Common NFVI Telco Taskforce
Antwerp Face-To-Face Sessions

Fu Qiao, China Mobile
Mark Shostak, AT&T
Mike Fix, AT&T

Reference Implementation (RI)

September 2019

THE LINUX FOUNDATION
Discussion Outline

Mission & Vision
- Objectives

Progress & Timeline
- RI Alignment

Project Proposal (CNTT RI)
- Goals | Scope | Documentation
- Committers | Contributors

OVP Framework & LFN Expectation of CNTT-RI

Working Items for CNTT-RI
- Installer Options
- Pipeline
- Workgroup

Keys For Success

*Challenges: Narrow Committer List, Req Gathering Templates, Release Alignment, OVP Framework Process, ...?
**Vision**

**Mission**

RI will represent a physical manifestation of a defined Reference Architecture which is used for lab build-out in support of NFVI+VNF verification and validation.

**Objectives**

- **Create NFVI reference implementations (RI)** based on the Common NFVI reference model (RM) and reference architectures (RA) defined in CNTT (Common NFVI for Telco Taskforce).
- Work with upstream communities and related projects in OPNFV to identify / create required test cases and test frameworks.
- Work with relevant projects in OPNFV to integrate, deploy and test RIs as part of OPNFV release lifecycle*.
- This project intends to follow and adhere to the gates and quality criteria used by OPNFV.
Progress & Timeline

**POST-PARIS**

**Project Proposal (RI)**

**Goal:**
Initiate project proposal to formalize communication, define deliverables, and review lab and test project opportunities

**Define & Align:**
- Draft Validation Strategy
- Submitted Project Proposal
- Defined deliverables
- Created communications model between CNTT and OPNFV & CVC
- Straw Man **Infrastructure Labs Requirements** (High Level)

**Align:**
- Project Deliverables
- Straw Man Lab Requirements
- OVP Validation Framework
- PDF/SDF Modifications needed

**Define:**
- Delivery dates and resources needed for Lab Setup and TC Refinement

**POST-PARIS**

**Delivery Alignment**

**Goal:**
Community alignment on foundational aspects and straw man; begin deeper dive into Implementation Roadmap

**Align:**
- Project Deliverables
- Straw Man Lab Requirements
- OVP Validation Framework
- PDF/SDF Modifications needed

**Define:**
- Delivery dates and resources needed for Lab Setup and TC Refinement

**POST-ANTWERP**

**Define Reference Implementation (Details)**

**Goal:**
Task Force alignment on detail for Reference Implementation, including Labs and Validation Scenarios

**Define & Align**
- Receipt Initial Reference Architecture(s) (Detail) with supporting Open Source solutions
- Deliver Lab and Test Case Requirements
- Begin to incorporate / align with Reference Implementation delivery:
  - Deployment (Lab Setup)
  - Validations (OVP / CVC)

**Timeline:**
- **July 9 – August 25**
  - Submit Project Proposal – 8/23
- **August 26 – September 27**
  - Approved Project Proposal – 9/17
  - Launch RA – 9/27
- **October 1 – March 31**
  - Setup Lab – 1/31*
  - Complete RI Validations – 3/31*
Project Proposal

Reference Implementation for Common NFVI Telco Taskforce (CNTT)

**Project Name:** CNTT-RI

**Proposed Release Schedule:** First Release in 1Q2020

**Project Lead:** Mike Fix (AT&T) - active CNTT member

**Goal:** Landing place for CNTT’s work in OPNFV; Triggering the following progress within OPNFV

---

**Flow**

- **Installer projects**
  - Scenarios

- **Testing Projects**
  - Verification cases

---

**As RI in CNTT’s Context**

**As Verification in CNTT’s Context**
Project Proposal

**OPNFV Scope**

- The project will act as the landing space for CNTT RA within OPNFV, and will be a starting point for creating RIs.

- **Translate RAs into deployable scenario descriptions**, which can be considered as Reference Implementations for CNTT

- Work with OPNFV and upstream projects to close any gaps in identified RA/RI components

- Work with installer & test projects to generate scenarios for installation, testing, and RI test cases

- Work with LFN Compliance and Verification Committee (CVC) to integrate RI, test cases/frameworks with OVP Framework.
Project Proposal

**OPNFV Documentation**

- **(Long term) Available on official OPNFV Documentation portal and will include:**
  - Reference implementation description for Common NFVI
  - Pointers to CNTT Reference Architectures with related install/config information
  - Test requirements for Common NFVI
  - Gaps discovered from while integrating, deploying and testing RIs
- **(Present) All documentation generated by CNTT-RI project will reside in CNTT main repository.**
Project Proposal

Dependencies

• Project relies on the progress of CNTT.
• Parallel work can be done while CNTT finalizes the first RA

Planned Deliverables

• Reference implementation for Common NFVI
• Test Requirements for Common NFVI

Committers & Contributors*

• Mike Fix (Michael.Fix@att.com)
• Fu Qiao (fuqiao@chinamobile.com)
Project Proposal – Community Feedback

**Logistics & Execution**
- Documentation Repo (short-/long-term)
- Weekly Meeting
- Resolving Issues / Different Proposals (meetings, online issue mgmt.)
- Creation of Epics (Lab, Requirements, Test Plan)
- Release Alignment

