Example Used In This Exercise*

- Based on the ONAP 5G wiki use case: https://wiki.onap.org/pages/viewpage.action?pageId=15992190

I am interpreting this diagram as "Service 1", "Service 2", and "Service 3" being instances of 3 separate Service types. To each of these "top level" Service Types, the NSI A and NSI B look like public networks. I.e., ONAP will not be asked (at least in this timeframe) to instantiate any "private" NSI "on the fly" for the use of any "top-level" Service instance.

I am filling in the "Service 3" definition by assuming that it "points to" both "NSI A" and "NSI B" (see red arrows left)

Though I understand that it is possible to have two NSSI:RAN **types**, for this use case example I am interpreting "NSSI:RAN1" and "NSSI:RAN2" as being **two instances** of the **same type**: "NSSI:RAN". I am interpreting "NSSI:Normal" and "NSSI:Premium" as instances of two separate types that use two separate sets of core network functions as shown below.

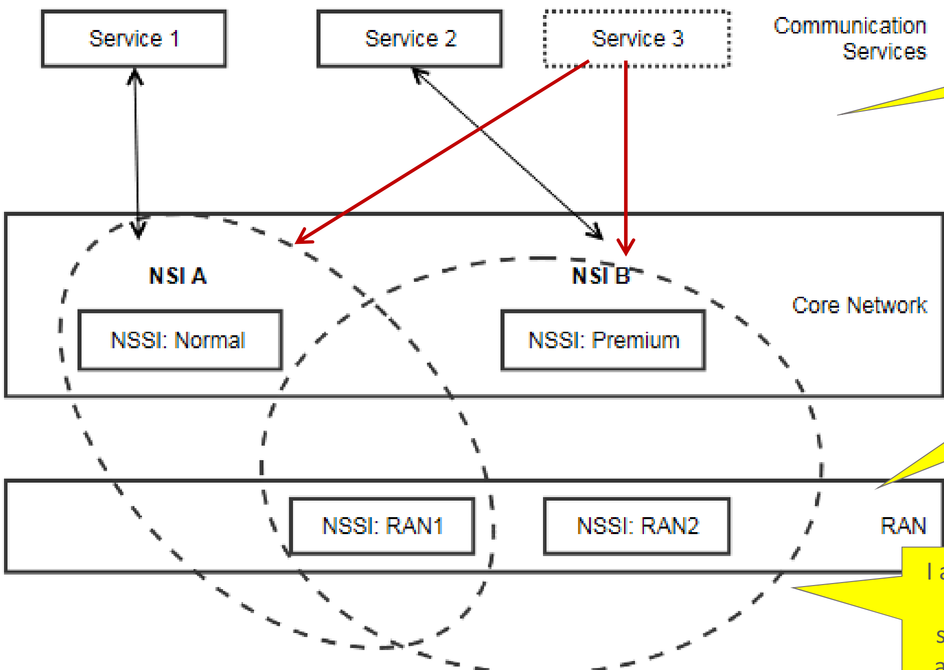
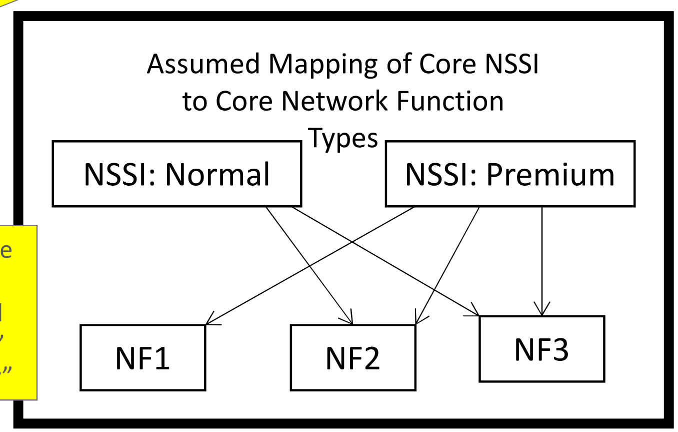
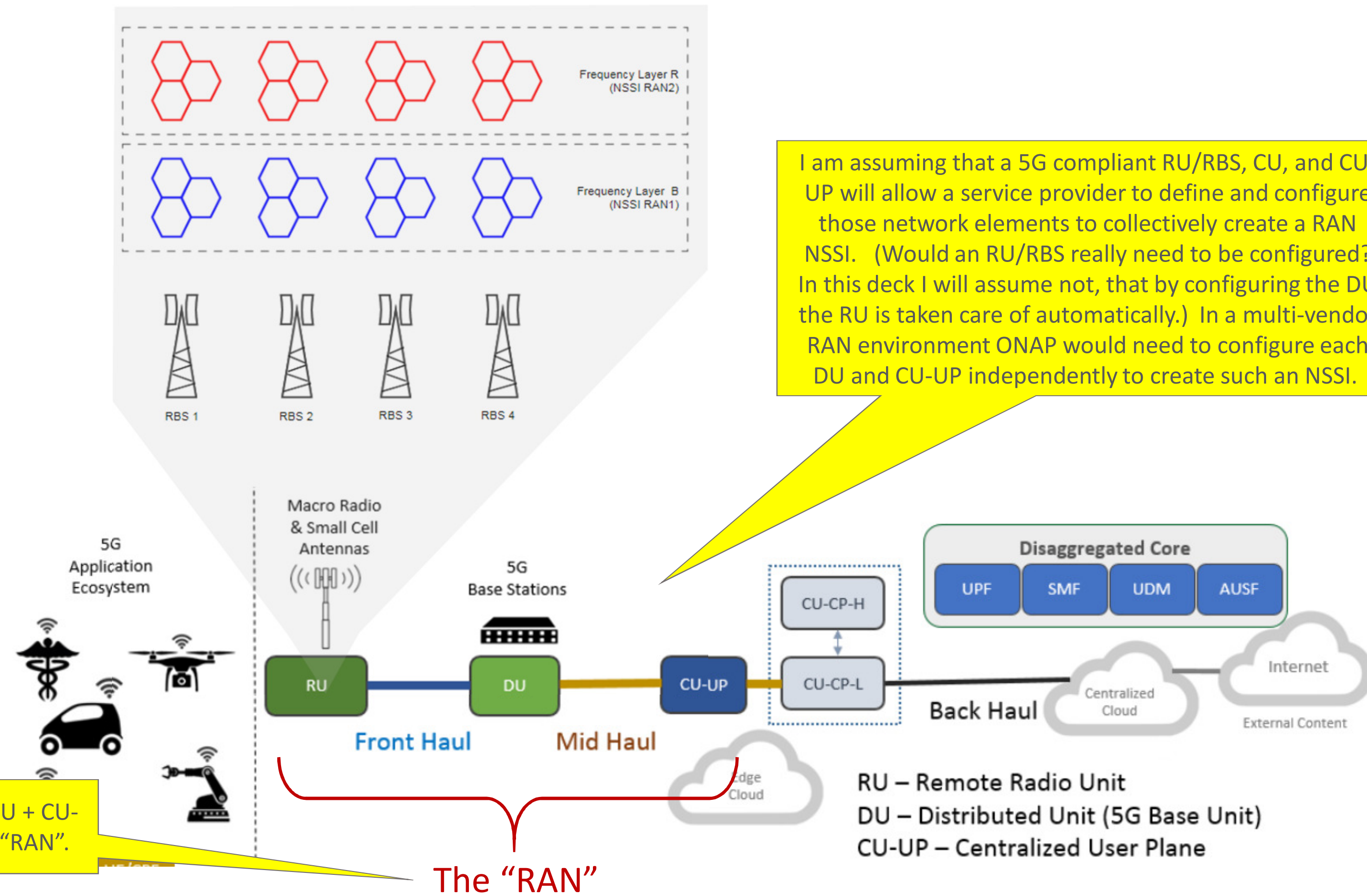


Figure 1: Illustration of how communication services ("Slice as a service") are implemented using NSIs which in turn use one or more NSSIs per network type (core and RAN).

