

Comparison of ETSI and ONAP/ECOMP Orchestration Boundaries and Proposed Deployment Options

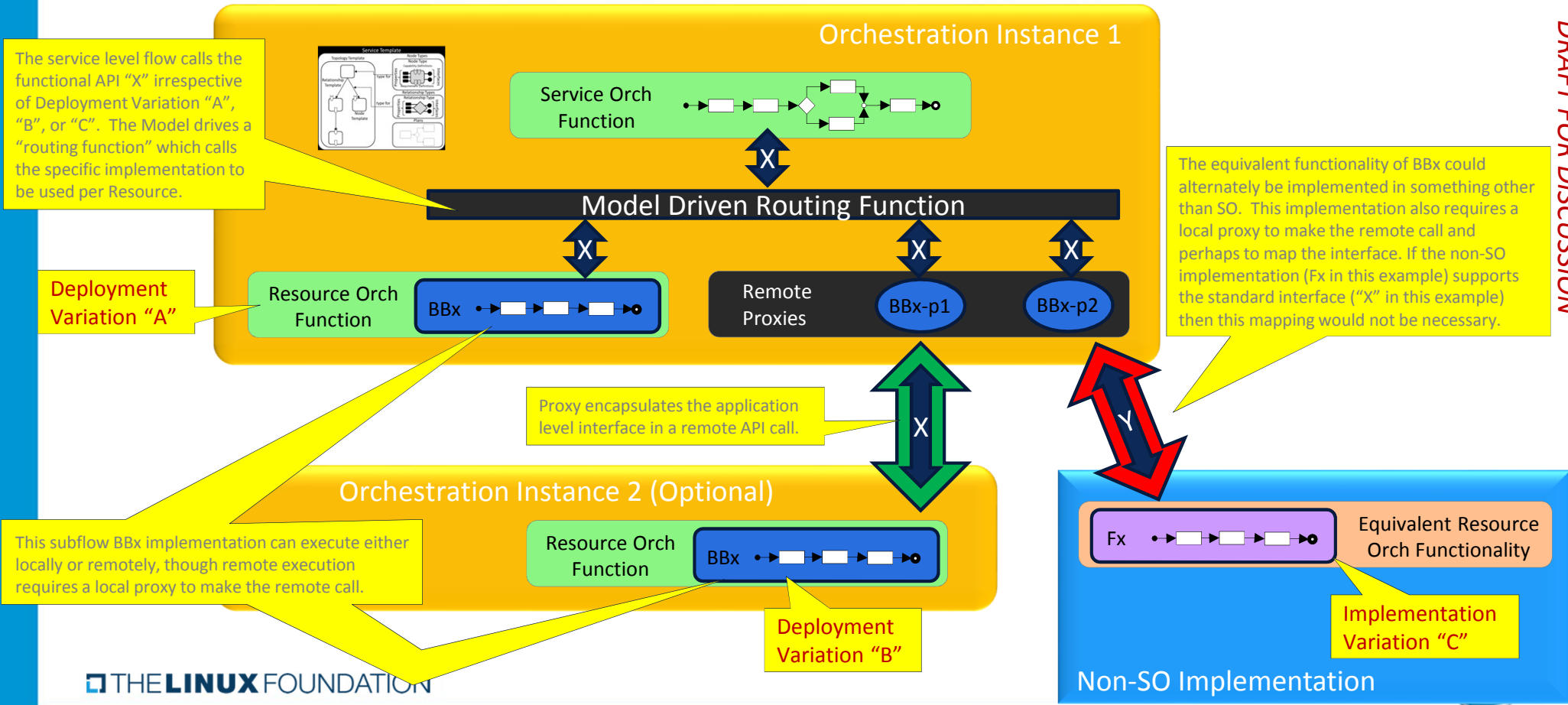
Gil Bullard, AT&T
February 2, 2018

DRAFT FOR DISCUSSION

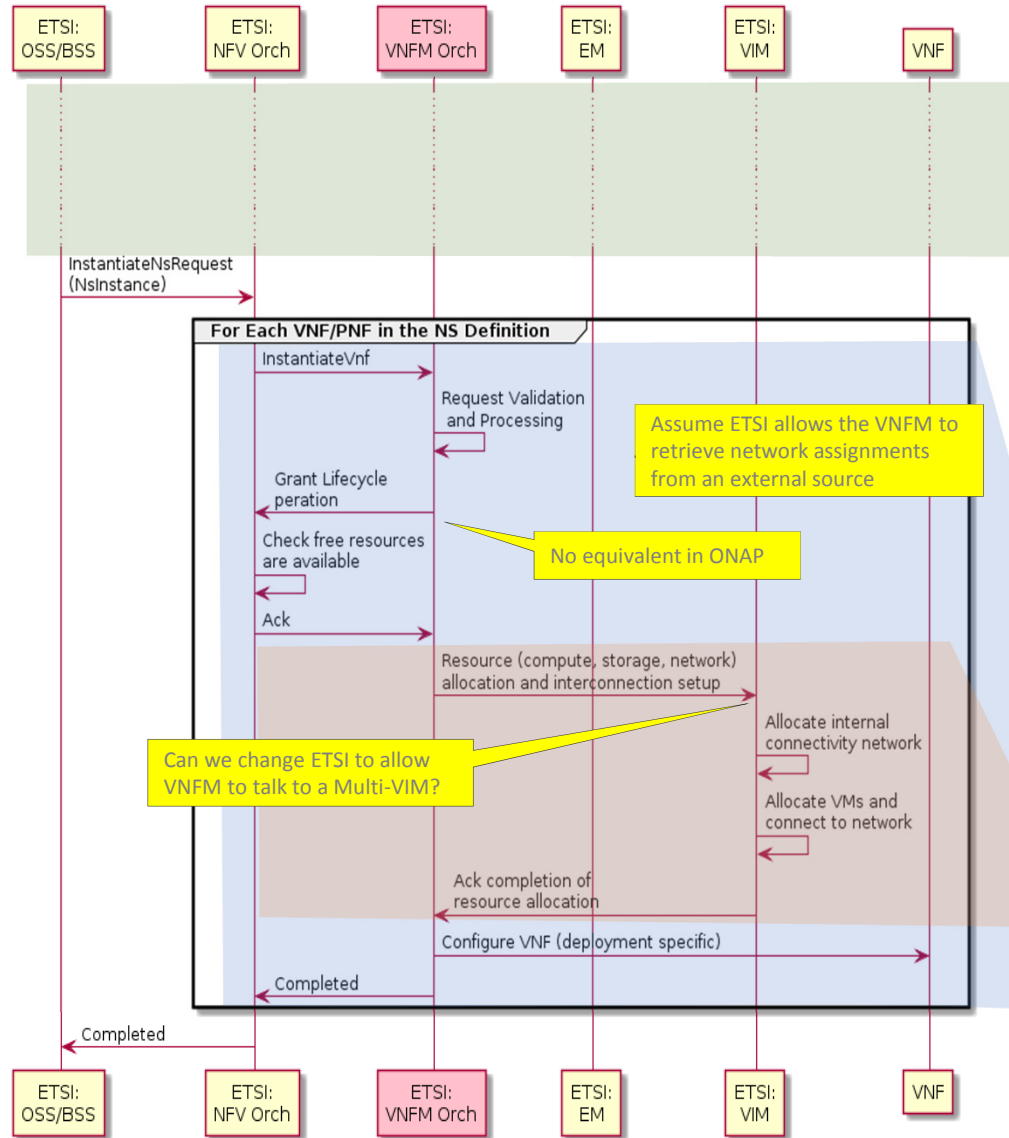


Distributed Orchestration Implementation Approach (Single ONAP Instance) Service/Resource Orchestration Federation Example

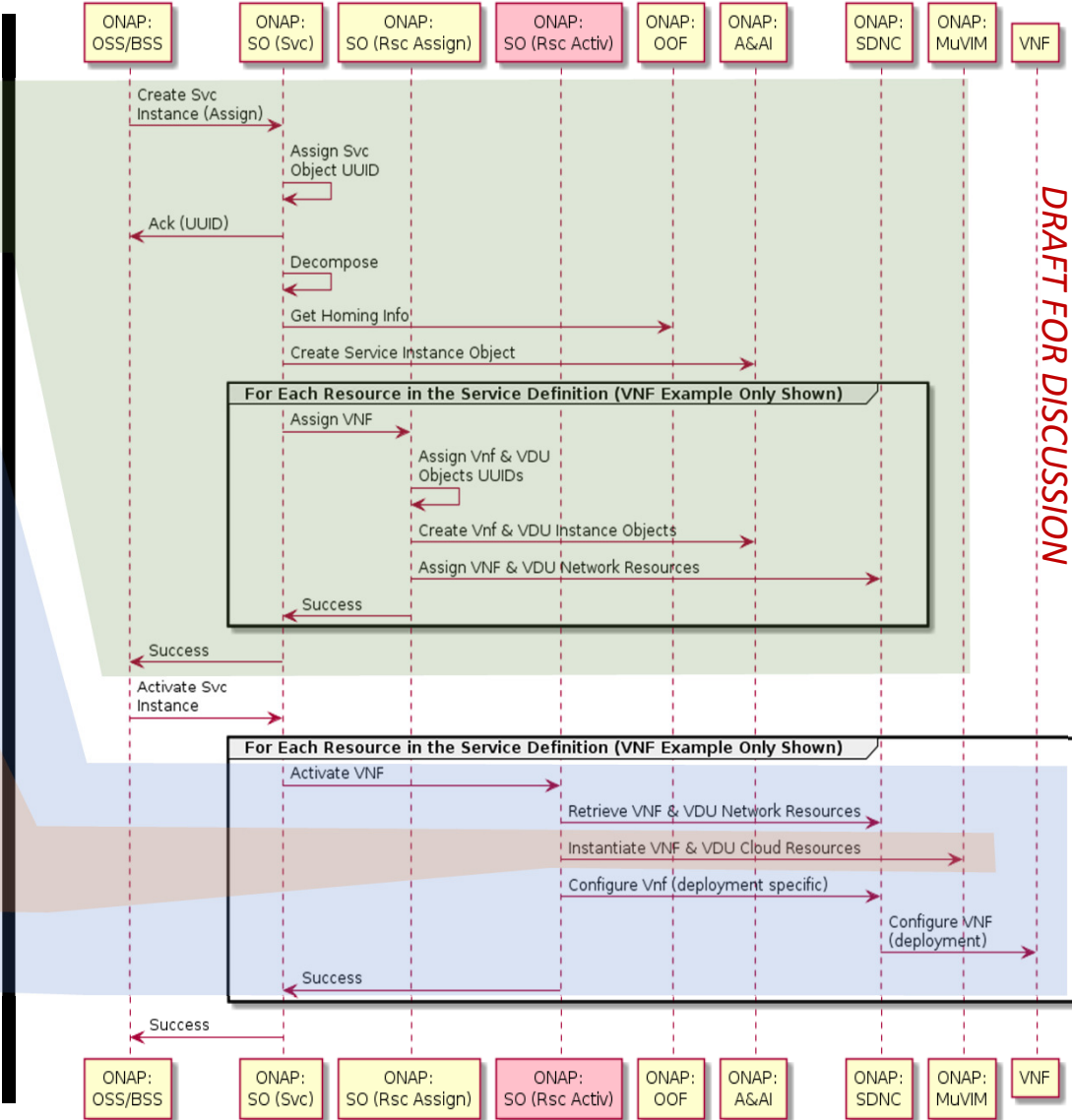
DRAFT FOR DISCUSSION



ETSI MANO B.3.1.2 and B.3.2.2 VNF Instantiation Flow (Figures B.9 and B.11)

