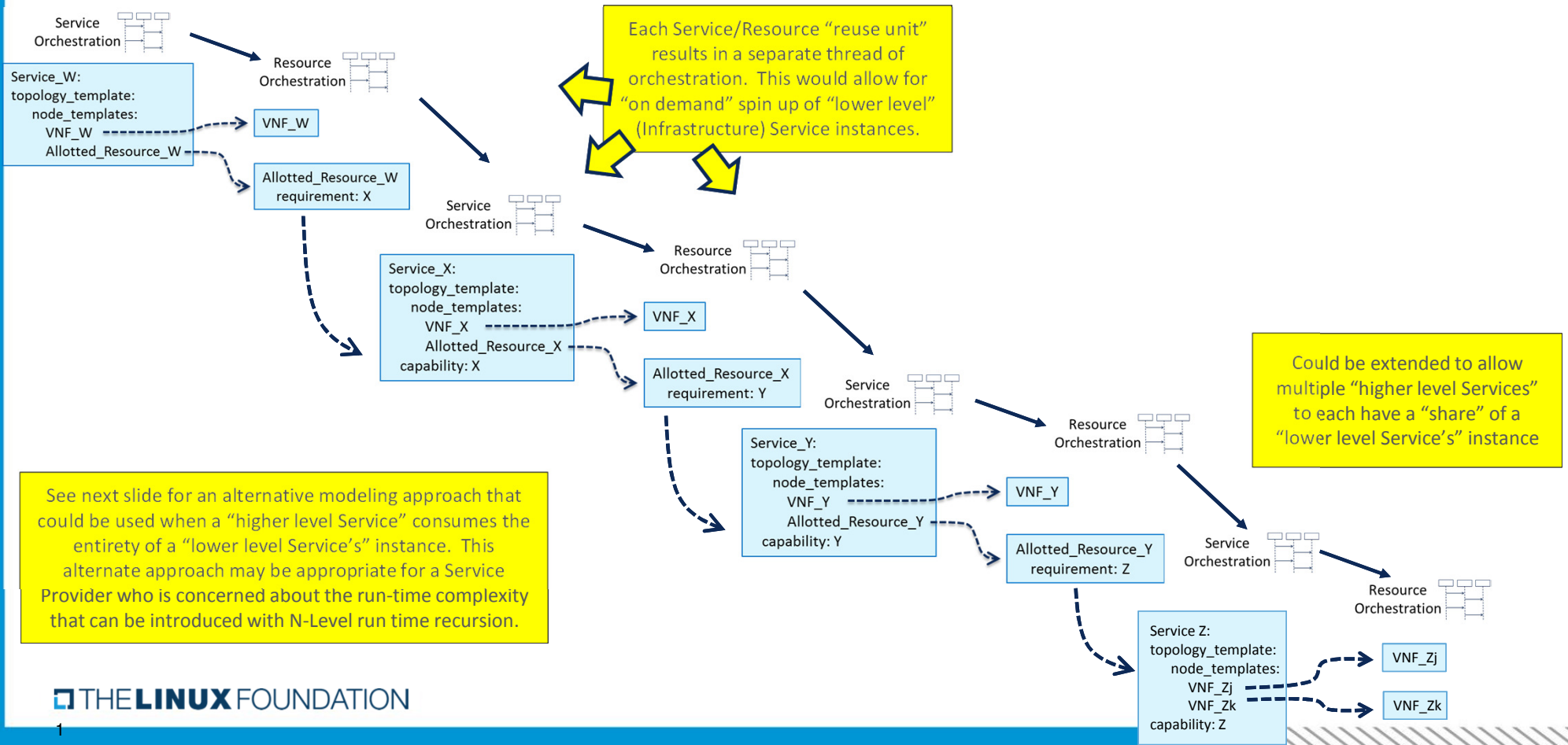


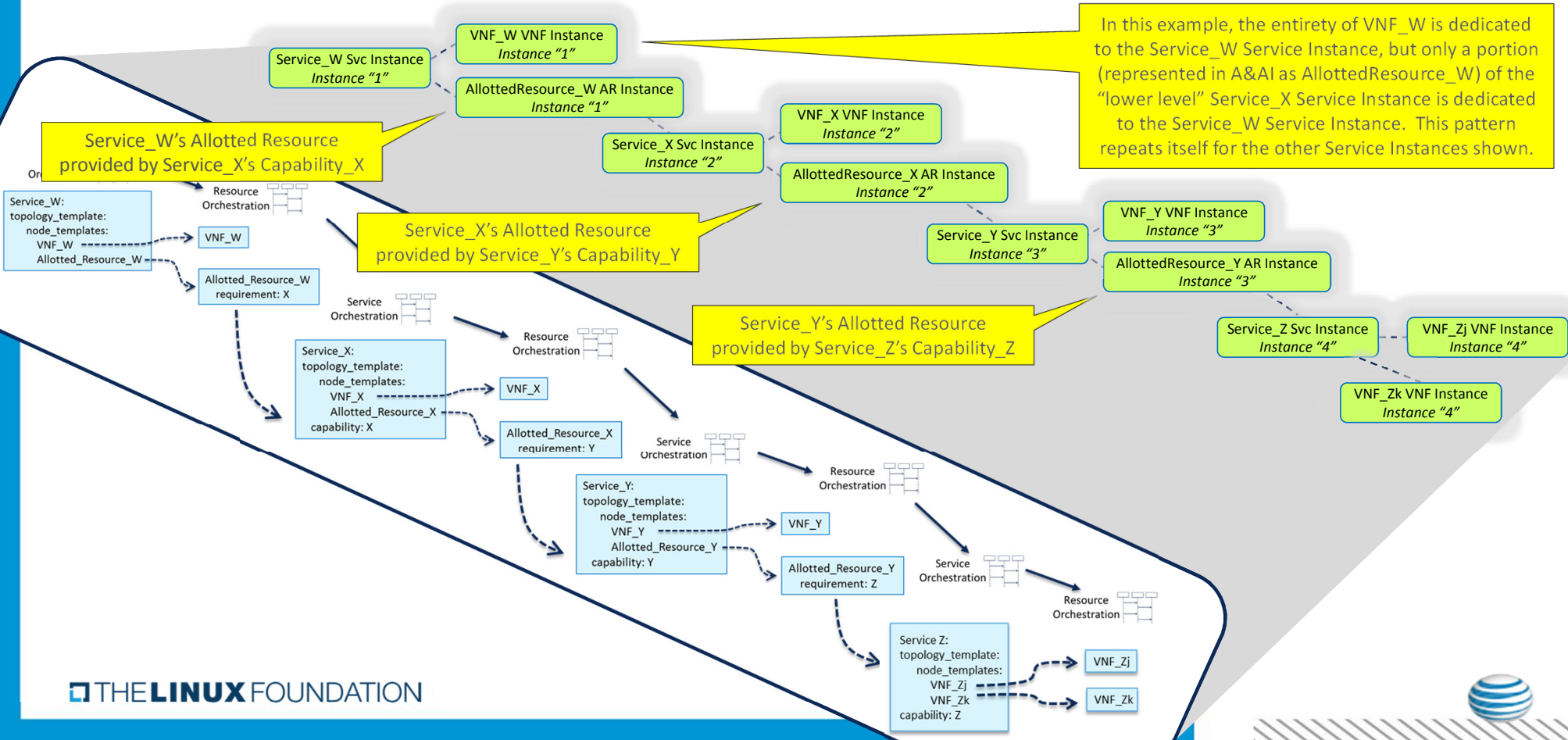
# N-Level Run Time Nesting? Let The Service Providers Decide

