Common NFVI Telco Taskforce

Technical F2F Work Shop – January 13-16, 2020

RC Workstream: Key Updates

Facilitator: Mike Fix, Cedric Olivier, Rajesh Rajamani, Kanag

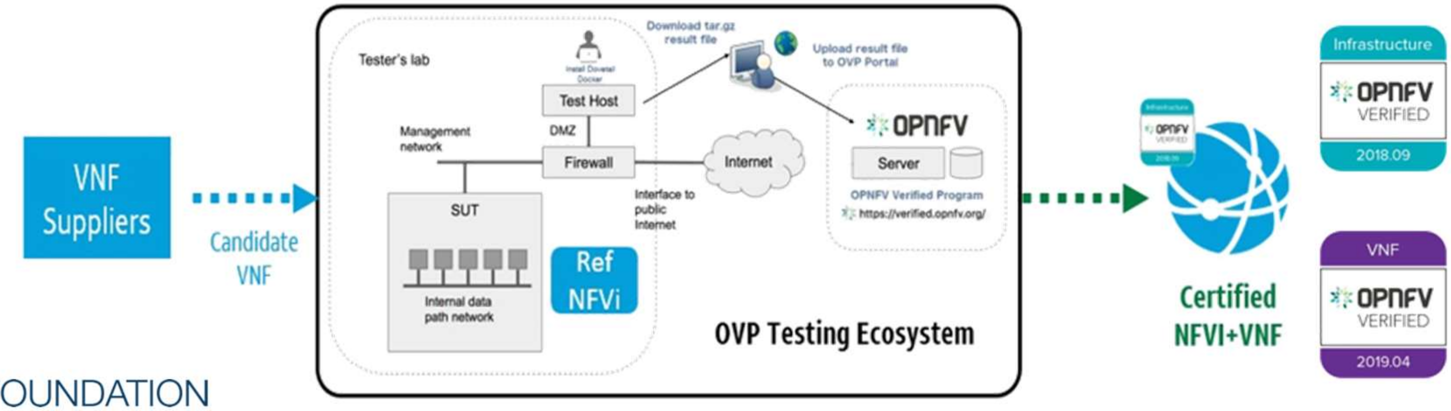
Content & MVP Targets

Progress to Date | Key Accomplishments

Mike

Objectives

- Re-write(WIP) for Certification Process/Frameworks/Badging
- Onboard Additional Team Support (Frameworks), Methodologies, Badging
- Stage Jenkins Hosts & Prep Health/Smoke Suites
- Perform initial compliance validations
- Identify Gaps in initial Alpha RI Release and expected compliance validations
- **Initial Badging Framework for NFVI | VNFs**
- **Certification Process Drafted**
- **Automation Tool Chain Framework**



Mike

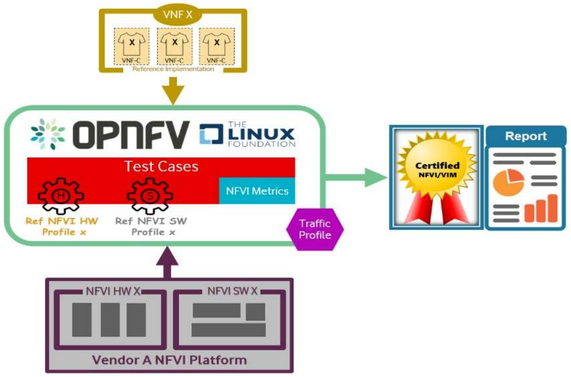
Level Set on MVPs

Initial Badging Framework for NFVI | VNFs

- ✓ **Lab** setup
- ✓ **Compliant** requirements
- ✓ **Execution** empirical, verification & validation
- ✓ **Evidence** meeting qualifications
- ✓ **Governance (Badging) reviews** and badging

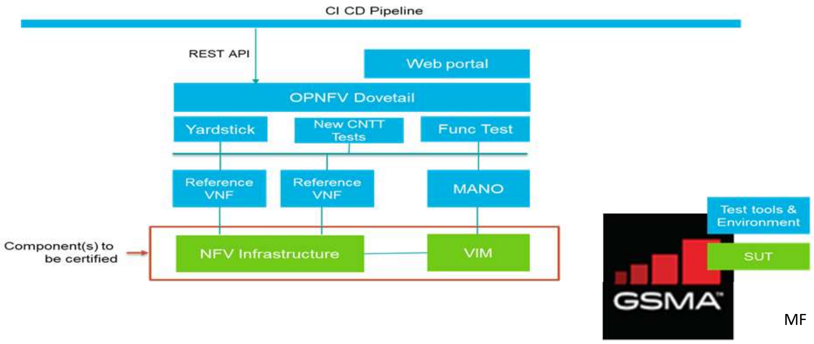
Certification Process Drafted

- ✓ **Certified Lab** utilized
- ✓ **Test Case Traceability** to req's
- ✓ **Execution** complete & passing
- ✓ **Results Collation** normalized & centralized
- ✓ **Evidence** meeting qualifications
- ✓ **Governance** reviews of Entry/Exit criteria and certification



Automation Tool Chain Framework

- ✓ **Refactor** existing OVP toolchain
- ✓ **Versatile** test harnesses using standard interfaces and services
- ✓ **Supplier Integration enabling** VNF testing using Supplier Apparatus
- ✓ **Adaptable** & Portable Tool Chaining across releases



MF

THE LINUX FOUNDATION

Progress: Initial Content Creation

Mike

Initial Content

NFVI

- ✓ Ch01: Introduction
- ✓ Ch02: NFVI E2E C&V Framework Requirements
- *alpha* ◕ Ch03: NFVI Test Case Requirements
- *alpha* ◕ Ch04: NFVI TC Traceability to RA Requirements.

VFN

- ✓ Ch05: VNF E2E C&V Framework Requirements
- *alpha* ◕ Ch06: VNF Test Case Requirements
- *alpha* (!) Ch07: VNF TC Traceability to RM Requirements.

Dev

- *alpha* ◕ Ch08: E2E Framework Integration.
- *alpha* ◕ Ch09: NFVI Tests Traceability to TC Requirements.
- *alpha* (!) Ch10: VNF Tests Traceability to TC Requirements.
- *alpha* ◔ Ch11: Gap analysis & Development

“RI-alpha & RC-pre alpha Phase”

Delivered - Snezka MVP

- Defined Certification
- Provide NFVI and VNF Certification Methodology
- Outline E2E Frameworks for Tools, Badges, and Process
- Included Gaps Identified During Installation

Next Release

- Include Normalized Results Template & Repo
- Finalize TC Requirements and Traceability
- Reach consensus on Entry & Exit Criteria

THE LINUX FOUNDATION

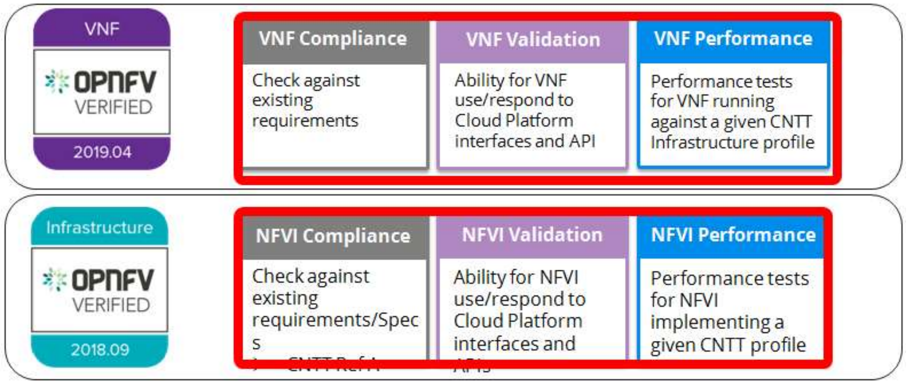
GSMA

MF

Progress: Badging Requirements

Kanag

Badging Requirements



Check against existing requirements

Ability for VNF use/respond to Cloud Platform interfaces and API

Performance tests for VNF running against a given CNTT Infrastructure profile

Check against existing requirements/Specs

- CNTT Ref A

Ability for NFVI use/respond to Cloud Platform interfaces and APIs

Performance tests for NFVI implementing a given CNTT profile

Badging - granting of certification by the OVP to Suppliers of NFVI+VNF upon demonstrating testing confirms:

- NFVI adheres to CNTT RA/RM requirements.
- VNFs pass interoperability tests on target NFVI with acceptable levels of stability and performance.

