



Integrating External Controllers with ONAP

AT&T Labs

Motivation

- Some service providers may want to leverage an alternative to an ONAP “out of the box” Controller (e.g., SDNC, GenNFC) for some subset of Resources. For example:
 - There may exist some VNFs that come with their own “Controller” that the service provider wants to use
 - There may exist some VNFs that leverage capabilities that the corresponding ONAP model (and hence ONAP Controller) does not yet support
- We see the need for a flexible approach that allows ONAP service providers to easily customize their ONAP instance to integrate External Controllers, as needed
- However, we recommend that “External Controllers” would “plug in” to ONAP in such a way to avoid adding complexity to the ONAP architecture

Proposed High Level Approach

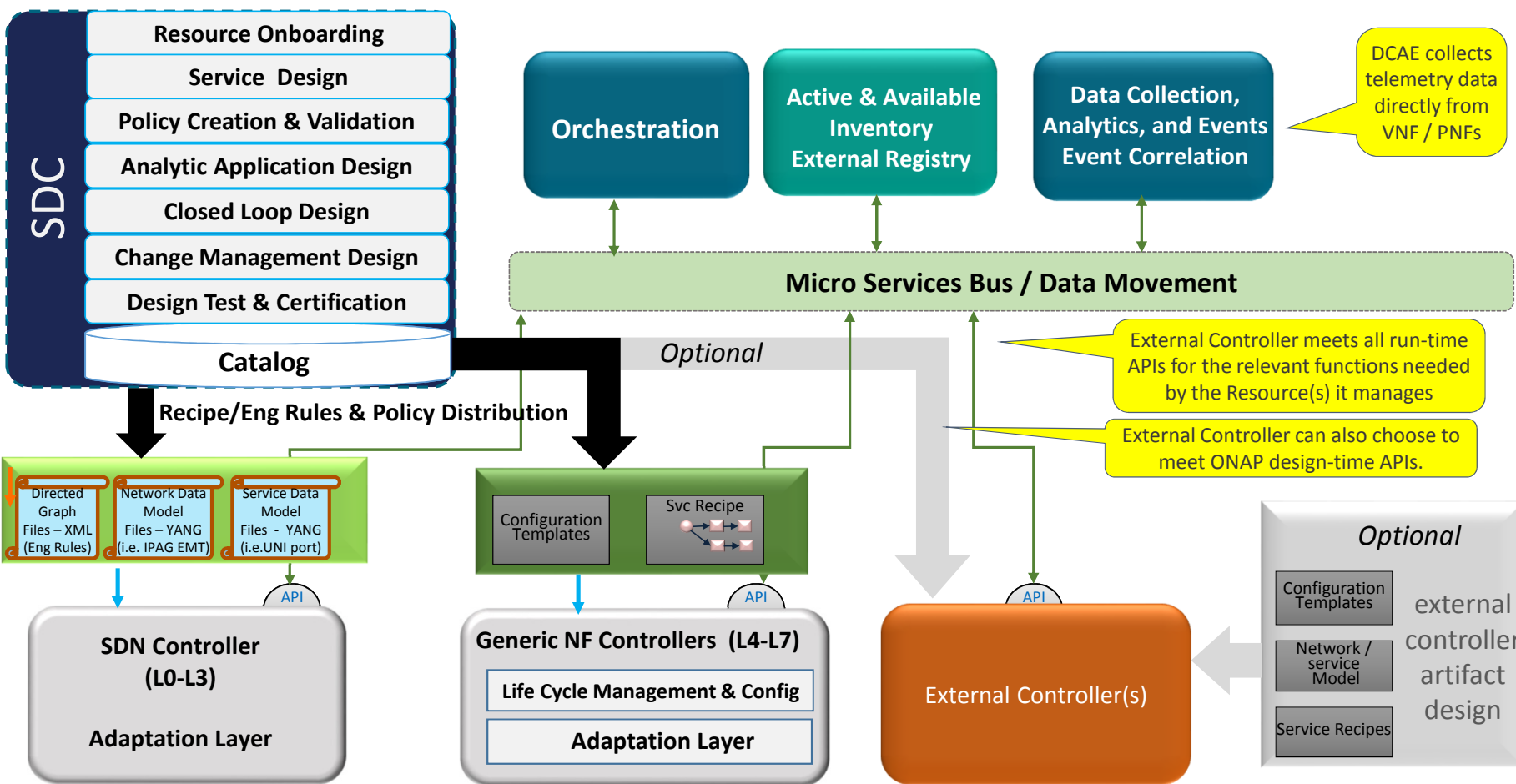
- Note that ONAP can support external components being “plugged in”. E.g., VNFM. So there is no technical reason why ONAP couldn’t support External Controller being “plugged in”.
- If a VNF vendor feels the need to provide their own “External Controller” because their VNF has needs that are not supported by the model-driven ONAP (out of the box) Controllers:
 - The first preference should be for that VNF vendor to contribute the needed capabilities to the ONAP CCSDK to fill the functional gaps. I.e., this would eliminate the need for an “External Controller”. Rather, the VNF would be managed by the ONAP Controller. (The remainder of this deck assumes that the VNF vendor is either not willing or not able to do this.)
 - The second preference would be that the VNF vendor would use the ONAP CCSDK to create the “base” of their External Controller, and add their “secret sauce” to it.
 - The third (and last) preference would be for the VNF vendor to custom-build an External Controller.
- To avoid complexity of the architecture, ONAP should only provide adaption to “plug in” external components that are of a class that exhibit well-defined functionality and APIs. E.g., ONAP can support a VNFM adaptor because the VNFM meets the ETSI standard.
- However, there currently exists no well-defined specification for a “Controller”. Thus, we will consider the ONAP Controller itself to be the de facto standard of “what is a Controller”, including an “External Controller”.
- Therefore it would be the responsibility of the External Controller vendor to adapt itself to meet the ONAP (out of the box) “Controller” definition. I.e., the External Controller must:
 - Meet the respective ONAP (out of the box) Controller (at least run-time) APIs for the functionality they support (either natively or through adaptation).
 - Call the necessary other ONAP APIs (e.g., A&AI) proper to the role of the respective ONAP (out of the box) Controller (see subsequent slide)