## Service\_W Modeling Example 1

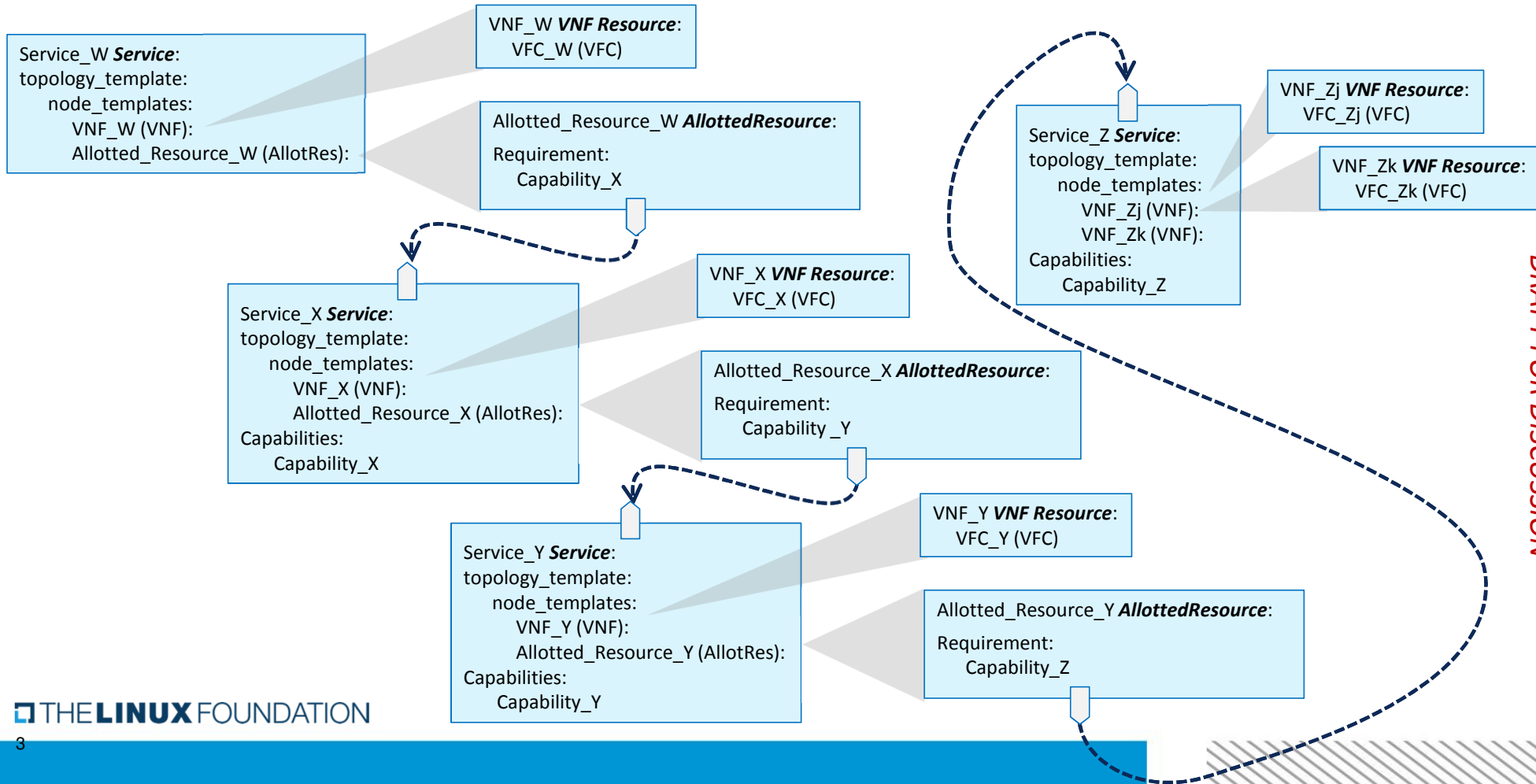


# A&AI Instance Representation of Service\_W Example 1

DRAFT FOR DISCUSSION

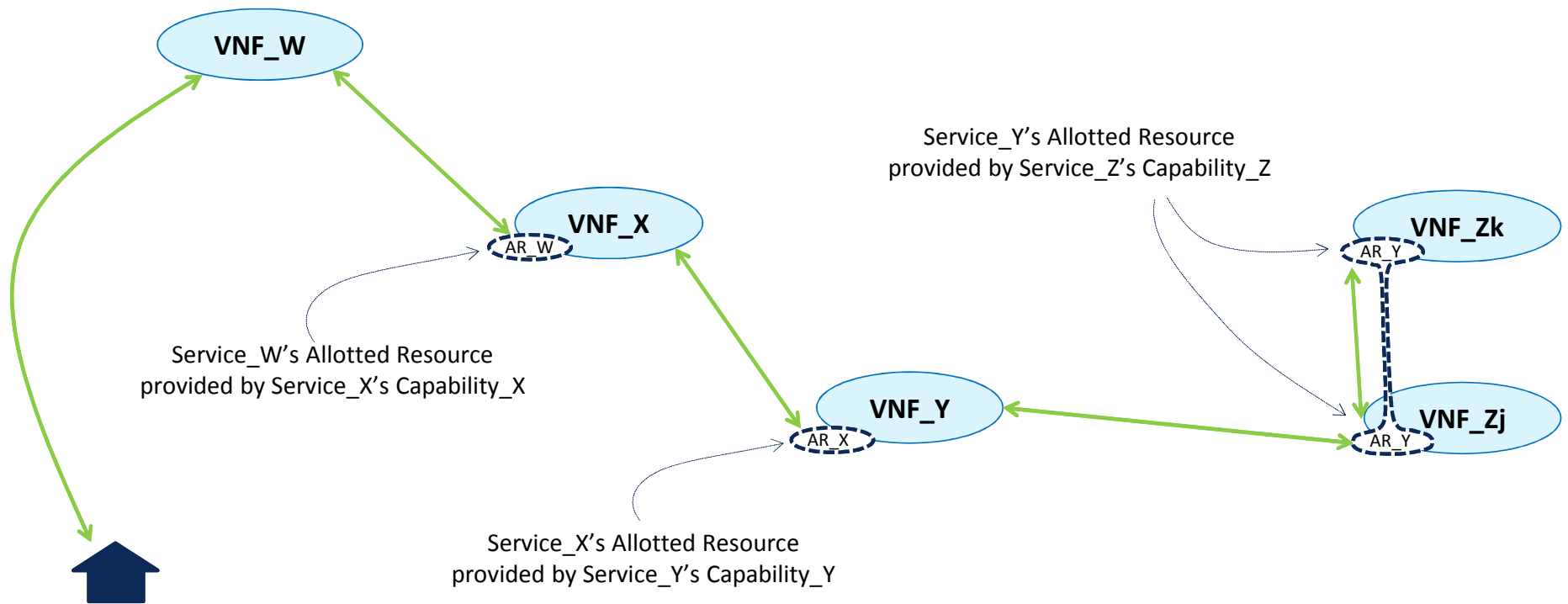


# Topological Model for Service\_W Example 1



DRAFT FOR DISCUSSION

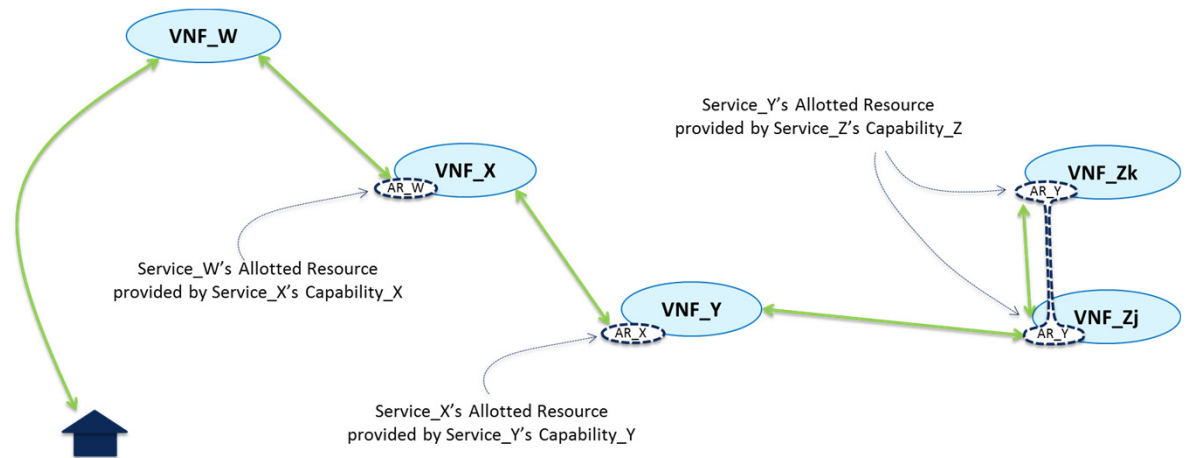
# “VNF Chaining” Data Flow for Service\_W Example 1



DRAFT FOR DISCUSSION

# Modeling Network Latency Homing Constraints for Allotted Resources

If Service\_W is sensitive to network