

# Current Mode OPNFV/CNTT Gaps/Issues/Scope Definition

The CNTT and OPNFV communities agree to follow the premises below and subscribe to the associated RACI chart for scope boundaries:

1. CNTT defines specifications and provides only the specifications to OPNFV
  - CNTT: Reference Model, Reference Architecture, Reference Implementation
  - OPNFV: Reference Implementation, Reference Compliance
  - OVP: Testing, Badging
2. OPNFV performs builds from the Reference Implementation specifications.
3. "Cookbook" artifacts are the domain of OPNFV, CNTT should participate in the management of the cookbooks, and selects the Test cases to be executed.
4. A release management method including feedback loops is needed between OPNFV and CNTT. LFN/OPNFV has an existing meeting in which some members of the CNTT community will participate.
  - CNTT Participants that will participate include [Scot Steele](#), etc
5. Cross Community Collaboration must increase without overloading the participants.
  - Recommendation: WS Leads should identify WS participants that can attend the other community TSC meetings
  - Regularly held cross community meetings should be held. Community Chairs should schedule/delegate to community participants to implement/manage

## General group agreement

- The RACI doc and other docs related to the organization are all living docs.
- Long term there will need to make a decision related to the final home for CNTT project work. – [Vincent Danno](#)
- We need to see how successful "1" arrangement is. no need to rush in thinking about "2" and we still didn't figure out "1". **I STRONGLY SUGGEST NOT TO RUSH INTO THIS WITHOUT FIGURE OUT "1" AND PROVE MODEL WORKS.** [Rabi Abdel](#)

## Questions need to be answered: [Current Mode OPNFV/CNTT RACI Chart](#)

- [Cedric Ollivier](#) *How can it help the short term issues? an implementation is needed for the conformance on the VNF side and we need additional relevant resources in both sides: specifications (CNTT) and OPNFV (OPNFV)*
- [Cedric Ollivier](#) *How would be leverage on it in a long term view? We shouldn't reinvent the wheel every release - the current model works and the separation between CNTT and OPNFV is clear for a few main CNTT contributors*
- What happens when CNTT develops a requirement in, say, RI-1?
  - does CNTT goes directly to prospective OPNFV project and get it implemented (via that project PTL).
    - how will the implementation link back to CNTT releases?
    - [Cedric Ollivier](#) : *CNTT RC lists the deliverables (containers/tags/testcases) from OPNFV. What's the additional point here? Could we detail the gap targeted?*
    - Will each project have their own release cycle (with appropriate baggings) or will OPNFV have an overall release cadence ? how does the model work?
    - [Cedric Ollivier](#) *I think the model works already.*
      - *CNTT is in charge of test case description/selection and playbooks (see RC) and then CNTT selects the test cases from OPNFV according to them.*
      - *OVP should implement the test result verification from the archive produced by CNTT playbook - Compared to first OVP model the test case selection is now in RC*
  - or will CNTT has to follow some kind of process to get their requirements implemented?
    - [Cedric Ollivier](#) : *we may consider the classical opensource model here and then the answer could be straightforward: by putting developers*
    - [Cedric Ollivier](#) : *opensource cannot follow a subcontractor model by design*
  - or will OPNFV itself regularly get CNTT releases and find the right OPNFV projects to implement any new requirements.
    - *via CIRV? Cedric Ollivier: CIRV is mostly an umbrella for incubation projects. It should be done at the project level possibly between CNTT RI/RC leaders and the PTLs/core developers*
  - what is the role of OPNFV TSC on all this? who is accountable for what?
  - we need to talk about accountability and make it clear who is accountable for what!
- How does OVP fit into this matrix?
- Given the GSMA process is it mostly a consumer and ratifier of the output from CNTT, or does it have a role in providing feedback? I.e. how can this be set up to be a two way street?

## [Scott Steinbrueck](#) proposal 6/9/2020

The challenge is how to organize the content in an efficient way that reduces complexity for the authors/implementors to create, while also producing a quality product that is simple for the end user base to navigate, read and comprehend.

Seeking agreement to the following premise:

- RA-1 is the **requirements and specs**
- RI-1 is an **example implementation** ("example" word is used because no company is required to follow this exact lab setup / installer approach – it's an example)

[Cedric Ollivier](#)

Does it mean that RI-1 would be strictly removed from CNTT repository?

Yes why not considering Airship as an implementation amongst others which tries to implement the CNTT requirements (RC is currently rather verified thanks to Functest SUT and Field trials).

I would agree to fully remove RI1 from CNTT but CNTT needs a reference implementation somewhere which passes RC successfully for the next steps (conformance of VNF/CNF).