Suppliers seeking NFVI & VNF certification furnish the following:

Category	OVP/CVC Expectation	Supporting Artifact(s)
Lab	Delivered test lab conforms to RI-x lab requirements for SUT	Bare-metal H/W Validations
Compliance	Installed software conforms to RM/RA requirements for components & options	Manifest S/W Validations
Validation	FR Validation of Component and API functional behavior meets requirements	API & Platform Test Results
Performance	NFR Validation of Component, Interface, and API, results are within baseline tolerance	Performance Test Results
Results Reporting	Test Results published into centralized and common repository & portal	Normalized Results per Standards
Release Notes	Supplier provides concluding remarks, links to artifacts, having met exit criteria for testing	Release Notes

Progress: Badging Requirements.. Test Results

Kanag

Categorization. Test suites – Functional/Platform or Performance based

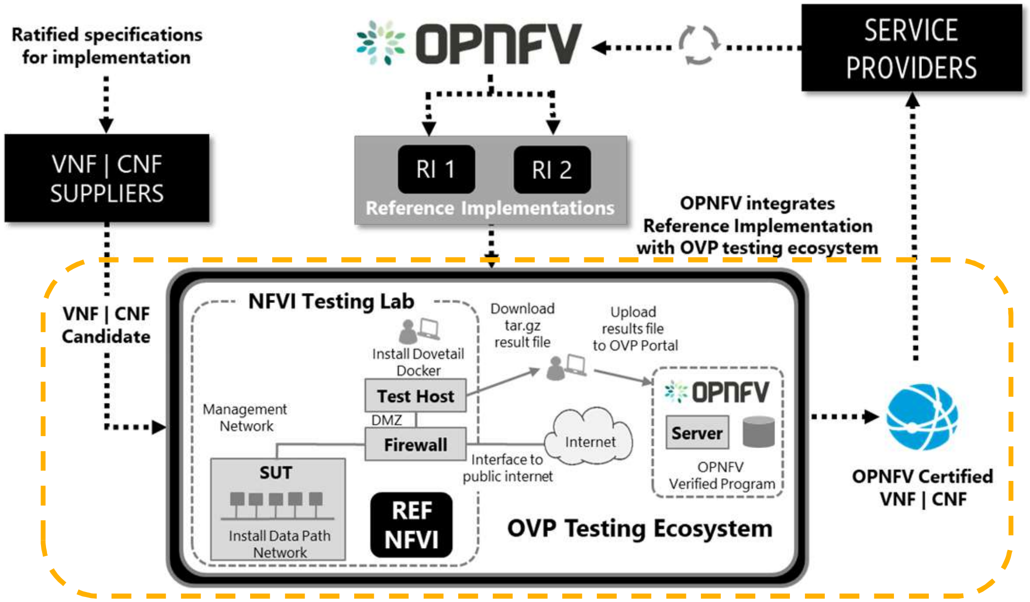
Results. Test results communicated as boolean (pass/fail), or Measurements Only

- **Functional Pass/Fail** - assertions in a test script verify the FR met its stated objective delivered by the developer
- **Performance-based Pass/Fail** - compares measured results with NFR KPIs &/or Reference VNF KPIs
- **Measurement Results -** baseline measurements when no benchmarks available to compare

Collation | Portal. Criteria applied to collation and presentation of test-result data:

- RA number and name (e.g. RA-1 OpenStack)
- Version of software tested (e.g. OpenStack Ocata)
- Normalized results will be collated across all test runs (i.e. centralized database)
- Clear time stamps of test runs will be provided.
- Identification of test engineer / executor.
- Traceability to requirements.
- Summarized conclusion if conditions warrant test certification (see Badging Section).
- Portal contains links to certification badge(s) received.

Reference Certification



Ratified specifications for implementation

VNF | CNF SUPPLIERS

OPNFV

SERVICE PROVIDERS

RI 1

RI 2

Reference Implementations

OPNFV integrates Reference Implementation with OVP testing ecosystem

VNF | CNF Candidate

NFVI Testing Lab

Install Dovetail Docker

Test Host

Management Network

DMZ

Firewall

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 Certified VNF | CNF

Rajesh

Objective

Deliver community certified NFVI | VNFs | CNFs to the Service Provider Marketplace

Goals

- Provide uniform approach for NFVI | VNF | CNF certification process, lifecycle, & badging
- Certify NFV | VNF | CNF on infrastructure, instantiation, tear-down, performance, & resiliency
- Provide VNFs | CNFs with effective & efficient intake & onboarding for Lab Management
- Ensure test framework can be reused for Manifest, Empirical, and Interoperability validations for new distributions

Target Delivery (pending alignment)

March | April 2020 (v 1.0 - Alpha)
Aligns with Reference Architecture # 1 (OpenStack)

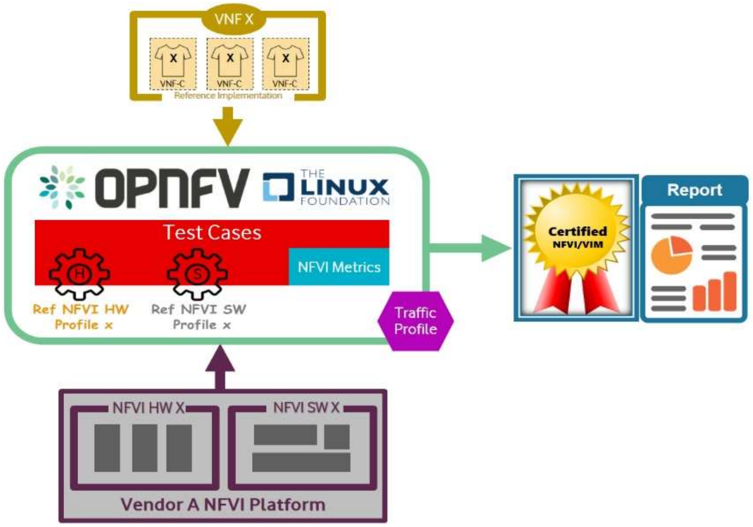
Progress: Certification Process Framework

Rajesh

Certification Process Framework

Core Principles

- **Certification fulfilled** by the OPNFV Verified Program (**OVP**), under the Linux Foundation Networking (LFN) umbrella
- Program **overseen** by the Compliance Verification Committee (**CVC**) providing tracking and governance
- **NFVI and VNFs supplied** by vendors **must adhere** to Reference Model (RM) and Reference Architecture (RA)



By Definition

- **Verification** conformance that NFVI is delivered per implementation specifications
- **Validation** testing performed confirms the actual output of a product meets the expected or desired outcome, or behavior
- **Certification** issuance of NFVI/VNF badges in recognition of the successful completion of verification and validation testing

Certification and issuance of NFVI+VNF badges

Infrastructure
OPNFV VERIFIED
2018.09

✓ Utilization of target RM/RA-x certified RI lab
✓ Traceable test cases to requirements
✓ Adoption & Execution of XTesting for RC pre-alpha validations

✓ Collation of Normalized Results in Centralized Repository
✓ Entry and exit criteria satisfied
✓ Required artifacts supplied to the OVP

VNF
OPNFV VERIFIED
2019.04

THE LINUX FOUNDATION

GSMA

RR

Reference Certification Achievements | Targets for Alpha

NFVI Compliance

Infrastructure OPNFV VERIFIED 2018.09

VNF OPNFV VERIFIED 2019.04

Mike

Scope & Test Strategy

- **Manifest Verifications** verify NFVI matches hardware and software profile specifications for RM/RA
- **Empirical Validations** baseline NFVI and Ref/Golden VNFs behaviors for future comparison
- **Interoperability Validation** performed leveraging VVP/CVC test suites to ensure VNF can be spun up, modified, or removed, on the target NFVI

Not In Scope

- VNF functional testing
- ONAP as a MANO for VNFs
- Validating VNF's ability to be upgraded
- Georedundant and Load Testing

RI-Alpha & RC-Pre Alpha Release

Xtesting and Xtesting CI meet Requirements for verification, compliance and certification:

- Assembly of multiple heterogeneous test cases
- OPNFV Releng Jenkins jobs to verify RI
- Test case results and logs for third-party certification review
- Deploy local CI/CD toolchains to verify RI compliance

