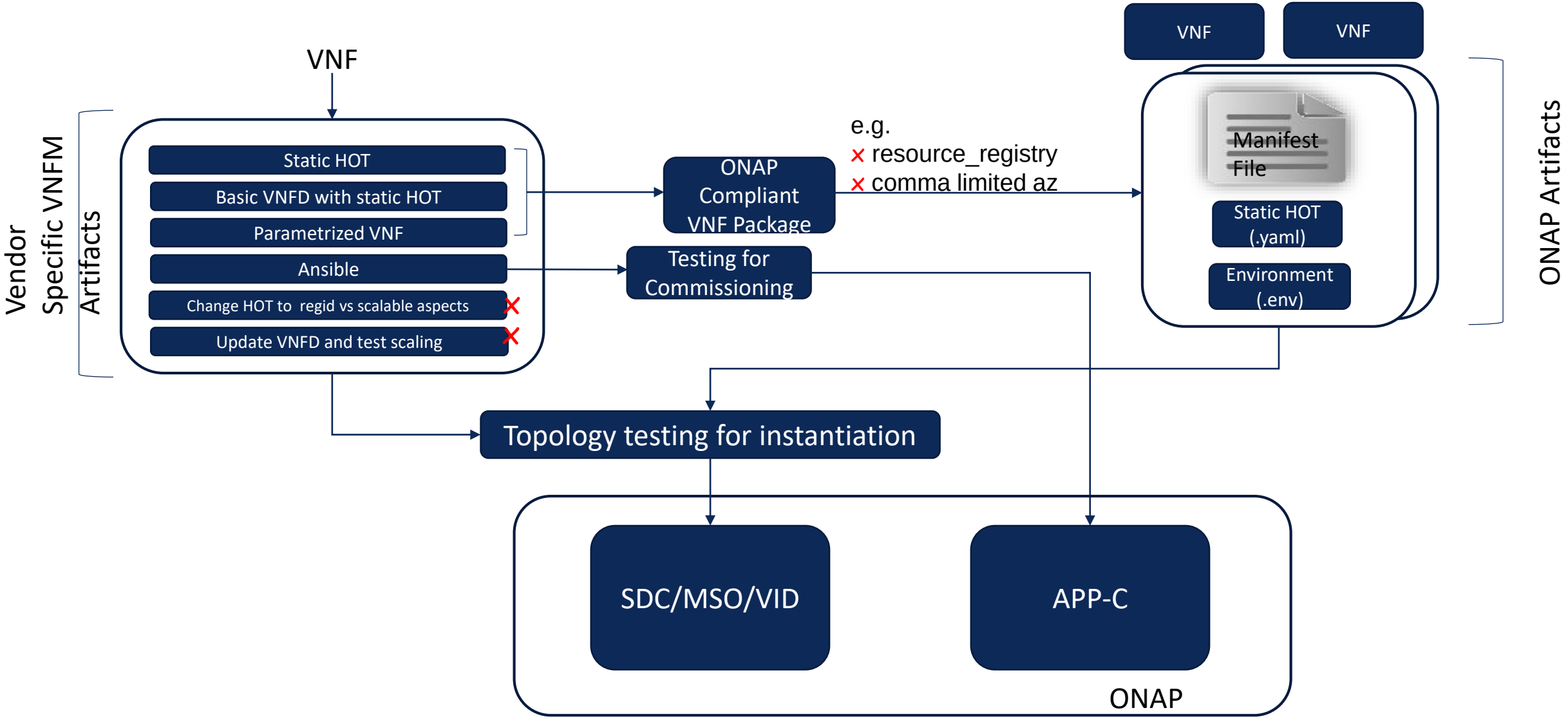
CBAM – ONAP integration high level architecture: component interaction

1. Publish it's existence & APIs
2. Upload images
3. Trigger operation via portal
4. Trigger operation via VF-C NBI
5. Download package
6. Create cross VNF resources
7. Search for Nokia VNFM
8. Trigger operation & track progress via S-VNFM API
9. Trigger operation & track progress via CBAM API
10. Allocate resources, (re)configure VNF
11. Send LCN containing structural changes
12. Modify VNF structure