latency between VNF\_W and the VNF\_X that hosts AR\_W, then the homing algorithm will need to select only VNF\_X instances that meet the Service\_W constraint. However, we don't want to write any homing (or any other) policies for Service\_W in terms of the internal structure of the underlying "lower order" Service type.

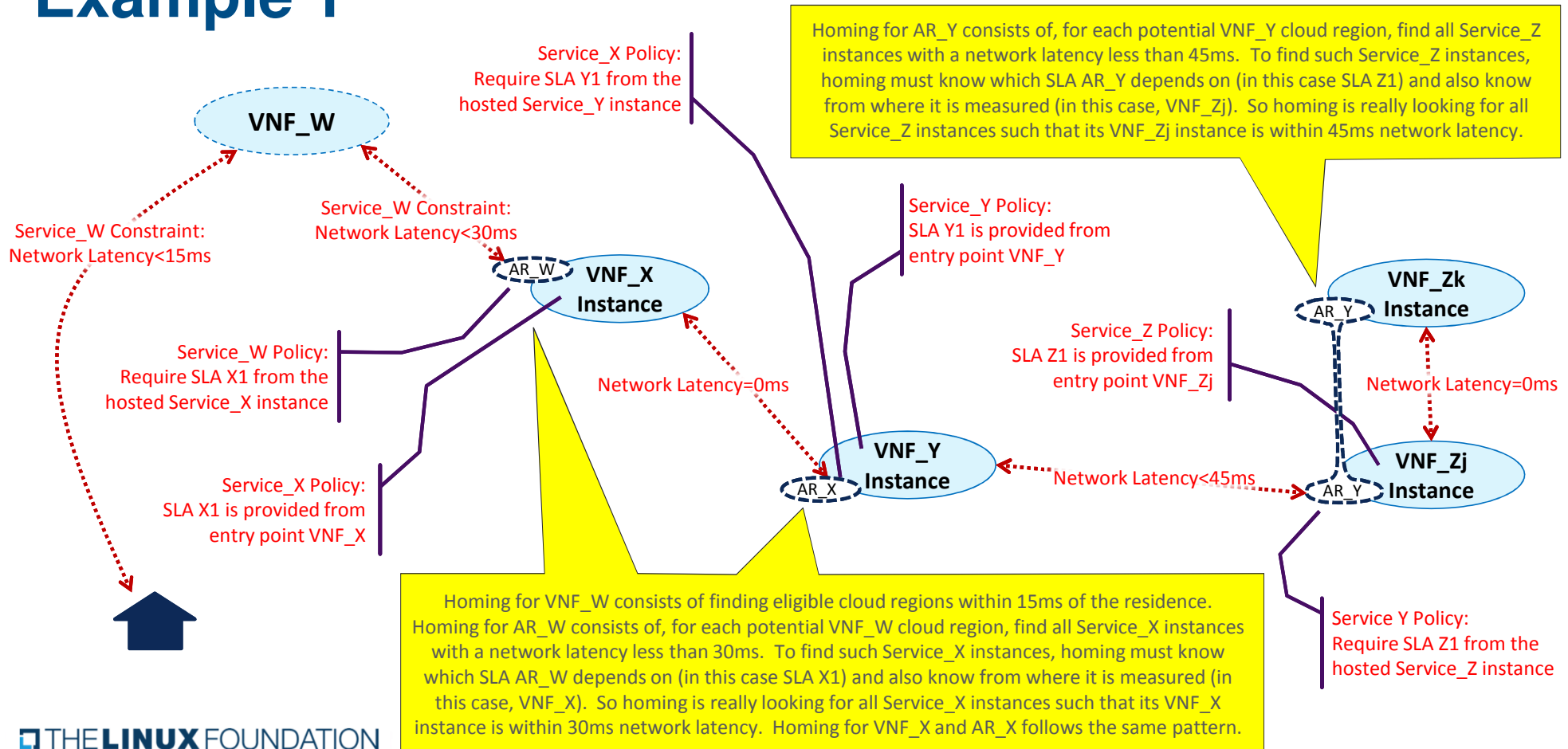


We can instead write the network latency constraint in terms of two policies, one a Service\_W policy and one a Service\_X policy.