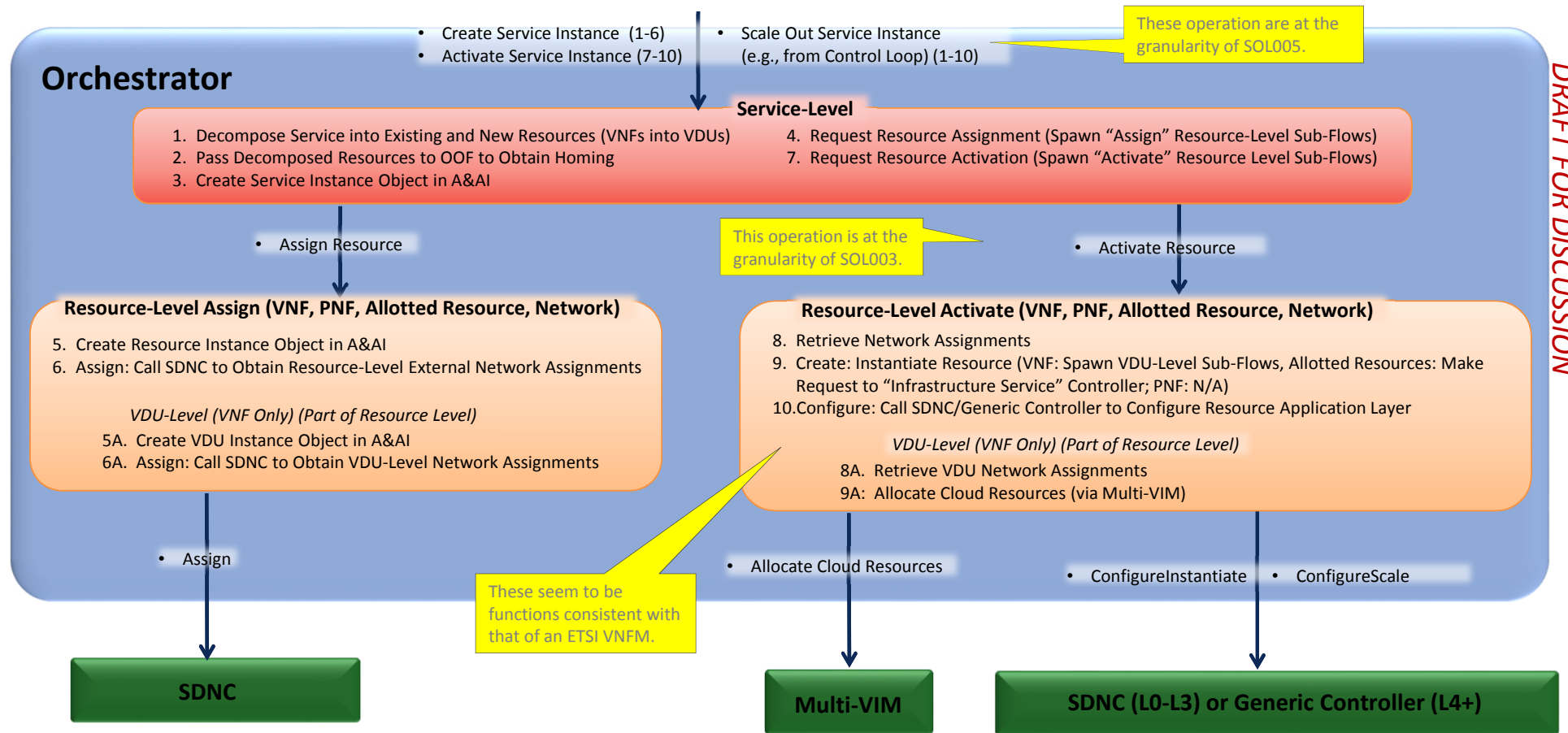


ONAP/ECOMP Instantiation Approach



DRAFT FOR DISCUSSION

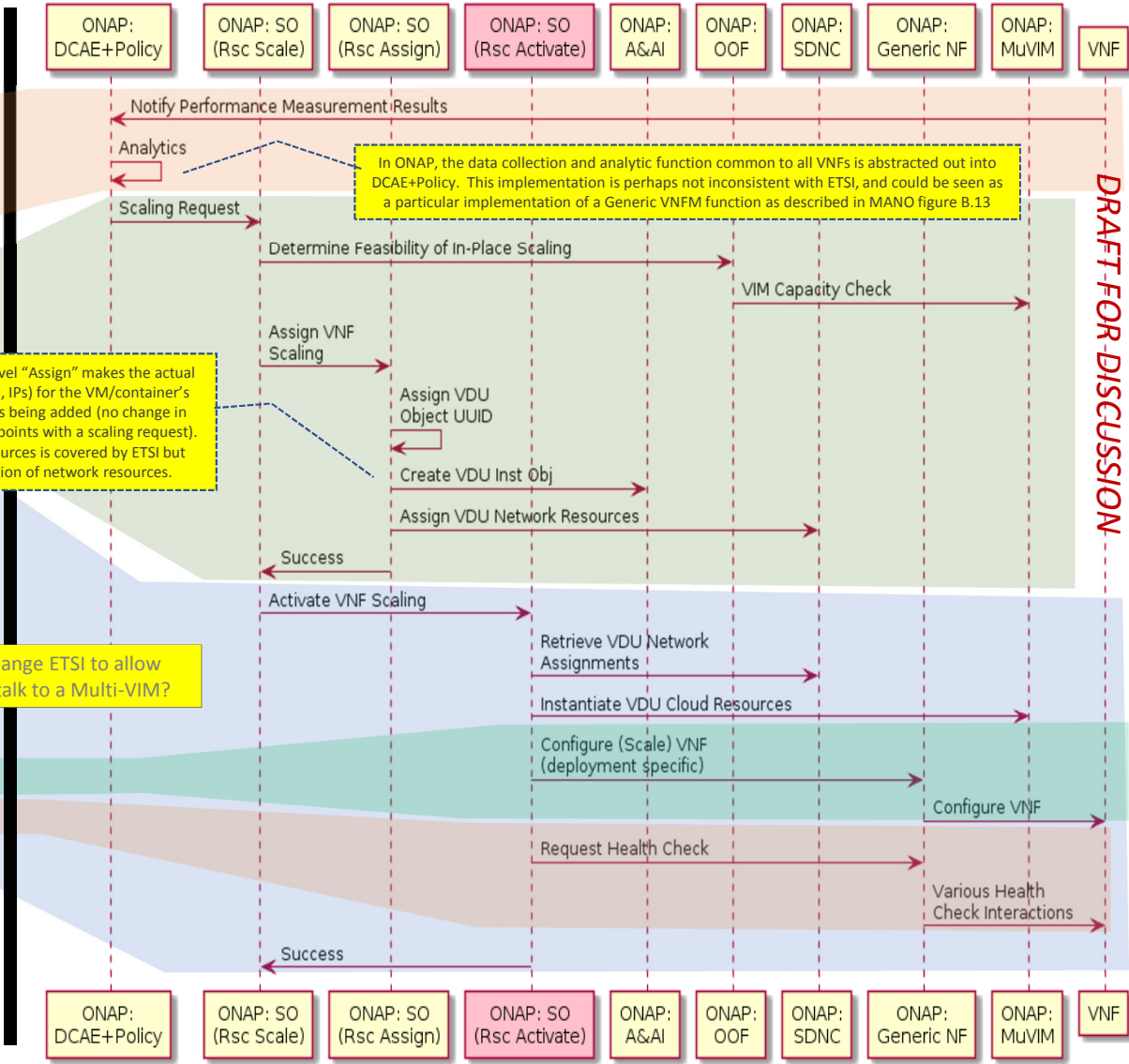
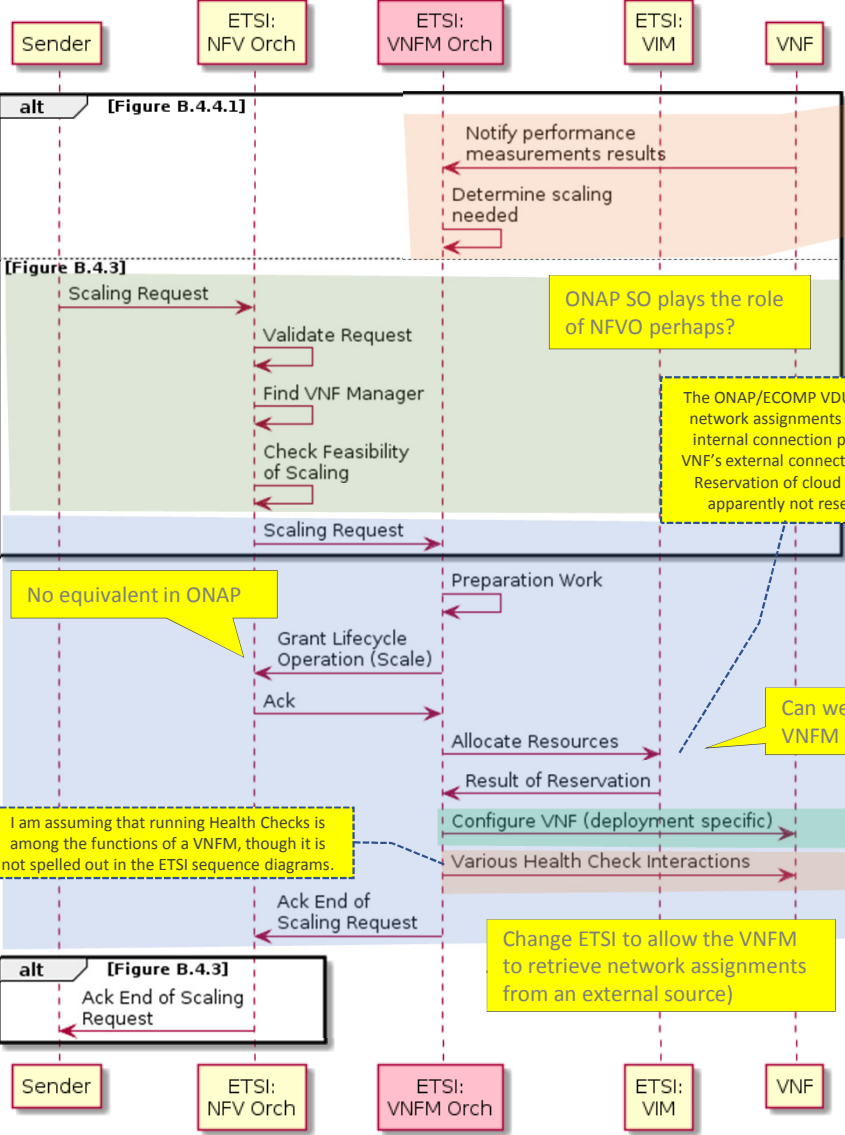
Orchestration Architecture View (Service Instantiate Example)



ETSI MANO B.4.3 and B.4.4.1 Scaling Flow (Figures B.12 and B.13)

ONAP/ECOMP VNF Horizontal Scaling Approach

DRAFT FOR DISCUSSION



No equivalent in ONAP

ONAP SO plays the role of NFVO perhaps?

