VNF Onboarding

As mentioned in VNFM section of ETSI-MANO the VNF Onboarding is key to ability of Service Providers to take OPNFV beyond labs into field for trials and eventual integration with their architectures like ECOMP from AT&T and others from providers like China Mobile. Thus this is a critical piece for Operators and Vendors to collaborate on a global basis learn and try standards that work in real mission/business critical networks. Thus current work in OPNFV will bring contributions for OPNFV an upstream like Service Catalog, Instance Catalog to compile VNFs from different sources, will be key for Onboarding methods and helping generic and Vendor specific variations to address Carrier Grade features.

Analysis of VNF Onboarding support goals in OPNFV will benefit from a common understanding of the role of VNF Onboarding in the end-to-end product lifecycle for VNFs and services built from them. Below is an overview of this, with extra detail in the phase in which VNF Onboarding requires particular focus. This description is a work in progress and expected to be refined through discussions in the MANO WG, as well as with upstream communities/SDOs.

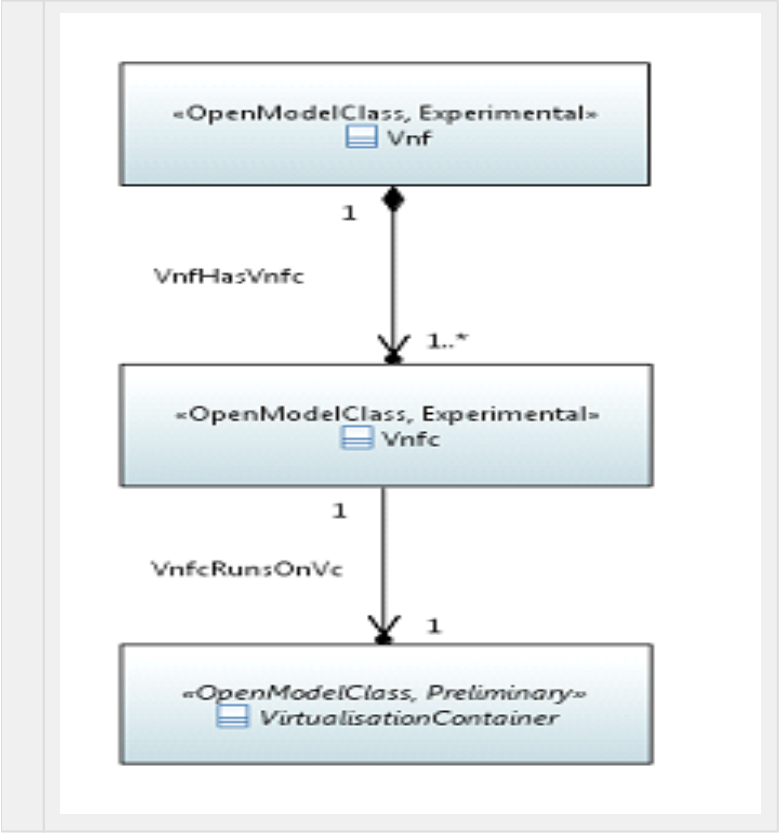
Lifecycle Phase	Activity	Description	Relation to OPNFV Projects	Notes (feel free to add yours)
VNF development	Developer creates and packages VNF	Developers create VNF packages that conform to package and NFV environment requirements of particular service providers (SPs) or onboarding systems. Devs may use IDEs that support standardized, pre-validated VNF package artifacts such as: ▪ NFVO consumes TOSCA VNFD. And in case VNFM provided with Juju, Juju bundle can be mapped to NSD , while Juju charm to VNFD ▪ VNF Descriptor (VNFD) and elements such as described for Tacker, ▪ Application configuration/state data model and APIs (e.g. via YANG)	Models: developer tools; VNF package and service environment requirements expression	For Declarative TOSCA VNFD simple Profile
Onboarding	VNF package import to onboarding system	Developer (or SP who obtains the package) uploads VNF to onboarding system, which performs basic package validation. The package contains or references (e.g. through models or build scripts) all artifacts needed to bring up the VNF in a specified NFV environment.	Models: VNF package import and validation Artifacts in OPNFV Parser: Information/data model (template/blueprint) validation	For CSAR and Modeling refer tosca-primer-v1.0-cnd01
	VNF basic testing	In a compatible NFV environment, the SP verifies basic functionality of the VNF, per tests supplied/described by the developer in the package. This could include verifying test claims (signed evidence that specific tests succeeded in specific NFV environments), executing tests (package-referenced or SP-required) in compatible NFV environments.	Models: VNF package support for test-related metadata	
	VNF incorporation into service design	SP applies the VNF to specific service blueprints, modifying the VNF package as needed, e.g. with service logic or for use with a specific application control system.	Models: service model/blueprints	
	VNF testing in service context	In compatible NFV environments (lab, pre-production, production), the SP verifies functionality of the VNF as part of services		
	VNF import into catalog	SP imports the VNF into a production catalog system, which further distributes the VNF as required/compatible with NFV environments of the SP.	Domino: distribution of VNF /service related policies	
	VNF is upgraded	Cycle back through the previous steps		
Production	VNF is deployed into production environment	The SP's NFVO/VNFM systems deploy the VNF per the requirements of the VNF/service /end-user.	Models: NFVO/VNFM support for standard VNF package deployment; package attributes defining deployment reqs of VNF /service/end-user	

	VNF resources are managed per VNF/service/user requirements	The VNFM monitors the VNF and adjusts resources as needed per requirements of the VNF/service/end-user.	Models: VNFM support for VNFD lifecycle hooks; package attributes defining resource management reqs of VNF/service /end-user	
	VNF upgrade			
	VNF migration			
	VNF suspension/resumption			
	VNF termination			
For all 3 above LC phase we need	Catalog	Catalog for (VNF and NS) Dev Ops, On boarding and Production	Catalog for OPNFV	
Market Place Catalog considerations	OPEN-O & Intel have considered this.			