I am interpreting the wiki use case as NSI:B requiring two slice segment instances of RAN. I will assume that one is for "Primary" use and the other for "Secondary"



Example Used In This Exercise (Continued)



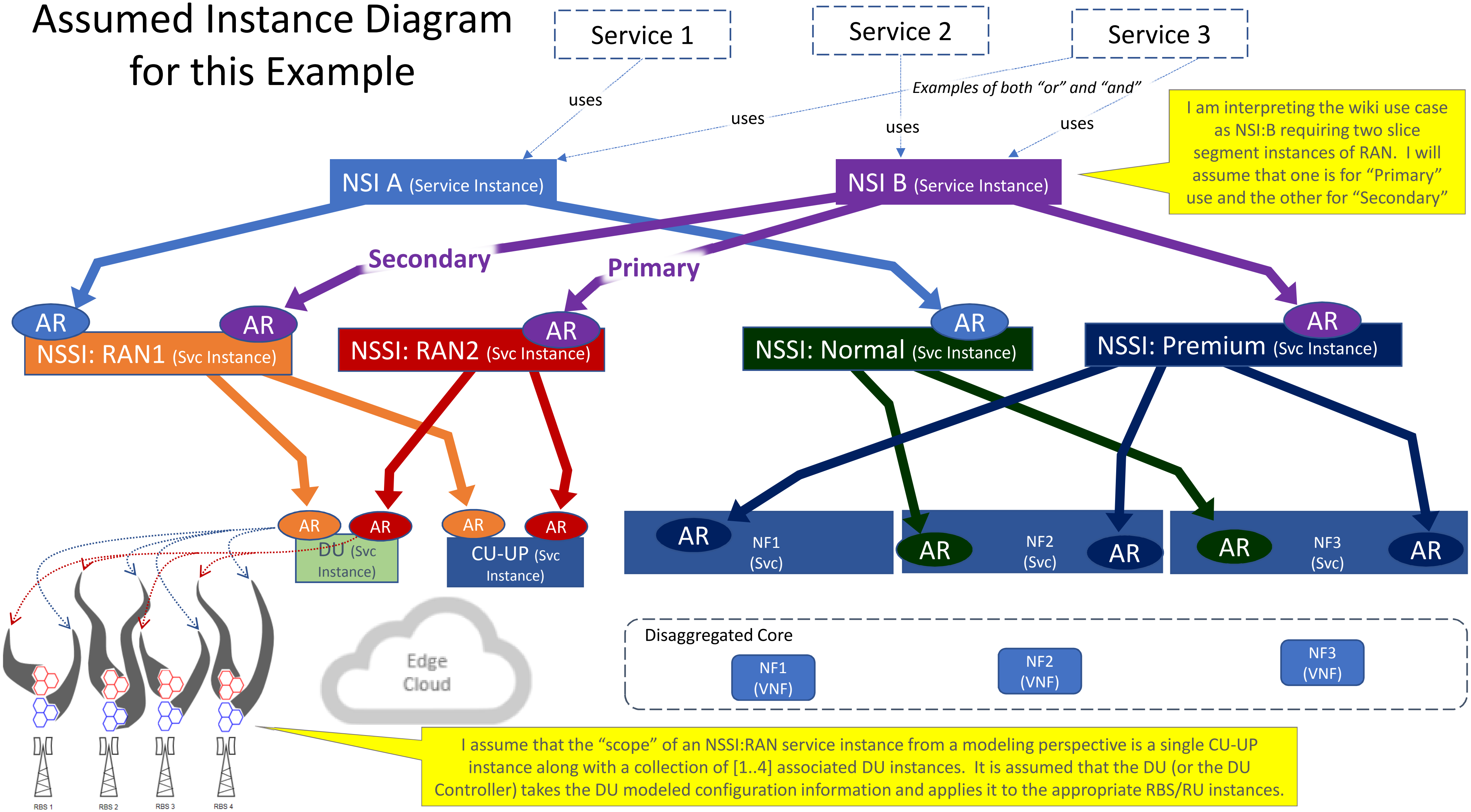
I am assuming that a 5G compliant RU/RBS, CU, and CU-UP will allow a service provider to define and configure those network elements to collectively create a RAN NSSI. (Would an RU/RBS really need to be configured? In this deck I will assume not, that by configuring the DU the RU is taken care of automatically.) In a multi-vendor RAN environment ONAP would need to configure each DU and CU-UP independently to create such an NSSI.

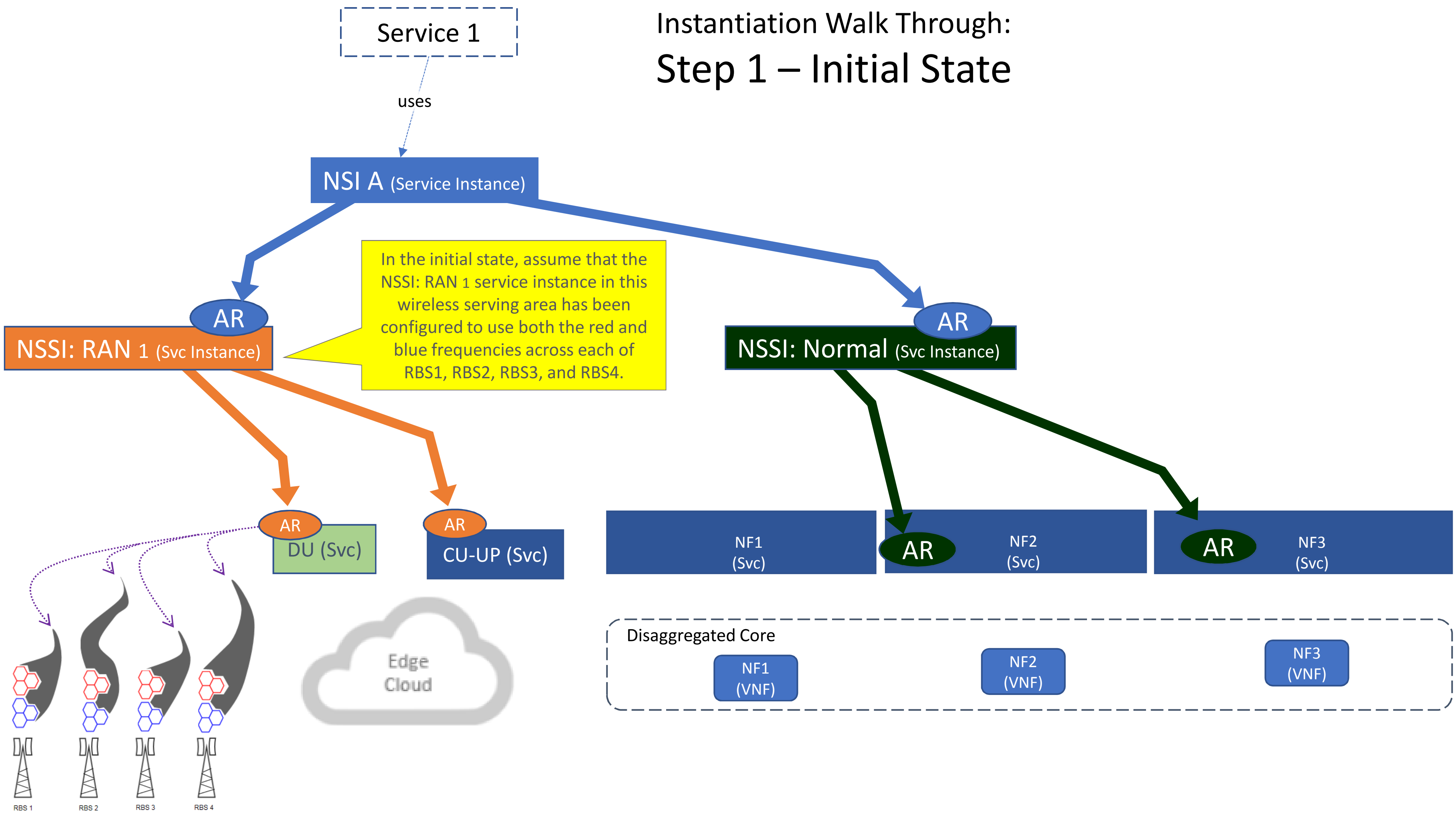
I am considering the RU/RBS + DU + CU-UP to collectively comprise the "RAN".

Example Used In This Exercise (Continued) – Other Assumptions

1. Vendors Provide Descriptors Which Can Be Mapped into NSSI Types:
   The network element vendor will provide a descriptor that includes the configuration parameters that drive the network behavior of the NSSIs configured on that device. These descriptors will be onboarded into SDC.
2. Defining NSSI Types:
   SDC will allow the designer to define new "NSSI **Types**", each type driving a different network behavior. For example a designer may define an NSSI type of "Low Latency RAN NSSI". In the example used in this slide, it is assumed that an SDC designer has created an NSSI type of "NSSI:RAN". It is assumed that
3. Defining NSSI Type Data Mappings:
   When instantiating an NSSI instance at **run time**, the ONAP Controller must populate the values for the network element configuration parameters (see #1 above) appropriately to obtain that desired network behavior for the NSSI instance. To facilitate this, it is necessary that at **design time** the NSSI Type designer must specify the input attribute **tags** and associated **allowed values** that must be populated at **run time** in an NSSI instantiation request. The designer must also specify the mapping of those input data attribute values to these per-network element configuration parameter values, or specify algorithms for deriving how to populate those values (e.g., default values to populate). For example, as part of the type definition of the "NSSI:RAN" type used in this example, the designer will be assumed to specify (among other things) that that the run-time user must identify the set of RBSs across which this NSSI instance will span (i.e., RBS1, RBS2, RBS3, and RBS4 in our example) and the frequencies associated with each of those RBS (i.e., red, blue). (See #6 below.)
4. Network Elements Support Multiple Slices:
   Each RAN network element can be configured to support multiple NSSI (RAN) instances. So from the RAN network element's perspective, when asked to create (its portion of) an NSSI (RAN) instance, it is in fact being asked to provide an "allotment" of its own functionality for use by this NSSI.
5. Slices and Capacity Sharing
   The network element doesn't necessarily reserve dedicated capacity (e.g., radio frequency) for a given NSSI that it supports, but it could. E.g., in the use case it is spelled out which frequencies the NSSI:RAN1 and NSSI:RAN2 instances use, and these frequencies (red/blue) differ between the two "NSSI:RAN" instances. The implication of this is that the radio frequencies to use for a particular NSSI:RAN instance can be configured. Thus I assume that at **design time**, the NSSI's "**Type**" could capture the attribute **tags** used to describe the radio frequency association. At **instantiation time**, the specific attribute **values** would be populated which indicate the actual frequencies associated with that NSSI instance. I assume that it is also possible to define NSSI:RAN types or instances that do not have dedicated radio frequencies set at instantiation time, but that the radio frequencies are shared. To demonstrate this, I am including a sub-variation (2b) in this exercise whereby NSSI:RAN1 and NSSI:RAN2 share the red frequencies.