**Collaboration with Test Projects**
- Tools Development (reuse)
- Test Cases (reuse + augment)

**Mechanics**
- Project Type (Feature v. Requirements)
Generate Installer Specs for NFVI required state based on Ref A, Ref M.
Align with CVC for Lab requirements (HW Specs) and OVP releases.
Spec, Create, and Accelerate development of Test Cases and Scripts for **NFVI Validation**.
Spec, Create, and Accelerate development of Test Cases and Scripts for **NFVI Performance**.
Req, Spec, Create, and Accelerate development of Test Cases and Scripts for **VNF Validation**.
Spec, Create, and Accelerate development of Test Cases and Scripts for **VNF Performance**.
Feeding in Requirement for ONAP VNFRQTS projects (such as T-Shirt Sizes, etc).
Working Items for CNTT-RI project

1. PDF/SDF modification requirement*

2. Test case gap analysis and new case development

3. Upstream gap analysis and development (identify if the current open-source OpenStack can fit into the RA defined in CNTT. If there are gaps, such work need to be developed within this project, and should probably go back to OpenStack Community)

4. Start talking with installer project, work out what should be done by CNTT-RI project and what should be done by installer project to accomplish RI
OPNFV Installer Options

• Airship for R1

• Do we need multi-installer support for CNTT-RI? What other choice do we have? Do CNTT have specific requirement for installer?

• Do we consider installer as part of the RI, or as just a tool to bring the RI up. This will decide whether our verification will include installer part, or we allow other commercial/open-source installer to pull up the verified RI? If so, what changes need to be done in OVP/dovetail?
OPNFV CI Pipeline

- OPNFV CI is important to generate RI in a continuous way. It is also important that Operators can reuse this CI to run the integration and verification within their own environment.

- Is current OPNFV CI good enough for CNTT RI?

- Bring OPNFV XCI project into scope. Is this also something we/OPNFV should adopt for rapid CI?
WG/Committee for long-term LCM

**Role of WG/Committee**

- Connection with CNTT as upstream
- Common platform for community member of OPNFV to learn progress of CNTT
- Common platform for CNTT related project (CNTT-RI, Airship, testing projects, dovetail, and etc.) to communicate and collaborate
- Platform to generate suggestions and feedbacks to OPNFV TSC for change in order to fit into the new tasks of CNTT

**How to run the WG/Committee**

- Use Monday tech-discussion call to drive CNTT related activities could be considered as a BOF for the WG
- Propose this WG/Committee to OPNFV TSC for official decision
- Set up consolidate meeting schedule and working items, invite active community members, related PTLs, to be solid members for the WG
Keys to Success

Success Factors

- Normalized & agreed upon **NFVI requirements template** for repeatable installer consumption
- **Automated test processes** for CVC verification framework
- OPNFV responsible for inventory & schedule of labs for verification
- **Establish Working Group** (WG) for LCM

How can our partners help?

- Support for critical path items
- Engaged in discussions & document reviews
- Contribute & adoption of artifacts
- Raise critical/major issues/gaps with content; take ownership to address
- Provide test cases
COMMON NFVI LIFECYCLE FRAMEWORK

Reference Model & Reference Architectures

SAMPLE NFVI STACK *

- VNF1
- VNF2
- CNF3
- VM
- GM
- C
- C

Virtualization/Container Layer
OS, KVM, VMWare, Kubernetes

Compute Network Storage

Bare Metal Install
Common Node Setup Ceph Setup

* For illustrative purposes only

Feedback for improvement of Reference Model & Reference Architectures

OPNFV integrates Reference Implementation with OVP testing ecosystem

Identified escaping defects for Root Cause Analysis

External Innovation

Owner TBD

Reference Architectures

OPNFV Verified VNF

TELCO OPERATORS

NFVI Testing Lab

Download tar.gz result file

Upload results file to OVP Portal

Server

OPNFV Verified Program

OVP Testing Ecosystem

VNF Candidate

Feedback for improvement of Reference Implementations

OPNFV verifies VNF

Sample NFVI Stack

VNF SUPPLIERS

RI 1 RI 2 Reference Implementations

CNVT community defined Reference Model & Reference Architectures

External Innovation

Community Standards

GSMA

Reference Model

RA 1 RA 2 RA 3 Reference Architectures

Reference Implementations

RI 1 RI 2

OPNFV

SUT

Install Data Path Network

REF NFVI

DMZ

Internet

Install Docker

Server

Download tar.gz result file

Upload results file to OVP Portal

OPNFV Verified Program

OVP Testing Ecosystem

Test Host

Internet
CNTT NFVI Reference Levels

L0: Infra Abstraction for VNFs
- Common across any IaaS/Cloud/VIM Technology choice
- Exposes virtual resources to VNFs in the form of profiles/compute flavors
- Defines set of capabilities and metrics of NFVI concerning VNFs

L1: Cloud Platform Agnostic Req
- Set of features of NFVI to deliver capabilities
- Act as functional requirements for NFVI Reference Architecture
- Common across any IaaS/Cloud/VIM Technology choice

L2: High Levels
- One instance of this artefact per technology choice (OpenStack, VMware, etc.)
- The content of each instance is common across vendors for the same technology choice
- Focus on high level components and interfaces (such as virtio for OpenStack)

L3: Component Levels
- One instance of this artefact per technology choice (OpenStack, VMware, etc.)
- The content of each instance is common across vendors for the same technology choice
- Focus on functional blocks and interfaces for interoperability between components

L4: High Levels Design
- There will be multiple instances of this artefact (one per distribution/vendor products)
- The content of each "instance" is specific to that distribution, and includes configuration specifics that conform to higher levels

L5: Low Levels Design
- As above but with more specific details relative to a specific version of a distribution
- Also, this layer will include CPU architecture specifics?