VNF deployment through ONAP

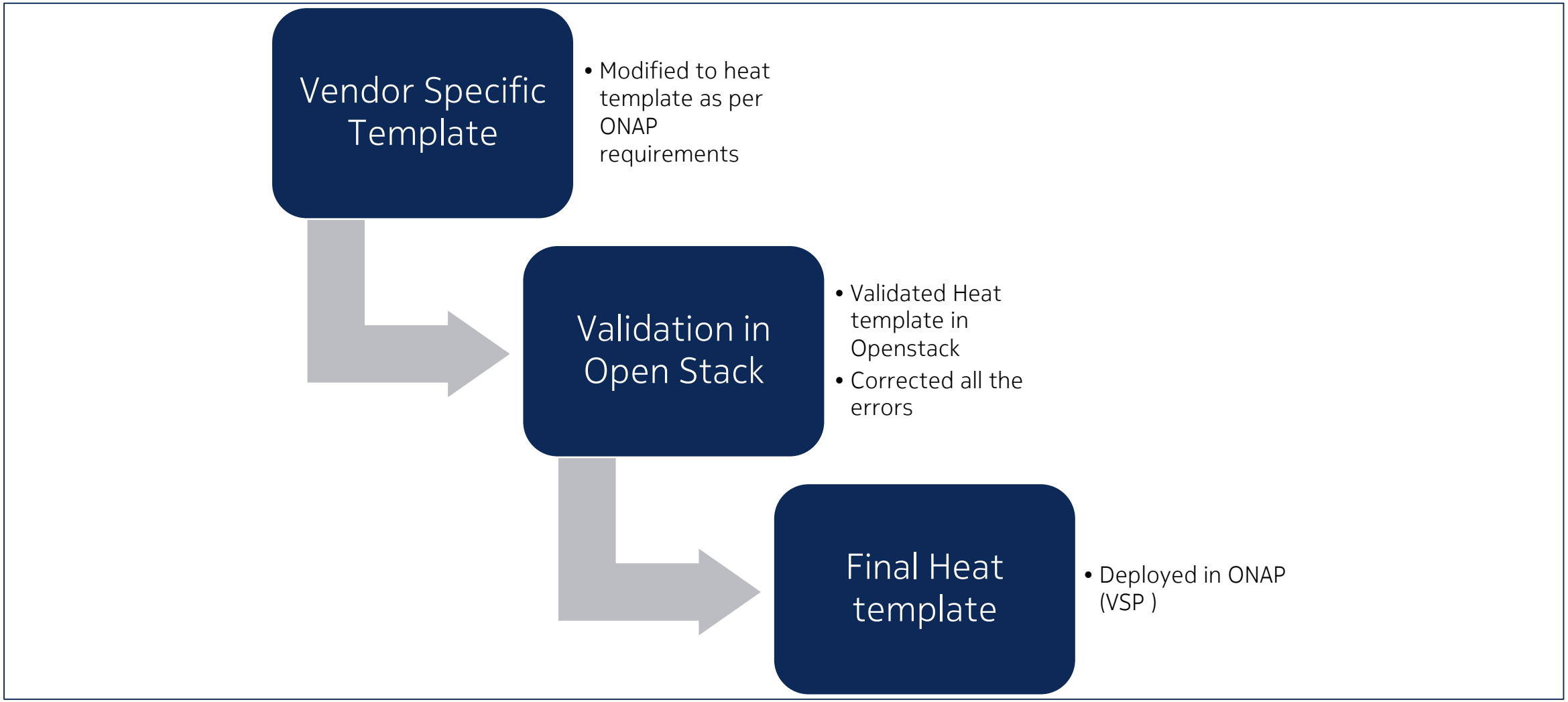
Heat Template Deployment



ONAP VNF Requirements:
https://onap.readthedocs.io/en/beijing/submodules/vnfrqts/requirements.git/docs/Chapter5.html

NOKIA

Customer Specific Templates to ONAP



Updates to Templates for ONAP Compliance

Resource Registry

- Removed vendor specific resource registry

Naming Convention

- Follow Naming convention

Additional Updates

- ResourceGroup does not deal with structured parameters (comma-delimited-list and json) – We replaced json with yaml format.
- Volumes created during runtime
- Internal Networks created during runtime
- External Networks pre-created and specified in env file

Nested config

- Removed Nested config
- Nested Heat is supported by ONAP*

User Data

- Updated User data- Meta Data

Scale-In/Scale – Out and other LCMs

- Currently removed, Handled seperately

Example updates

Example of resource group

```
type: OS::Heat::ResourceGroup
LB1:
   type: ALU::VMG::LB
the above one was for resource registry
MG1:
   type: ALU::VMG::MG
OAMA:
   type: ALU::VMG::OAM
resource_registry:
 ALU::VMG::OAM: VMG_OAM.template.yaml
 ALU::VMG::LB: VMG_LB.template.yaml
 ALU::VMG::MG: VMG_MG.template.yaml
CBAM - OAMA:
   type: ALU::VMG::OAM
```

To ONAP compliance

```
oam_server_0:
   type: OS::Nova::Serve
ONAP mg_work_server_0:
   type: OS::Nova::Server
```