Proposed High Level Approach (Continued)

- Insofar that an External Controller meets the ONAP definition for a “Controller”, there is no need for ONAP to support an “adaptor” for such an External Controller; rather the External Controller can “plug in” to ONAP directly.
- An External Controller would typically replace an ONAP Controller’s function only within a specific (set of) domain(s)
 - A “domain” in this context is a collection of one or more Resource types (see subsequent slide)
- Such domain specific External Controllers are outside ONAP and as such not integrated or tested during ONAP community integration test. Rather the service provider that customizes their ONAP instance to support an External Controller would take on this integration testing.
- The role of other ONAP modules doesn’t change in the presence of an External Controller:
 - SDC will be used for modeling the management of Resources and Services (including on-boarding, designing and full LCM)
 - SO will provide E2E orchestration services for all lifecycle management functions for all managed objects
 - Obtaining and managing cloud resources will be done through ONAP multi-cloud layer
 - A&AI will maintain inventory of all the managed Resources (controlled by External Controllers or ONAP Controllers)
 - DCAE will collect telemetry data directly from all managed objects / services (without going through the external controllers) and provide analytic services
 - Etc.

Integration of External Controllers – Target View

The registration of new External Controller instances needs to be worked through. Post-registration, the management of the mapping between Service/Resource types and the Controller instance for each type would be handled in the same manner as any "internal" ONAP Controller.