Assumed Instance Diagram for this Example

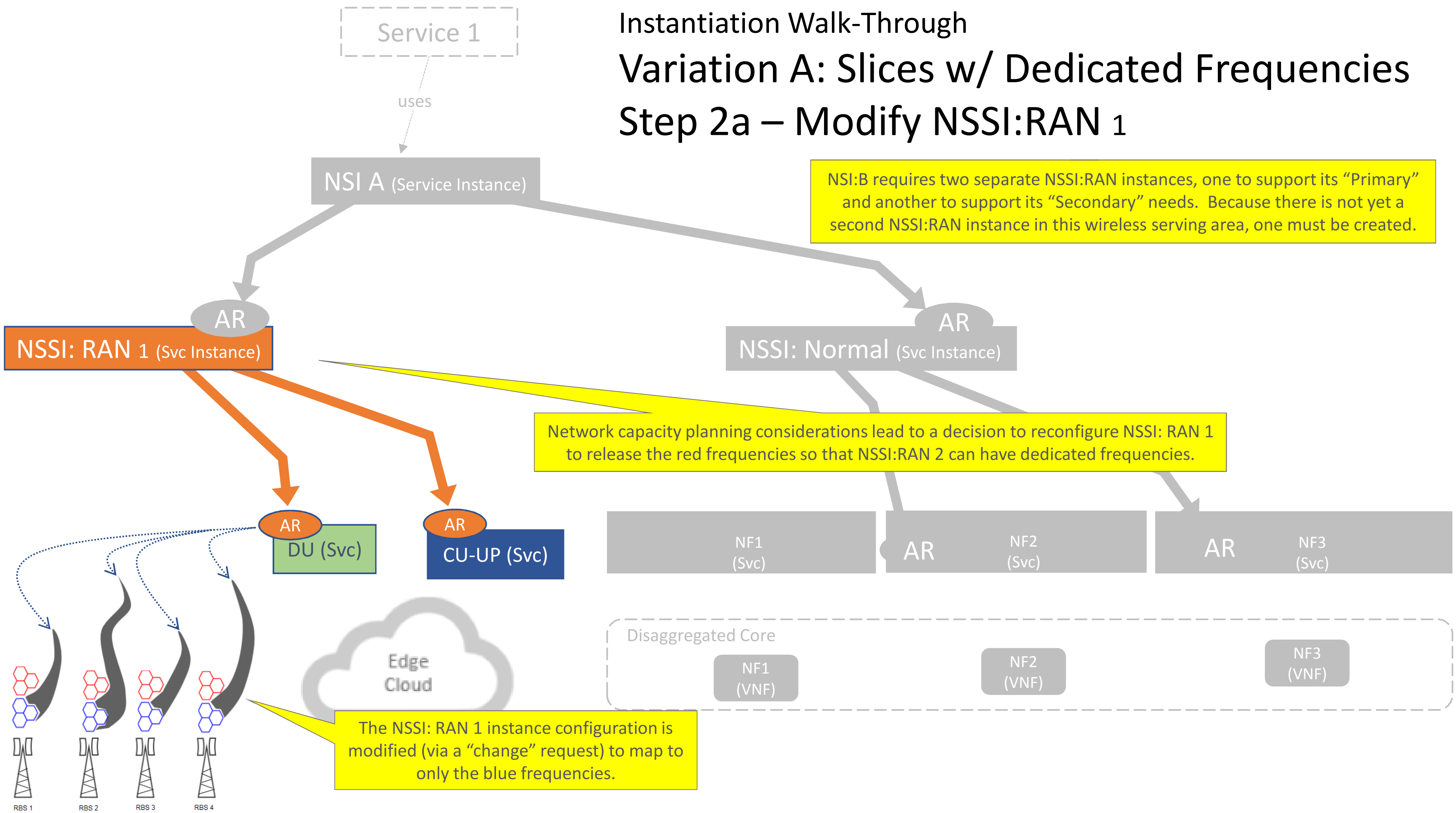



Instantiation Walk Through:
Step 1 – Initial State
Service 1
uses
NSI A (Service Instance)
AR
NSSI: RAN 1 (Svc Instance)
In the initial state, assume that the NSSI: RAN 1 service instance in this wireless serving area has been configured to use both the red and blue frequencies across each of RBS1, RBS2, RBS3, and RBS4.
AR
NSSI: Normal (Svc Instance)
AR
DU (Svc)
AR
CU-UP (Svc)
NF1
(Svc)
AR
NF2
(Svc)
AR
NF3
(Svc)
Edge
Cloud
Disaggregated Core
NF1
(VNF)
NF2
(VNF)
NF3
(VNF)
RBS 1
RBS 2
RBS 3
RBS 4