Run time Volume creation

```
volume_attachment:
   type: OS::Cinder::VolumeAttachment
   properties:
     volume_id: { get_param: cinder_id }
     instance_uuid: { get_attr: [necc0, instance_id] }
     mountpoint: /dev/vdj
```

Naming Convention:

```
oamACompact Flash1:
   type: OS::Cinder::Volume
   properties:
     name: { list_join: [ "-", [ { get_param: [ cbam, vnfId ] }, OAM-A-CompactFlash1 ] ] }
     size:  { get_param: [ cbam, extensions, oamCompactFlash1Size ] }
oam_comp_flash_01:
   type: OS::Cinder::Volume
   properties:
    name: { list_join: [ "-", [ { get_param: vnf_id }, OAM, { get_param: oamSlotId_0 },
CompactFlash1 ] ] }
    size: 2
```

User Data:

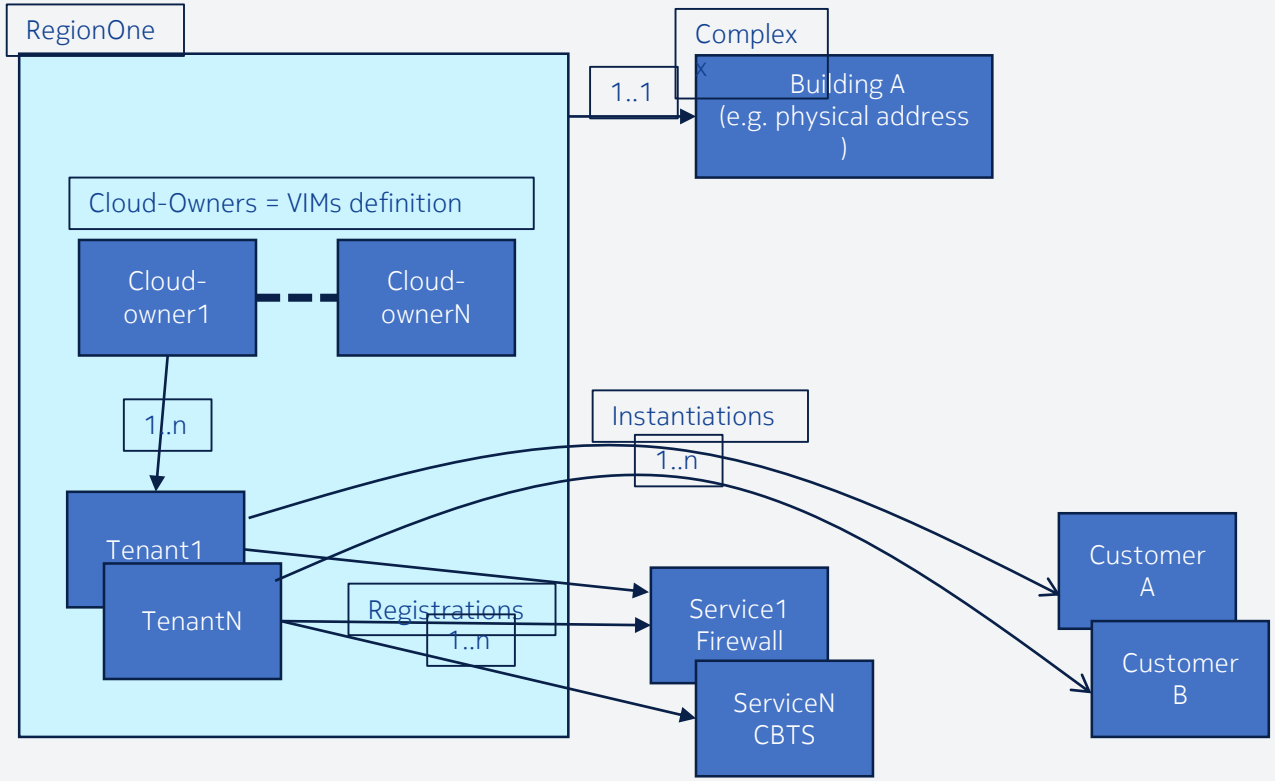
```
user_data:
      str_replace:
        template: "$oamSmbios"
        params:
          $oamSmbios: { get_param: [ resources, smbios ] }
    user_data_format: "RAW“
user_data:
      str_replace:
        template: "$smbios1"
        params:
          $smbios1: { get_param: smbios_1 }
    user_data_format: RAW\
```

