

# Data Model Interest Group 2018-10-01 Meeting Minutes

Recordings:

[audio\\_only.m4a](#), [zoom\\_0.mp4](#)

The main topic of discussion was the Annex A.5 SOL001 example and SOL001 constructs.

[Chesla Wechsler](#) shared info provided by [Thinh Nguyenphu](#) regarding configurable properties and modifiable properties:

Scope/Context:

- Team is bound by constraints of ETSI NVF
- Team is bound by constraints of OASIS Simple Profile 1.2
- Tried to duplicate info model in TOSCA

Configurable Properties:

- intended for consumption by the VNF instance itself
- can exist at the VNF level and the VNFC level
- takes effect immediately when sent to the instance

Modifiable Properties:

- intended for the VNFM
- VNFM will act on the modifiable properties based on where it is in the lifecycle and what scripts need them

VDU.Compute

- conflates the Compute and the software that will run on that compute into one node type
- restricts cardinality of VNFC to Compute to be 1:1
- it requires the orchestrator to have normative behavior for nodes of type `tosca.nodes.nfv.Vdu.Compute`
- it requires stating what are really VNFC's requirements of the `tosca.nodes.Compute` as capabilities of the `Vdu.Compute`, and the orchestrator would then have to map those to the compute

The spec informs the VNF vendor as to how to use the modifiable and configurable properties, but doesn't provide examples within the document. The properties are set by the vendor independently by deriving from the "base" types. Chesla interprets that to mean that, unlike HPA where there are common keys being defined, that type of approach hasn't yet been defined here. This bears investigation.

We went through the example to try to understand how a TOSCA orchestrator would process the sample. This section of the recording is filled with guesses and assumptions as to what is intended and would benefit from a subject matter expert to clarify the uncertainty.

We touched on VNF vs. PNF - the lifecycle management of them is different (Jacqueline). Different lifecycle management might take place whether it's associated with the PNFDevice or the NetworkFunction. Do the LCM operations of the NF (the application) interact with the LCM operations of the PNFDevice? Does this make any sense? Or is it simply that the standard NF interface has operations that are no ops for VNFs and others that are no ops for PNFs and workflows call the right operations. This is a work area - see how to model both a PNF and a VNF starting with the idea of modeling the NetworkFunction (commonalities) and capturing what makes PNFs different from VNFs. Need to at least be mentally orchestrating this. Could we work backwards from a PNF that works?