**By the way, it's becoming urgent to take CNTT results into account to find the best path (RI1 doesn't pass RC - we can't leverage on the existing resources allocated to Functest for the VNF conformance)**

**If FMO targets to improve the process, having an implementation fulfilling CNTT requirements is a very short term key issue**

- RC-1 is a test **conformance suite**

Scott Steinbrueck proposal 6/16/2020

## Big Picture

Cedric Ollivier

Sorry I'm lost here. It's introduced as an executive summary but I would rather consider that it increases so much the complexity both regarding the collaborative work and the end users.

If we target an executive summary it could rather be a minor change in the top CNTT document listing (it could be a new Executive summary bullet in between Governance and Technical specifications)

- RA1 requirements
- additional relevant requirement details (e.g. RA1 chapter 5)
- the related mandatory test cases (please see 3.4 Test Cases Traceability to Requirements)
- RC playbook (please see 4.3 NFVI Testing Cookbook)

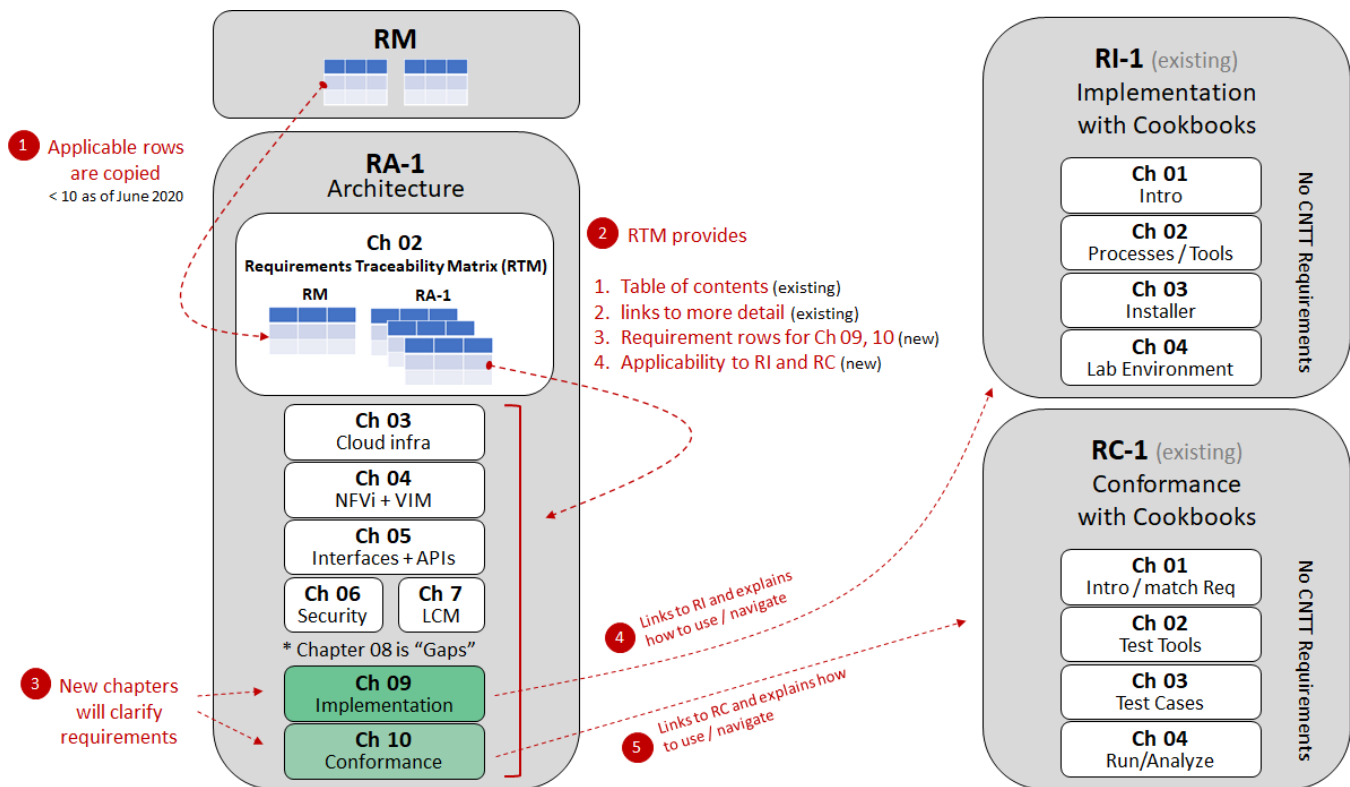
RA1 would decide for everything which is not right regarding the current work and false from a collaborative mindset.

I'm lost by considering RI1 as an example. Why should we increase the complexity of RA1 in that case?

