Common NFVI Telco Taskforce

Antwerp Workshop

Michael Bustamente, [AT&T]
Walter Kozlowski, [Telstra]
RM Chapter 7 -Security

Sept 2019

THE LINUX FOUNDATION

GSMA

Security Objectives
Secure by Design at delivery

Networking functions must be designed with common security principles and standards.

- Perform VNF security validation/certification against CNTT reference implementations
- Ensure the OPNFV Verification Program for NFVi and VNFs meet CNTT Security design principles and requirements

Security Areas

- Platform Security
- Workload Security
- VNF Security

Discussion Items

Core Infrastructure Initiative badging

Opportunities for collaboration and partnerships in Security

How security fits within OVP

Options for Code Quality Scanning

Where we are today?

The OPNFV Security has CII Best Practices badging

The program is following [Linux Foundation (LF)]() [Core Infrastructure Initiative (CII)]()

> "Best Practices badge is a way for Free/Libre and Open Source Software (FLOSS) projects to show that they follow best practices. Projects can voluntarily self-certify, at no cost, by using this web application to explain how they follow each best practice. The CII Best Practices Badge is inspired by the many badges available to projects on GitHub. Consumers of the badge can quickly assess which FLOSS projects are following best practices and as a result are more likely to produce higher-quality secure software."

This shows a level of trust between various Linux Foundation projects and our users, we use security-conscience development processes that produce high quality software

- https://bestpractices.coreinfrastructure.org
- https://wiki.opnfv.org/display/security/Security+Home

Who is using the Core Infrastructure initiative badging?

Sponsored by the Linux Foundation

Widespread adoption

Includes:

- **OPNFV**
- **OpenStack**
- **ONAP**
- **Kubernetes**

kubernetes
HYPERLEDGER
node JS
LibreOffice
curl://
GitLab
Prometheus
pkgsrc
Xen Project
openstack
NVM
Zephyr
Syncope
libcloud
OpenSSL
BRLCAD
Nextcloud
GnuPG
GNU Make
blender
OPNFV

What are our opportunities?

Other LNF projects are developing functional test lifecycles for security compliance with OVP - user stories, slotting, backlog, etc (scrum team)

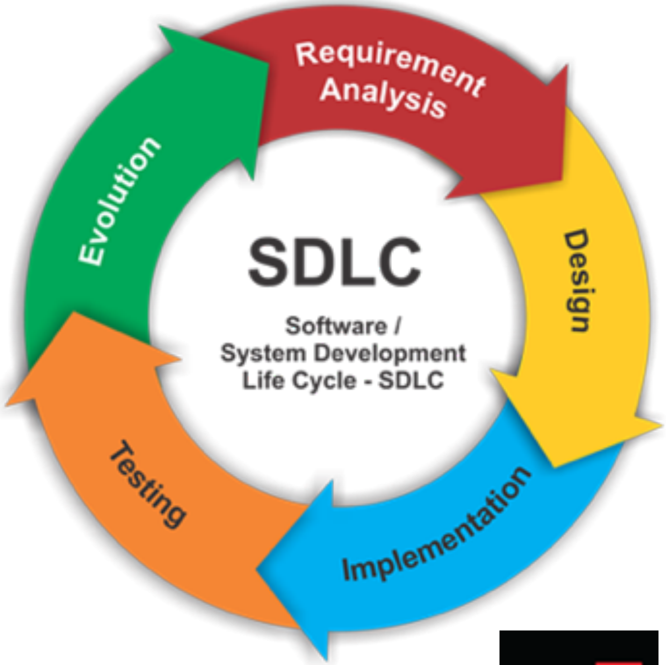
How do we best replicate that process in OPNFV? How do we become an active participant in the OPNFV Security Working Group, what is the state of the Security Working Group?

Lots of people are using OVP to validate/verify/certify “things”
How do we manage the end to end certification across multiple communities,
Especially with a focus on managing and oversight