The ONAP/ECOMP VDU-Level "Assign" makes the actual network assignments (e.g., IPs) for the VM/container's internal connection points being added (no change in VNF's external connection points with a scaling request). Reservation of cloud resources is covered by ETSI but apparently not reservation of network resources.

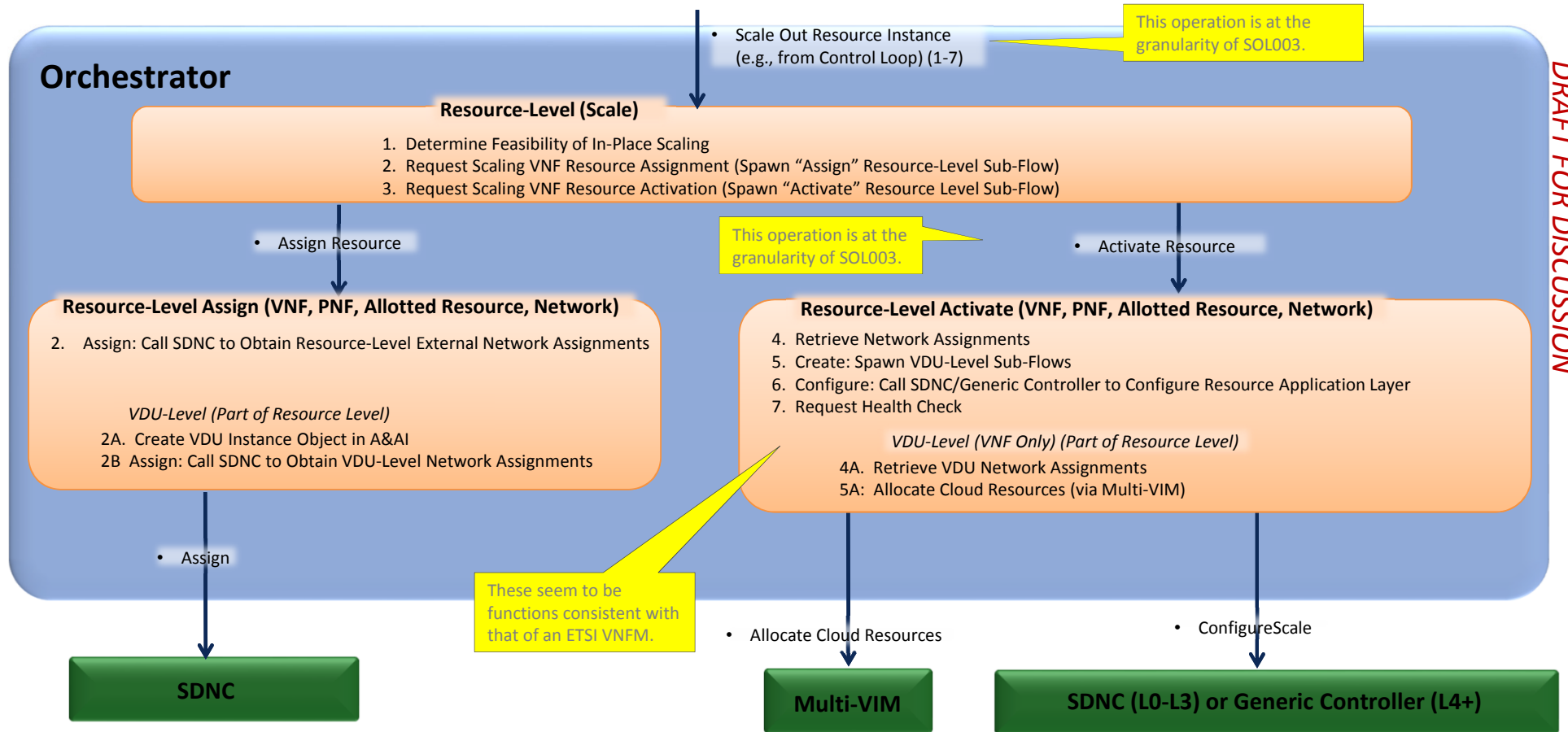
Can we change ETSI to allow VNFM to talk to a Multi-VIM?

Change ETSI to allow the VNFM to retrieve network assignments from an external source)

I am assuming that running Health Checks is among the functions of a VNFM, though it is not spelled out in the ETSI sequence diagrams.

In ONAP, the data collection and analytic function common to all VNFs is abstracted out into DCAE+Policy. This implementation is perhaps not inconsistent with ETSI, and could be seen as a particular implementation of a Generic VNFM function as described in MANO figure B.13

Orchestration Architecture View (VNF Scale Out Example)

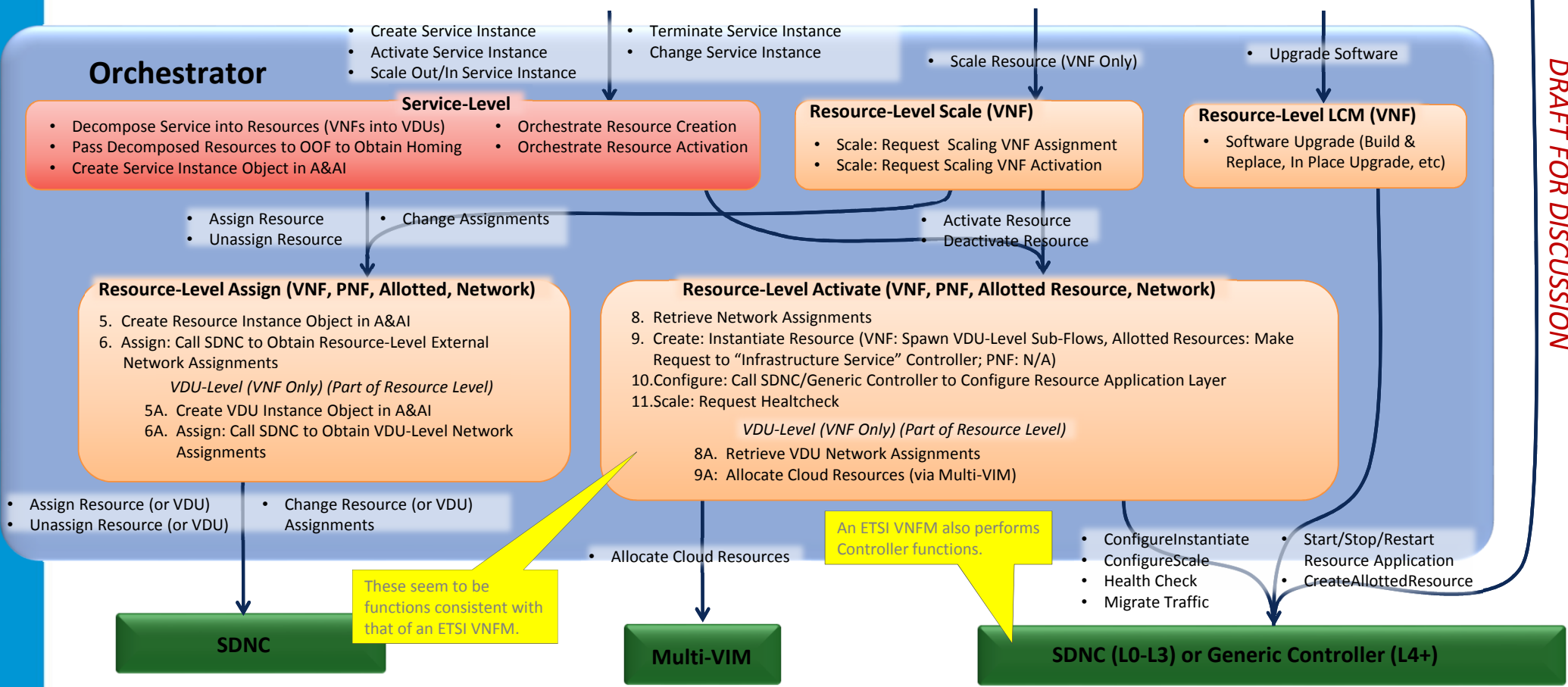


DRAFT FOR DISCUSSION

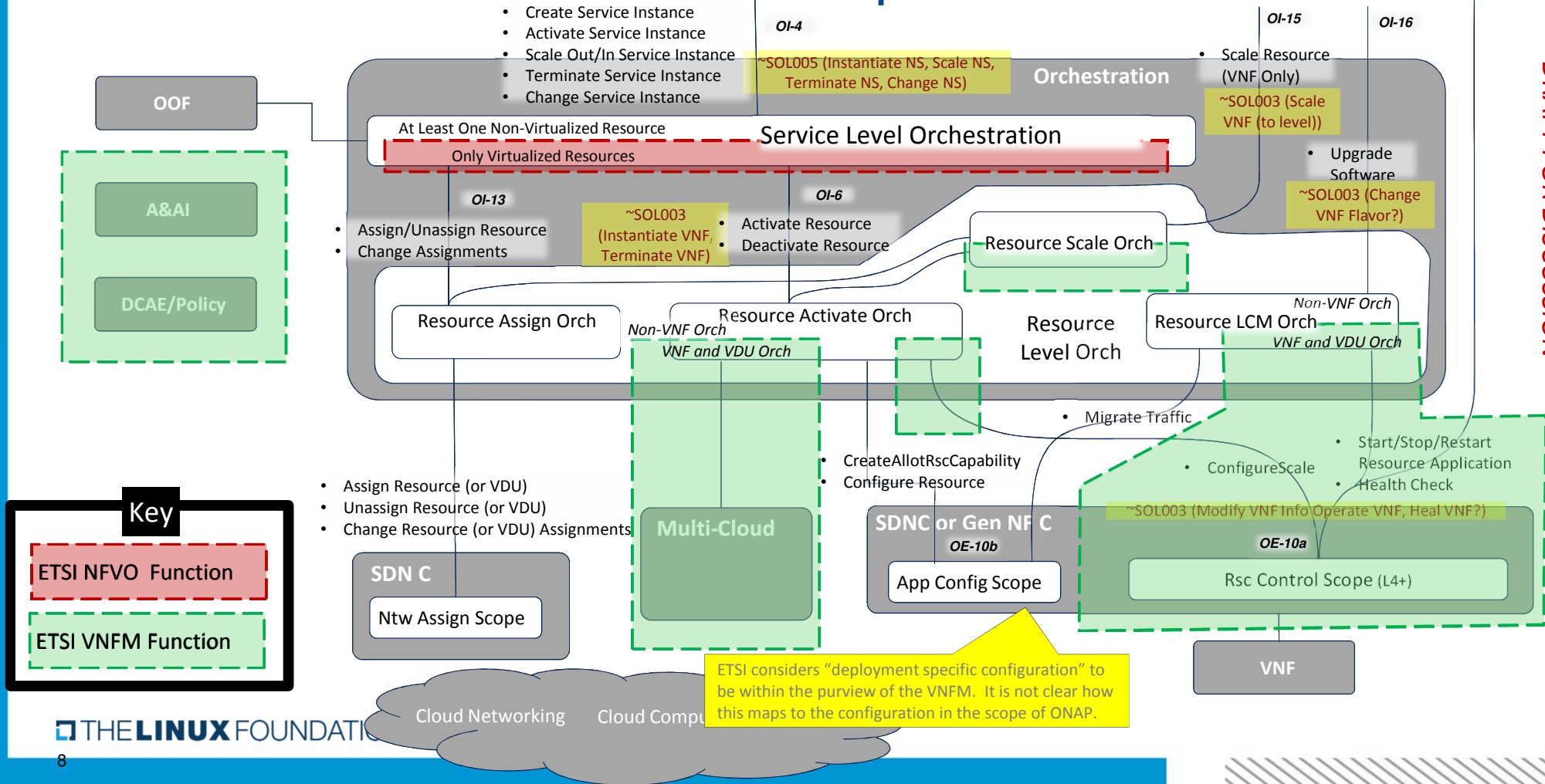
Orchestration Architecture View (General Case)