Specifically, we will define the concept of an "SLA" that the lower order service will advertise. We will give the "higher order" Service a policy as to which SLA it requires from the "lower order" Service type. We will have the "lower order" Service type have a policy which indicates from which VNF the SLA is measured (mirroring the data path)

# Modeling Network Latency for Service\_W

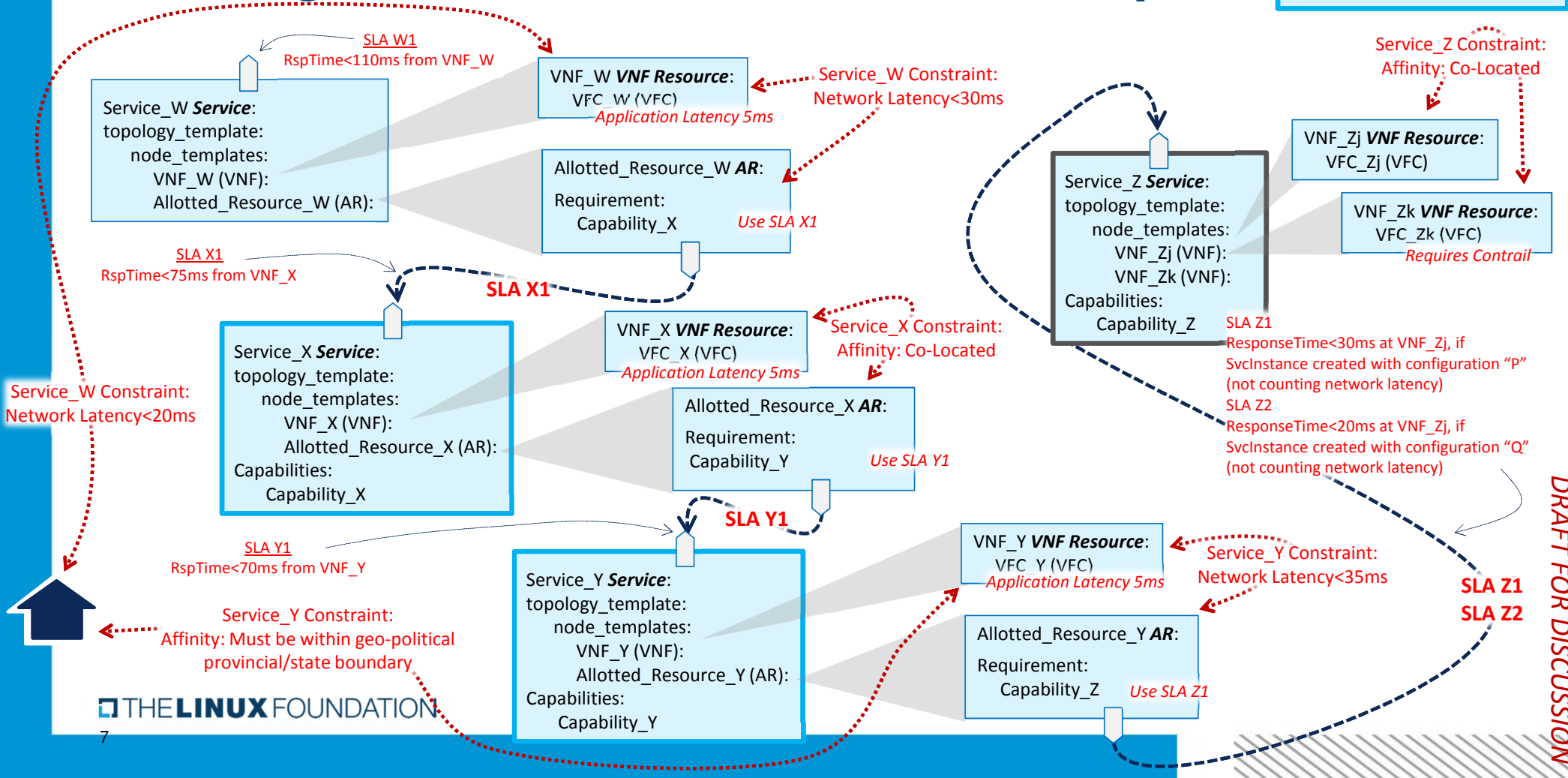
## Example 1



# Homing Policies for Service\_W Example 1

Key

"Lower Level Service Type" that can be instantiated in real time on an "on demand" basis





# Decomposition Structure for Service\_W Example 1

Svc Type	Rsc Type	AR Capab Svc	SLA Policies	Homing Constraints	Capab Svc Struct
Service_W			W1: RspTime<80ms from VNF_W	Ntw Latency: VNF_W <-> AR_W < 30ms	
Service_W	VNF_W			Ntw Latency:Residence <-> VNF_X < 15ms	
Service_W	AR_W	Service_X		Require SLA X1 from Service_X instance	

AR\_W Capab Svc Struct

Svc Type	Rsc Type	AR Capab Svc	SLA Policies	Homing Constraints	Capab Svc Struct
Service_X			X1: RspTime<45ms from VNF_Y	Affinity: VNF_X, AR_X Co-Located	
Service_X	VNF_X				
Service_X	AR_X	Service_Y	Require SLA Y1 from Service_Y instance		

AR\_X Capab Svc Struct

Svc Type	Rsc Type	AR Capab Svc	SLA Policies	Homing Constraints	Capab Svc Struct
Service_Y			Y1: RspTime<40ms from VNF_Y	Ntw Latency: VNF_Y <-> AR_Y < 45ms	
Service_Y	VNF_Y			Affinity: Residence, VNF_Y within state boundary {CA, OR, MA, RI, NH}	
Service_Y	AR_Y	Service_Z	Require SLA Z1 from Service_Z instance		

AR\_Y Capab Svc Struct

Svc Type	Rsc Type	AR Capab Svc	SLA Policies	Homing Constraints	Capab Svc Struct
Service_Z			Z1: <30ms with config "Q" Z2: <20ms with config "P"	Affinity: VNF_Zj, VNF_Zk Co-Located	
Service_Z	VNF_Zj				
Service_Z	VNF_Zk				