Instantiation Walk-Through

Variation A: Slices w/ Dedicated Frequencies

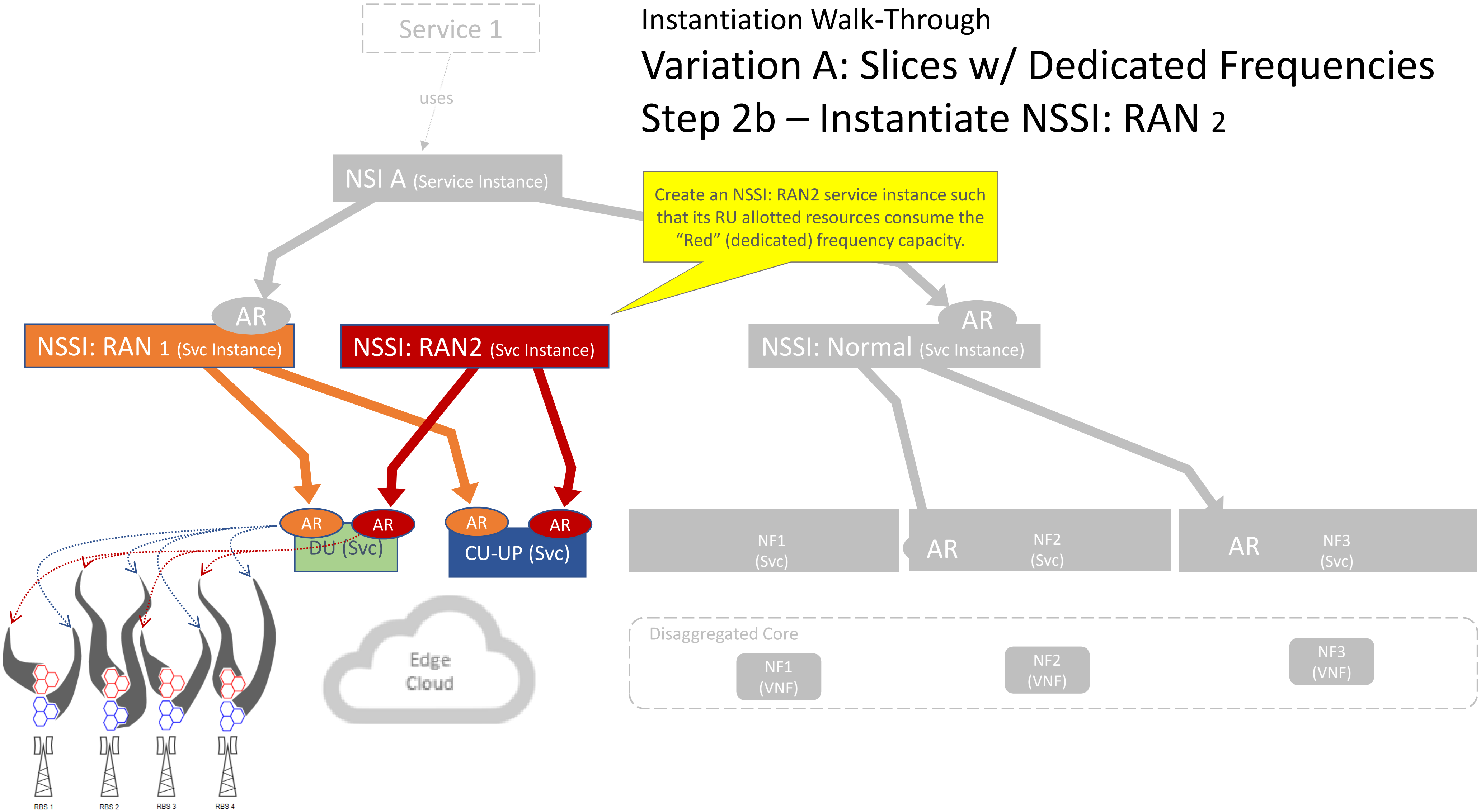
Step 2a – Modify NSSI:RAN 1

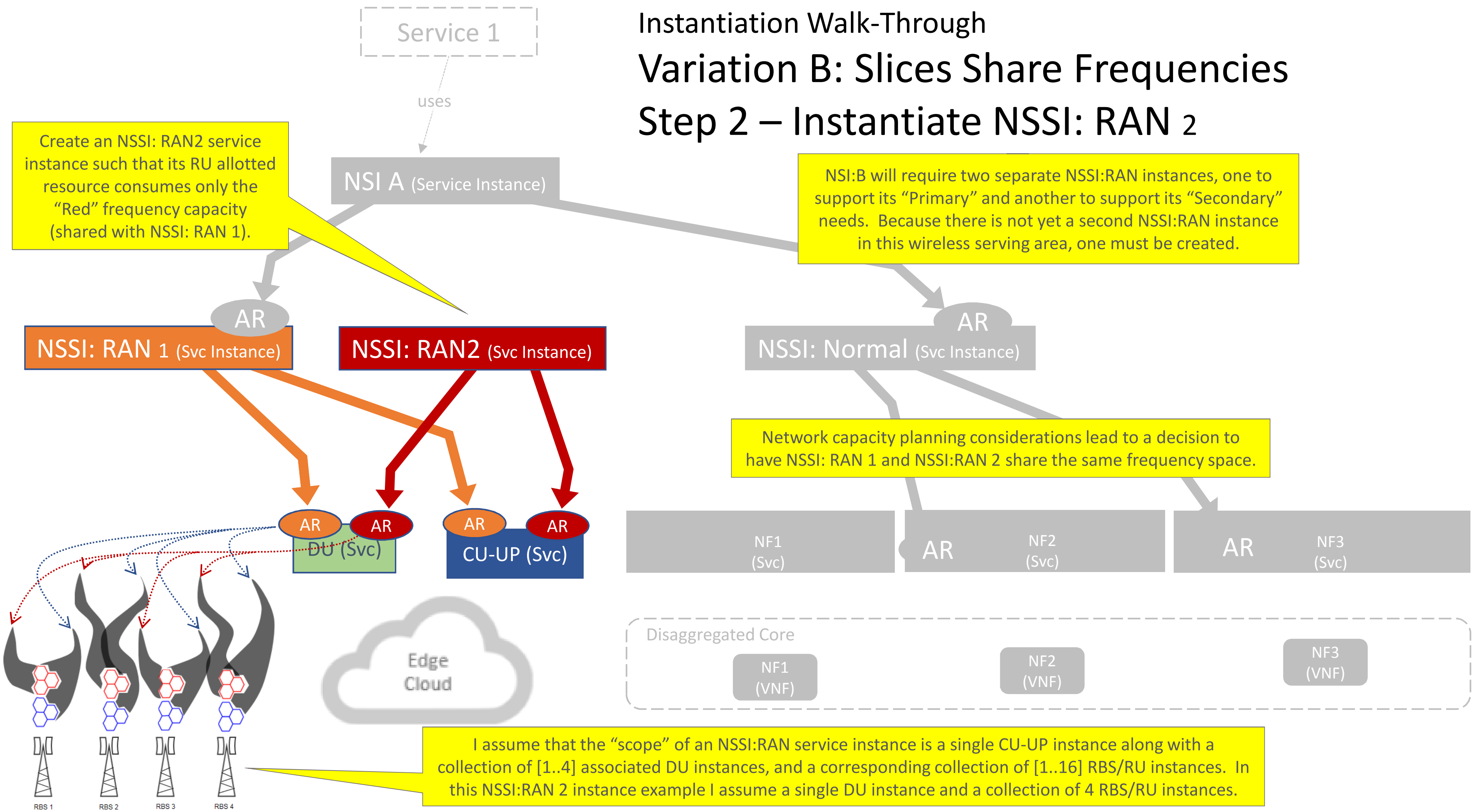


Instantiation Walk-Through

Variation A: Slices w/ Dedicated Frequencies

Step 2b – Instantiate NSSI: RAN 2

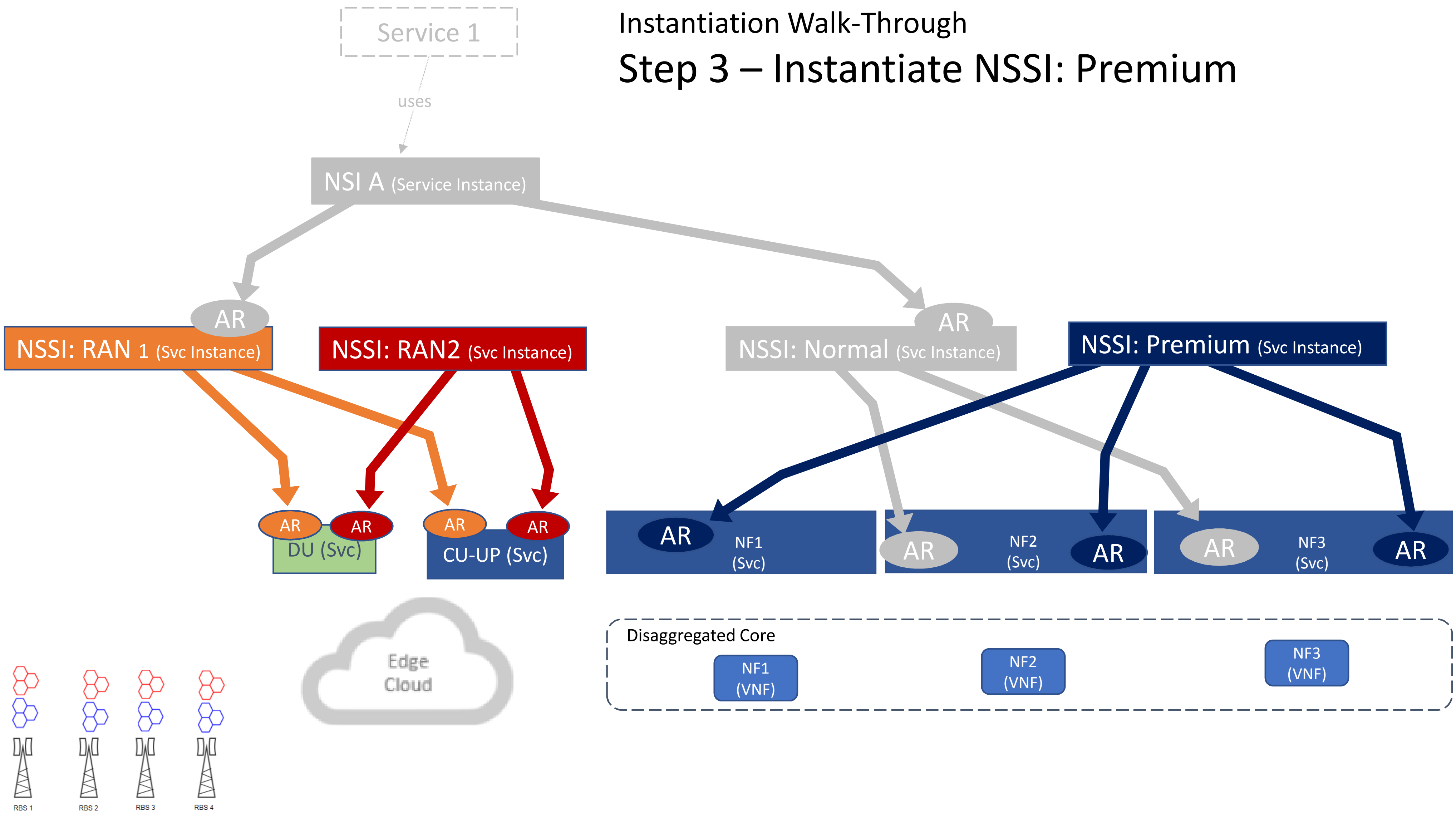




Instantiation Walk-Through
Variation B: Slices Share Frequencies
Step 2 – Instantiate NSSI: RAN 2
Service 1
uses
NSI A (Service Instance)
Create an NSSI: RAN2 service instance such that its RU allotted resource consumes only the "Red" frequency capacity (shared with NSSI: RAN 1).
NSI:B will require two separate NSSI:RAN instances, one to support its "Primary" and another to support its "Secondary" needs. Because there is not yet a second NSSI:RAN instance in this wireless serving area, one must be created.
AR
NSSI: RAN 1 (Svc Instance)
NSSI: RAN2 (Svc Instance)
AR
NSSI: Normal (Svc Instance)
Network capacity planning considerations lead to a decision to have NSSI: RAN 1 and NSSI:RAN 2 share the same frequency space.
AR
AR
AR
AR
DU (Svc)
CU-UP (Svc)
NF1 (Svc)
AR
NF2 (Svc)
AR
NF3 (Svc)
Edge Cloud
Disaggregated Core
NF1 (VNF)
NF2 (VNF)
NF3 (VNF)
RBS 1
RBS 2
RBS 3
RBS 4
I assume that the "scope" of an NSSI:RAN service instance is a single CU-UP instance along with a collection of [1..4] associated DU instances, and a corresponding collection of [1..16] RBS/RU instances. In this NSSI:RAN 2 instance example I assume a single DU instance and a collection of 4 RBS/RU instances.