DCAE collects telemetry data directly from VNF / PNFs

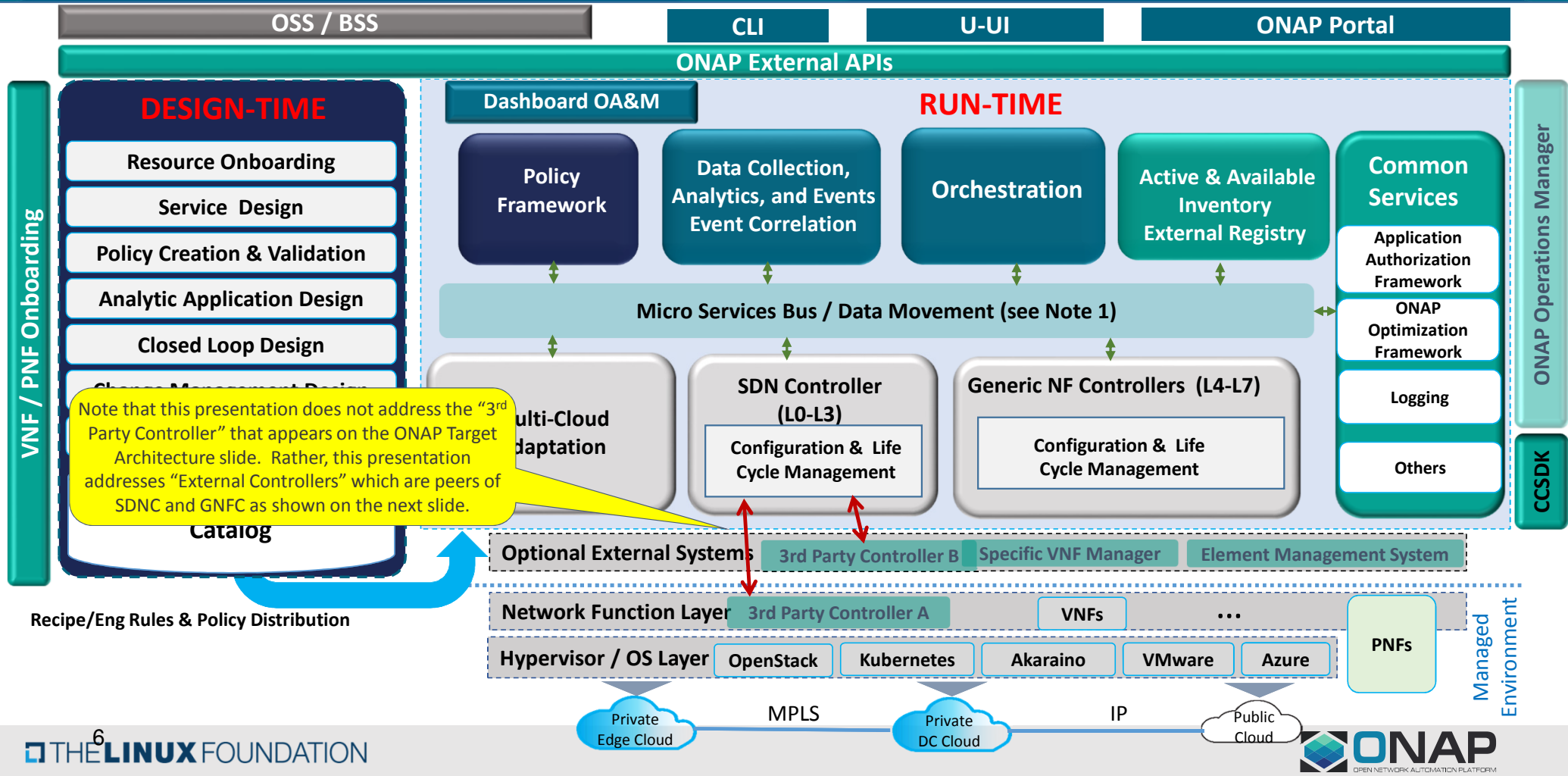
External Controller meets all run-time APIs for the relevant functions needed by the Resource(s) it manages

External Controller can also choose to meet ONAP design-time APIs.

Optional
 Configuration Templates
 Network / service Model
 Service Recipes
 external controller artifact design

