VNF SDK Casablanca Release Planning (M1)

- 1 Overview
- 2 Scope
  - 2.1 What is this release trying to address?
  - 2.2 Minimum Viable Product
  - 2.3 Functionalities
    - 2.3.1 Epics
    - 2.3.2 Stories
  - 2.4 Longer term roadmap
- 3 Release Deliverables
- 4 Sub-Components
- 5 Architecture
  - 5.1 High level architecture diagram
  - 5.2
  - 5.3 Platform Maturity
  - 5.4 API Incoming Dependencies
  - 5.5 API Outgoing Dependencies
  - 5.6 Third Party Products Dependencies
- 6 Testing and Integration Plans
- 7 Gaps
- 8 Known Defects and Issues
- 9 Risks
- 10 Resources
- 11 Release Milestone
- 12 Team Internal Milestone
- 13 Documentation, Training
- 14 Other Information
  - 14.1 Vendor Neutral
  - 14.2 Free and Open Source Software

Overview

Project Name	VNFSDK
Target Release Name	Casablanca
Project Lifecycle State	Incubation
Participating Company	Huawei, Intel, AT&T, Gigaspaces, China Mobile, Wind River, Orange, Amdocs, Nokia, ZTE, IBM, VMWare, Cisco, ARM

Scope

What is this release trying to address?

VNF SDK's mission is to simplify the process of developing and onboarding VNFs and expanding ONAP's VNF ecosystem. For Service Providers, VNFSDK will reduce the time and level of testing required to identify, select, and onboard a VNF. For vendors, VNFSDK will reduce integration efforts by defining a standard for VNF packaging with widespread operator acceptance. VNFSDK can be used in a validation/conformance testing program in the future.

The Casablanca release will enhance existing functionality, introduce security, integrate with ONAP use-case flow and progress towards being carrier grade with high quality.

1. Enhance dovetail integration so we can use it in LFN Compliance/Verification testing
    a. Improvements to function test for better integration into the framework
2. Incorporate vnfreqs testable requirements
    a. consistent support for HEAT/TOSCA VNFs (as defined in VNFReqs) - may not be 100% common
        i. NB. HEAT tests will be developed by VVP, but may be implemented in VNFSDK for VNF Compliance Testing
3. modular marketplace framework to make it easier to plug in tests (maybe configurable per application such as compliance testing, operator-vendor engagement, 3rd party enhancements, etc.)
4. Support SDC on boarding tests (maybe allow SDC to replace refrepo portal, but use vnfmarket-be for their own onboarding test engine
5. Package data model enhancements (internal and onboarding models?)
6. VES: multivim enhancements, PNF registration, notification, and RAN metrics (5G use case)
7. Enhance HPA, SOL-004 certificate, and other carry-over items

Use Cases

1. VNF provider uses VNFSDK for VNF Compliance Testing
2. Support operator use of VNFSDK for vendor engagement/acceptance testing
3. 3rd Party Lab uses VNFSDK for extended VNF testing (may include functional, non-functional, and/or performance tests developed by 3rd party labs)
4. HPA
5. Enhance Security

Minimum Viable Product

Describe the MVP for this release.

1. Support the VNF packaging model & VNF Requirements defined in Beijing
2. Support VNF Compliance under LFN CVC umbrella
3. HPA

Functionalities

List the functionalities that this release is committing to deliver by providing a link to JIRA Epics and Stories. In the JIRA Priority field, specify the priority (either High, Medium, Low). The priority will be used in case de-scoping is required. Don't assign High priority to all functionalities.