# Decomposition and Homing Approach

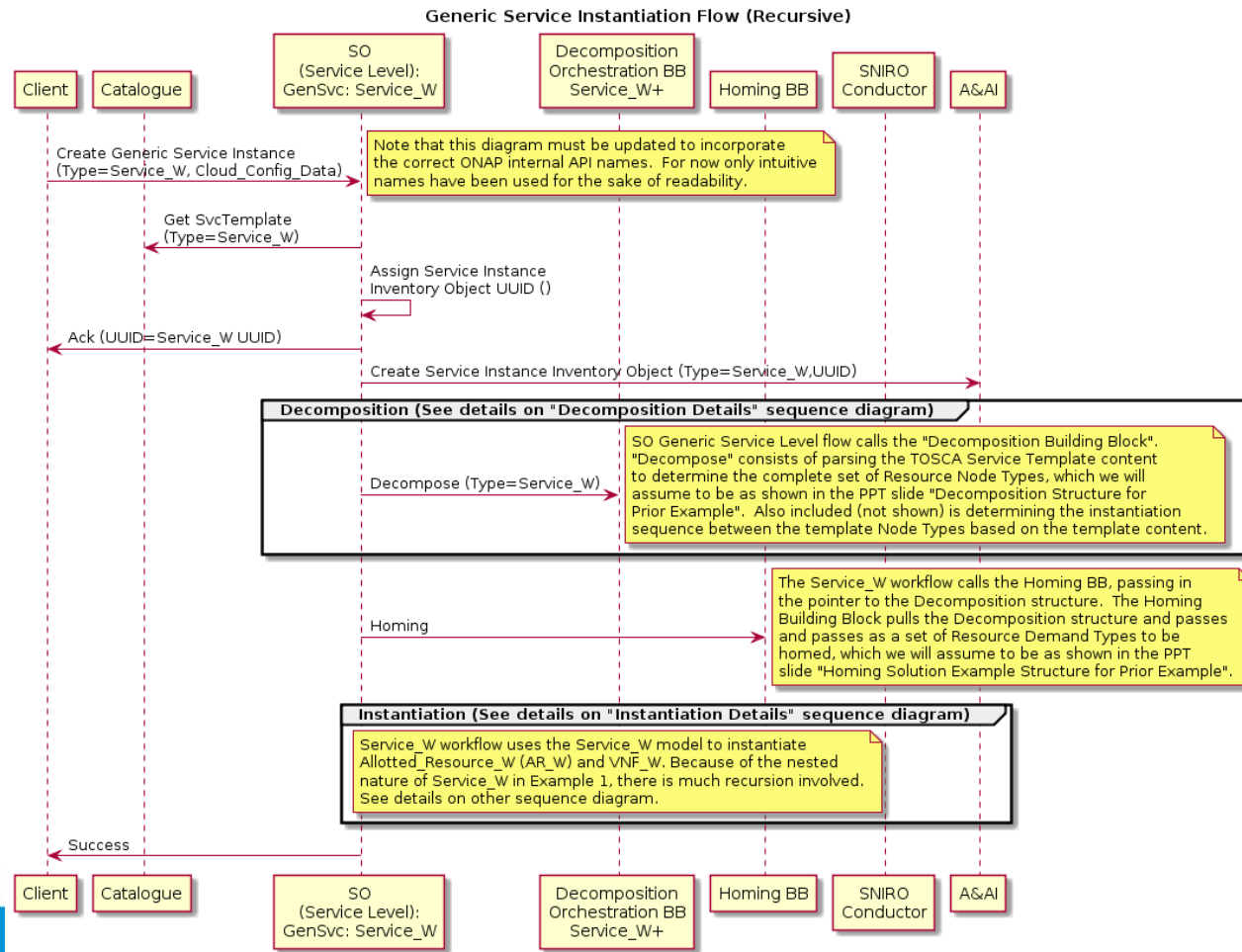
Note that, from a Service\_W perspective, the goal of homing is to find a Service\_X instance which meets the Service\_W “Ntw Latency” constraint of “VNF\_W <-> AR\_W < 30ms”. This would require decomposition to create the Service\_W rows in the decomposition example. If such service instance is found, then homing is complete. However, if no such Service\_X instance exists, homing can determine that a new one should be created “on demand.”

Creation of a new Service\_X instance would require decomposition of Service\_X (i.e., the Service\_X rows only) for a second homing attempt. From the Service\_X perspective, the goal of homing is to find a Service\_Y instance which meets the Service\_X “Affinity” constraint that “VNF\_X, AR\_X Co-Located” and such that the “Ntw Latency” constraint of “VNF\_W <-> AR\_W < 30ms” is also met. (Note that the network latency of AR\_W is measured from the Capability\_X SLA, which is in turn measured from VNF\_X.) Thus, in order to solve the Service\_X homing problem, consideration must be given to the Service\_W constraints. If homing finds no such Service\_Y instance, it can determine that a new one should be created “on demand.”

From this point the recursion pattern is set: for nested Services such that the “lower level” Services can be instantiated “on demand”, it is necessary to solve the homing problem holistically. Thus, we will opt in the subsequent slides for SO to do a full decomposition prior to a single homing attempt.