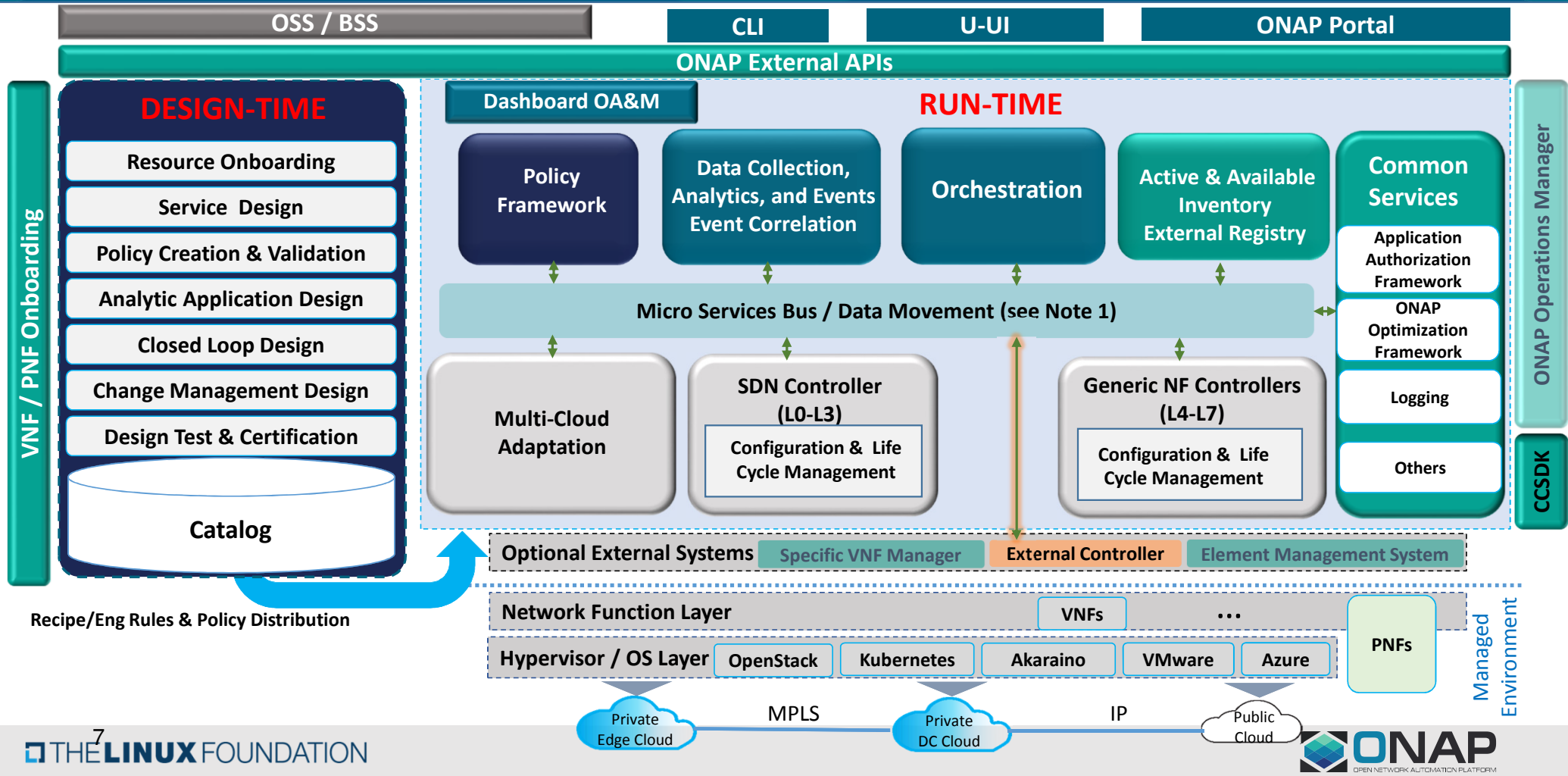
ONAP Target Architecture

(High-Level Functional View)



ONAP Target Architecture

(High-Level Functional View)



Next Steps: Define What Is an ONAP Controller

- In order for the proposed approach to work, it is necessary to publish a well-defined specification of the functionality that an ONAP Controller supports and the APIs it exposes to meet that functionality. Among other things, this definition should include the following points...
- What is the “scope” of an ONAP Controller:
 - ONAP components interact with ONAP Controllers primarily at the “Resource-level”, within a “Service context”.
 - ONAP Controllers expose Resource-level LCM APIs such as: configure NF, activate NF, restart NF, run NF health check, upgrade NF software, etc.
 - For example, SO interactions with Controllers are fundamentally at the Resource level, where a “Resource” is either of Network Function (a VNF, PNF, or Allotted Resource (“my share” of a NF) or a Network (e.g., “my share” of a WAN)
 - E.g., for an ONAP Service X that is comprised of 3 dedicated VNF types: A ,B, and C, a different GNFC type/instance could be used to configure each VNF type. The VNF A GNFC is able to configure that VNF in the context of the Service X.
 - Hence, ONAP Controllers are more properly thought of as being “Resource-level Controllers”. The locus of “Service-level Control” in ONAP is understood not to be a Controller but rather the collective of interacting components within ONAP itself.
- How does an SDNC differ from a GNFC:
 - The ONAP Target Architecture diagram represents ONAP Controllers as being of two basic “types”
 - We use the term “SDNC” for a ONAP Controller instance that manages Network Function Resources that provide L0-3 functionality and/or manages (non-Network Function) Resources that are Networks.
 - We use the term “GNFC” for an ONAP Controller instance that manages Network Function Resources that provide L4+ functionality.
 - Because both SDNC and Gen NFC perform common LCM functions for Network Function Resources, they should both be implemented from a common CCSDK (separate presentation).

What is an ONAP Controller? (Continued)