Please refer to link for OPNFV End User Advisory Board's describing user story and terminology.

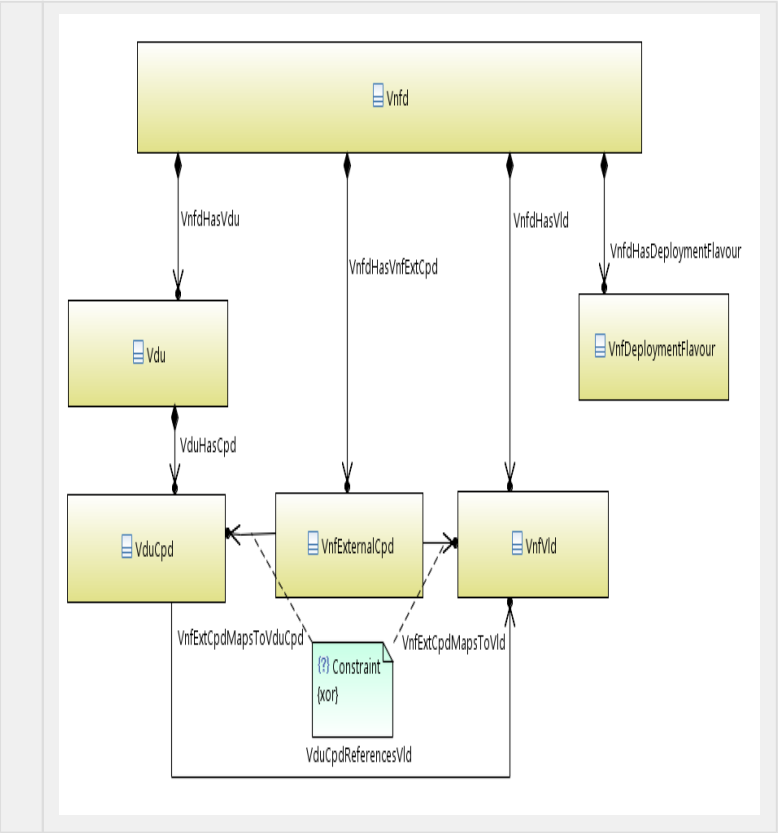
Following is a **user story** from TMF forum that is enabling NFV Ecosystem for **VNF Procurement and Operations** depicting the life cycle of a VNF..

VNF Supplier
•Develop - [•Design, •Develop, •Test] -
VNF Supplier/Service provider (procurement)
•Deliver - [•Package, •Validate, •Accept and catalogue (onboard)]
Service provider (Service design)
•Deploy - [•Combine, •Assemble, •Configure (software)]
Service provider (Service Delivery)
•Use - [•Service design, •Configure (service), •Instantiate]
Service provider (Service Assurance)
•Manage - [•Monitor, •Update, •Upgrade]
Service provider (Service Decommissioning)
•Retire - [•Migrate User]
VNF Logical Model can be described through UML.
The Vnf is a Class and has one or more Vnfc components and runs on one Virtual Contianer Vc. (Refer Dia below Basic VnfLogiclaModel)



Now all these begins with a Model of NFV and a high level VNFD as UML is described herein. We need some common agreed Model for VNF to even start on-boarding and managing it. Note ETSI NFV has started with Papyrus model from ONF we can borrow from it to drive YAML files for both TOSCA and YANG. Some simple UML mapping for Vnfd should be agreed as common baseline. This is all part of Eclipse Modeling Framework(EMF) and all in open source. In fact the Model and/or Movie project can be of great help to OPNFV realize the using ODL/ONOS the Connection Point Descriptor besides the VNFD we are describing to bridge the gap in topology modeling and orchestration. The ETSI NFV should publish its work items for 2016 sooner for us to leverage and keep synch, otherwise the gap will keep widening and fracturing the efforts.

Figure : VNFD high-level structure



To understand at high level use flowing Glossary:

ID	Identifier
VNFD	Virtual Network Function Descriptor
CP	Connection Point
CPD	Connection Point Descriptor
DSL	Domain Specific Language
Flavor	Specifies Vnf characteristics.
LCM	Life Cycle Management
NIC	Network Interface Controller
VDU	Virtual Deployment Unit
VL	Virtual Link
VLD	Virtual Link Descriptor
QoS	Quality of Service

Note: Additional user stories will be reviewed on the proposed basis above and can be revised as we try match with current and evolving OPNFV architecture framework.

Note: Bryan Sullivan has taken up VNF On boarding with Cloudify "hello world sample" and we plan to review his inputs and answer Margaret who has raised a valid question as what user stories for VNF on boarding we will be working on to figure out what architecture is needed to support it MANO stack above VIM. Telco-Case-Study.pdf

A sample user scenario for Smart City is available in public of TMF forum and listed below for reference and discussions. (to be proposed review under NetReady & Domino)

User Stories - Smart City NFV Ecosystem

Other user stories are welcome to be discussed in SPC/Polestar WG https://wiki.opnfv.org/pages/viewpage.action?spaceKey=EUAG&title=Pain+Points.

Plus we can always help technical evaluation in MANO WG to support how we map this into OPNFV projects and/or Upstream to realize the user stories.