- Start/Stop/Restart Resource Application
- Health Check

DRAFT FOR DISCUSSION



ONAP Orchestration Arch Compared to ETSI

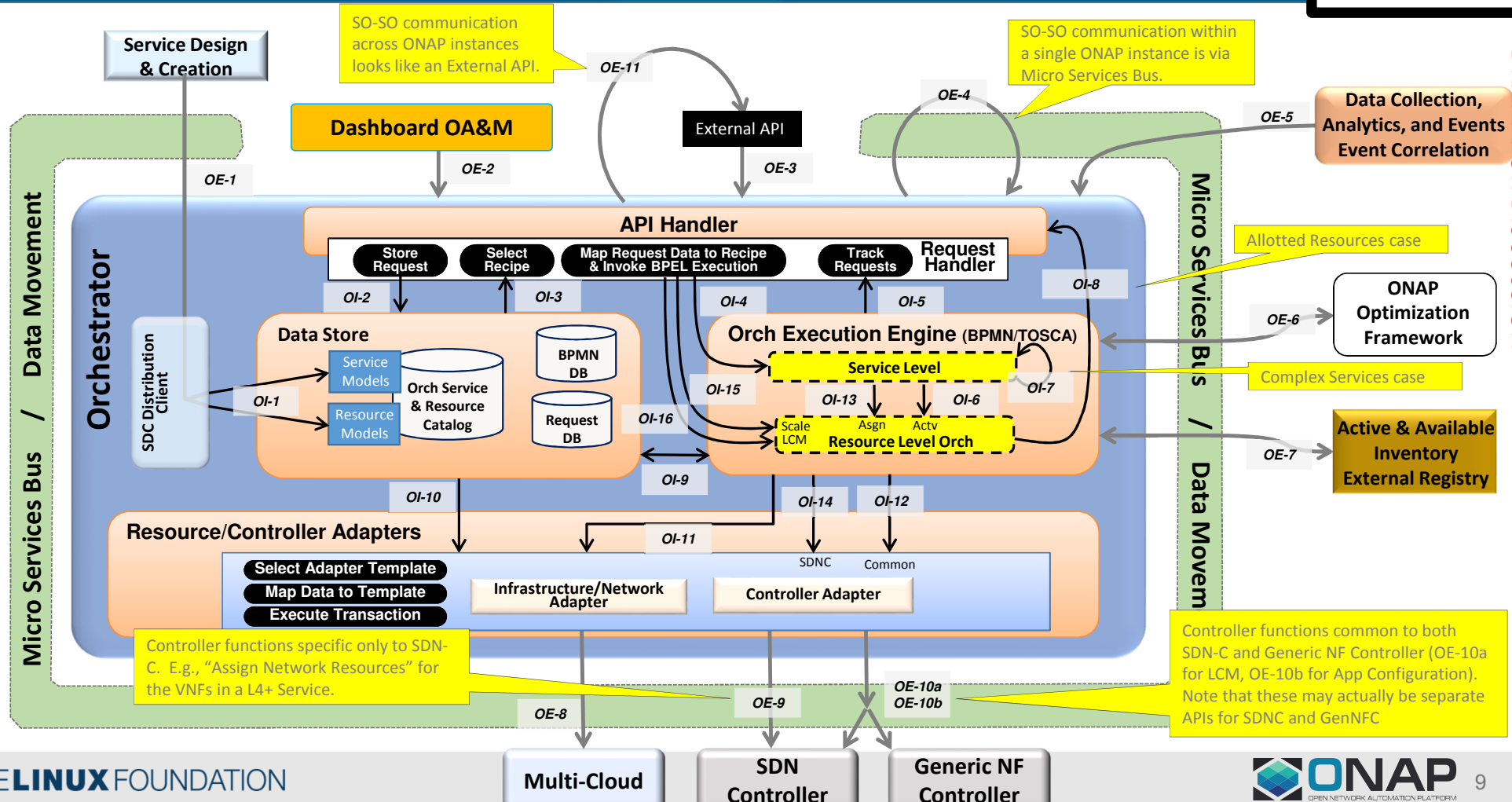


DRAFT FOR DISCUSSION

Orchestrator Functional Internal View

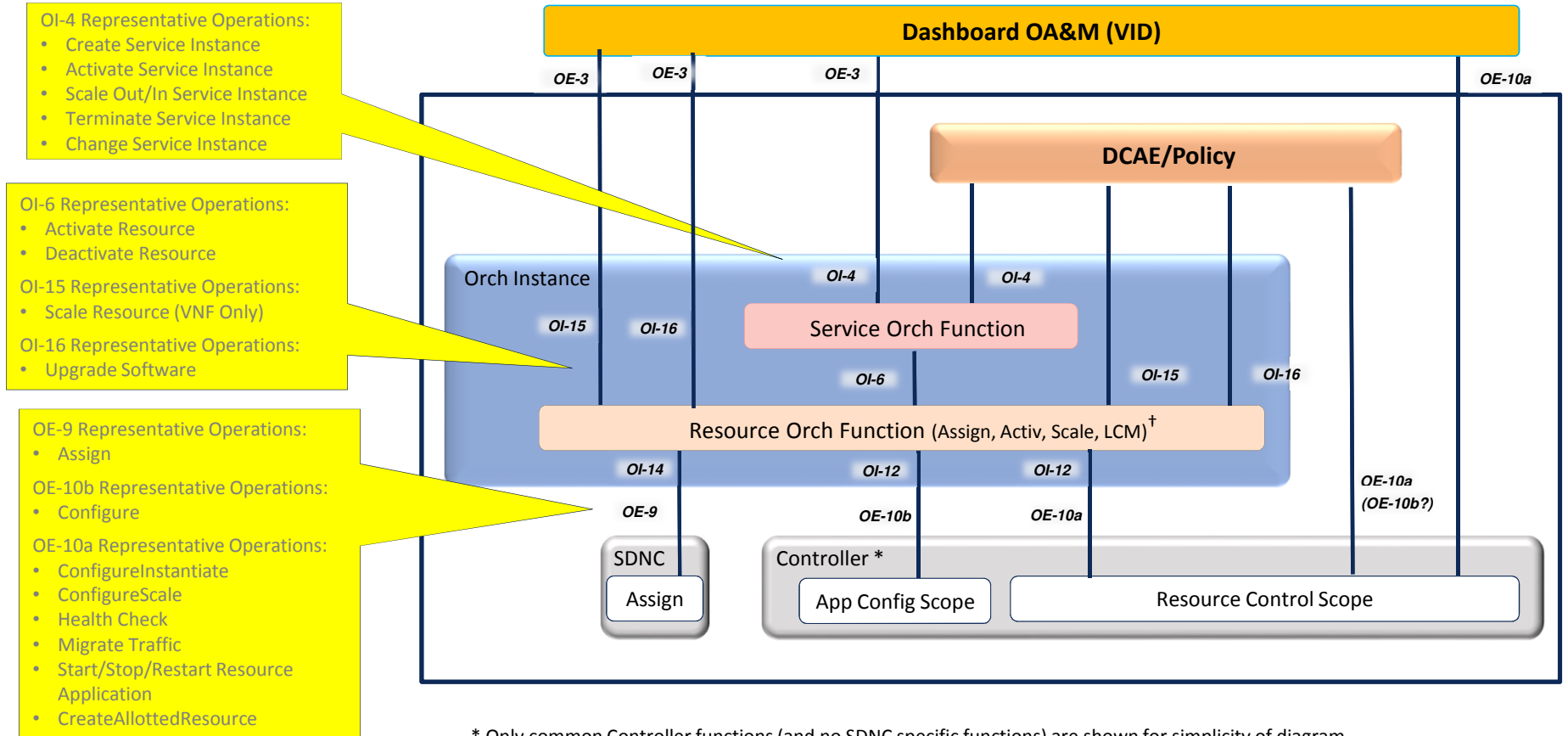
APIs labeled on slide relative to SO for reference only.

Key	
OE-x	SO External API
OI-x	SO Internal API



Orchestration Standard Model

DRAFT FOR DISCUSSION

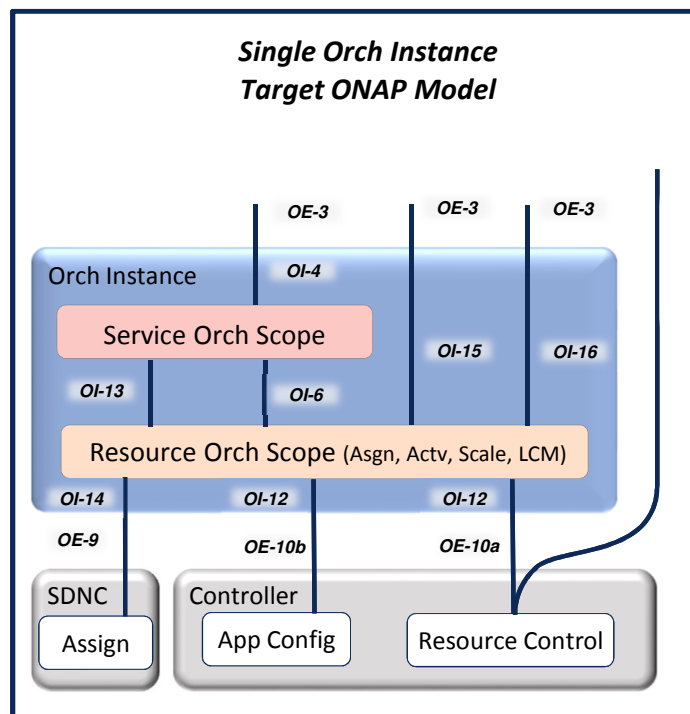


* Only common Controller functions (and no SDNC specific functions) are shown for simplicity of diagram

[†] Resource Orchestration Function (Assign) not represented because SDNC specific functions not shown



Implementation Pattern 1 (Single Orchestration Instance)



Routing Function not shown for simplicity. (See next slides.)

