Explanation of ObjectModelExampleVDU_VNFC.

Caveats

This is one person's (Chesla Wechsler, AT&T) interpretation of reading the ETSI NFV specs, in an effort to understand the different between VDU, VNFC, and the execution environment.

It's not intended to be exemplary or propose anything new.

Note that I think in terms of INSTANCES, which is then has to sort of look back at the DESCRIPTORs to see how we got here.

Beliefs

The VDU is a descriptor. You don't instantiate a VDU per se. You USE It to guide instantiation of VNFCs (which is the software function that runs on an execution environment).

The VDU is 1:1 with the software image, i.e., a VDU has one software image. Not sure if this matters, but my point is that it's the VDU that has the image, not the VNFC.

The software image may contain software for more than one type of VNFC.

When the VDU is used as a descriptor by the orchestrator to instantiate a VNFC instance, only one type of VNFC is *activated* within the image.

The VNFC is 1:1 with the execution environment, i.e., one and only one instance of a VNFC runs on an execution environment.

The execution environment can take any of these incarnations:

1) Bare metal
2) Virtual machine on bare metal
3) Container on bare metal
4) Container on virtual machine on bare metal

A VDU's software image used would likely vary by execution environment, because not all environments have the same requirements and capabilities (e.g., should a guest OS be included).

Description of the Diagram

The diagram contains one VNF descriptor, myVNF.

That VNF has two types of VNFCs, the FOO and the OAM.

It may be advantageous to have the ability to deploy the FOO as one big monolithic beast running on, perhaps, bare metal.

It may also be advantageous to have the ability to deploy the FOO as a cluster of instances, because not everyone has the same performance requirements as the folks who need the beast.

Regardless of how you deploy FOO, an instance of myVNF only requires one instance of the OAM function and it's pretty lightweight.

So, I have two deployment flavors, Monolithic and ClusterOfVNFCs.

Monolithic

The Monolithic one is only ever deployed in one size, therefore I have only the one InstantiationLevel.

The VDU related entities are used in the following way for this deployment flavor.

- I only need one vduProfile and one vduLevel for FOO Massive VDU
- The vduProfile tells me that the FOO Massive VDU has a min and max number of instances (of VNFCs) of 1
- The vduLevel tells me (unsurprisingly) that my InstantiationLevel uses only 1 instance
- In my simple example, I have two vduProfiles for OAM, just to show that's where affinity rules belong. I don't use the 2nd one however.

ClusterOfVNFCs

The ClusterOfVNFCs has three sizes, small, medium, and large, so therefore there are three InstantiationLevels.

The VDU related entities are used in the following way for this deployment flavor.

- I need three vduLevels for FOO VDU, each corresponding to an InstantiationLevel
- My vduProfile allows me a range of 2-10 instances, which my vduLevels use.
- I've allowed room for growth in the vduProfile, in that even the large instantiation could see two more instances added (possibly reusing the vduLevel for small, but I'm not sure if this is some other type of scale level object).
- Same comments about the OAM VDU