Epics

Key	Summary	T	Created	Updated	Due	Assignee	Reporter	P	Status	Resolution
VNFSDK-286	VNF Test Platform		Jul 11, 2018	Jun 12, 2019		Unassigned	None	=	CLOSED	Done
VNFSDK-285	Dovetail integration		Jul 11, 2018	Jun 13, 2019		Unassigned	None	=	CLOSED	Done
VNFSDK-280	VES Multivim enhancements, PNF registration, notification, and RAN metrics		Jun 22, 2018	Mar 12, 2020		Unassigned	None	=	CLOSED	Not Done
VNFSDK-279	Package data model enhancements		Jun 22, 2018	Mar 12, 2020		Unassigned	None	=	CLOSED	Not Done
VNFSDK-278	Support SDC vendor engagement		Jun 22, 2018	Jul 31, 2019		Unassigned	None	˅	CLOSED	Done
VNFSDK-277	Modular test framework		Jun 22, 2018	Oct 11, 2018		Unassigned	None	=	CLOSED	Done
VNFSDK-276	Incorporate vnfreqs testable requirements for TOSCA		Jun 22, 2018	Jul 31, 2019		Unassigned	None	^	CLOSED	Done
VNFSDK-274	Enhance dovetail integration so we can use it in LFN Compliance/Verification testing		Jun 20, 2018	Jul 31, 2019		Unassigned	None	=	CLOSED	Done
VNFSDK-153	Align with VNF Reqs		Jan 15, 2018	Aug 08, 2018		Unassigned	None	=	CLOSED	Duplicate
VNFSDK-152	HPA support for VNF packaging/validation		Jan 15, 2018	Aug 12, 2023		Unassigned	None	=	CLOSED	Done
VNFSDK-135	Update pkgtools /marketplace to support VNF packaging model from R2		Dec 17, 2017	Oct 29, 2019		Unassigned	None	=	CLOSED	Done

11 issues

Stories

Key	Summary	T	Created	Updated	Due	Assignee	Reporter	P	Status	Resolution
VNFSDK-307	Adapt functest to the new architecture		Aug 23, 2018	Sep 19, 2018		Unassigned	None	=	CLOSED	Done
VNFSDK-305	Enable package validation as part of VTP		Aug 20, 2018	Sep 17, 2018		Unassigned	None	=	CLOSED	Done
VNFSDK-304	Add gRPC Interface between Refrepo and OCLIP		Aug 20, 2018	Sep 19, 2018		Unassigned	None	=	CLOSED	Done
VNFSDK-299	Update VES client documentation		Aug 08, 2018	Aug 08, 2018		Unassigned	None	=	CLOSED	Done
VNFSDK-292	pkgtools - retire aria and replace it with openstack /tosca-parser		Jul 25, 2018	Aug 23, 2018		Unassigned	None	=	CLOSED	Done
VNFSDK-275	Add validation abilities to test cases		Jun 20, 2018	Dec 30, 2018		Unassigned	None	=	CLOSED	Done
VNFSDK-272	Set up package model sandbox for Papyrus		Jun 06, 2018	Aug 08, 2018		Unassigned	None	⌄⌄	CLOSED	Done
VNFSDK-264	VNFTEST to support instantiation of multiple vf-modules per VNF		May 14, 2018	Aug 08, 2018		Unassigned	None	=	CLOSED	Done
VNFSDK-257	Add shutdown method to VES Agent		Apr 17, 2018	Sep 21, 2018		Unassigned	None	=	CLOSED	Done
VNFSDK-185	implement metadata enhancements		Jan 15, 2018	Aug 08, 2018		Unassigned	None	=	CLOSED	Done
VNFSDK-184	agree on metadata enhancements (with Marketplace team)		Jan 15, 2018	Aug 08, 2018		Unassigned	None	=	CLOSED	Done
VNFSDK-175	implement vnfreqs packaging requirements in validation		Jan 15, 2018	Dec 30, 2018		Unassigned	None	=	CLOSED	Done
VNFSDK-174	implement vnfreqs packaging requirements in pkgtools		Jan 15, 2018	Sep 14, 2018		Unassigned	None	=	CLOSED	Done
VNFSDK-173	evaluate impact of vnfreqs		Jan 15, 2018	Sep 17, 2018		Unassigned	None	=	CLOSED	Done
VNFSDK-168	support Logging Enhancements		Jan 15, 2018	Jul 31, 2019		Unassigned	None	=	CLOSED	Done
VNFSDK-165	refine and enhance model for Casablanca		Jan 15, 2018	Sep 19, 2018		Unassigned	None	=	CLOSED	Done
VNFSDK-164	update validation scripts to support new model		Jan 15, 2018	Aug 08, 2018		Unassigned	None	^^	CLOSED	Done
VNFSDK-163	align with SDC team on pkg model for Beijing		Jan 15, 2018	Aug 08, 2018		Unassigned	None	^	CLOSED	Done
VNFSDK-149	Model - create a VNF tosca template with HPA support (vCPE)		Jan 11, 2018	Aug 29, 2018		Unassigned	None	^	CLOSED	Duplicate
VNFSDK-144	enhance packaging tools to support security features in SOL004		Jan 05, 2018	Aug 27, 2018		Unassigned	None	=	CLOSED	Done