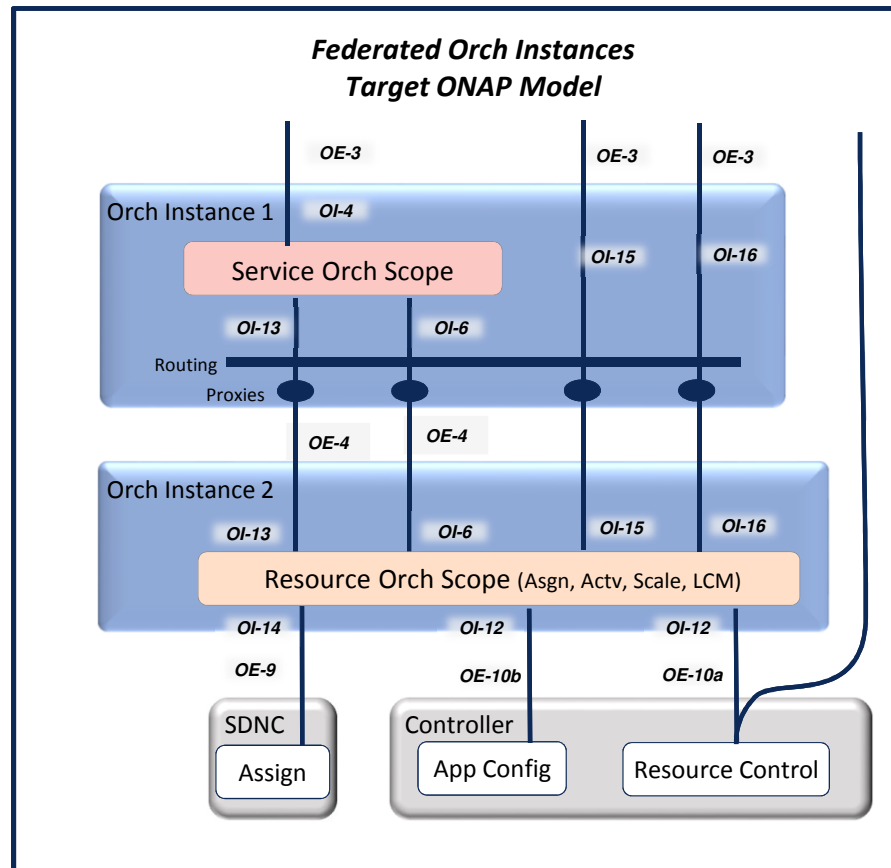
DRAFT FOR DISCUSSION



Implementation Pattern 2 (Federated Orchestration Instances with Homogenous Service Example)

Example Uses:

- Resource Orchestration by Resource type specific Orchestrators
- Resource Orchestration by Orchestrators per geographic domain
- Combination of the above



Service Example Used for Patterns 3 and 4

In the following two examples a single Service type ($Service_A$) decomposes into two Resource types ($Resource_1$ and $Resource_2$).

In the first example, $Resource_1$ is supported via the same SO/Controller instance as $Service_A$ whereas $Resource_2$ is supported via a (set of) remote SO/Controller instance(s).

Service_A **Service:**
topology_template:
node_templates:
 Resource₁ (Any):
 Resource₂ (VNF or PNF):

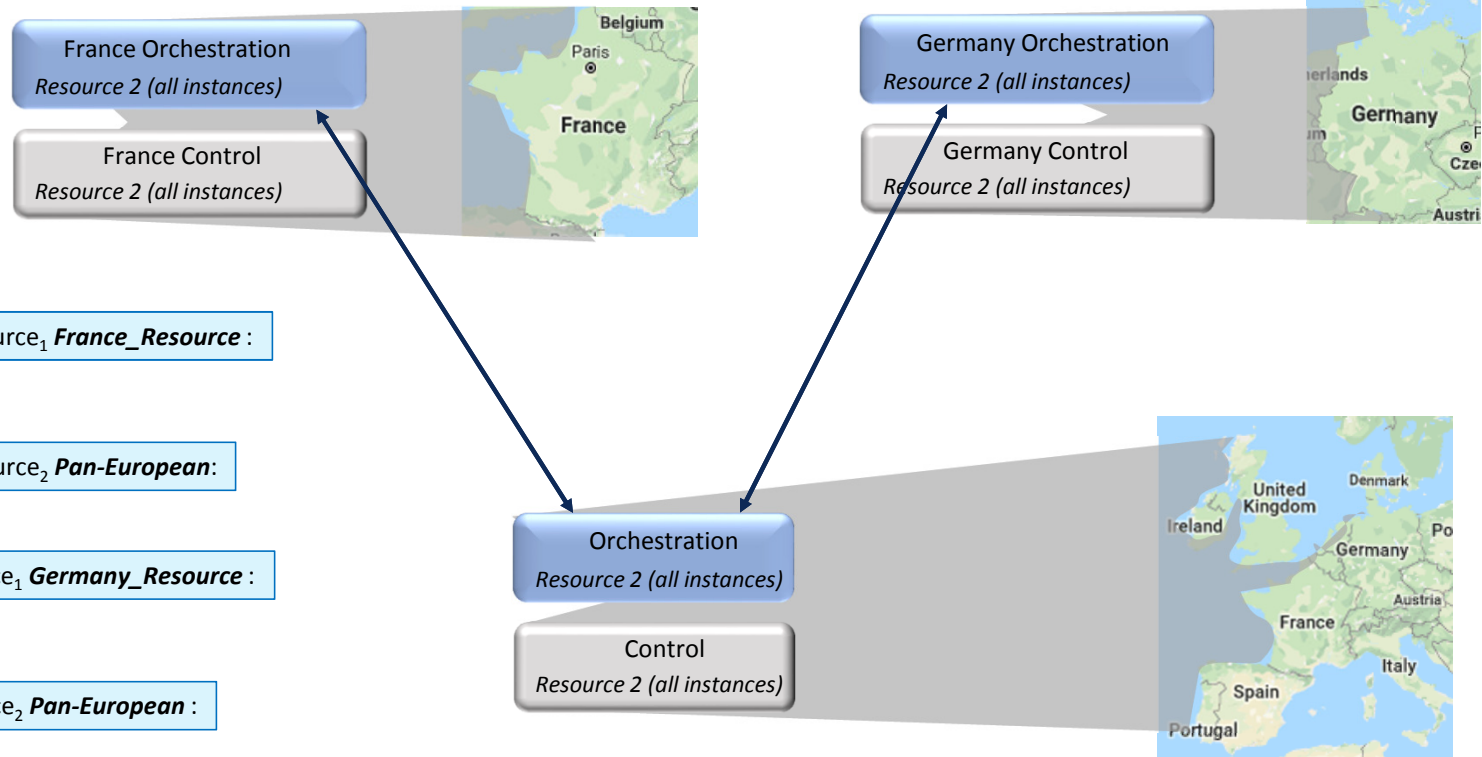
Resource₁ **Any Resource:**

Resource₂ **VNF or PNF Resource:**

In the second example, $Resource_1$ is supported via the same SO instance as $Service_A$ whereas $Resource_2$ is supported via a VNFM proxy. It is due to this second example that $Resource_2$ is assumed to be either a VNF or a PNF. In both examples, $Resource_1$ could be any Resource type, including an Allotted Resource. However, see Implementation patterns 5 and 6.



Pattern 3 Use Case – Sub-Orchestrators by Type



Service_A **France_Service**:
 topology_template:
 node_templates:
 Resource₁ (Any):
 Resource₂ (VNF or PNF):

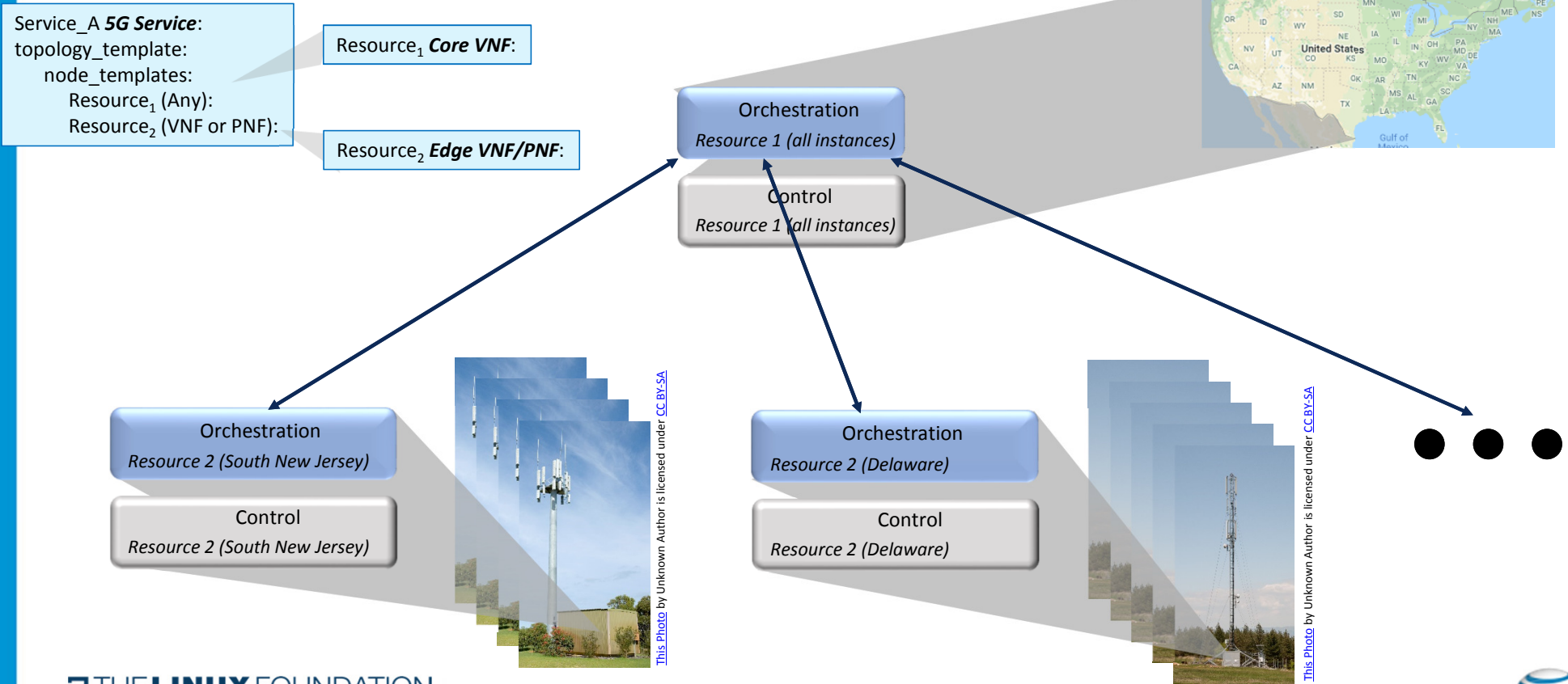
Resource₁ **France_Resource** :
 Resource₂ **Pan-European** :