If the overall work is about the separation between CNTT and OPNVF, everyone should know that RC1 Baldy and older is only driven by the CNTT part:

- testcase descriptions, selection and integration are part of CNTT RC
- OPNVF implements the test cases and offers the deliverables (e.g. containers) as asked by CNTT (see first bullet)

**In my view, Ch 10 is just complexifying the current model which should firstly be demonstrated as false to avoid any sophism as it seems here.**



## Proposal Summary

### Introduction

CNTT was formed in order to create a single spec for Openstack (1-stream) and a single spec for Kubernetes (2-stream). The audience is the end-user companies that will become conformant to these specs. CNTT needs to ensure that the end-user base can easily navigate and consume the documentation.

### Problem statement

In the current CNTT documentation format, CNTT's end user base will be challenged to figure out what they need to follow in order to be conformant to CNTT requirements and specs – should they follow what's in RM or RA-1 or both ? Maybe go look in some chapters of RI-1 or RC-1. It's a bit all over the place right now, hard to navigate.

### Solution statement

Let's make it easier for the end user to **navigate** !

Let's make it easier for the end user to **understand what they need to do** !

- **RA-1**: Consolidate requirements into the RA-1 as a single document; any other documentation can reference this content as a single source. Some existing RI-1 and RC-1 content should be moved into RA-1; remaining RI-1 and RC-1 content may remain or be removed altogether.
- **RI-1** only describes an example implementation of RA-1.
- **RC-1** only describes a test suite that is capable of verifying and validating the requirements in RA-1.

Document	Owner	What
----------	-------	------

RA-1	CNTT	<ul style="list-style-type: none"> <li>Ch 02 is "Requirements Traceability Matrix" (proposal already in progress)</li> <li>(new) Ch 09 – Implementation Requirements</li> <li>(new) Ch 10 – Conformance Requirements</li> </ul>
RI-1	OPNFV	<ul style="list-style-type: none"> <li>("Cookbooks") Describes lab environment, installer, tools, and any other process</li> <li>No CNTT requirements</li> </ul>
RC-1	OPNFV	<ul style="list-style-type: none"> <li>("Cookbooks") Describes test tools, test cases created, running the tests, analyzing results.</li> <li>Describe how the created test cases match to CNTT RA-1 Ch 10/02</li> <li>No CNTT requirements</li> </ul>

## Deeper Detail of the Proposal

### RA-1

What will the enhanced RA-1 look like ?

- **Chapter 02**
  - Section 2.2 (stays "Reference Model requirements")
    - Need to include relevant RM requirements here going forward. This will be a copy/paste, but there are < 10 and easy to manage since RM really doesn't change that much. This is one big step in consolidating requirements.
    - Strict approval process for those requirements can enforce aligning with RM requirements.
  - Section 2.3 (rename as "Architecture and **ecosystem Requirements**")
    - Requirements will be only "must" (non-"must" will be moved to 2.4)
    - **Add a column** "RI-1 Applicability" to existing tables, indicating "yes" or "no" if applicable to the Implementation
    - **Add a column** "RC-1 Applicability" to existing tables, indicating "yes" or "no" if applicable to the Conformance
    - **Add a column** "RC-1 Category" to existing tables, indicating type of test in the Conformance
    - **Note:** The existing tables in this section serve as a "requirements Table of Contents", with a column "Traceability" that links to more detailed information about that specific requirement. Maybe this column should be renamed to "Detailed information" or equivalent in order to avoid confusion.
    - Consider adding reference to OpenSource "project/test script" that covers this requirement for completeness (requirements fulfillment with project/test name, etc)
  - Section 2.4 (rename as "Architecture and **ecosystem Recommendations**")
    - Requirements will be only "may" or "should" (this section will remain "Recommendations" as the title states)
    - **Note:** The existing tables in this section serve as a "requirements Table of Contents", with a column "Traceability" that links to more detailed information about that specific requirement. Maybe this column should be renamed to "Detailed information" or equivalent in order to avoid confusion.
  - Link to [RA-1 Ch 02](#)
  - Link to current [RTM draft proposal](#)
- **Chapter 09** will be a new chapter for "Implementation Requirements"
  - **Introduction**
    - Describe implementation intent, reference to RI-1 and its purpose
  - **Requirements**
    - Explain and refer to Traceability Matrix in Ch 02
    - Implementation specifications
      - Include OSTK setup / parameter / configuration details (references to RA-1 chapters as needed)
      - Include min hardware specs needed, software versions
  - **Implementation Examples**
    - hardware configurations, SDN, no SDN
  - **Installer Selection**
    - state that this is an end-user choice, provide link to RI-1 for an example
- **Chapter 10** will be a new chapter for "Conformance Specs"
  - **Introduction**
    - Describe conformance intent, reference to RC-1 and its purpose
  - **Requirements**
    - explain and refer to Traceability Matrix in Ch 02
  - **Verification**
    - Make sure the manifests match the RA-1 configurations/specs
  - **NFVi Validation**
    - **OSTK Services.** Make sure the OSTK services are correctly configured
    - **OSTK API.** Make sure the OSTK APIs are matching the correct version / microVersion per RA-1 Ch 02/05
    - **NVFI Validation using Golden VNF.** Make sure the infrastructure conforms to RA-1 Ch 02 requirements.
  - **Badging**
    - Description (not yet clear which group actually documents the requirements, so keep this as a placeholder section)
    - Include links to NFVi and VNF badging details in OVP

### RI-1

What will the RI-1 look like based on proposed changes ?