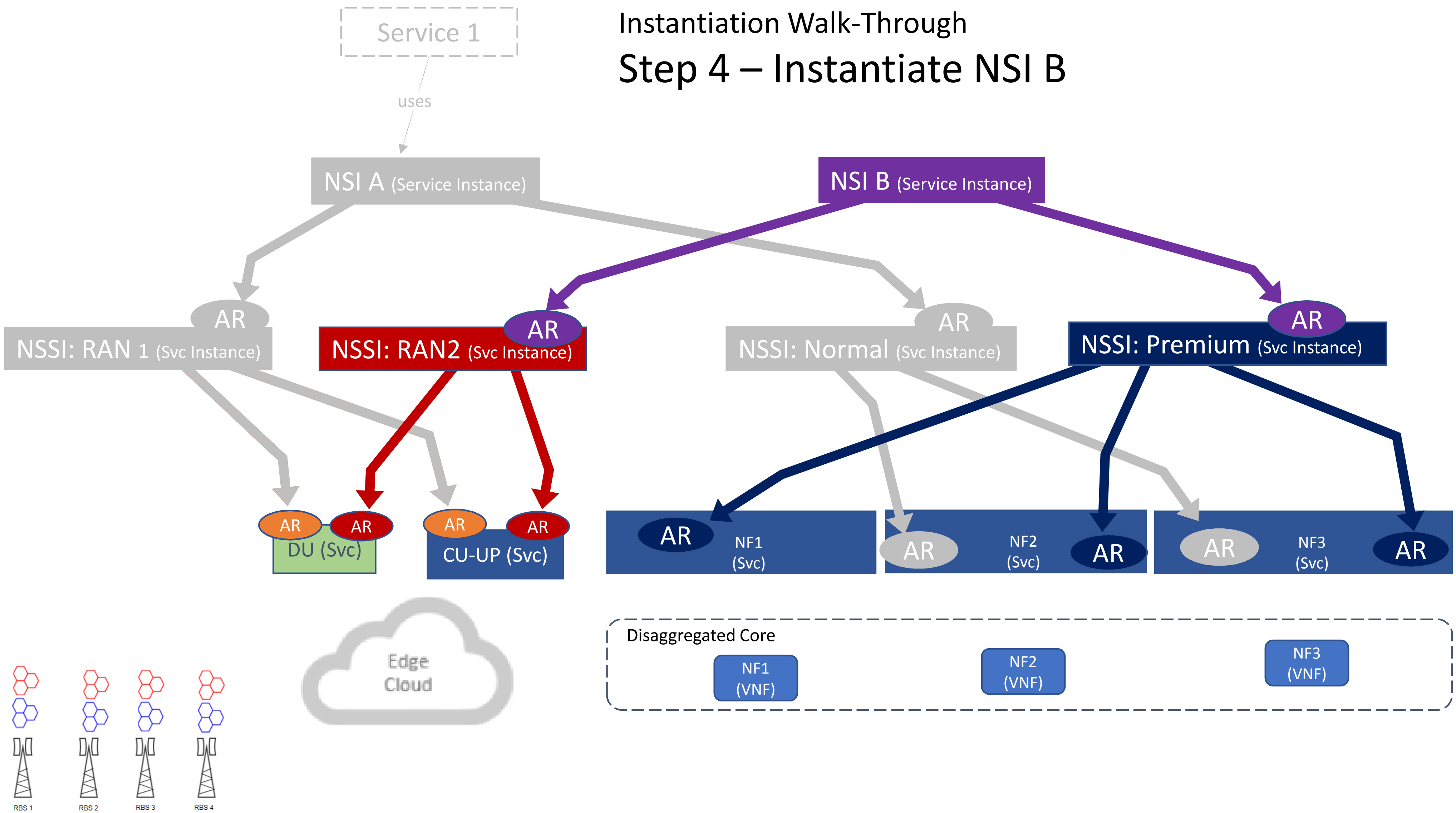
Instantiation Walk-Through

Step 3 – Instantiate NSSI: Premium



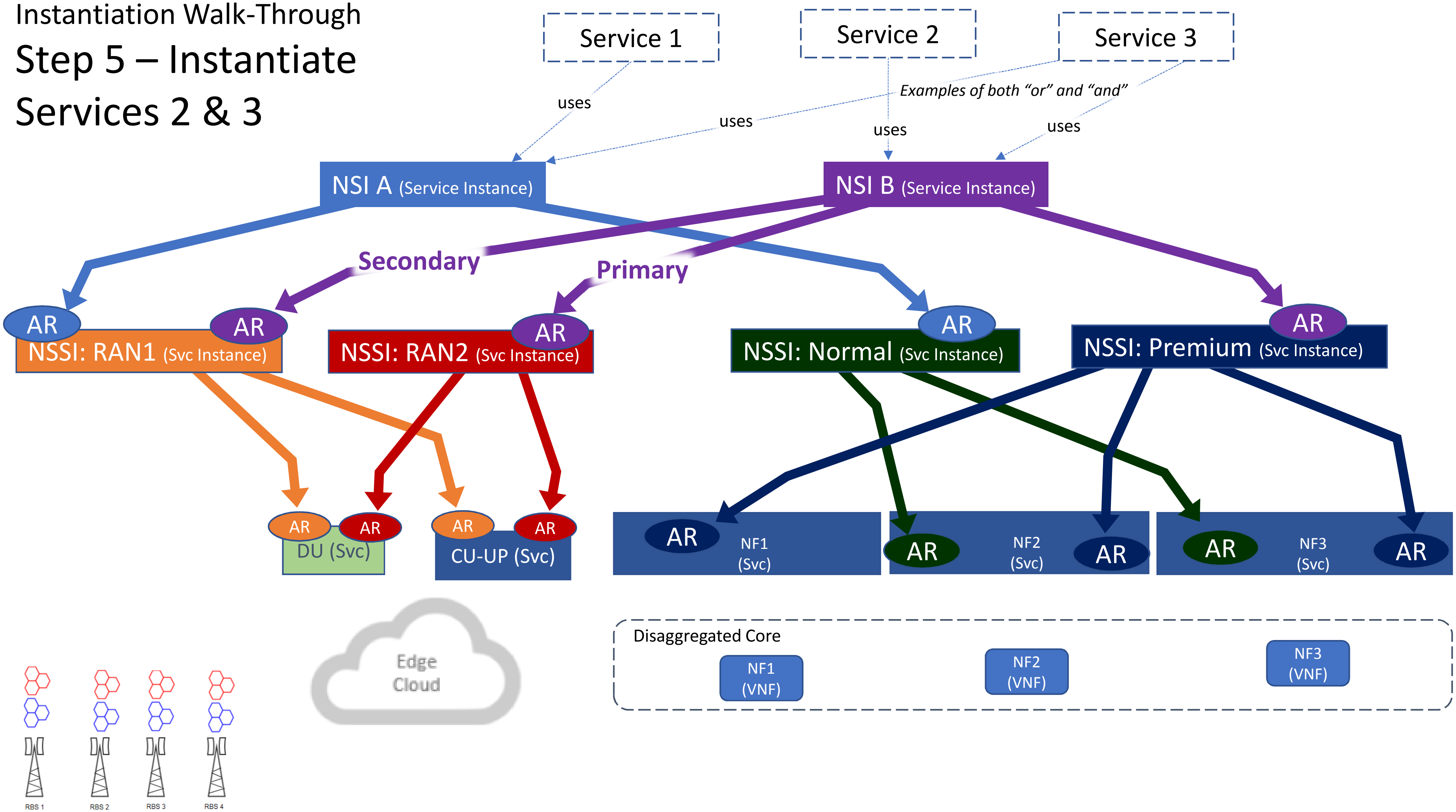
Instantiation Walk-Through

Step 4 – Instantiate NSI B



Instantiation Walk-Through

Step 5 – Instantiate Services 2 & 3

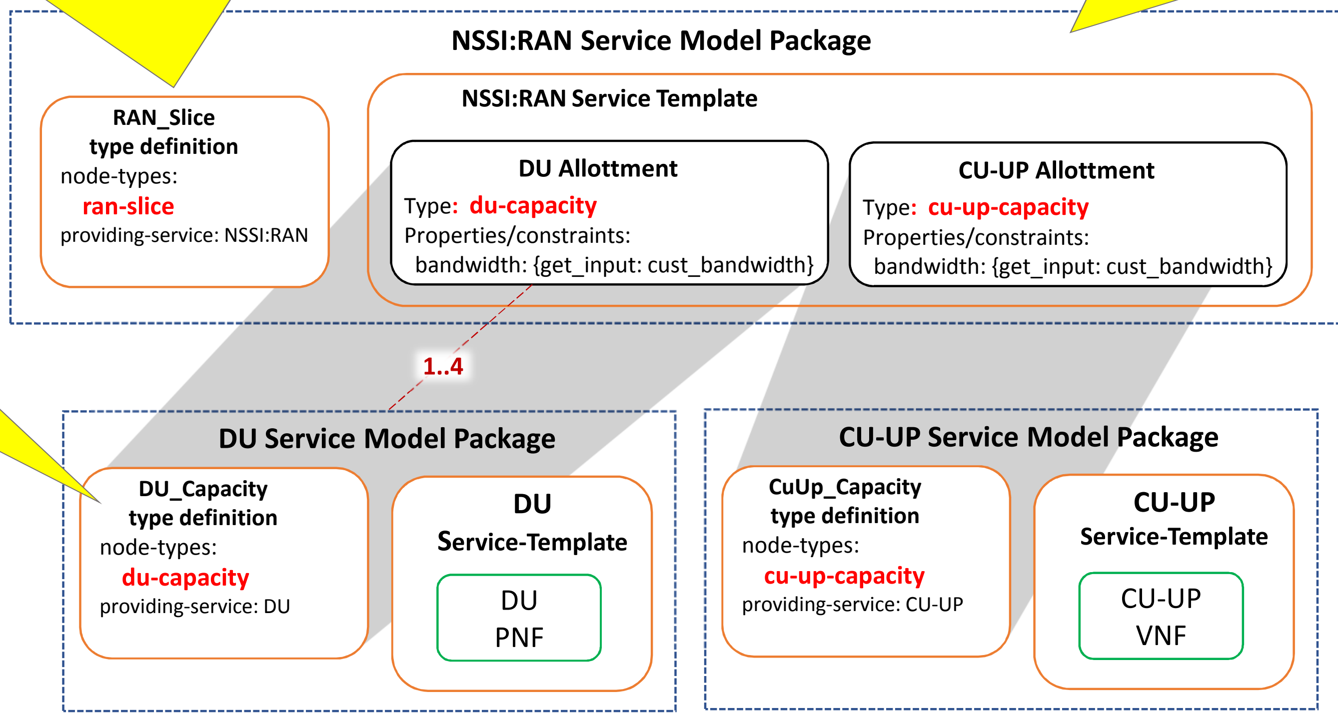


Design Time Model for the NSSI:RAN "Higher Level" (RAN Segment) Service Type and the Corresponding "Lower Level" (RAN NF) Service Types that it Uses

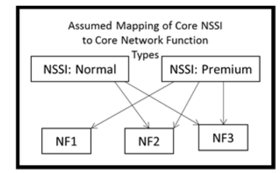
The "higher level" service's type definition would capture the attribute tags and allowed values to collect at run time as part of an instantiation request, such as the DUs instances