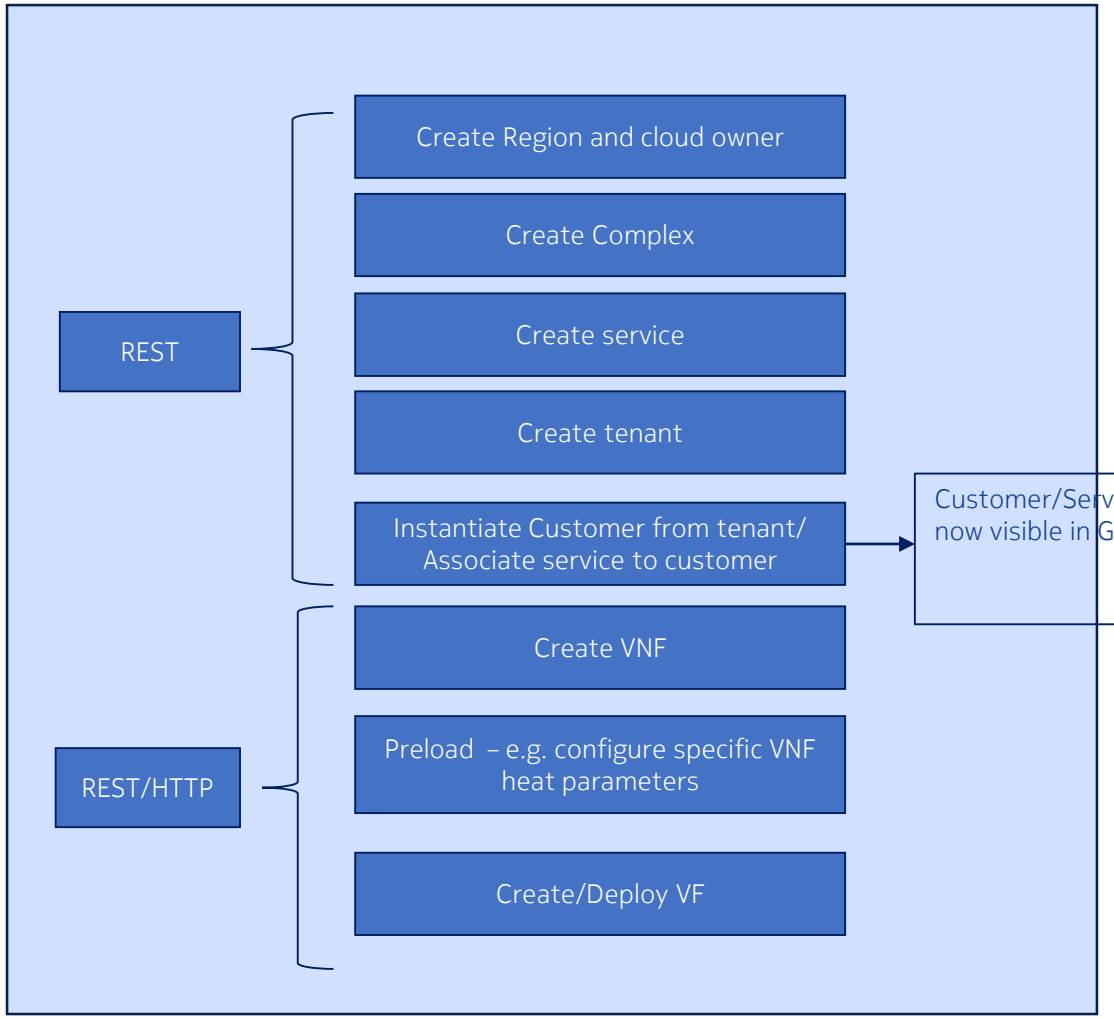
base_cmg_cp.env

base_cmg_cp.yaml

Nokia	vHSS	vCSCF	vMME	vSAE-GW
Radisys	vMRF			
Clearwater	vIMS			

Deployment of the Service/VNF





Service instance

Action	UUID	Invariant UUID	Name	Version	Category	Distribution Status	Last Updated By	Tosca Model
Deploy	e3cce766-d901-41e8-aa8d-1f890fa5c151	4f3c45f0-da9e-4774-926d-a92407942eca	demoVFW	1.0	Network L1-3	DISTRIBUTED	jm0007	
Deploy	76229bf3-26d0-45f3-92cd-9ecb59107ef4	95105c19-615f-435b-ad91-80d933f84435	demoVLB	1.0	Network L1-3	DISTRIBUTED	jm0007	
Deploy	12171814-7a33-46eb-b00d-df5749744e53	35d01ae9-be5a-42e8-b78f-43d3639a536d	Service	1.0	Network L4+	DISTRIBUTED	jm0007	

Status: COMPLETE - Service Instance has been created successfully.