Showing 20 out of 23 issues

Longer term roadmap

Indicate at a high level the longer term roadmap. This is to put things into the big perspective.

- Support VNF lifecycle tests

Release Deliverables

Indicate the outcome (Executable, Source Code, Library, API description, Tool, Documentation, Release Note...) of this release.

Deliverable Name	Deliverable Description
Compliance Tool	Executable
Model	UML info model and TOSCA model
Packaging tool	Executable
Reference Repo/Marketplace	Executable
	API : Deliver API for external Project. The API will deliver such as uploading/downloading/Updating/etc functionality
Validation Tools	Executable
Lifecycle Test Framework (future)	Executable/Library
Function Test Framework	Executable
VES Agent	code/libraries that can be included in VNFs

Sub-Components

List all sub-components part of this release.
Activities related to sub-components must be in sync with the overall release.

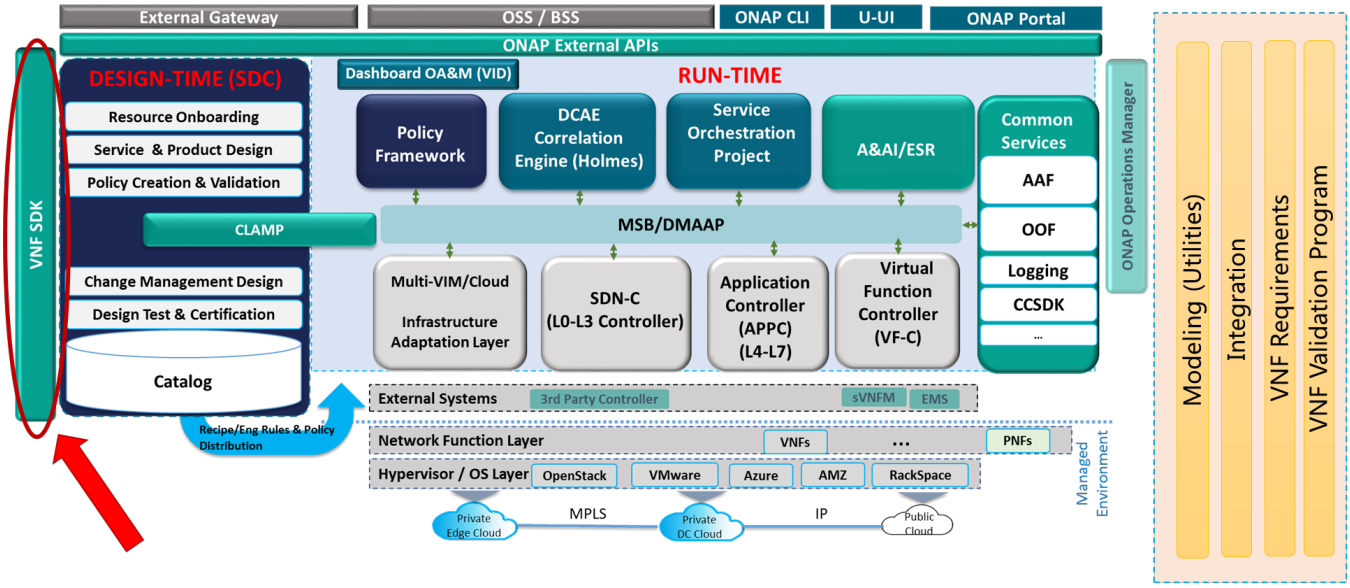
Sub-components are repositories and are consolidated in a single centralized place. Edit the Release Components name for your project in the centralized page.