- The “special case” of “assign” (?):
 - It is only an SDNC that performs the “assign” functionality for all Network Functions (L0+) and Network
 - The “assign” function is called by Orchestration to ensure that ONAP knows the instance values for all of the “deployment data” attributes for that Resource. E.g., it is as a result of the “assign” interaction that SDNC assigns instance values for the IP Addresses that correspond to the external connection points of a given Network Function (e.g., VNF) or Network (e.g., WAN VLAN).
 - By “deployment data” we mean that set of data that must be configured at instantiation time in order for the Resource to be further managed (configured) by ONAP.
 - Note that ONAP hasn’t typically employed the term “deployment data” but rather this term is being adopted for this presentation from the ETSI common usage and is meant to convey the same meaning as ETSI assumes.
 - It is believed that in some cases this “assign” function could require service-level context that spans across multiple Resource instances within a given Service Instance.
 - For this reason, ONAP currently also supports a service-level “assign” interaction with SDNC. This was created due to the potential existence of “service-level” assignments. However, today not much is done with this call.
 - It is also for this reason that the same SDNC instance performs the “assign” function for all Resource instances for a given Service instance.
 - It is believed that this scope into the “service-level” space is unique to the “assign” function, and not shared for other LCM operations. E.g., for a given Service instance, a different Controller instance could be used to “configure” and “activate” each VNF instance that comprise that Service instance

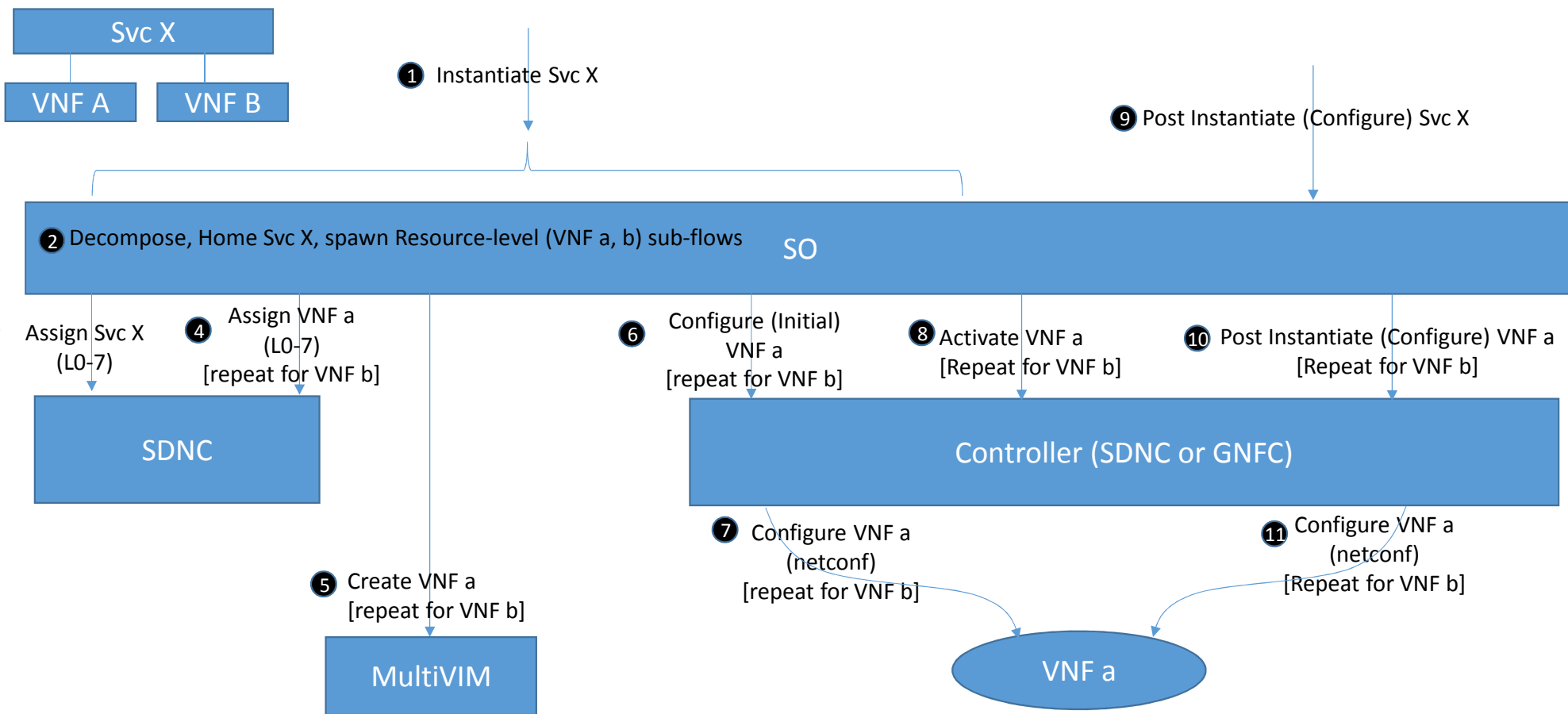


ONAP

OPEN NETWORK AUTOMATION PLATFORM

Backup Slides

ONAP Processing for Example Service "X" (Target Architecture Assumed)



SDNC “Assign” Function for Example Service Y