100 %


```
05/26/17 17:35:46 HTTP Status: OK (200)
{
  "request": {
    "requestId": "7c02df8c-8f6f-4e7a-ac3a-cd426473a7f5",
    "startTime": "Thu, 25 May 2017 21:35:34 GMT",
    "requestScope": "service",
    "requestType": "createInstance",
    "requestDetails": {
      "modelInfo": {
        "modelCustomizationName": null,
        "modelInvariantId": "35d01ae9-be5a-42e8-b78f-43d3639a536d",
        "modelType": "service",
        "modelNameVersionId": "12171814-7a33-46eb-b00d-df5749744e53",
        "modelName": "Service",
        "modelVersion": "1.0"
      },
      "requestInfo": {
        "billingAccountNumber": null,
        "callbackUrl": null,
        "correlator": null,
        "orderNumber": null,
        "productFamilyId": null,
        "orderVersion": null,
        "source": "VID",
        "instanceName": "DemoInstance",
        "suppressRollback": false
      },
```


Close

Create Service Instance

Service Name:	service
Service Invariant UUID:	dd322b17-d21f-4ec7-9ee6-6e76736952e7
Service Version:	1.0
Service UUID:	55cf72d4-ec11-41cc-8001-48fe6b1a4ae9
Service Description:	service
Service Category:	Network L4+

User Provided Data (* indicates required field)

Instance Name: *	DemoInstance
Subscriber Name: *	Select Subscriber Name
Service Type: *	
Suppress Rollback on Failure:	

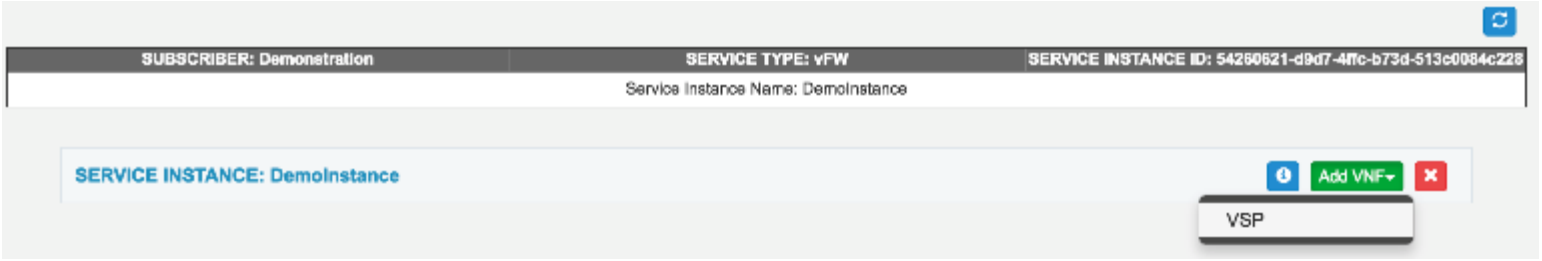
Enter Data and **Confirm** to Create **Service Instance**

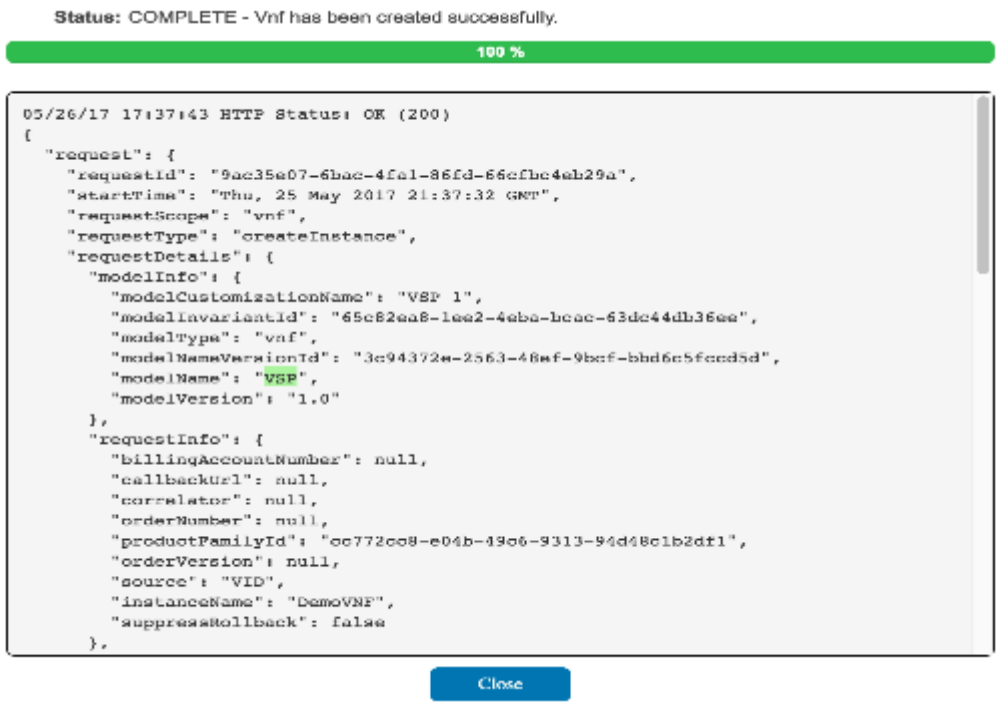
Cancel to Return to Previous Page.
Data entered will be lost