Architecture

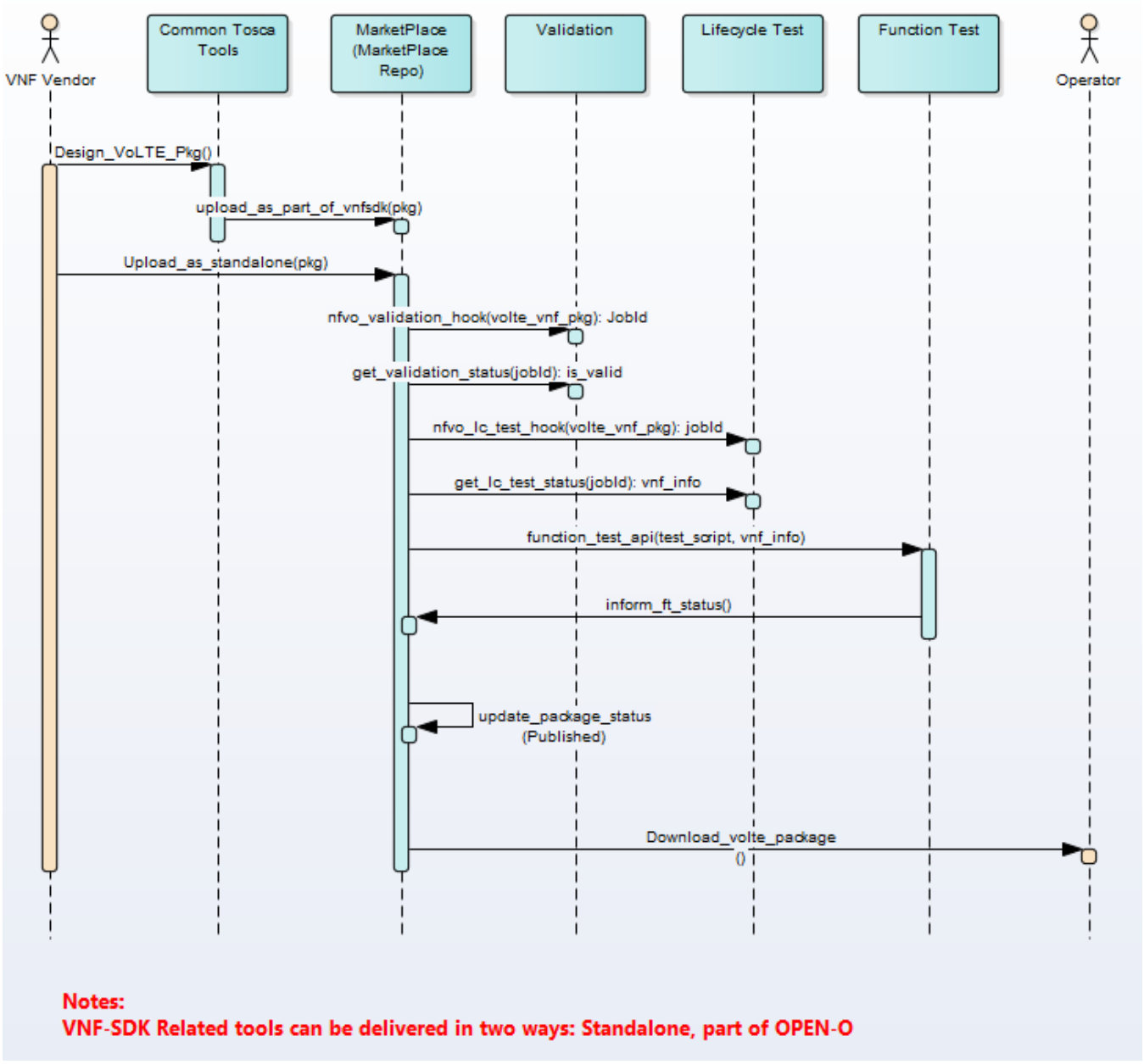
High level architecture diagram

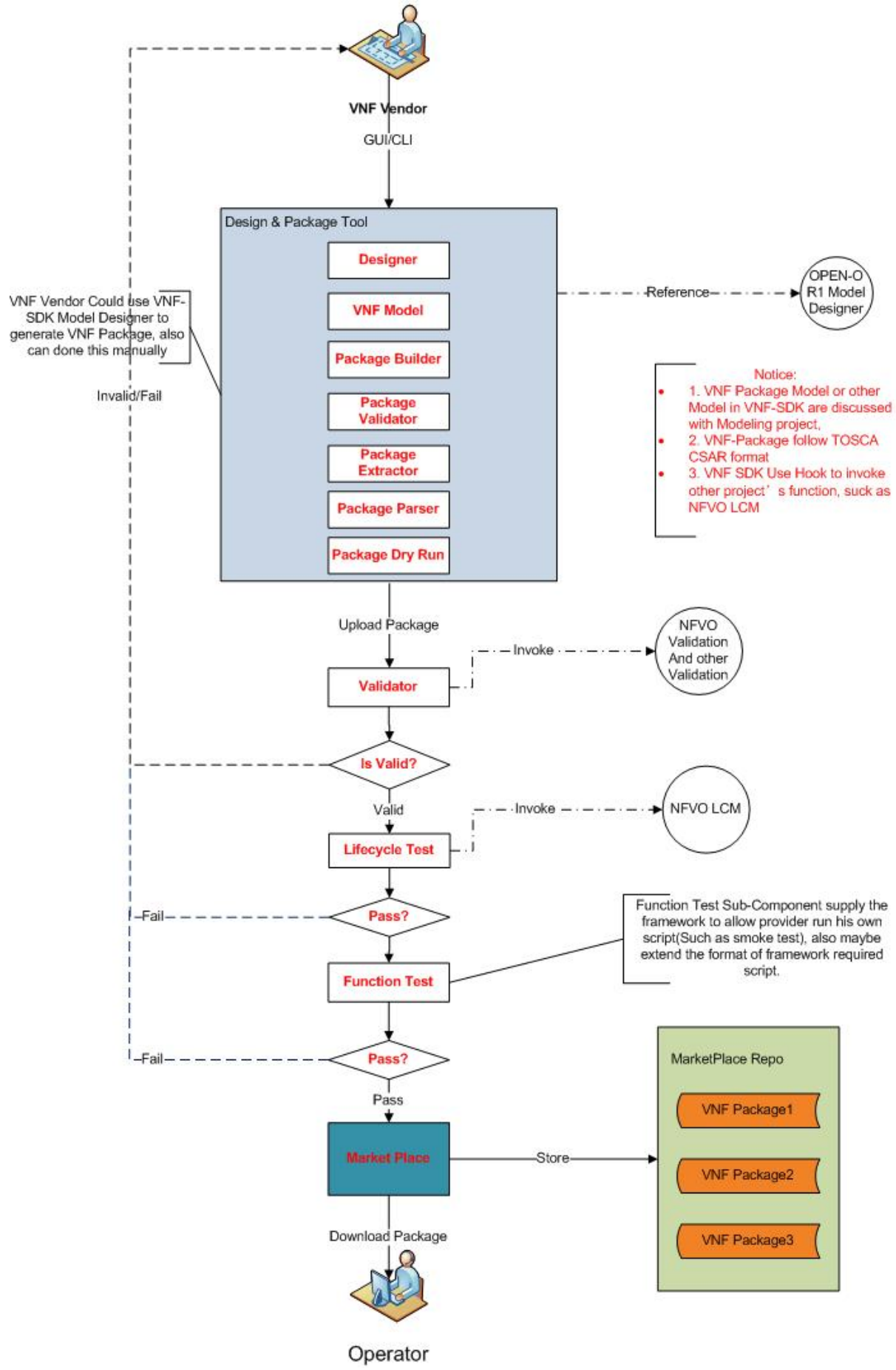
At that stage within the Release, the team is expected to provide more Architecture details describing how the functional modules are interacting.