Framework

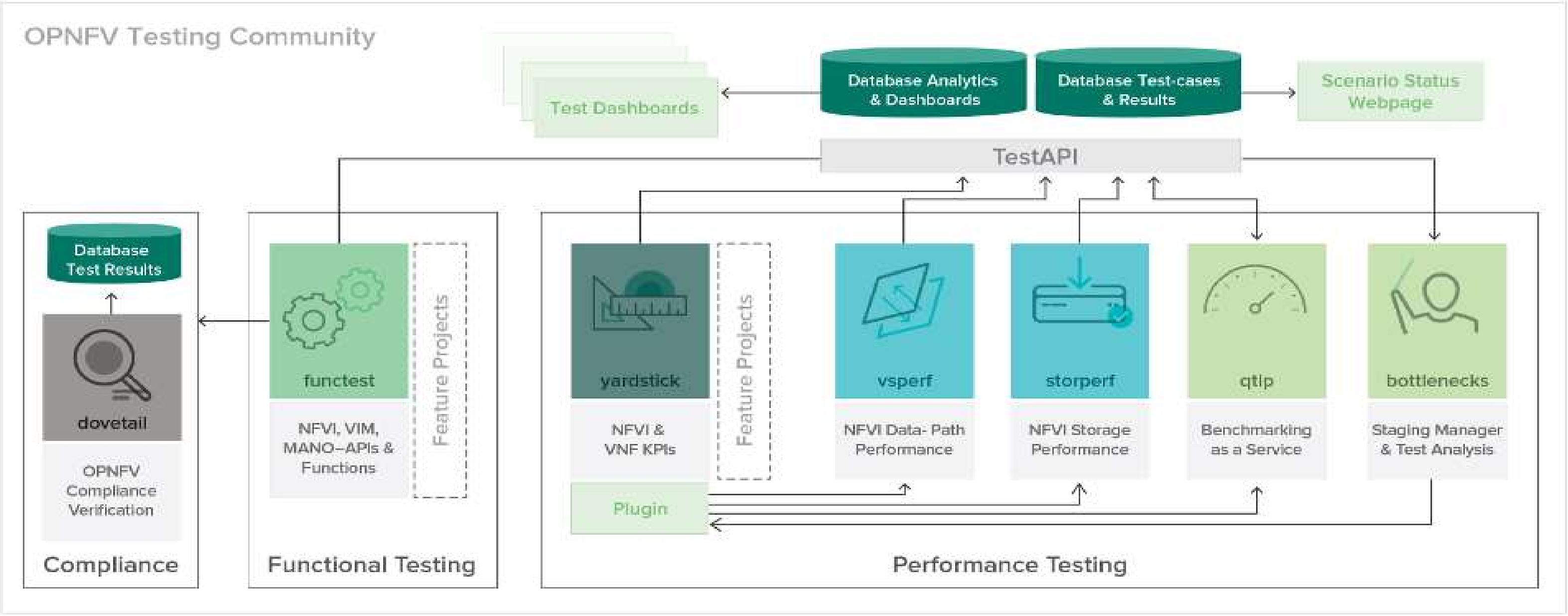
Infrastructure OPNFV VERIFIED 2018.09

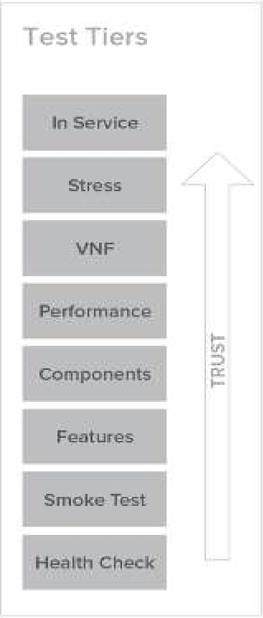
VNF OPNFV VERIFIED 2019.04

Cedric

- ✓ NFVI+VNF verifications performed against well defined profiles
- ✓ Utilizing common CICD frameworks across RI and RC via CIRV project:
  - ✓ RI verification (jjb in releng)
  - ✓ RC compliance and certification (local CI toolchains)
- ✓ Test cases delivered as [Docker containers]

- ✓ Automated CIRV deployments: **PMO**: local component deployment, **FMO**: Jenkins triggered deployments
- ✓ OPNFV test cases launched automatically for NFVI verification
- ✓ Test results stored in a [centralized test database] along with all artifacts (reports, logs, etc.) to [an S3 storage service]

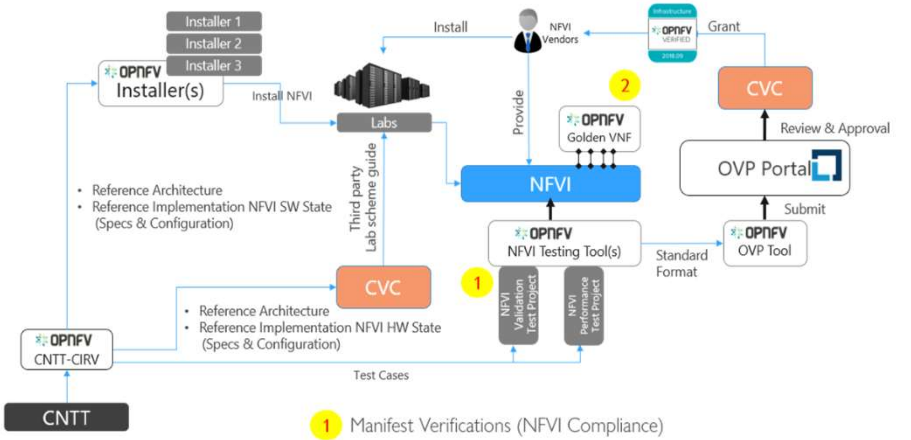




CO

Certification Process | Gap Analysis

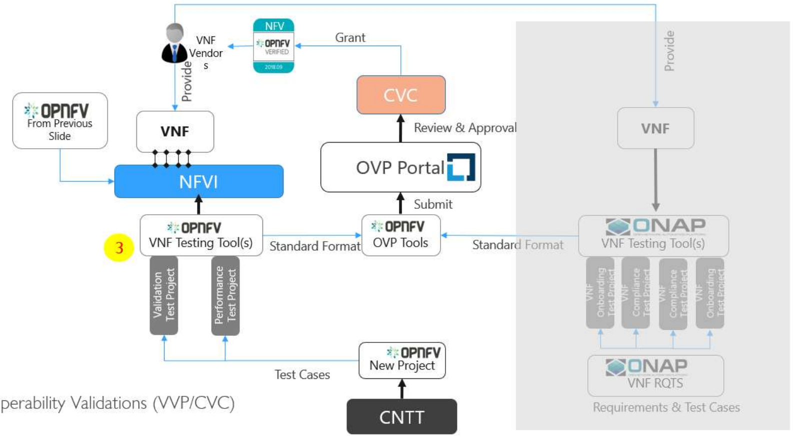
Cedric



Installer 1
Installer 2
Installer 3
OPNFV Installer(s)
Install NFVI
Labs
Install
NFVI Vendors
Infrastructure OPNFV VERIFIED 2018.09
Grant
CVC
Review & Approval
OVP Portal
Submit
OPNFV OVP Tool
Standard Format
2
OPNFV Golden VNF
Provide
NFVI
OPNFV NFVI Testing Tool(s)
Third party Lab scheme guide
• Reference Architecture
• Reference Implementation NFVI SW State (Specs & Configuration)
CVC
1
NFVI Validation Test Project
NFVI Performance Test Project
• Reference Architecture
• Reference Implementation NFVI HW State (Specs & Configuration)
OPNFV CNTT-CIRV
Test Cases
CNTT

1 Manifest Verifications (NFVI Compliance)

2 Empirical/Baseline Validations (NFVI Compliance)



3 Interoperability Validations (VVP/CVC)

Process

- ➢ **Certification based on successful delivery of:**
  1) **Manifest Verifications** – confirming NFVI delivered per RI-x requirements.
  2) **Baseline Validations** – measuring FR/NFR behaviors using reference VNFs
  3) **Interoperability Validations** – confirming capabilities, stability and perf
- ➢ **For NFVI Certification:**
  - Vendor NFVI/VIM images under Test will be installed and configured
  - For Phase 1, OPNFV **FuncTest** tests used for Compliance and Validation
- ➢ **For VNF Certification.:**
  - Vendor VNF images are ingested by the CICD pipeline, implemented in lab
  - VNF on-boarding and lifecycle operations validation is performed using upstream projects such as VNFSDK and VPP
- ➢ Test results submitted to OVP for review by committee. If results are validated, the vendor's application for certification is approved and the badge(s) awarded.