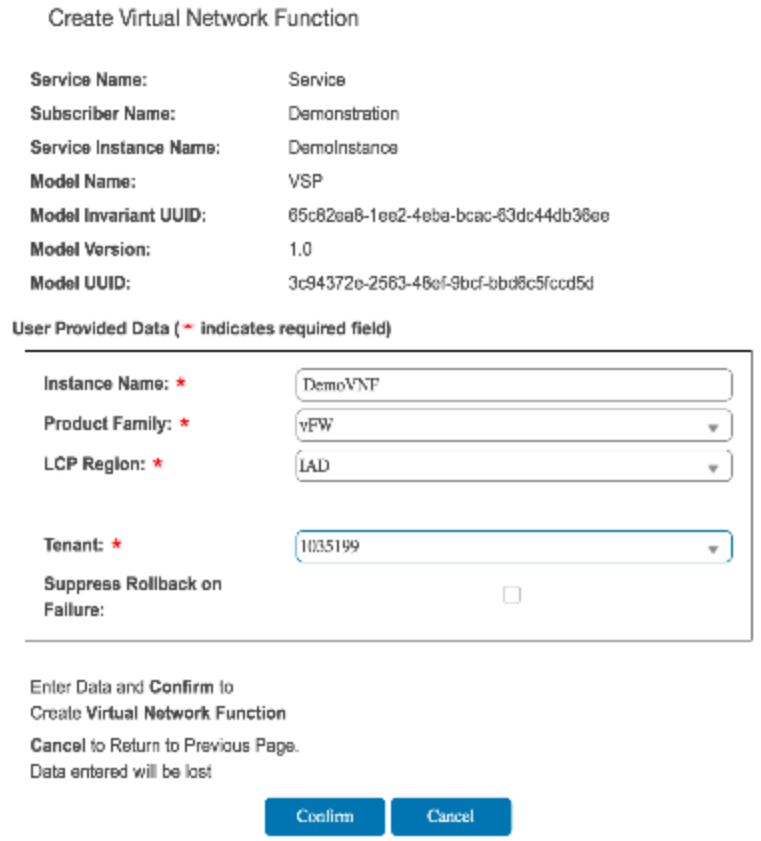
Confirm Cancel

NOKIA

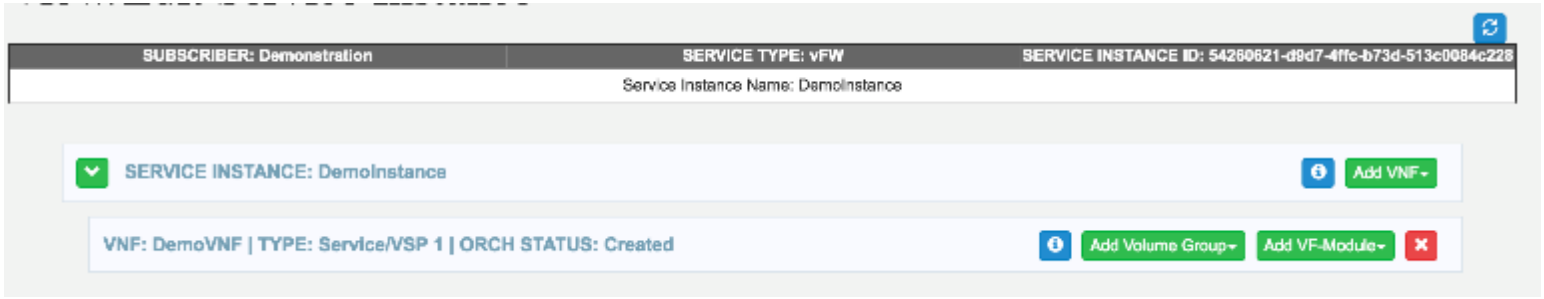
VNF Deployment

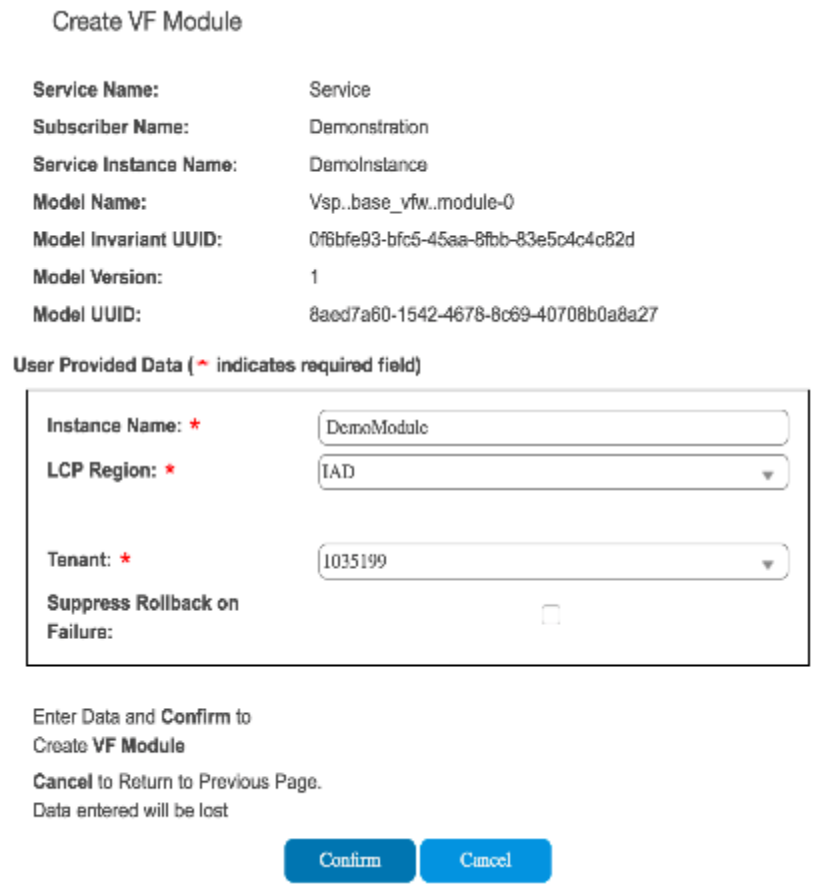




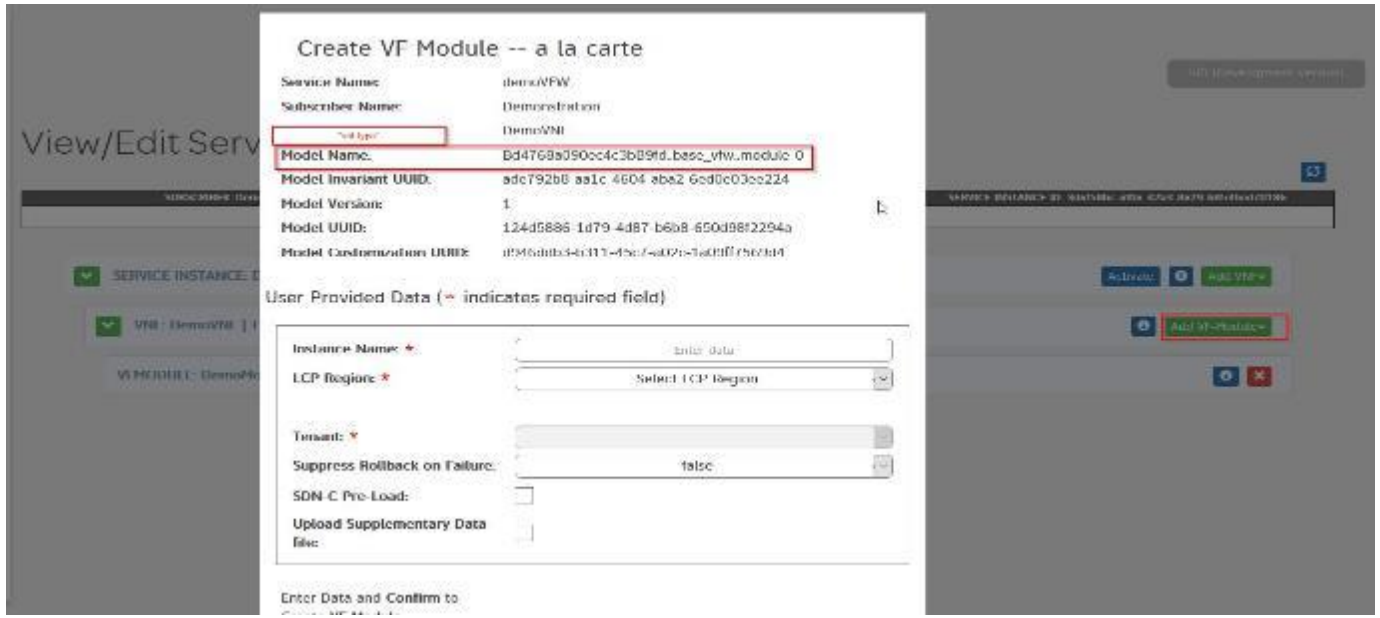


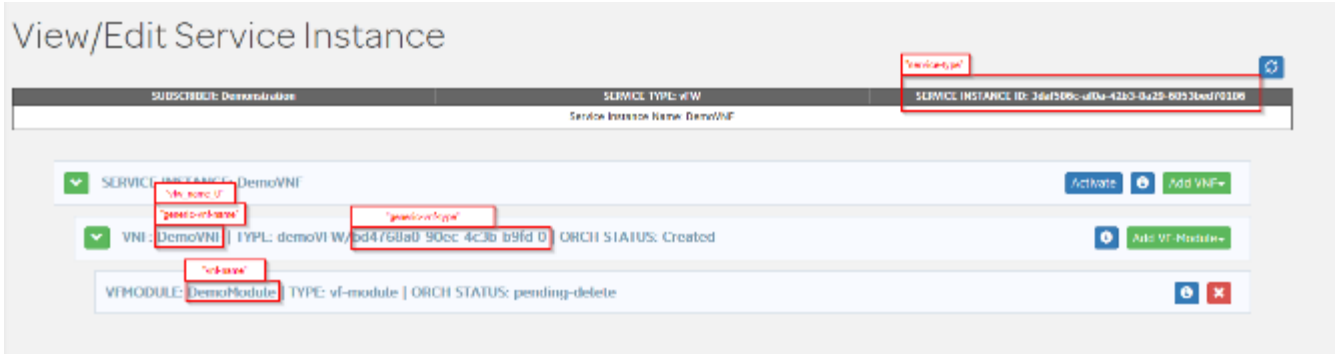
VF Deployment





VF Deployment





Topology Pre-Load
Post Script

POST /operations/VNF-API:preload-vnf-topology-operation

Response Class

Model | Model Schema

(preload-vnf-topology-operation)output-TOP {
  VNF-API:output (object[(preload-vnf-topology-operation)output], *optional*)
}
(preload-vnf-topology-operation)output {
  VNF-API:response-code (Some response-code, *optional*),
  VNF-API:svc-request-id (Some svc-request-id, *optional*),
  VNF-API:response-message (Some response-message, *optional*),
  VNF-API:ack-final-indicator (Some ack-final-indicator, *optional*)
}

Response Content Type application/json

Parameters

Parameter	Value	Description	Parame Type