that will participate in the slice, the frequency(ies) to use for this slice, and any other data required to pass to the "lower level" service types.

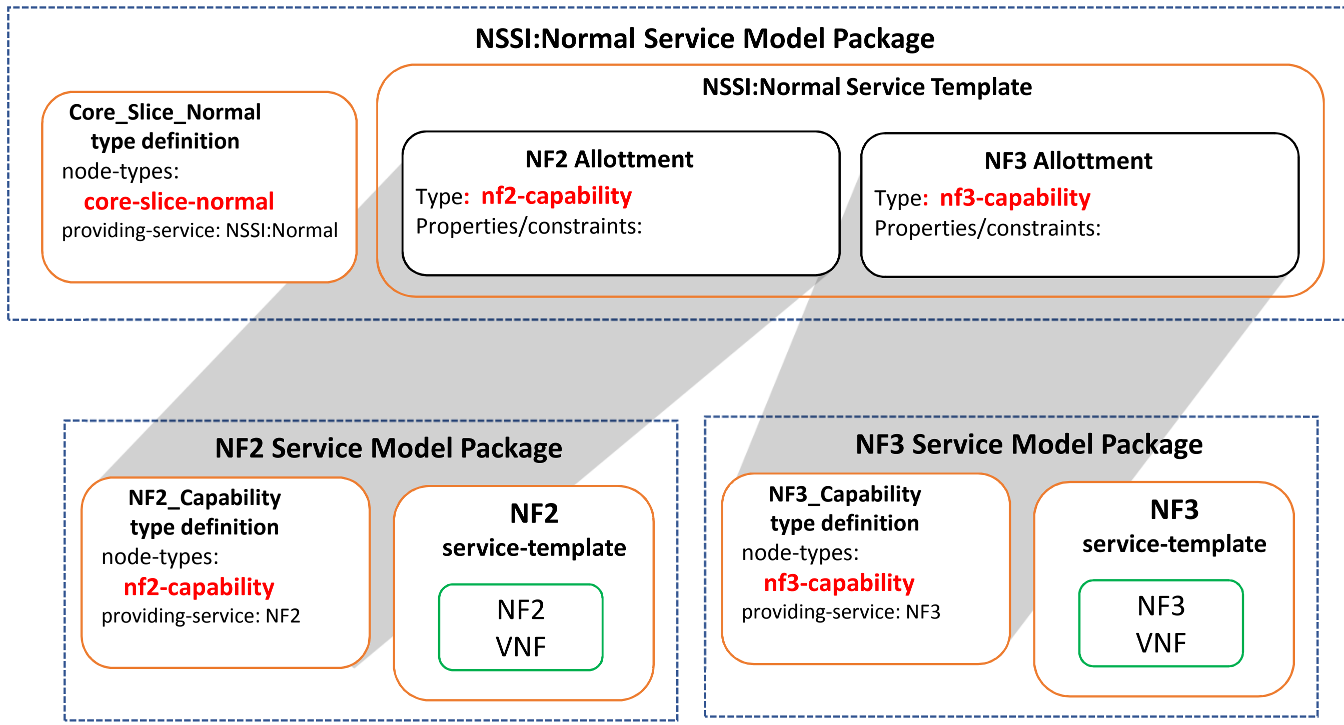
I assume that the "scope" of an NSSI:RAN service instance is a single CU-UP instance along with a collection of [1..4] associated DU instances. I assume that the impacted RBSs would be the universe of those RBSs managed by the DU.

The "lower level" service's type definition for the DU would capture the attribute tags and allowed values to receive at run time as part of an instantiation request for the portion of the NSSI configuration (slice configuration) that is the responsibility of the DU.