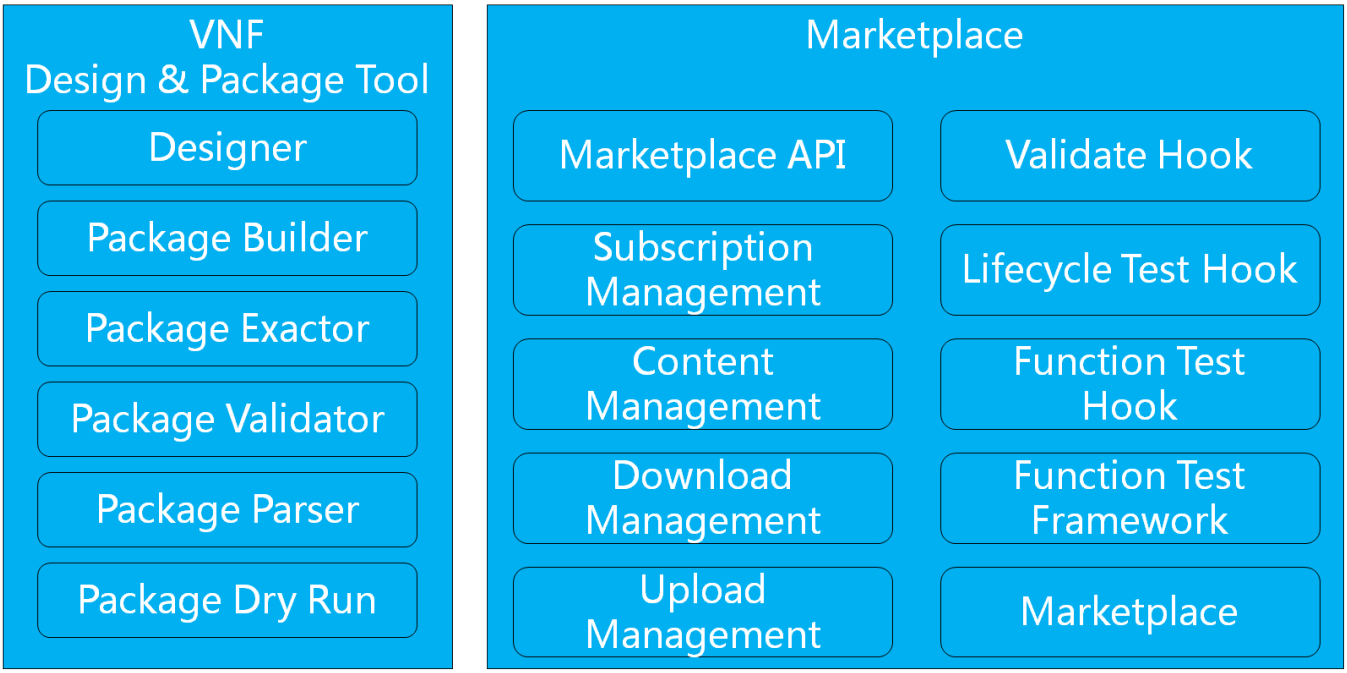
Indicate where your project fit within the ONAP Archiecture diagram.



Block and sequence diagrams showing relation within the project as well as relation with external components are expected.