Service_B **Germany_Service**:
 topology_template:
 node_templates:
 Resource₁ (Any):
 Resource₂ (VNF or PNF):

Resource₁ **Germany_Resource** :
 Resource₂ **Pan-European** :



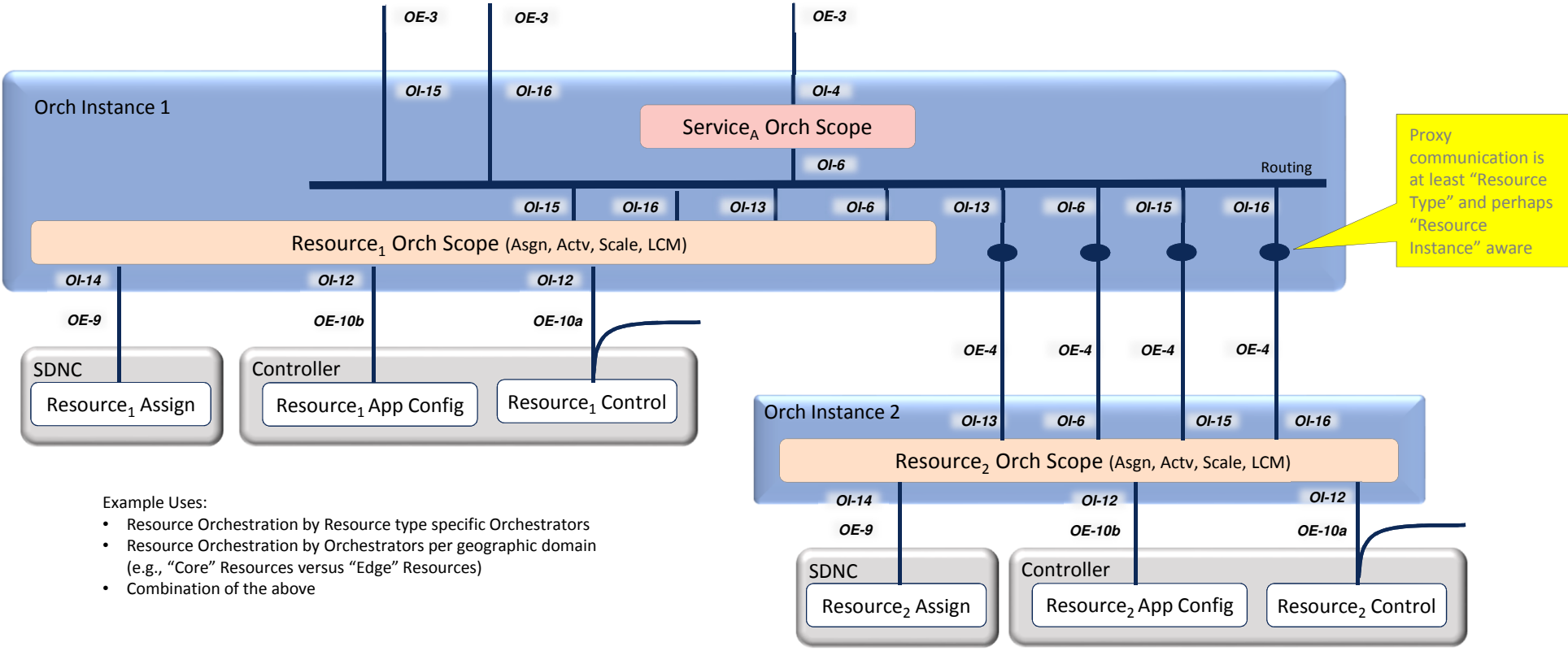
Pattern 3 Use Case – Sub-Orchestrators by Type + Instance



DRAFT FOR DISCUSSION



Implementation Pattern 3 (Federated Orchestration Instances with Mixed Service Example)



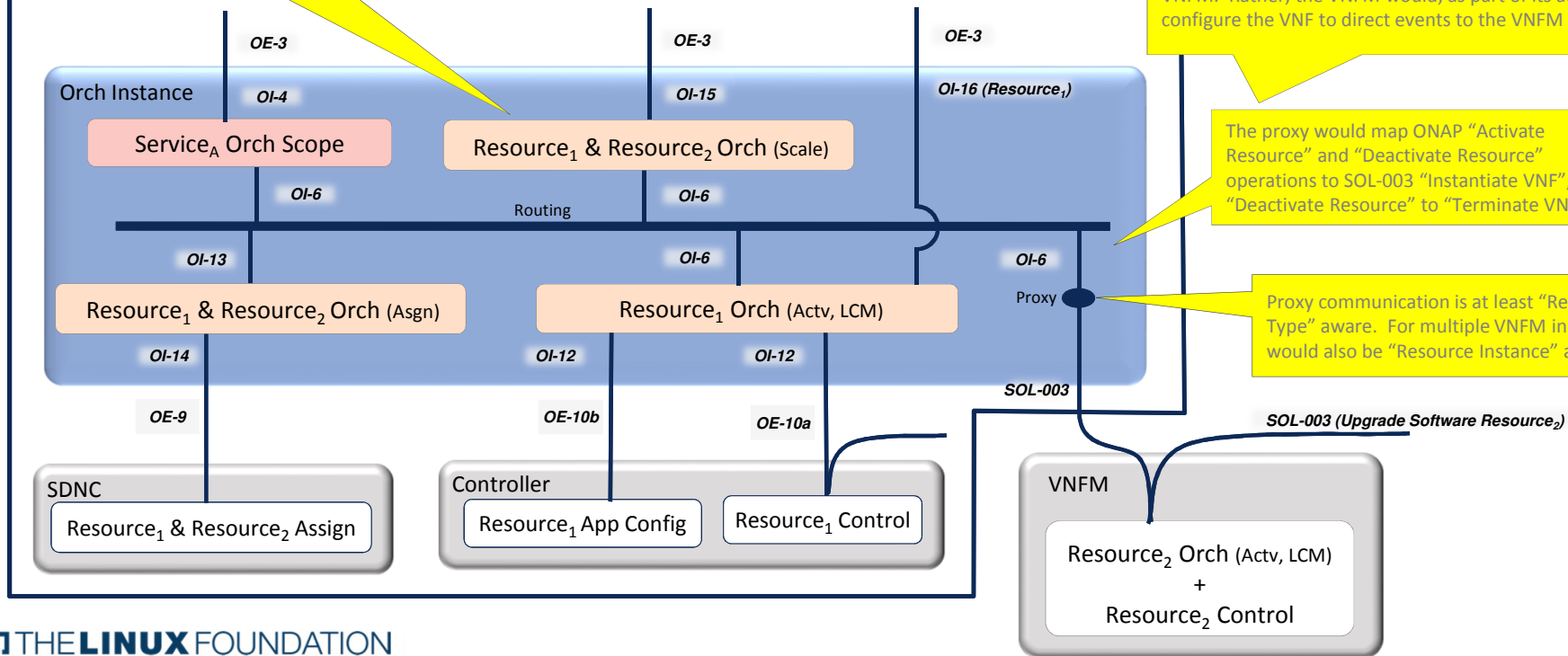
- Example Uses:
- Resource Orchestration by Resource type specific Orchestrators
 - Resource Orchestration by Orchestrators per geographic domain (e.g., "Core" Resources versus "Edge" Resources)
 - Combination of the above



Implementation Pattern 4 (VNFM Plug in with Mixed Service Example)

"Scale" workflow calls the "Assign" sub-flow then the "Activate" function. This latter function is either a sub-flow (Resource₁) or a VNFM implementation (Resource₂).

Single Orch Instance Alternate VNFM Proxy Model



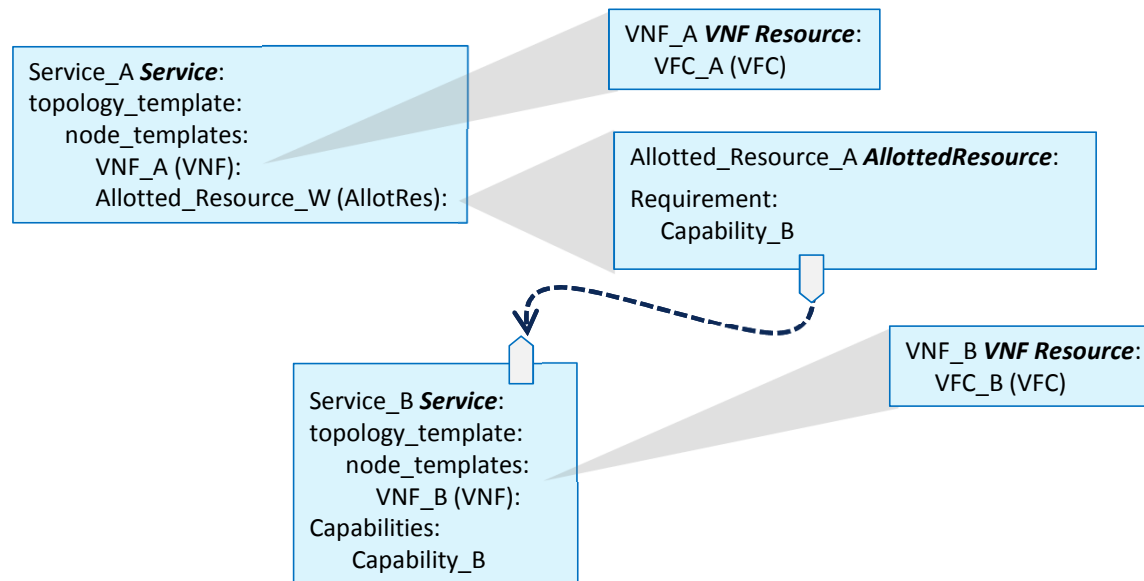
For Resource₂, Change VNF Flavor, Start/Stop/Restart Resource Application, Health Check are not performed via ONAP, but are performed by the VNFM direct interaction with the VNF and/or the EM. Thus, Resource₂ activation (VNFM) would not configure any DCAE collector to receive events from a VNF managed by a VNFM. Rather, the VNFM would, as part of its activation, configure the VNF to direct events to the VNFM itself.

The proxy would map ONAP "Activate Resource" and "Deactivate Resource" operations to SOL-003 "Instantiate VNF", and "Deactivate Resource" to "Terminate VNF".