# Generic Service Level Flow for Service\_W Example 1

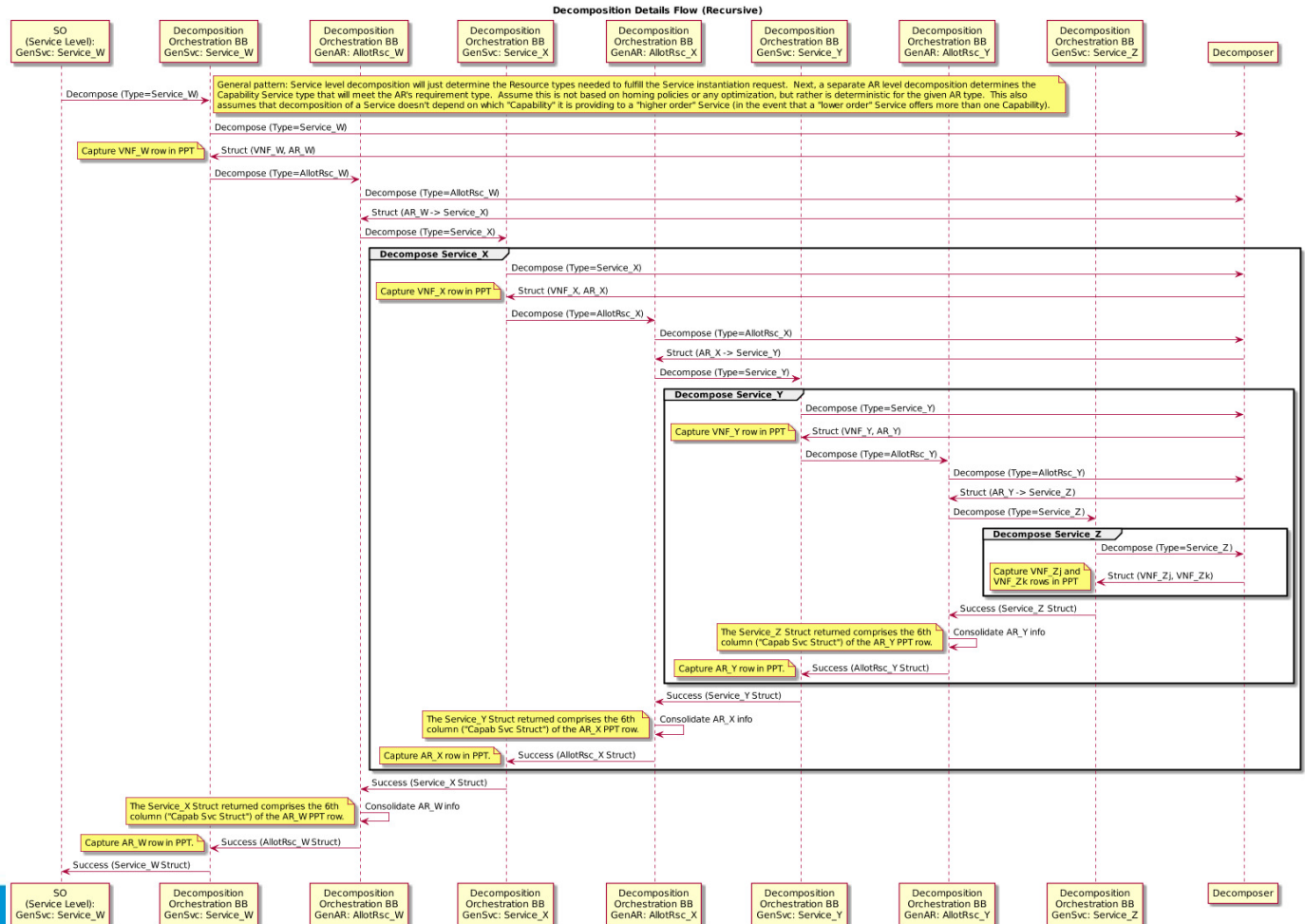


onap\_uc\_Generic\_Service\_Recursive\_p1.html



# Decomposition Detail Flow for Prior Example

DRAFT FOR DISCUSSION



onap\_uc\_Generic\_Service\_Decomposition\_p1.html

# Homing Solution Example for Service\_W Example 1

Service Type	Resource Type	Allotted Resource Capability Service	Capability Service Struct	Homing Solution
Service_W	VNF_W			Cloud_Region_1
Service_W	Allotted_Resource_W	Service_X		Instantiation_Needed

Service Type	Resource Type	Allotted Resource Capability Service	Capability Service Struct	Homing Solution
Service_X	VNF_X			Cloud_Region_2
Service_X	Allotted_Resource_X	Service_Y		Instantiation_Needed

Service Type	Resource Type	Allotted Resource Capability Service	Capability Service Struct	Homing Solution
Service_Y	VNF_Y			Cloud_Region_2
Service_Y	Allotted_Resource_Y	Service_Z		Service_Z_Instance_327

Service Type	Resource Type	Allotted Resource Capability Service	Homing Solution
Service_Z	VNF_Zj		As Exists
Service_Z	VNF_Zk		As Exists