(preload-vnf-topology-operation)input-TOP	{"vnf-parameter-name":"repo_url","vnf-parameter-value":"https://nexus.onap.org/content/sites/raw"}], "vnf-topology-identifier": { "service-type":"11ce5bc8-cf1c-4f65-b01f-f3d37df483e5",		body

Topology Pre-load
Post Script

POST http://172.16.1.57:8282/restconf/operations/VNF-API:preload-vnf-topology-operation

X-FromAppId: API client

Authorization: Basic YWRtaW46S3A4Yko0U1hzek0wV1hsaGFrM2VIbGNzZTJnQXc4NHZhb0dHbUp2VXkyVQ==

Accept: application/json

X-TransactionId: 0a3f6713-ba96-4971-a6f8-c2da85a3176e

Cache-Control: no-cache

Postman-Token: e1c8d1ec-4cd9-5744-3ac9-f83f0d3c71d4

Content-Type: application/json


```
{
   "input": {
      "vnf-topology-information": {
         "vnf-topology-identifier": {
            "service-type": "3daf586c-af0a-42b3-8a29-6853bed70186",
            "vnf-name": "DemoModule3",
            "vnf-type": "Vsp..base_vfw..module-0",
            "generic-vnf-name": "DemoVNF",
            "generic-vnf-type": "bd4768a0-90ec-4c3b-b9fd 0"
         },
         "vnf-assignments": {
            "availability-zones": [],
            "vnf-networks": [],
            "vnf-vms": []
```

Updates to Templates for ONAP Compliance

Resource Registry

- Removed vendor specific resource registry

Naming Convention

- Follow Naming convention

Additional Updates

- ResourceGroup does not deal with structured parameters (comma-delimited-list and json) – We replaced json with yaml format.
- Volumes created during runtime
- Internal Networks created during runtime
- External Networks pre-created and specified in env file

Nested config

- Removed Nested config
- Nested Heat is supported by ONAP*

User Data

- Updated User data- Meta Data

Scale-In/Scale – Out and other LCMs

- Currently removed, Handled seperately

Thank you

Please let us know if you have questions

NOKIA