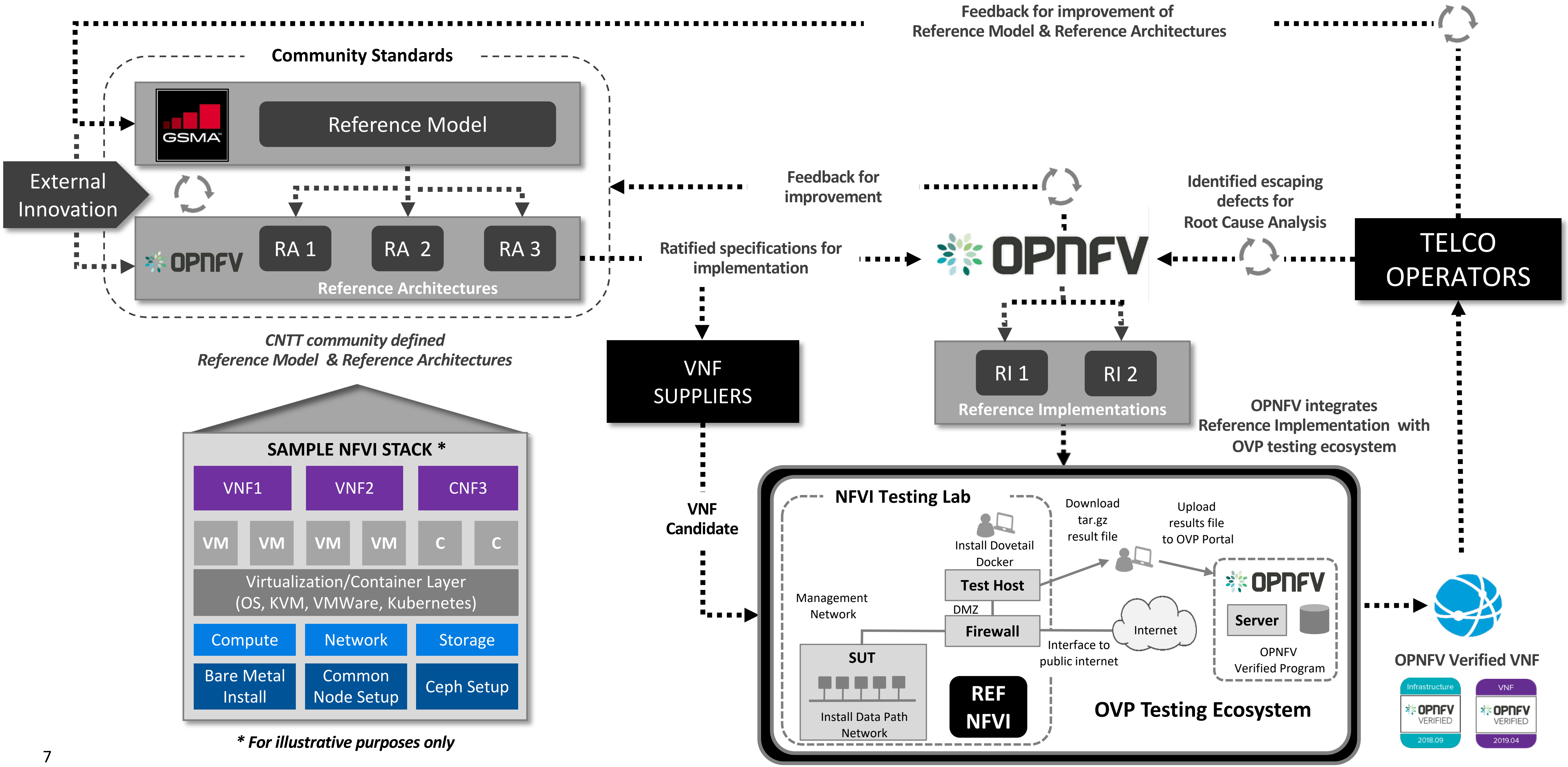
- Platform/NFVI
  - For Reference Implementation 1, work with OpenStack

- Workload/Operator
  - CNTT member’s best practices (Sweet Spot)

- VNF
  - Partner with ONAP and with VNF vendors on development practices



CNTT NFVI LIFECYCLE FRAMEWORK



Feedback for improvement of
Reference Model & Reference Architectures

Community Standards

GSMA

Reference Model

External Innovation

OPNFV

RA 1

RA 2

RA 3

Reference Architectures

CNTT community defined
Reference Model & Reference Architectures

Feedback for improvement

Ratified specifications for implementation

OPNFV

Identified escaping defects for Root Cause Analysis

TELCO OPERATORS

VNF SUPPLIERS

RI 1

RI 2

Reference Implementations

OPNFV integrates
Reference Implementation with
OVP testing ecosystem

SAMPLE NFVI STACK *

VNF1

VNF2

CNF3

VM

VM

VM

VM

C

C

Virtualization/Container Layer
(OS, KVM, VMWare, Kubernetes)

Compute

Network

Storage

Bare Metal Install

Common Node Setup

Ceph Setup

VNF Candidate

NFVI Testing Lab

Install Dovetail Docker

Test Host

DMZ

Firewall

Management Network

SUT

Install Data Path Network

REF NFVI

Download tar.gz result file

Upload results file to OVP Portal

Internet

Interface to public internet

OPNFV

Server

OPNFV Verified Program

OVP Testing Ecosystem

OPNFV Verified VNF

Infrastructure
OPNFV VERIFIED
2018.09

VNF
OPNFV VERIFIED
2019.04

** For illustrative purposes only*

Software Code Quality

Do we have anything to consider with centralized security scanning to increase VNF onboarding velocity? How. do we manage self-attestation from the VNF vendors?

For example: Vendors use industry recognized software testing suites to perform:

- Automated static code review with remediation of Medium/High/Critical security issues. The tool used for static code analysis and analysis of code being released must be shared.
- Dynamic security tests with remediation of Medium/High/Critical security issues. The tool used for Dynamic security analysis of code being released must be shared.
- Penetration tests (pen tests) with remediation of Medium/High/Critical security issues.
- Methodology for ensuring security is included in the Agile/DevOps delivery lifecycle for ongoing feature enhancement/maintenance.

Workload Security

Do we integrate with the OPNFV End User Advisory Group and/or Yardstick?
Ability to ensure that the configuration of the control plane and NFVI meet general security policies of the service provider.

The ability of VIMs to leverage and be protected by open source tools and best practices for security hardening, can be verified with functional testing such as:

- real-time vulnerability scanning
- network intrusion and host intrusion
- security audits
- open source scanning for licenses, vulnerabilities, and known defects
- compute host software configuration
- image software configuration
- real-time installation from a local, pre-scanned package/repo/image/container cache

- https://wiki.opnfv.org/display/EUAG/Security+and+Policy

Appendix