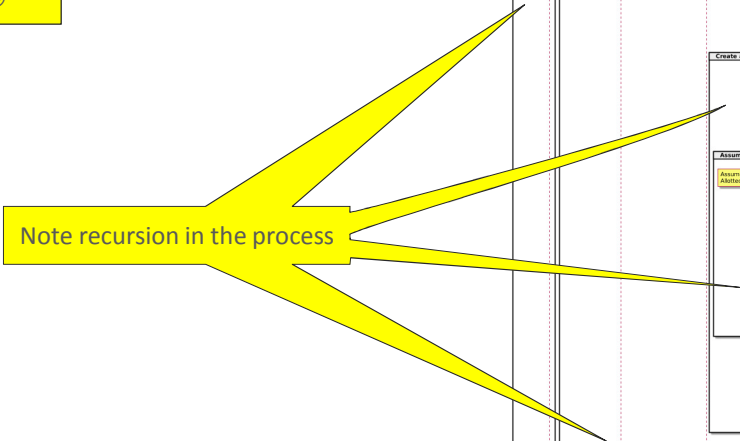
AR\_W Homing Structure

AR\_X Homing Structure

AR\_Y Homing Structure

# Instantiation Detail Flow for Service\_W Example 1

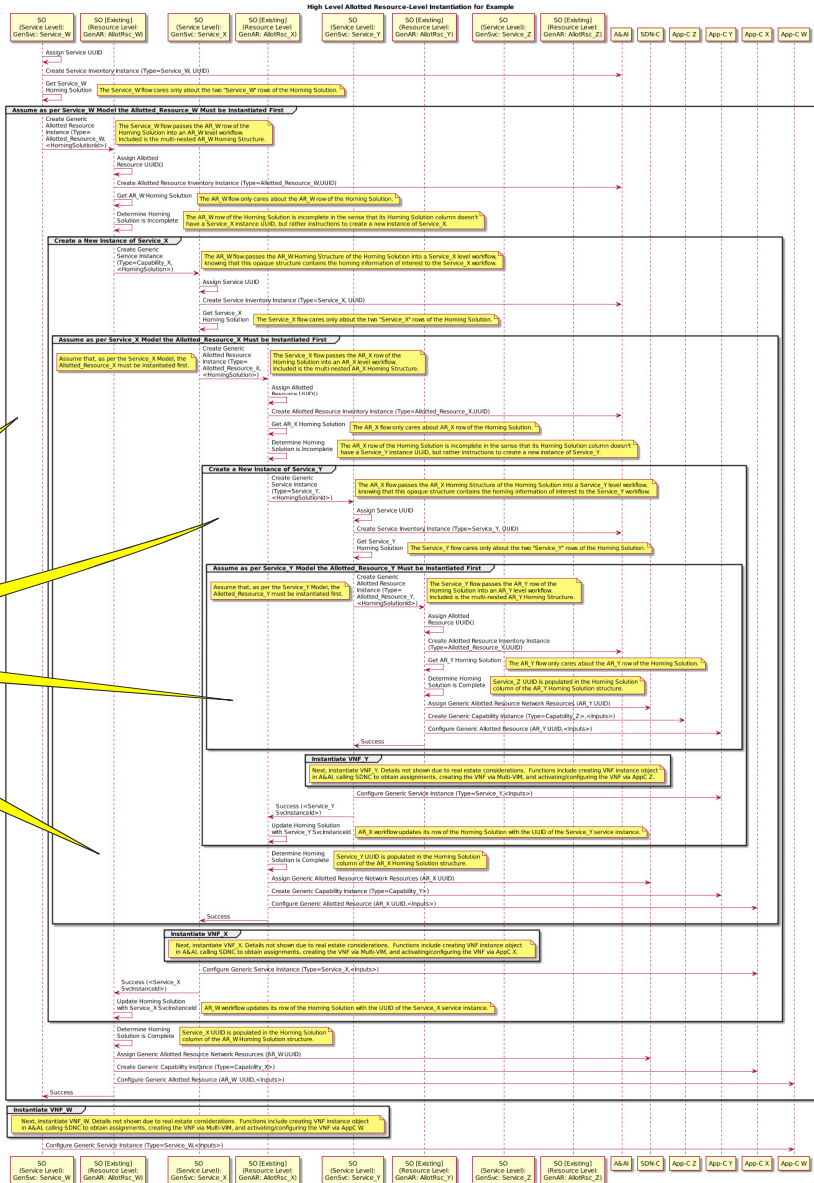
Scale PPT to 300% to view detail. 😊



Note recursion in the process



onap\_uc\_Generic\_Resource\_VNF\_Recursive.html



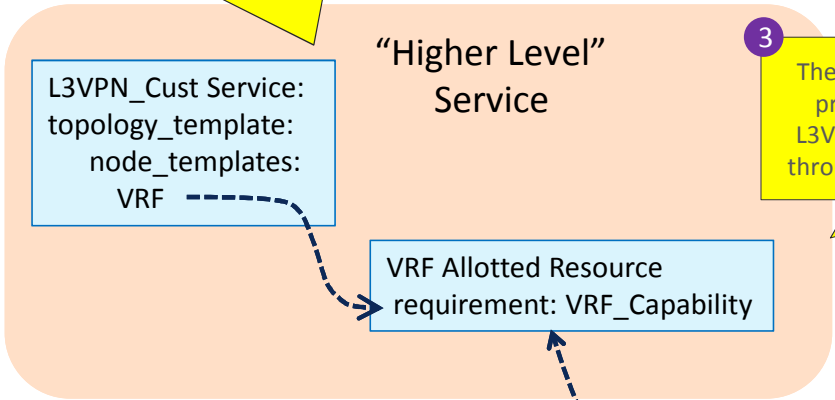
# Backup Slides



# Allotted Resources – vPE/VRF Example

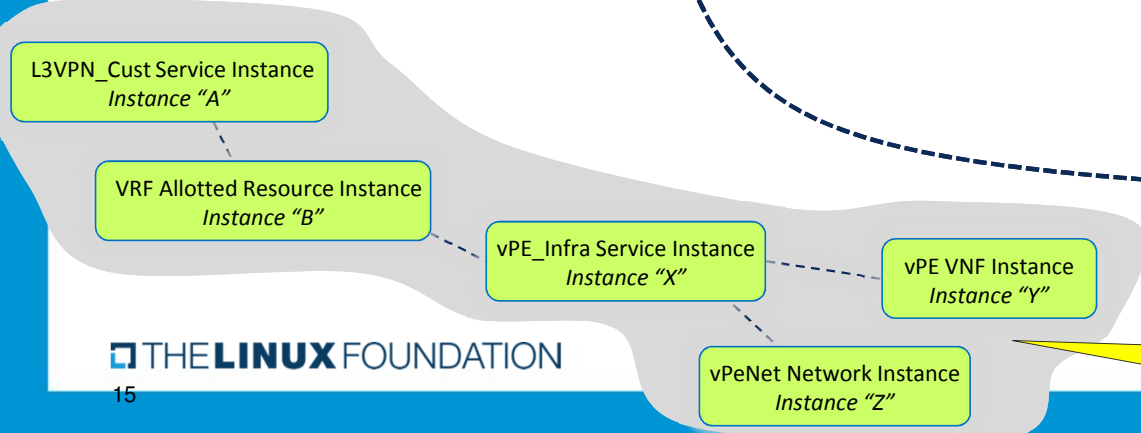
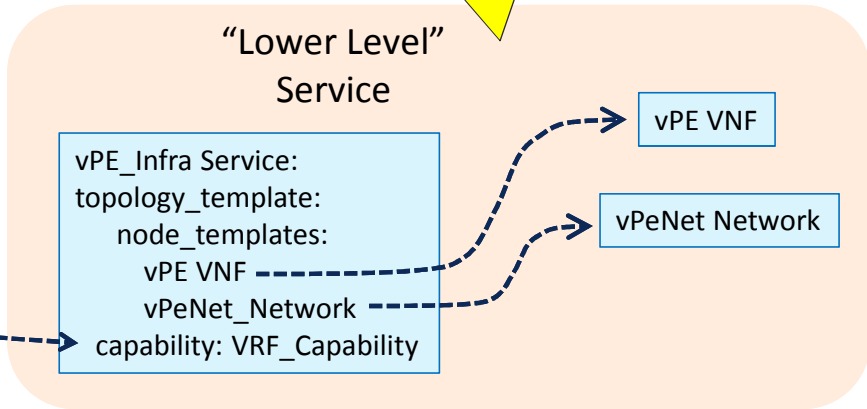
4 An instantiation request for a L3VPN\_Cust Service would result in a VRF being instantiated. That VRF would be “homed” to an existing vPE\_Infra Service instance (i.e., the vPE VNF instance on which this VRF will be configured).

1 Every Resource can be exposed as a Service. The ONAP model supports this today through the “Allotted Resource” construct. This concept of “Allotted Resource” does not seem to appear in the ETSI model. Perhaps this is due to ETSI seemingly covering only instantiation of Infrastructure Services, and not instantiation of end Customer Services.



3 The vPE\_Infra Service exposes a capability to provide “VRFs” (a “VRF\_Capability”). The L3VPN\_Cust Service consumes this capability through its “VRF Allotted Resource” construct.