VNF Vendor
GUI/CLI
Design & Package Tool
Designer
VNF Model
Package Builder
Package Validator
Package Extractor
Package Parser
Package Dry Run
Reference
OPEN-O R1 Model Designer
VNF Vendor Could use VNF-SDK Model Designer to generate VNF Package, also can done this manually
Invalid/Fail
Notice:
1. VNF Package Model or other Model in VNF-SDK are discussed with Modeling project,
2. VNF-Package follow TOSCA CSAR format
3. VNF SDK Use Hook to invoke other project's function, suck as NFVO LCM
Upload Package
Validator
Invoke
NFVO Validation And other Validation
Is Valid?
Valid
Invoke
NFVO LCM
Lifecycle Test
Fail
Pass?
Function Test Sub-Component supply the framework to allow provider run his own script(Such as smoke test), also maybe extend the format of framework required script.
Function Test
Fail
Pass?
Pass
Market Place
Store
MarketPlace Repo
VNF Package1
VNF Package2
VNF Package3
Download Package
Operator



Anyone reading this section should have a good understanding of all the interacting modules.

- Platform Maturity

Refering to CII Badging Security Program and Platform Maturity Requirements, fill out the table below by indicating the actual level , the targeted level for the current release and the evidences on how you plan to achieve the targeted level.

Area	Actual Level	Targeted Level for current Release	How, Evidences	Comments
Performance	0	0	design-time component	• 0 -- none • 1 – baseline performance criteria identified and measured • 2 & 3 – performance improvement plans created & implemented
Stability	1	1	design-time component	• 0 – none • 1 – 72 hours component level soak w/random transactions • 2 – 72 hours platform level soak w/random transactions • 3 – 6 months track record of reduced defect rate
Resiliency	1	1	system run as-needed	• 0 – none • 1 – manual failure and recovery (< 30 minutes) • 2 – automated detection and recovery (single site) • 3 – automated detection and recovery (geo redundancy)
Security	1	1	increase test coverage to 50%. Implement https and aspects of SOL-004	• 0 – none • 1 – CII Passing badge + 50% Test Coverage + Critical and High vulnerabilities • 2 – CII Silver badge; internal communication encrypted; role-based access control and authorization for all calls • 3 – CII Gold
Scalability	0	0	not a runtime component. VNF SDK will be run as-needed in constrained environment.	• 0 – no ability to scale • 1 – single site horizontal scaling • 2 – geographic scaling • 3 – scaling across multiple ONAP instances

Manageability	1	1		• 1 – single logging system across components; instantiation in < 1 hour • 2 – ability to upgrade a single component; tracing across components; externalized configuration management
Usability	1	1	onap.readthedocs.io	• 1 – user guide; deployment documentation; API documentation • 2 – UI consistency; usability testing; tutorial documentation

API Incoming Dependencies

List the API this project is expecting from other projects.
Prior to Release Planning review, Team Leads must agreed on the date by which the API will be fully defined. The API Delivery date must not be later than the release API Freeze date.

Prior to the delivery date, it is a good practice to organize an API review with the API consumers.

API Name	API Description	API Definition Date	API Delivery date	API Definition link (i.e.swagger)
none				

API Outgoing Dependencies

API this project is delivering to other projects. API used by SDC project.

API Name	API Description	API Definition Date	API Delivery date	API Definition link (i.e. swagger)
Upload API	API for vendors to upload VNF packages to marketplace	M1	M3	API Documentation
Download API	API for downloading VNF Package from marketplace	M1	M3	API Documentation
Query VNF information	API to query information on VNFs uploaded to the marketplace	M1	M3	API Documentation

Third Party Products Dependencies

Third Party Products mean products that are mandatory to provide services for your components. Development of new functionality in third party product may or not be expected.
List the Third Party Products (OpenStack, ODL, RabbitMQ, ElasticSearch,Crystal Reports, ...).

Name	Description	Version
OPNFV Dovetail	To fill out	To fill out

In case there are specific dependencies (Centos 7 vs Ubuntu 16. Etc.) list them as well.

Testing and Integration Plans

Provide a description of the testing activities (unit test, functional test, automation,...) that will be performed by the team within the scope of this release.