Proxy communication is at least "Resource Type" aware. For multiple VNFM instances it would also be "Resource Instance" aware

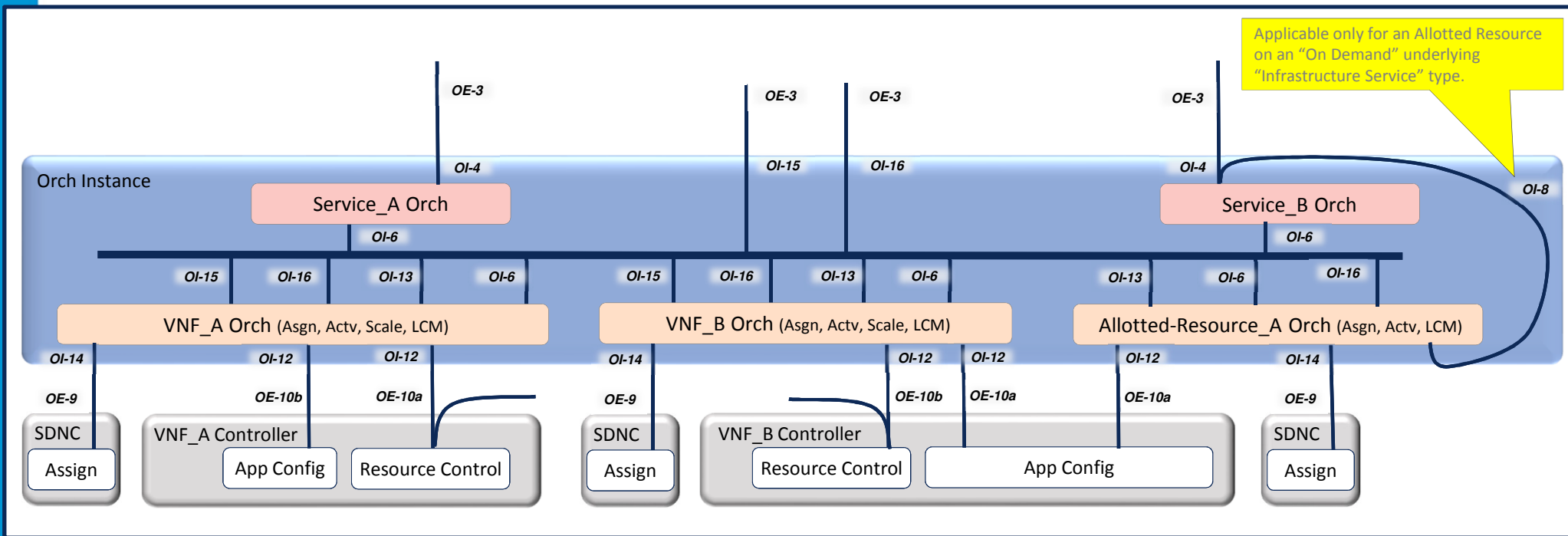


Service Example Used for Patterns 5 and 6



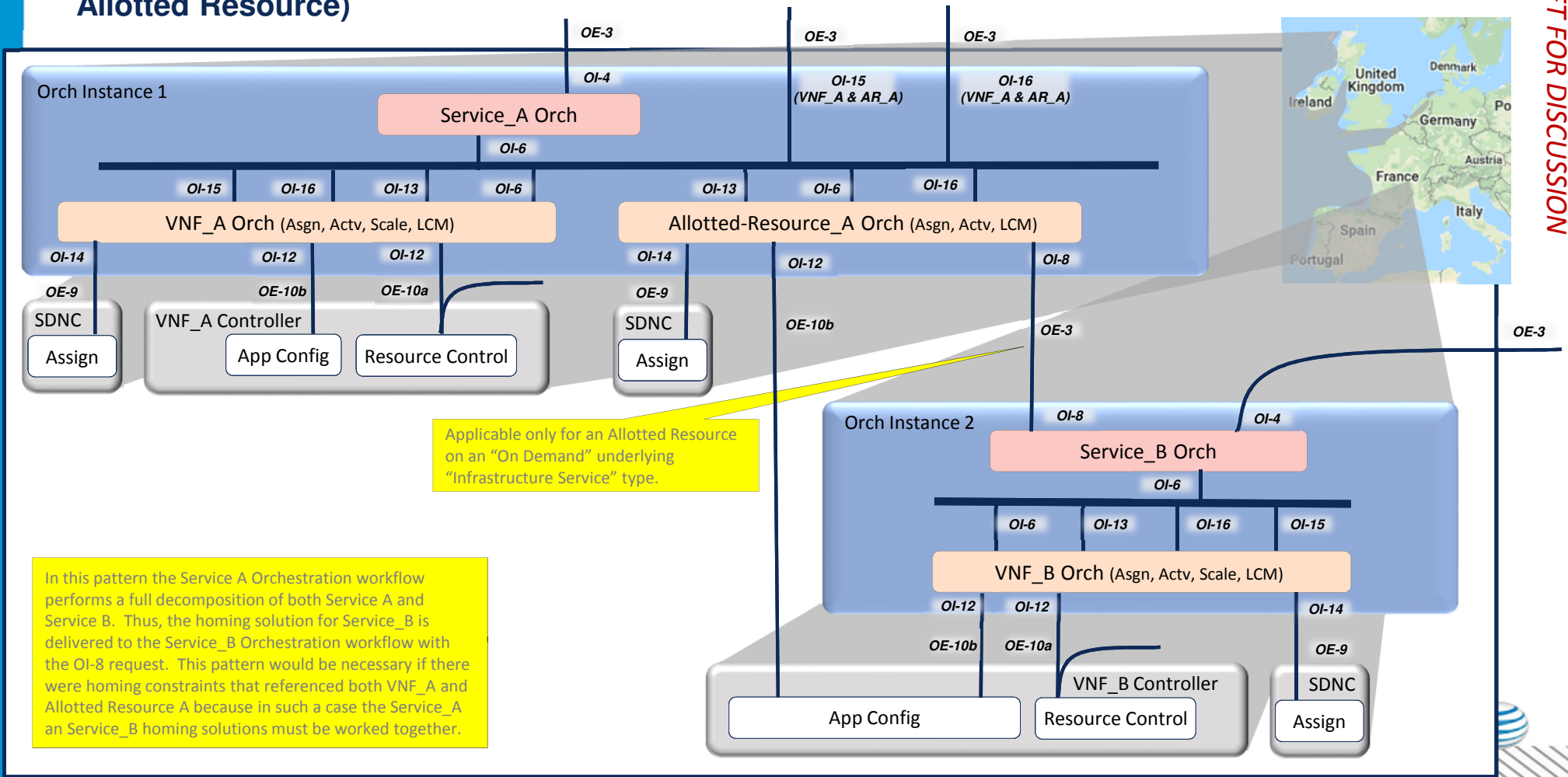
Implementation Pattern 5 (Single Orchestration Instance with “On Demand” Allotted Resource)

DRAFT FOR DISCUSSION



Implementation Pattern 6 (Federated Orchestration Instances with “On Demand” Allotted Resource)

DRAFT FOR DISCUSSION



Need to separate the Vodafone example from the 5G example:

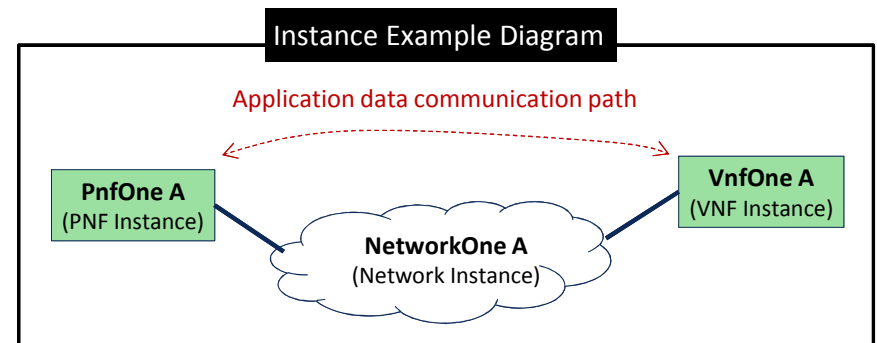
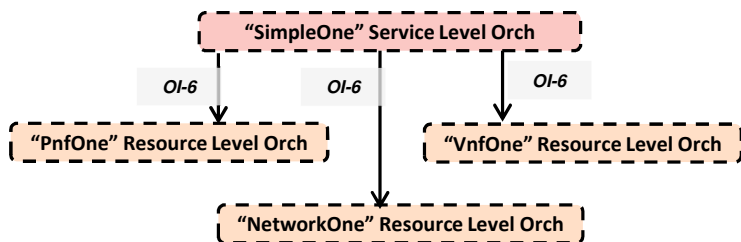
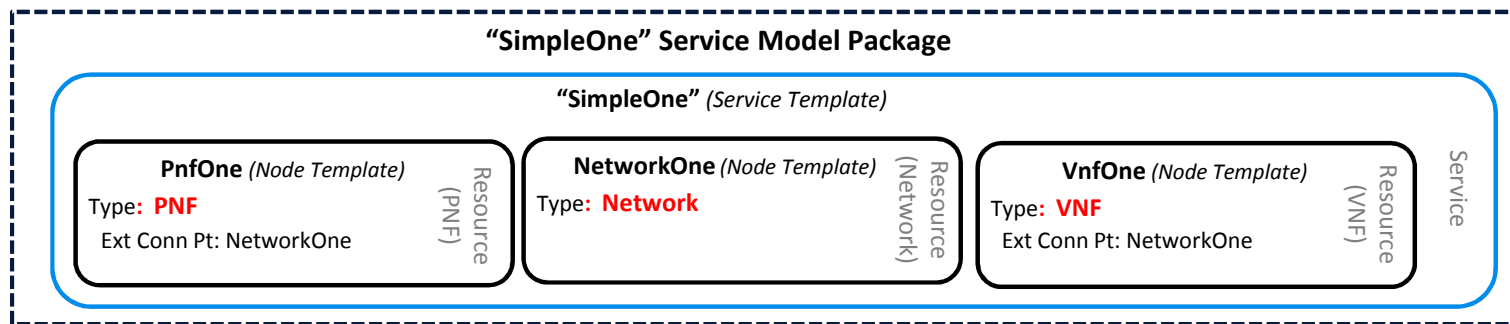
The Vodafone example says that there are Orchestration Instances that are differentiated by TYPE.

The 5G example is different Orchestration instances for the same service type.