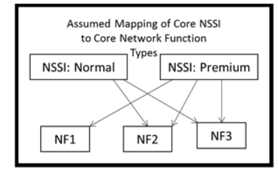


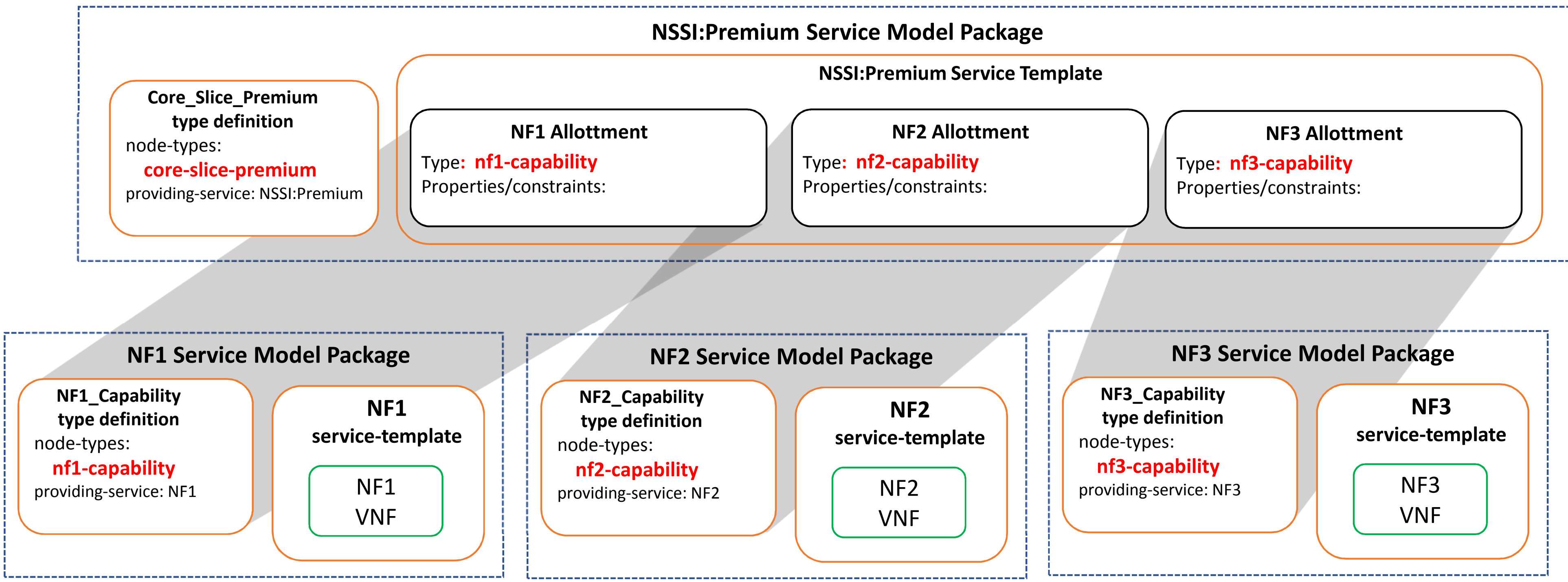
Design Time Model for the NSSI:Normal "Higher Level" (Core Segment) Service Type and the Corresponding "Lower Level" (Core NF) Service Types that it Uses



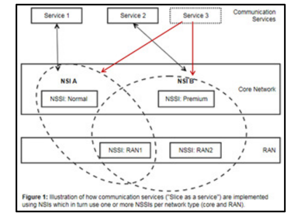


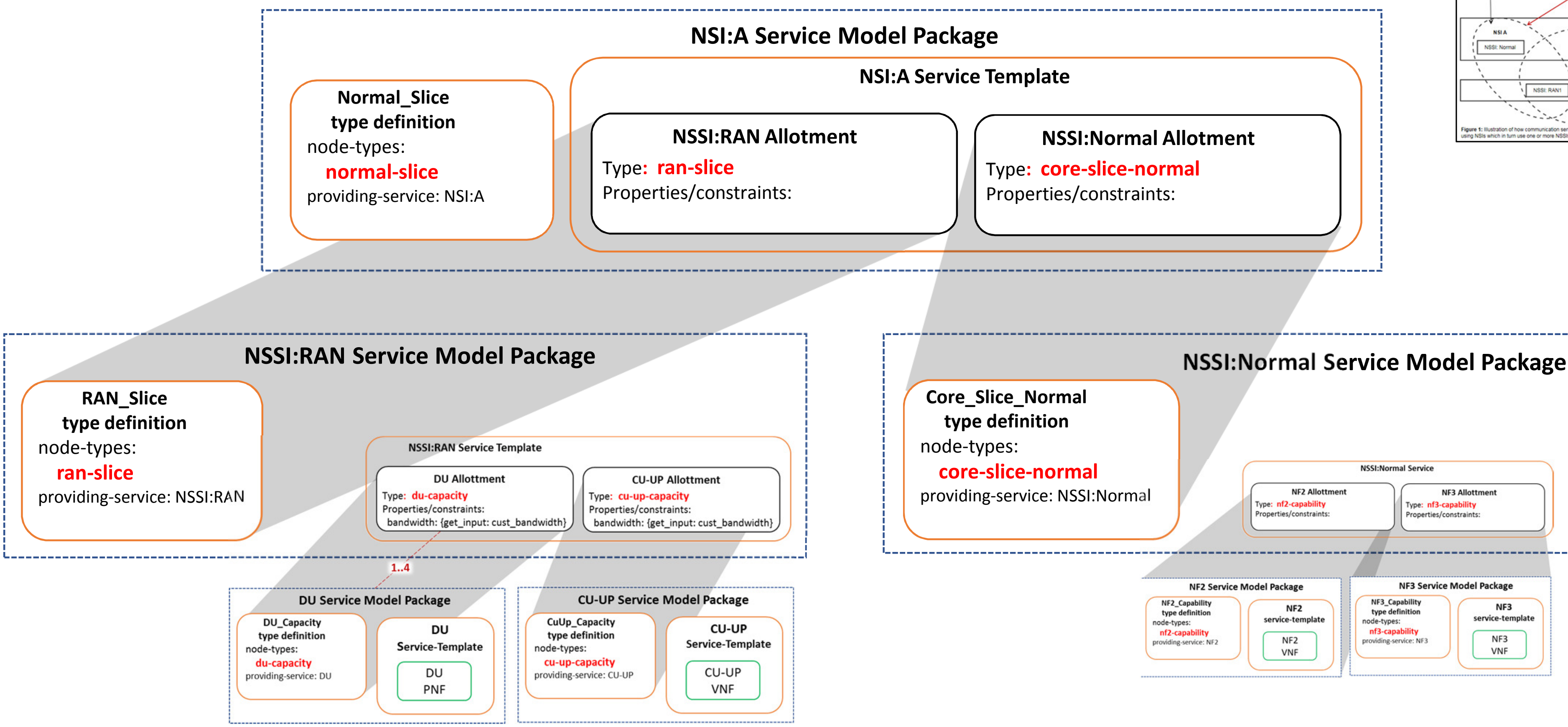
Design Time Model for the NSSI:Premium "Higher Level" (Core Segment) Service Type and the Corresponding "Lower Level" (Core NF) Service Types that it Uses



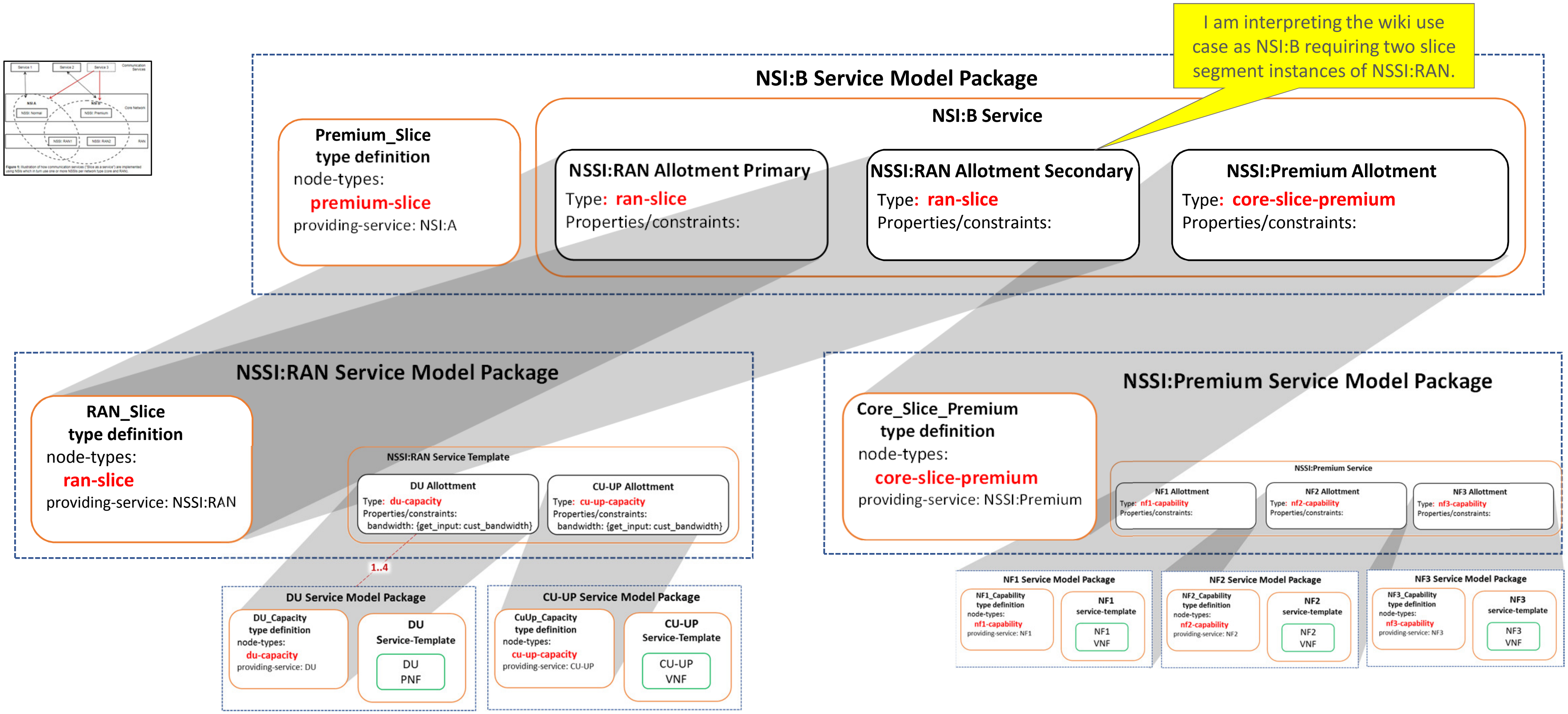


Design Time Model for the NSI:A "Higher Level" (E2E Slice) Service Type and the Corresponding "Lower Level" (RAN & Core Segment) Service Types that it Uses