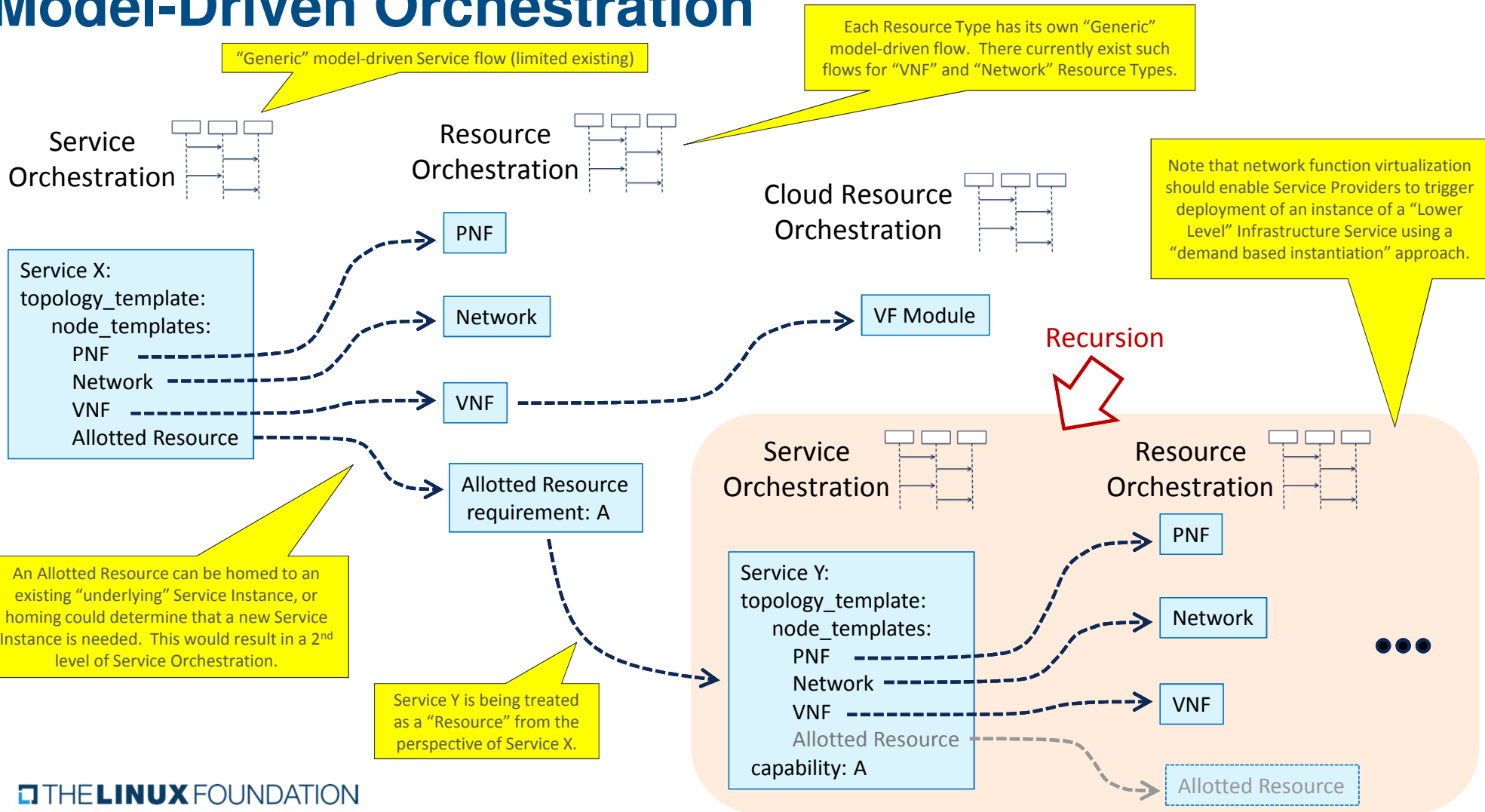
2 In this case, the vPE VNF has been packaged as an Infrastructure Service. An instantiation request for this vPE\_Infra Service would result in a new vPE VNF being instantiated.



4 In A&AI an actual instance object represents the Allotted Resource separate and distinct from the Services involved.



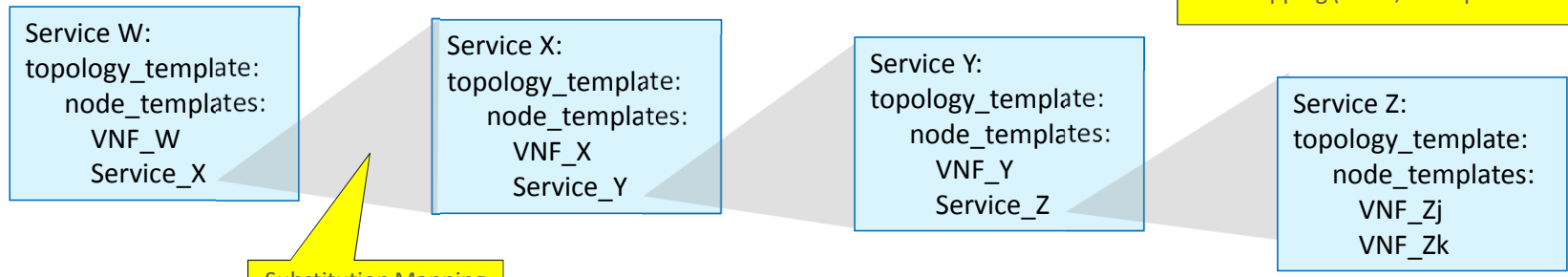
# Model-Driven Orchestration



# N-Level Run Time Nesting? Let The Service Providers Decide

## Service\_W Modeling Example 2

For the case whereby a “higher level Service” consumes the entirety of a “lower level Service’s” instance, SDC should support the Design Time ability to construct an “upper” Service Definition from other Services definitions via substitution mapping (a.k.a., “Compile Time Nesting”)



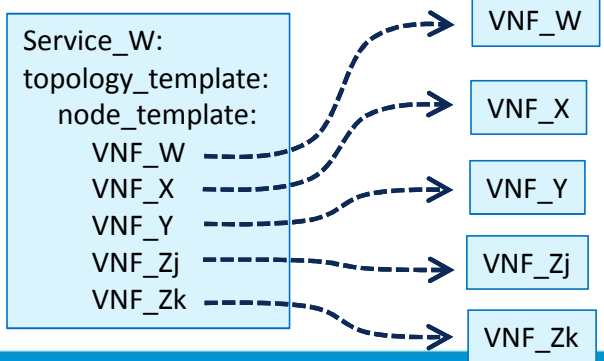
Substitution Mapping

Design Time

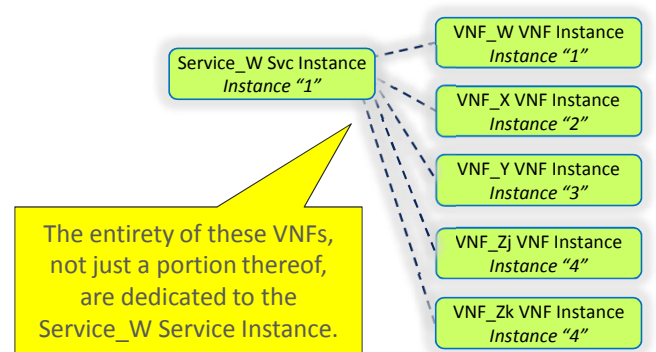
Distribution Time & Run Time



The “lower level Services” would not be visible at Distribution Time. Hence a “flattening” of the run-time orchestration would result.



A&AI Instance Representation of Service\_W Modeling Example 2



The entirety of these VNFs, not just a portion thereof, are dedicated to the Service\_W Service Instance.