Gaps

- ➢ No automated means for Manifest (s/w) or Lab (h/w) Validations
- ➢ Need support vehicle for Installer changes (to meet RI-x specs)
- ➢ Normalization and Centralization of Results Alignment needed
- ➢ Storage and Performance scenarios (or tools) missing
- ➢ Need Installers which are OpenStack release agnostic
- ➢ Need priority on lab support with proactive monitoring

Exceptions: Certifications From the OVP Process

Cedric

Step 1:	Step 2:	Step 3:	Step 4:	Step 5:	Step 6:
Submission of Participation Form	Testing	Submission of Results	Notification of Reviewers	Community Review of Test Results	Grant of Use of Program Marks
Gap	Gap	Work in Progress	Work in Progress	Work in Progress	Work in Progress

OPNFV Verification Program (OVP) is a five step process resulting in the issuance of three badges: NFVI, VNF, and Lab

CNTT seeks to align with the OVP process, noting two Gaps above in process:

- Step 1 for client (NFVI/VNF) participation in badging and certification
- Step 2 for Test(ing) & tools utilized for testing.

Mitigating Process Gaps:

- **Step 1:** Friendly and Controlled Introductions in 2020, using key learnings to create formal participation mechanism
- **Step 2:** Use results from RI-alpha compliance validations with FuncTest and XTesting/XTestingCI and assess feasibility as a platform to be Dovetail compliant

THE LINUX FOUNDATION

GSMA

CO

Results: Compliance

Cedric

With RI-alpha

- Released first CNTT API compliance containers from OPNFV FuncTest
- Compliance containers forbid skipping tests for mandatory services/capabilities
- Commands deploy the full CI/CD toolchain within minutes
- Tests run for Compliance Verification include:
  - benchmarking
  - healthcheck
  - vnf (tests mainline API interopability.

✓ **Conclusion**: Functest confirmed the RI System Under Tests (SUT) is compliant (aka Pass) per RA1 Chapter 5 (API) feature capability and exposure, with the caveate using Ocata vs. Pike.

Observations

- OpenStack Helm (OSH) doesn't support live migration and resize server for Ocata
- Compliance checks for Live Migration and Resize Servers blacklisted
- Metrics needed for API / data-plane benchmarking

Recommendations (for RC-alpha)

- Implement Stein, OSH supported
- Update RM/RA-1 OpenStack documentation for Stein
- Leverage XTesting to wrap the RI deployment calls, simplifying the RI “cookbook”

Next Steps

Mike

Current Status

Completed:

- RM | RA | RI Requirements
- Lab Requirements
- Initial Lab Secured
- S/W Deployed | Config
- Smoke Test | Sanity
- Continuous Deployment (with errors)

In-Progress

- Manifest > PDF & IDF
- Complete Lab PoC & Deliver Lab
- Create Cookbook & RI Topology artifact
- Implement PoC Key Learnings

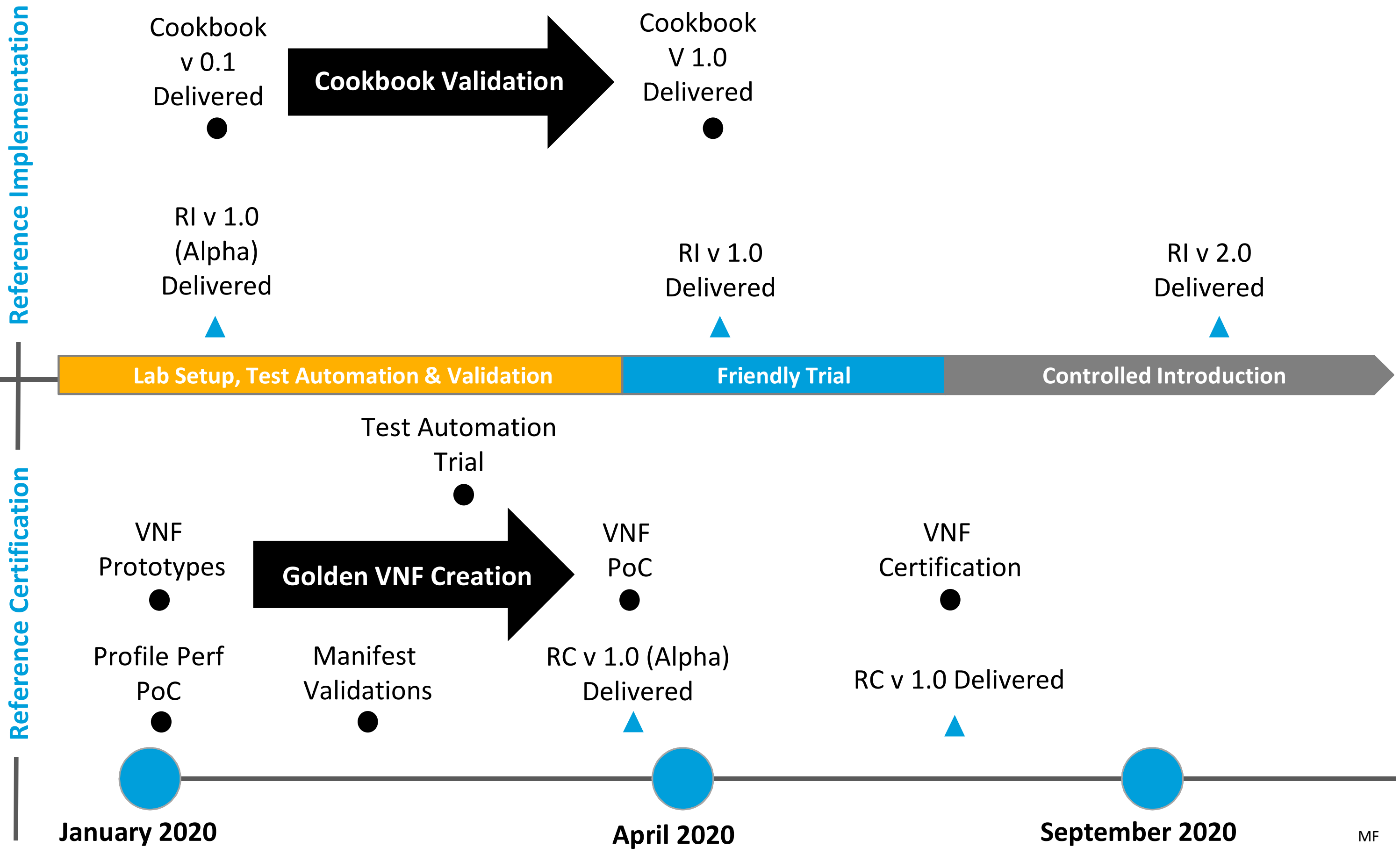
Completed:

- Define & Vet Verification Methodology
- Stage Jenkins Hosts
- Prep Health/Smoke Suites
- Completed API test harness setup
- Initial API Compliance Validation

In-Progress

- Design & deliver VNF Prototypes
- Integrate upstream community performance & storage test cases

Note: All events & dates are pending community alignment



Reference Implementation

Cookbook v 0.1 Delivered ●

Cookbook Validation

Cookbook V 1.0 Delivered ●

RI v 1.0 (Alpha) Delivered ▲

RI v 1.0 Delivered ▲

RI v 2.0 Delivered ▲

Lab Setup, Test Automation & Validation | Friendly Trial | Controlled Introduction

Reference Certification

Test Automation Trial ●

VNF Prototypes ●

Golden VNF Creation

VNF PoC ●

VNF Certification ●

Profile Perf PoC ●

Manifest Validations ●

RC v 1.0 (Alpha) Delivered ▲

RC v 1.0 Delivered ▲

January 2020

April 2020

September 2020

MF

Reference Certification Challenges

Rajesh

- Availability of OVP | CNTT resources & active engagement
- **OVP's ability to scale to demand, & alignment with CNTT objectives, including:**
  - Fully automated ecosystem & badging process
  - Intuitive and efficient VNF on-boarding processes
  - Lab optimization & rationalization
  - Lab-use management & control strategy
    - Augmentation of OVP processes to include third party certification platforms & process
    - Clear and efficient third party certification requirements
  - Audits of lab certifications confirming availability, state (current), & accessibility
- Normalized test results with a centralized repository
- Manifest validations require an automated check of requirements

CNTT will maintain ownership of the Reference Certification until a satisfactory level of support, stability, & maturity is attained

Appendix

Framework

- High level architecture of NFVI verification & compliance testing fram
- High level architecture of VNF verification & compliance testing frame
- NFVI+VNF verifications will be performed against well defined instance types consisting of a HW and SW Profile, Configured Options, and Applied Extensions (See image.)