Describe the plan to integrate and test the release deliverables within the overall ONAP system.
Confirm that resources have been allocated to perform such activities.

We will enhance unit tests and CSIT tests for validation tools, packaging tools, and certification tools.

Gaps

This section is used to document a limitation on a functionality or platform support. We are currently aware of this limitation and it will be delivered in a future Release.
List identified release gaps (if any), and its impact.

Gaps identified	Impact
To fill out	To fill out

Known Defects and Issues

Provide a link toward the list of all known project bugs.

Key	Summary	T	Created	Updated	Due	Assignee	Reporter	P	Status	Resolution
VNFSDK-874	Python Code Updates: Vnfsdk(VNFSDK)		May 15, 2023	May 15, 2023		Unassigned	Matthew Watkins		OPEN	Unresolved
VNFSDK-862	Kohn Release Note not updated		Nov 30, 2022	Dec 05, 2022		Unassigned	Thomas Kulik		OPEN	Unresolved
VNFSDK-832	Sonar issues present in ONAP vnfsdk validation		Mar 20, 2022	Mar 20, 2022		Unassigned	None		OPEN	Unresolved
VNFSDK-785	Issue in VTP profile GET operations		May 28, 2021	May 31, 2021		Unassigned	None		IN PROGRESS	Unresolved
VNFSDK-769	Remove PM references from PM_Dictionary_validator		May 06, 2021	May 06, 2021		Unassigned	None		SUBMITTED	Unresolved
VNFSDK-768	VNFSDK marketplace CSIT is testing old postgres image		May 05, 2021	Sep 19, 2022		Unassigned	None		OPEN	Unresolved
VNFSDK-405	Make validation of CSAR case insensitive		May 08, 2019	Sep 19, 2022		Unassigned	None		OPEN	Unresolved
VNFSDK-344	Support HEAT based VNFpackage uploads in Market place		Dec 11, 2018	Sep 19, 2022		Unassigned	None		OPEN	Unresolved

8 issues

Risks

List the risks identified for this release along with the plan to prevent the risk to occur (mitigation) and the plan of action in the case the risk would materialized (contingency).

Risk identified	Mitigation Plan	Contingency Plan
To fill out	To fill out	To fill out

Resources

Fill out the Resources Committed to the Release centralized page.

Release Milestone

The milestones are defined at the Release Level and all the supporting project agreed to comply with these dates.

Team Internal Milestone

This section is optional and may be used to document internal milestones within a project team or multiple project teams. For instance, in the case the team has made agreement with other team to deliver some artifacts on a certain date that are not in the release milestone, it is erecommended to provide these agreements and dates in this section.

It is not expected to have a detailed project plan.

Date	Project	Deliverable
To fill out	To fill out	To fill out

Documentation, Training

- Highlight the team contributions to the specific document related to he project (Config guide, installation guide...).
- Highlight the team contributions to the overall Release Documentation and training asset
- High level list of documentation, training and tutorials necessary to understand the release capabilities, configuration and operation.
- Documentation includes items such as:
  - Installation instructions

- Configuration instructions
- Developer guide
- End User guide
- Admin guide
- ...

> **Note**
>
> The Documentation project will provide the Documentation Tool Chain to edit, configure, store and publish all Documentation asset.

Other Information

- Vendor Neutral

If this project is coming from an existing proprietary codebase, ensure that all proprietary trademarks, logos, product names, etc. have been removed. All ONAP deliverables must comply with this rule and be agnostic of any proprietary symbols.

- Free and Open Source Software

FOSS activities are critical to the delivery of the whole ONAP initiative. The information may not be fully available at Release Planning, however to avoid late refactoring, it is critical to accomplish this task as early as possible.
List all third party Free and Open Source Software used within the release and provide License type (BSD, MIT, Apache, GNU GPL,...).
In the case non Apache License are found inform immediately the TSC and the Release Manager and document your reasoning on why you believe we can use a non Apache version 2 license.

Each project must edit its project table available at Project FOSS.

Charter Compliance

The project team comply with the ONAP Charter.