Design Time Model for the NSI:B "Higher Level" (E2E Slice) Service Type and the Corresponding "Lower Level" (RAN & Core Segment) Service Types that it Uses



Design Time Model for the Service1 “Higher Level” Service Type and the Corresponding “Lower Level” (E2E Slice) Service Type that it Uses

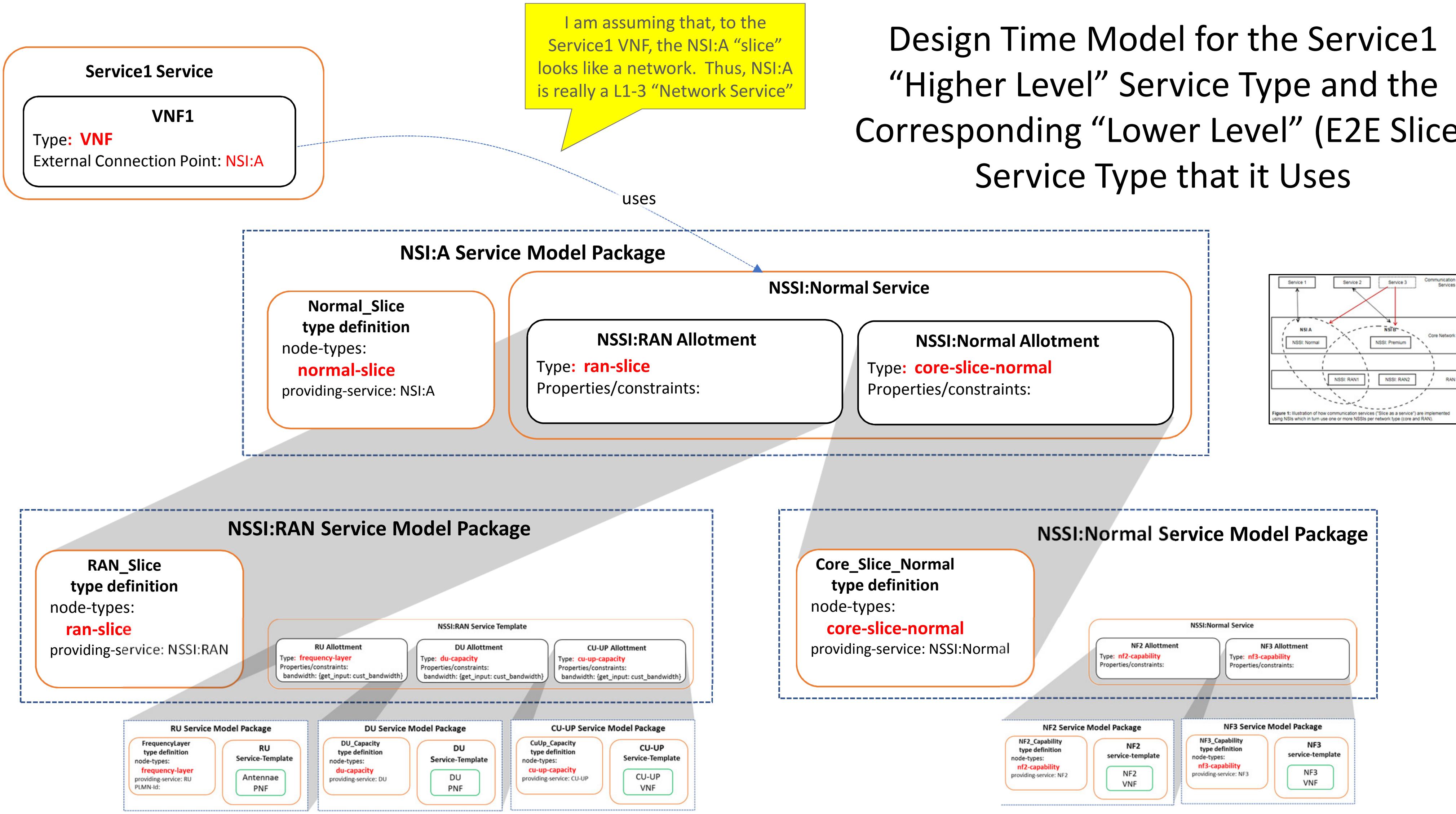
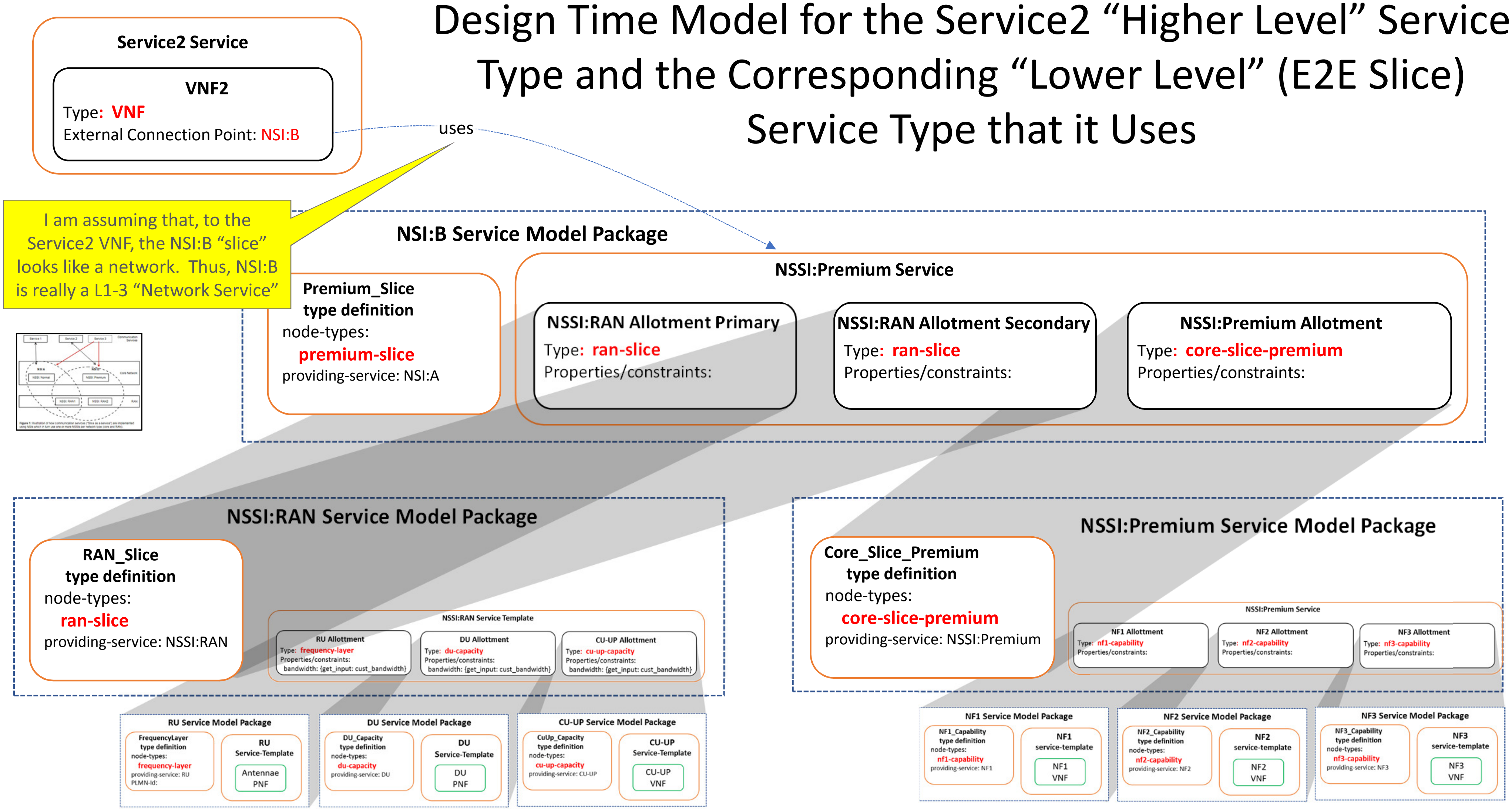


Figure 1: Illustration of how communication services ("Slice as a service") are implemented using NSIs which in turn use one or more NSSIs per network type (core and RAN).

Design Time Model for the Service2 "Higher Level" Service Type and the Corresponding "Lower Level" (E2E Slice) Service Type that it Uses



Design Time Model for the Service3 “Higher Level” Service Type and the Corresponding “Lower Level” (E2E Slice) Service Types that it Uses