Cedric

- **Assigned to Cedric**

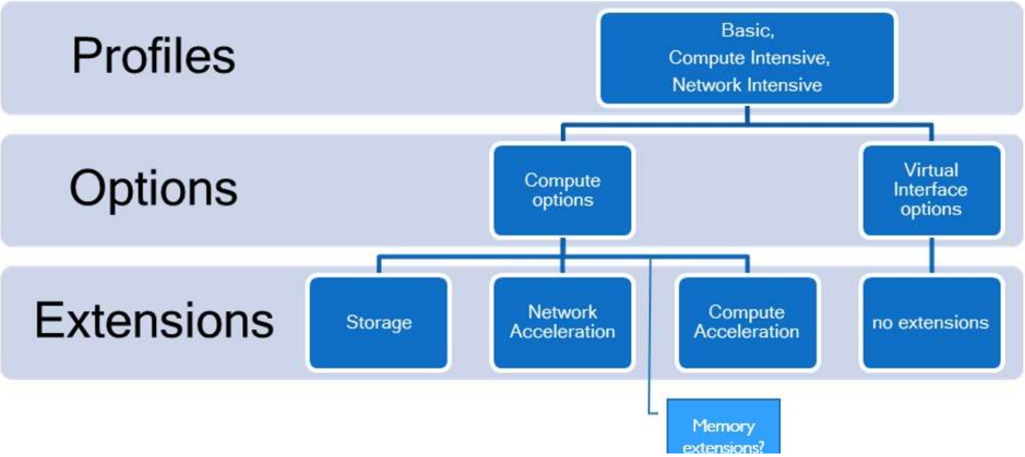
[Alpha RI and Pre-Alpha RC](#)

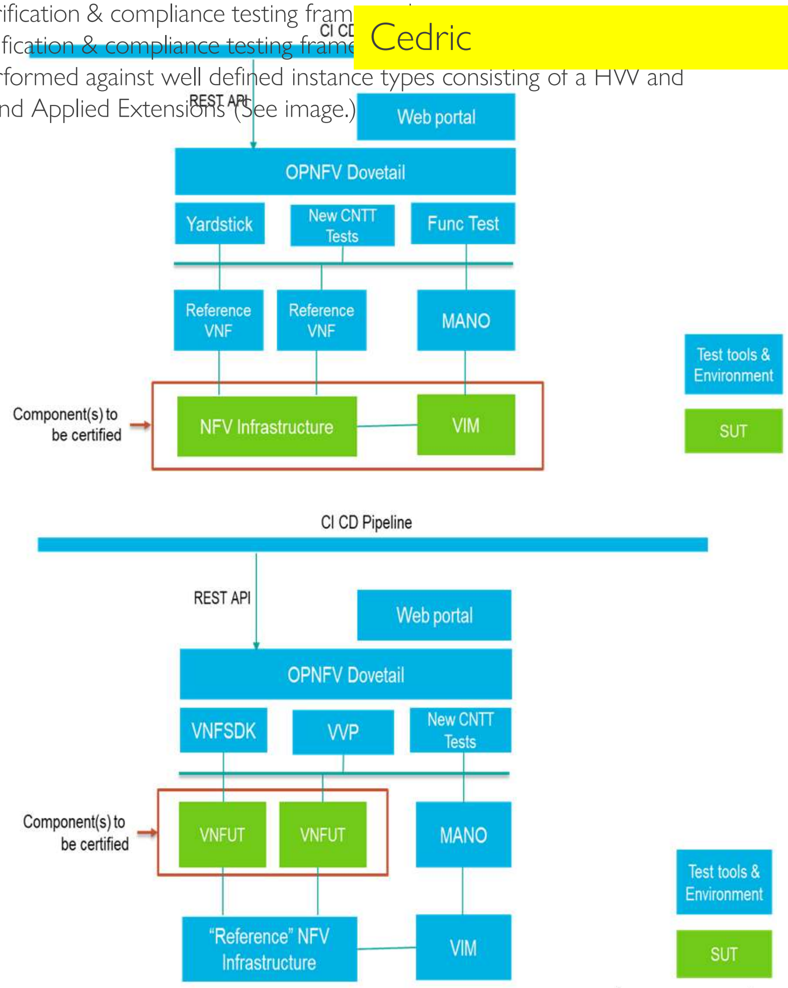
Leverage existing OPNFV testing knowledge (projects) and experience (history).

Utilize a common CICD framework across RI and RC. More specifically, implement:
- RI verification (jjb in releng)
- RC compliance and certification (local CI toolchains)

OPNFV has a complete CI/CD toolchain continuously deploying and testing NFVI utilized for the CIRV project.

From a Process perspective, Jenkins triggers:
- CIRV deployments (PMO: local component deployment; FMO: automated deployments)
- Runs the OPNFV test cases part of the NFVI verification
- Publishes all test results in the centralized test database and all artifacts (reports, logs, etc.) to an S3 compatible storage service (PMO: all results and logs for certification are provided today via the CIRV and Daily Jobs (Functest); FMO: incorporation of additional test projects)





NFVI Compliance

Infrastructure OPNFV VERIFIED 2018.09

VNF OPNFV VERIFIED 2019.04

Cedric

- **Assigned to Cedric**

In general, NFVI+VNF compliance process steps include:

•Based on VNF's requirements, NFVI profile is selected - B, C, N
•NFVI readiness is checked for certification.
•The test VNFs are on-boarded using automation scripts on the NFVI.
•VNF on-boarding is validated by running functional tests to ensure that the on-boarding is successful.
•VNF performance tests are executed and NFVI KPIs are recorded during the tests.
•KPI comparison is run to compare NFVI KPIs with Golden KPIs, which serve as a reference for NFVI certification.
•If NFVI KPIs meet Golden KPIs, NFVI is certified and granted a certification badge.
•If NFVI KPIs do not meet Golden KPIs, no certification is provided.

For RI-Alpha and RC-pre-alpha release:

Xtesting* and Xtesting* CI meet the CNTT requirements about verification, compliance and certification:

- smoothly assemble multiple heterogeneous test cases
- generate the Jenkins jobs in OPNFV Releng to verify CNTT RI
- deploy local CI/CD toolchains everywhere to check compliance with CNTT
- dump all test case results and logs for third-party certification review
- The CI/CD approach (methodology) will be detailed in additional PRs.

All test cases must be delivered as Docker containers embedding the Xtesting Python package and their test case execution description files as required by Xtesting.

- Docker containers simply enforce that the test cases are delivered with all runtime dependencies.
- All test projects should offer Xtesting CI description files (see Functest site.yml) to ease deploying plug-and-play CI/CD toolchains in a few commands

*Xtesting a simple framework to assemble sparse test cases and to accelerate the adoption of CI/CD best practices. By managing all the interactions with the CI/CD components (test scheduler, test results database, artifact repository), it allows the developer to work only on the test suites without diving into CI/CD integration.
*Xtesting CI tool deploys anywhere plug-and-play CI/CD toolchains in a few commands.

THE LINUX FOUNDATION

GSMA

Framework

Cedric

Principles

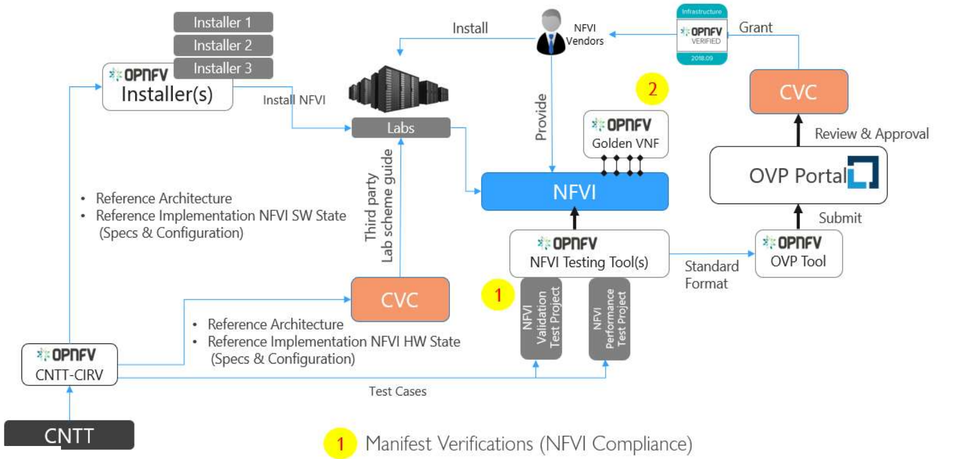
NFVI+VNF verifications will be performed against well defined instance types consisting of a HW and SW Profile, Configured Options, and Applied Extensions.