Show with a map



Simple Service, No Allotted Resource Example

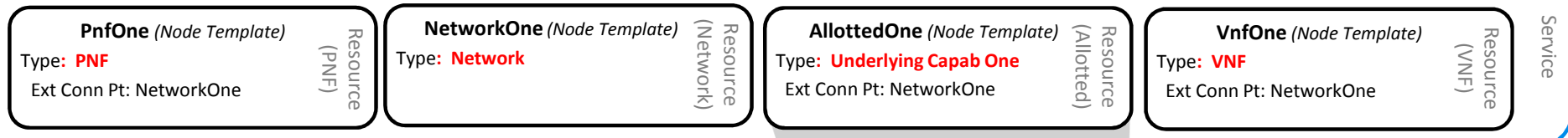


Simple Service with Allotted Resource Example

DRAFT FOR DISCUSSION

"SimpleTwo" Service Model Package

"SimpleTwo" (Service Template)



Assume policy indicates that if no appropriate "InfraOne" Service Instance is available upon receipt of a "SimpleTwo" service instantiation request, it is possible to dynamically "spin up" an associated "InfraOne" service instance "on the fly" in a suitable location. This InfraOne service instance would not be dedicated to this initial "SimpleTwo" service instance, but could be shared by future service instances of "SimpleTwo" or any other "higher order service" that requires an "Underlying Capab One". Thus, we should think the triggering of a new "InfraOne" service instance as being an "on demand" infrastructure build-out.

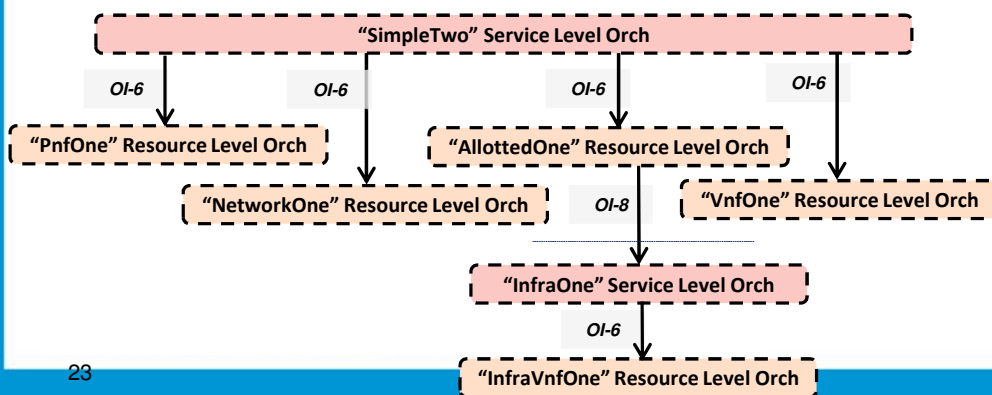
"InfraOne" Service Model Package

InfraOne Shared Resource (Node Type Definition)

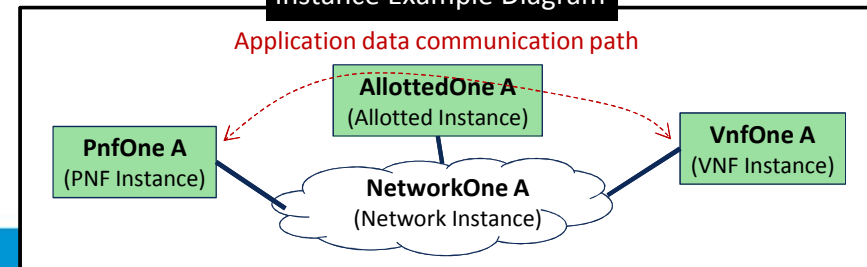
node-types:
Underlying Capab One
providing-service: InfraOne

"InfraOne" (Service-Template)

InfraVnfOne
Type: **VNF**
Ext Conn Pt: NetworkOne
Resource (VNF)



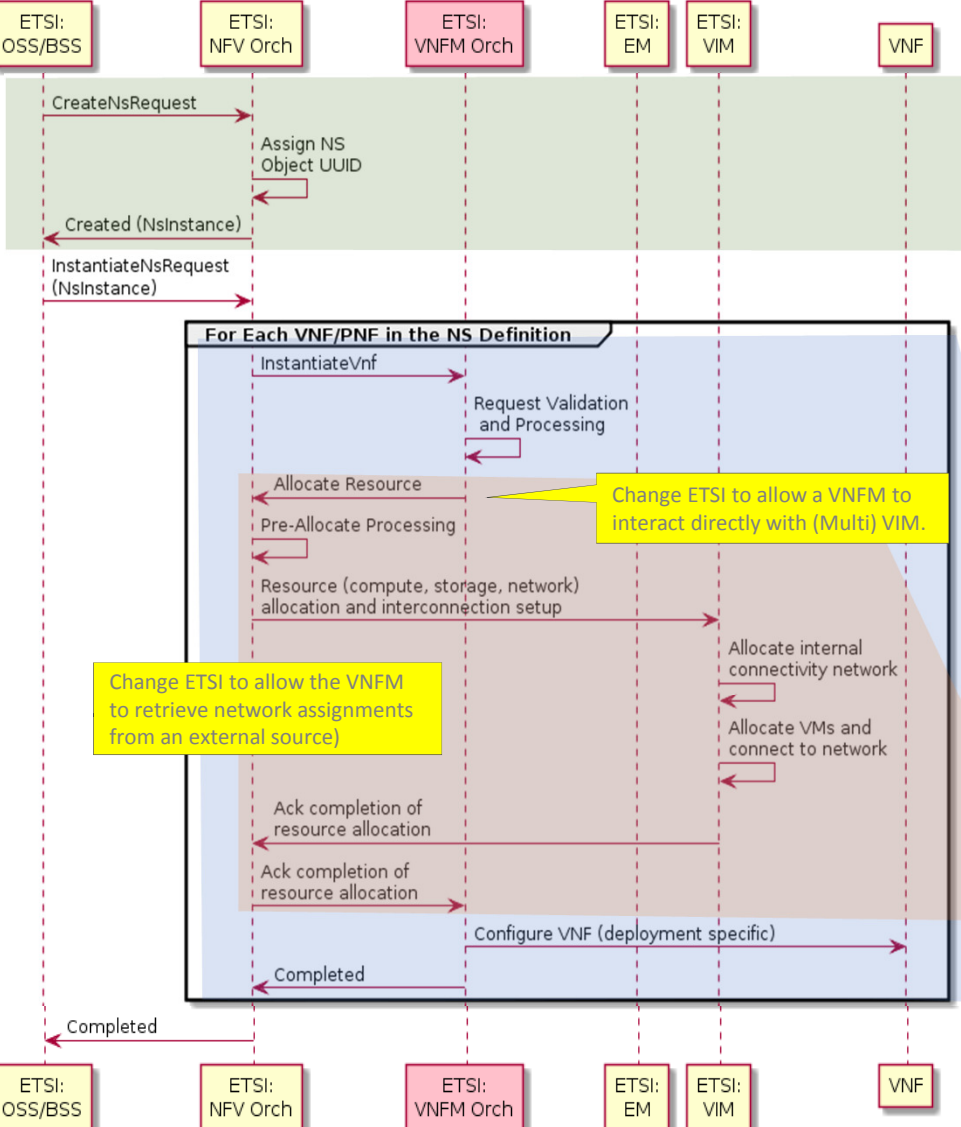
Instance Example Diagram



Backup Slides



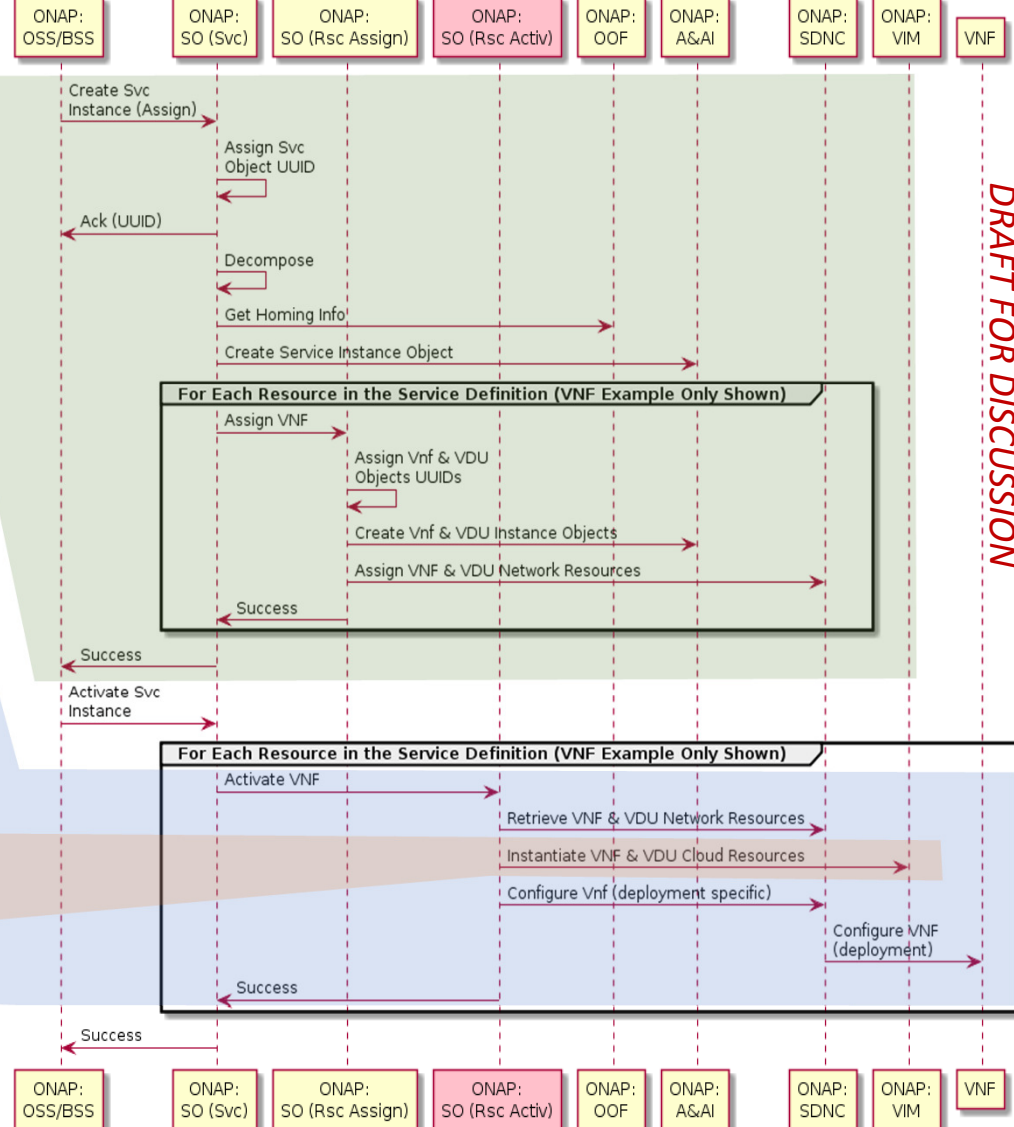
ETSI MANO B.3.1.2 VNF Instantiation Flow (Figure B.9)



Change ETSI to allow a VNFM to interact directly with (Multi) VIM.

Change ETSI to allow the VNFM to retrieve network assignments from an external source)

ONAP/ECOMP Instantiation Approach



DRAFT FOR DISCUSSION