

# Data Model Interest Group 2018-10-08 Meeting Minutes

Recordings:

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

We recapped some of the 10/1 meeting.

Also, an important point that was brought up by [Thinh Nguyenphu](#)

- the ETSI SOL001 team was expressly avoiding using TOSCA constructs in the NFV types which would force an orchestrator to be TOSCA based. An example of this is the `boot_order` (which determines an order that VMs need to be booted). Standard TOSCA semantics would express this with a dependency between the VNFCs such that one needed to come before the other. One caution here: a dependency might apply at a different state in the lifecycle, e.g., if A depends on B, is there some state that B has to be in prior to satisfying A's dependency on it? This likely needs to be encoded in the interface/operations.

Modifiable attributes and Configurable attributes are meant to be extended by the vendors. They map back into the SOL003 API spec. Modifiable attributes are set at runtime but they can have defaults. `ModifyVNFCInfo` can update the data record so that it can be read and used in commands like `instantiate`.

Create VNF creates just the data record for the VNF whereas `Instantiate VNF` results in actual cloud assets. The data record would have the values specified in the VNFD filled in.

We went through the sample from the spec "MySunshineDB" to clarify how to interpret it as a document of information and then how it might be expressed for a TOSCA orchestrator audience.

We touched briefly on deployment flavor and how IFA maps to TOSCA. This bears a conversation of its own because the VF Modules used in the ONAP TOSCA use the TOSCA groups to capture deployable units which can be used for instantiation and scaling. There should be an opportunity to layer the concept of deployment flavors over vf modules (e.g., modules `base + A + B` = deployment flavor small and `base + C + D` = medium, etc.).

We touched briefly on the wisdom of a new set of types (onap types). Mapping from an existing type into new node types could be fraught with peril (Thinh). Perhaps there is a way to rework the ETSI NFV types to make them more "standardly" TOSCA orchestrate-able.

Some of the impacts of avoiding the use of TOSCA semantics in the ETSI NFV types are as follows (Chesla's Opinion):

- The Internal TOSCA model **does** seek to be TOSCA-based to enable TOSCA orchestration within the platform and therefore uses standard TOSCA semantics.
- The ETSI NFV types do not provide a clear separation between the application provided by the vendors and the cloud infrastructure (they are conflated)
- Deriving from the ETSI NFV types would result in the the derived objects inheriting the non-standard TOSCA semantics, ignoring their
- Another point on derivation: The ETSI NFV types dive straight into, e.g., VNFD, rather than a level of abstraction like `NetworkFunction` from which concrete node types can be derived. See below for additional detail on this one.

Is ONAP going to mandate TOSCA orchestration? The Task force is going to look into how to use the model end to end. If TOSCA is to be used by the platform the internal model can exploit TOSCA semantics. That doesn't require the ETSI NFV types to have to use TOSCA semantics as long as, once onboarded, it can be mapped to the internal model to the extent that components are able to use it and it supports operator's needs without violating the vendor "rules" as to how to use its NF.

We talked through the concept of having a `NetworkFunction` root node from which concrete node types derive. This avoids branching the TOSCA types in a way that would preclude `NetworkFunction-capability-based` service topology usage. That is, if there were a `NetworkFunction` node type from which two Router implementations are derived, one could be physical and one could be virtual, but both have capabilities needed by the service. The orchestrator could choose either one. If instead you branch your types into VNF and PNF node types, the type hierarchy is broken and the service cannot be written at the `NetworkFunction` level to express the use of a Router in the topology.

Some actions (too many!):

See if the VNFC can be teased out of the compute to create that separation between NFV and cloud assets.

Express the SOL001 Annex A.5 in proposed model.

Look through the node types to see if there are improvements we could make to make them more internally useful (but this would move them towards TOSCA semantics which is not the intent?)