Leverage existing OPNFV testing knowledge (projects) and experience (history).

Utilize a common CICD framework across RI and RC. More specifically, implement:

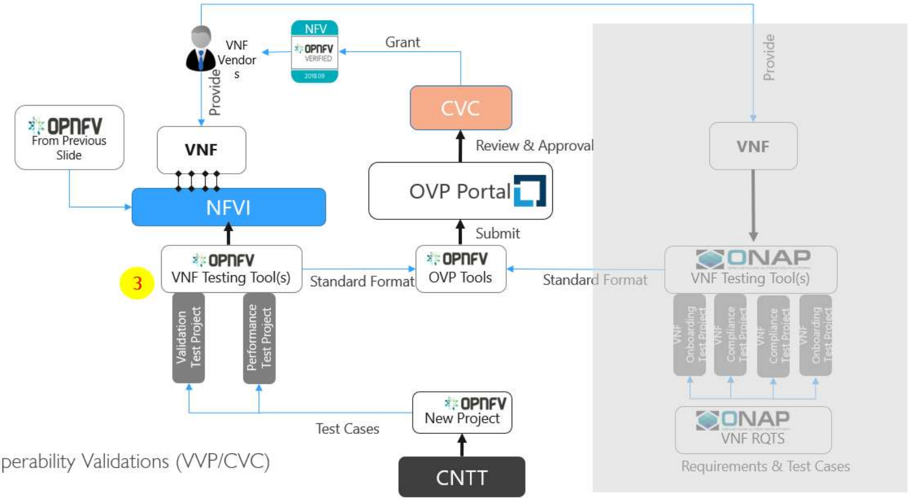
- RI verification (jjb in releng)
- RC compliance and certification (local CI toolchains)

OPNFV has a complete CI/CD toolchain continuously deploying and testing NFVI utilized for the CIRV project.



1 Manifest Verifications (NFVI Compliance)

2 Empirical/Baseline Validations (NFVI Compliance)



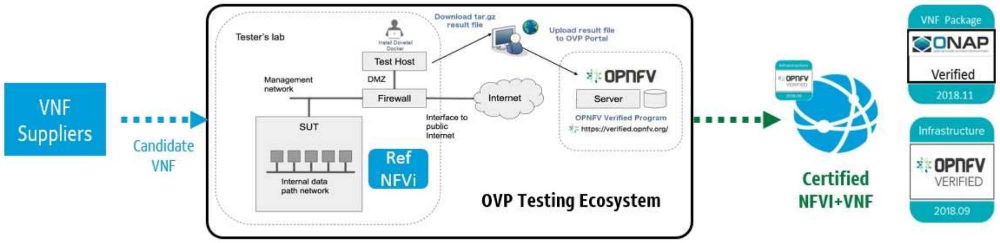
3 Interoperability Validations (VVP/CVC)

Jenkins triggers:

- CIRV deployments
  - **PMO**: local component deployment
  - **FMO**: automated deployments
- Runs the OPNFV test cases part of the NFVI verification
- Publishes all test results in the centralized test database and all artifacts (reports, logs, etc.) to an S3 compatible storage service
  - **PMO:** all results and logs for certification are provided today via the CIRV and Daily Jobs (Functest)
  - **FMO:** incorporation of additional test projects)

Certification Process | Gap Analysis

Cedric



1. The process starts with VNF or NFVI vendors providing VNF images, NFVI/VIM images or NFVI hardware components for certification. While self-certification by VNF/NFVI vendors (with final validation by CVC) is allowed, the following example describes the use of a 3rd party OVP certified test labs for performing VNF certification

2. For VNF Certification:

   - The VNF images are ingested by the CI CD pipeline implemented in the certification lab and stored in local image repositories.
   - VNF on-boarding and lifecycle operations validation is performed against an implementation of CNTT reference architecture, leveraging existing OPNFV Intake Process. Upstream projects such as VNFSDK and VPP will define features/capabilities, test scenarios, and test cases to augment existing OVP test harnesses to be executed via the OVP Ecosystem.

3. For NFVI Certification:

   - Vendor NFVI under Test will be installed and configured
   - For Phase 1 of the CNTT RI, OPNFV Yardstick, FuncTest and Bottlenecks tests will define test scenarios and test cases needed for NFVI certification.

4. The test results are submitted on the OVP portal and the results will be reviewed by a committee of reviewers. If the tests are satisfactorily validated, the vendor's application for certification is approved and the marks are granted to the vendor. For a more detailed description of the OVP process, please refer to the documentation by following the link below

https://www.lfnetworking.org/OVP/

Exceptions: Certifications From the OVP Process

Cedric

NFVI

- NFVI needs to be validated and expected to be in a required state. This state would be determined by running tests as described in CNTT RI. Once the target NFVI passes these tests, it would become a candidate for NFVI certification.
- NFVI+VNF validations consist of a three part process for Compliance, Validation, and Performance. Adherence to Security standards are equally important and addressed in Chapter 7 of CNTT RM.
- The three part verification process includes:
  - NFVI Manifest Validations
    - NFVI Verification (Compliance): NFVI is the SUT, ensuring NFVI is compliant with specs of RM and RA accomplished with Manifest Validations (performed via Echo Tests)
  - Empirical Baseline measurements against targeted VNF families
    - Empirical Validation with Reference VNF (Validation): NFVI is the SUT, ensuring NFVI runs with Golden VNFs and is instrumented to objectively validate resources through consumption and measurement
  - Candidate VNF verifications
    - Candidate VNF Certification (Validation & Performance): VNF is the SUT, ensuring VNFs operate with RM and RA leveraging VVP/CVP/VFNSDK Test Suites
  - Security: Ensures NFVI+VNF is free from known security vulnerabilities, utilizing industry standard cyber security frameworks (Refer to CNTT Chapter 7 Security for additional test/verification details) Validations are performed against an Infrastructure Profile Catalog, VNF performance profile, and targeted VNF class or family for baseline measurements.

VNF

- VNF certifications confirm capability of VNF to interact with the target NFVI with a high degree of stability.
- Certification begins with confirmation of the NFVI delivered and verified based on B/N/C:
  - Basic: for VNF that can tolerate resource over-subscription and variable latency.
  - Network Intensive: for VNF that require predictable computing performance, high network throughput and low network latency.
  - Compute Intensive: for VNF that require predictable computing performance and low network latency.
- Suppliers adhere to documentation standards for test cases:
  - Test case Name
  - Test case description
  - Virtual Network function Name
  - Network function Feature/functionality name
  - Test case input parameters
  - Test case result attributes
  - Test case version
- Validation is performed and Results Collated:
  - Interoperability validations for VNF functional testing defined.
  - Interoperability validations for VNF performance testing defined (IOPS, connection, threading, resource consumption).

THE LINUX FOUNDATION

GSMA

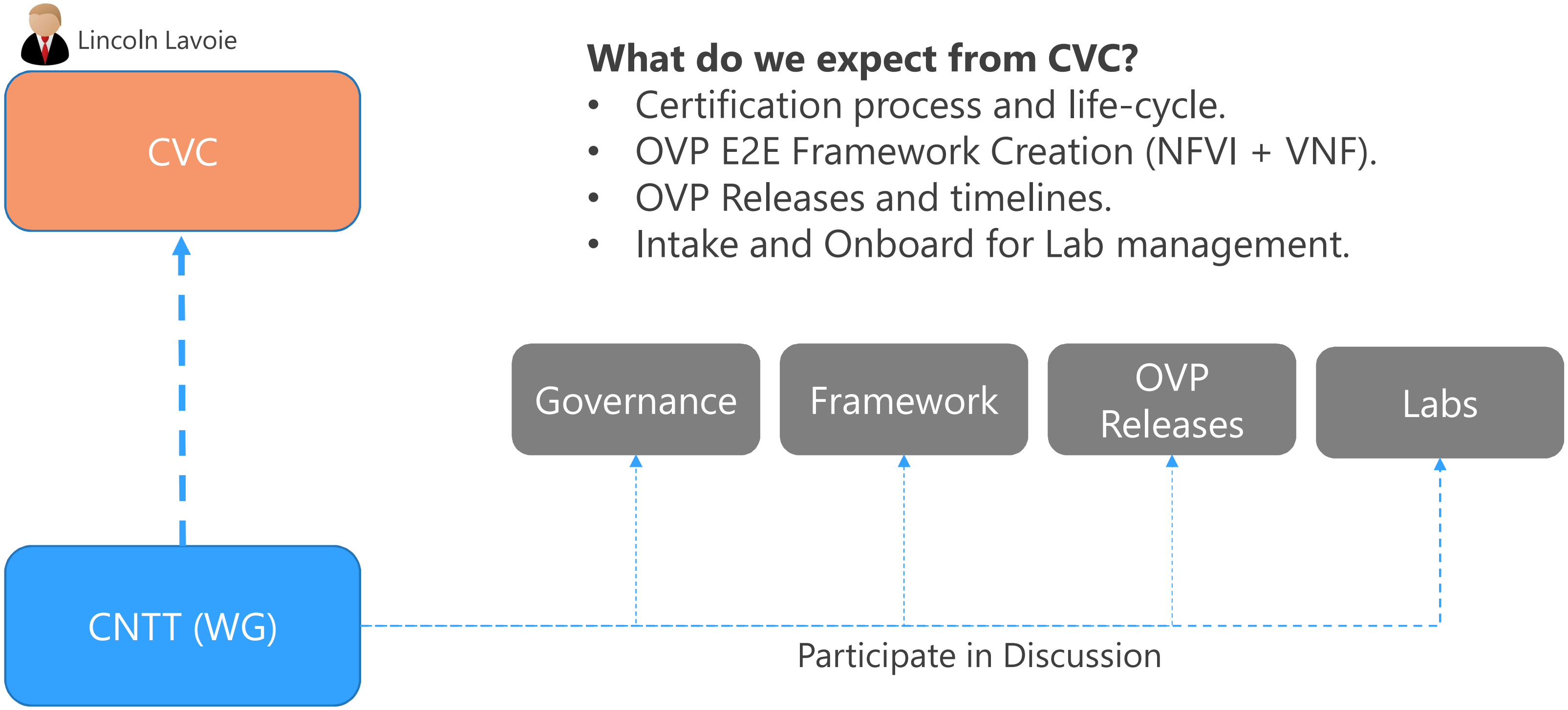
Reference Certification Approach | Outcomes

<table>
<tr><th>Actions Underway</th><th>Outcomes</th></tr>
<tr><td>Define requirements & criteria for profile based implementations & certifications</td><td>Clearly defined requirements to assess vendor software, hardware, and VNF solutions</td></tr>
<tr><td>Define badging requirements to achieve certification</td><td>Deliver compliant & stable VNFs | CNFs to CSPs</td></tr>
<tr><td>Define, align, and select NFVI | VNF | CNF test case requirements, covering:
• Infrastructure
• Instantiation
• Tear-Down
• Performance
• Resiliency
• Security</td><td>Ensure certification criteria and badging requirements are met through an robust & optimized test suite</td></tr>
<tr><td>Establish guidelines and/or processes for:
• Entry | Exit Criteria
• Test Categories
• Quality Assurance
• Test Case In-Take</td><td>Ensure proper level of structure & discipline exists within test ecosystem to effectively manage & scale to demand</td></tr>
<tr><td>Define & align across communities on the test framework & tooling</td><td>Optimized test ecosystem, designed to scale, manage, & perform VNF | CNF certifications</td></tr>
</table>

THE LINUX FOUNDATION

GSMA

Relationship with CVC

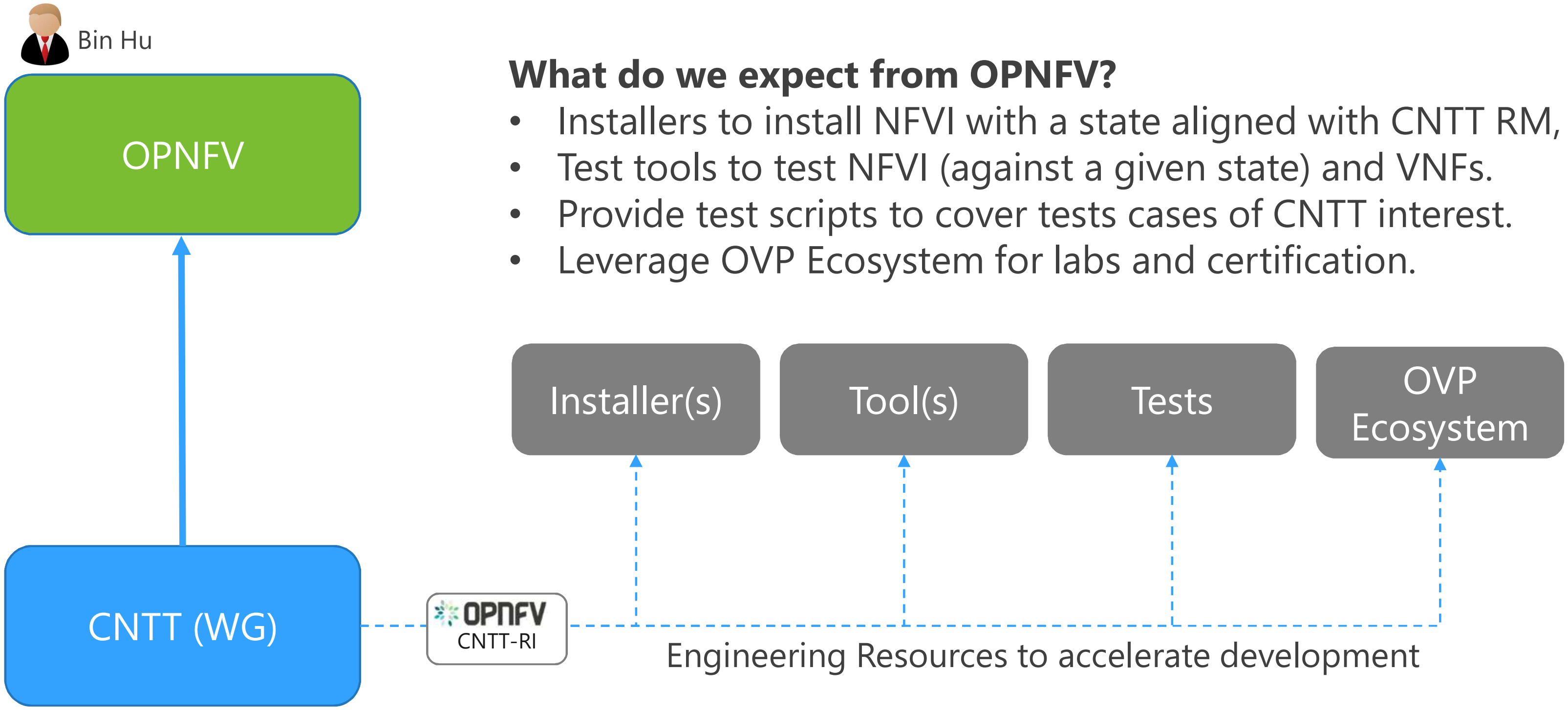


What do we expect from CVC?

- Certification process and life-cycle.
- OVP E2E Framework Creation (NFVI + VNF).
- OVP Releases and timelines.
- Intake and Onboard for Lab management.

- CNTT will work directly with CVC to align with governance
- Output of CNTT will be input to release scope, labs needs, and augment governance where needed

Relationship with OPNFV and OVP



What do we expect from OPNFV?

- Installers to install NFVI with a state aligned with CNTT RM, RA.
- Test tools to test NFVI (against a given state) and VNFs.
- Provide test scripts to cover tests cases of CNTT interest.
- Leverage OVP Ecosystem for labs and certification.

- CNTT will work directly with OPNFV via the RI Project
- Output of CNTT-RI will be RI requirements and test cases

MF

Chapter 8 Team: North Star

Mission

Ensure Implementation of CNTT Reference Model and Reference Architecture meets industry driven quality assurance standards for compliance, verification and validation.

Objectives

- **Data Driven** RA Verification and Validations
- **OPNFV, CVC, and OVP Processes used** to onboard and check for NFVI compliance
- **Entry** and **Exit** Quality **Standards** are satisfied
- Ensure **test harnesses** can be **ported** and utilized **across multiple distributions**

Guiding Tenets

- **Verification** and **Validations** determine NFVI+VNF compliance
- **Verification** signals conformance to design requirement specifications
- **Validations** signals compliance that output of a product meets the expected, or desired outcome
- **Certifications**, are out of scope as this measures adherence to development, however, no code is being delivered by testing
- OVP and CVC track and govern RM/RA verification

Scope

Scope & Test Strategy

- **Manifest Verifications** verify NFVI matches hardware and software profile specifications for RM/RA
- **Empirical Validations** baseline NFVI and Ref/Golden VNFs behaviors for future comparison
- **Interoperability Validation** performed leveraging VVP/CVC test suites to ensure VNF can be spun up, modified, or removed, on the target NFVI

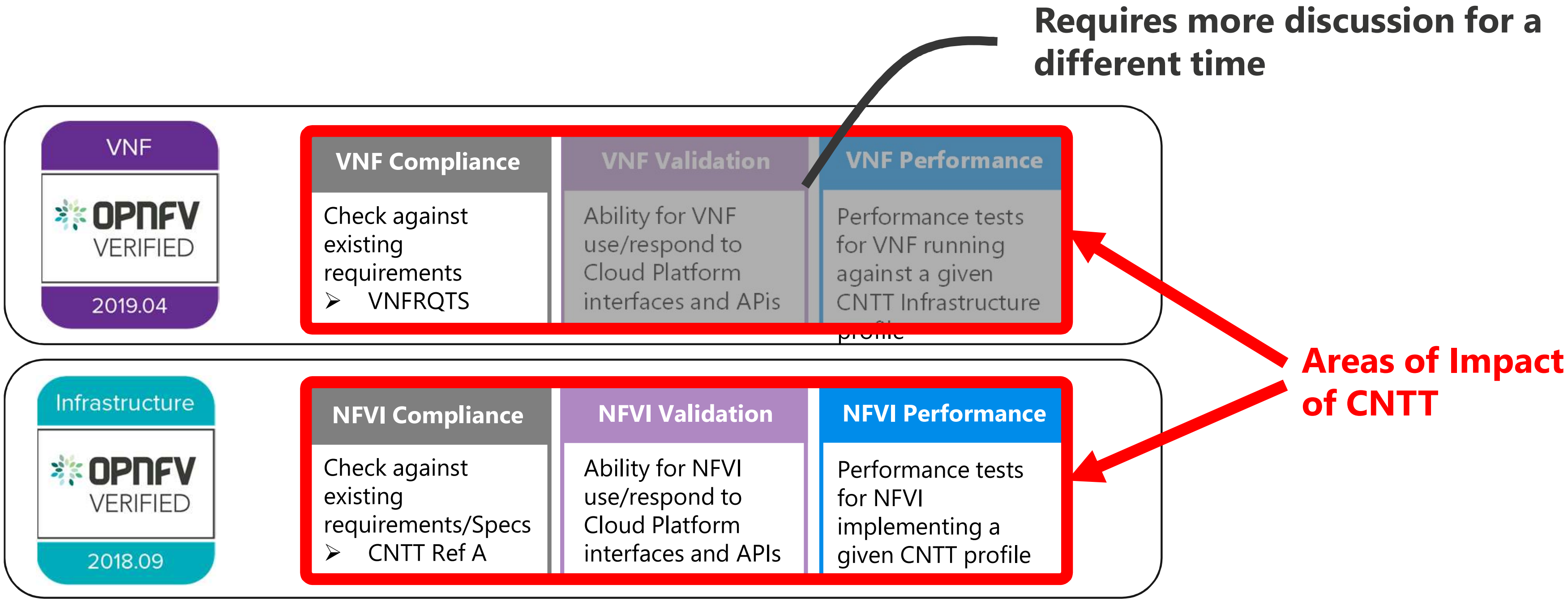
Not In Scope

- VNF functional testing
- ONAP as a MANO for VNFs
- Validating VNF's ability to be upgraded
- Georedundant and Load Testing

Different Distributions

Repeat the strategy of Manifest Verification, Empirical Golden VNF Validations, and Interoperability Testing for any new Distributions.

Verification & Validation Scope



Requires more discussion for a different time

VNF OPNFV VERIFIED 2019.04	**VNF Compliance** Check against existing requirements ➢ VNFRQTS	**VNF Validation** Ability for VNF use/respond to Cloud Platform interfaces and APIs	**VNF Performance** Performance tests for VNF running against a given CNTT Infrastructure profile
Infrastructure OPNFV VERIFIED 2018.09	**NFVI Compliance** Check against existing requirements/Specs ➢ CNTT Ref A	**NFVI Validation** Ability for NFVI use/respond to Cloud Platform interfaces and APIs	**NFVI Performance** Performance tests for NFVI implementing a given CNTT profile

Areas of Impact of CNTT

Example Tests

- VNF Interoperability Testing
- Security Testing
- Scalability Testing
- Fault Recovery Testing
- VNF Coexistence
- HA Testing

Test Category / Case Gap Summary

Projects Identified

- * Airship Installer
- Barometer
- Bottlenecks
- Doctor
- * Dovetail
- * Fuel
- FuncTest
- * High Availability
- NFVBench
- * Pharos
- SampleVNF
- VSPerf
- Yardstick

- **# Total OPNFV Projects = 31**
- # CNTT-NFVI = 13 (potential value)
- # 5/13 NA for Review – *Already Covered by Yardstick and Functest

Results

- # 6 Projects can be adopted (as is)
- # 4 We can add/augment TCs for gaps
- # 3 Create projects for new testing

MF

Assessment Strategy

- Select Project by activity, use, and maturity state
- Compare Against Test Categories
- Identify Gaps
- Form Professional Opinion – e.g. augment, adopt
- Solicit Strategic Partner Contributions

Test Categories

- **(Hardware Validations)** BareMetal – HW & O/S validations
- **(Component Validation and VNF Validation Config Only)** VNF Interoperability – validations
- **(Platform Stability)** Compute Component – validations
- **(Platform Resiliency)** Control Plane Component – validations

Next Steps

- **Review Projects** Identified during Antwerp not considered
- **Integrate TCs** from these projects into the delivery stream
- **Discuss augmenting existing test project**s, or create new
- **Onboard Spirent Test Case contributions** – **Where?**
- **Finalize** Test-/Use- Case Needs

Test Category / Case Gap Summary.. Continued

Project	Purpose	Recommendation
FuncTest	▪ Functional interoperability validations	✓ **ADOPT,** as an RI suite. Covers 2k+ Openstack Interoperability Validations ✓ **Augment** to include Baremetal testing for Manifest Validations
Yardstick	▪ VNF/Payload performance validations	✓ **ADOPT,** 62 TCs, leverages Shaker and YAML for test-case development ✓ **Augment** to perform POD restarts and HA for Maria/Ceph restarts
VSPerf	▪ vSwitch perf testing	✓ **ADOPT,** for OVS-DPDK validations with 32 perf and functional TCs ✓ **Setup** external packet generator to avoid latency caused by the tool.
DoveTail	▪ Automation framework	✓ **ADOPT,** with large number of test cases for conformance evaluation
Barometer	▪ Platform availability and NW usage validations	✓ **ADOPT,** for use of NFVI+VNF validations capturing Telemetry data ✓ **Augment** to include device specific resiliency testing and monitoring.
"**NEW**"	▪ **Baremetal Validations**	✓ **CREATE New** Baremetal Validations to verify engineering packages
"**Augment**"	▪ **Spirent Validations**	✓ **Augment** projects with 240 TC adds for load, scaling, cloud migration.
"**NEW**"	▪ **Chaos Toolkit**	✓ **CREATE New**, project to test POD resiliency by injecting chaos (failover)
Bottlenecks	• Stress Testing	✗ **Not recommended** with limited test sets and results categorization
Doctor	• Computer NFVI Fault Mgmt validations	✗ **Not recommended** with limited/no coverage for SDN, KVM, or containers
XTesting	• CICD tool chaining in CNTT validations.	? **REQUIRES POC** if CNTT NFVI requires ADOPTION and USE of tool chaining. ? **There are no specific TCs**, as XTesting is for chaining together CICD test projects, and not for NFVI validation.
NFVBench	• NFVI Perf Measurements (at physical hardware/host level)	✓ **ADOPT,** as a complement to vsperf and yardstick ✓ **Augment** to expand SRIOV and/or OVS-DPDK test cases.