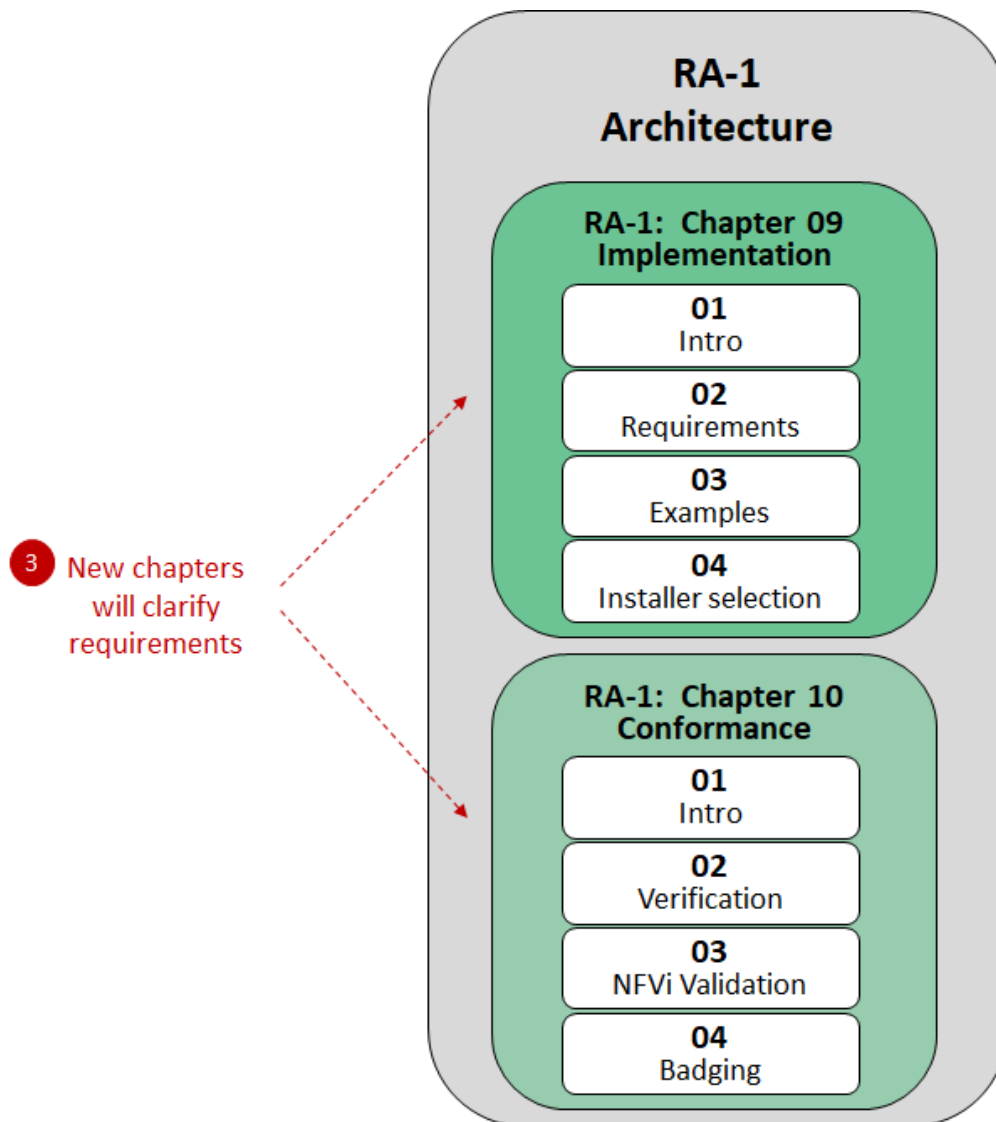
- This document does not contain any CNTT requirements.
- Describes the OPNFV lab environment with relevant links. ("cookbook")
- Describes the OPNFV Installer, usage and relevant links. ("cookbook")
- Describes any other tools used, provide links.
- Suggestion: Lab interaction (login, operational tasks, etc) should be explained elsewhere, provide a link.

## RC-1

What will the RC-1 look like based on proposed changes ?

- This document does not contain any CNTT requirements.
- Describes tools, processes and tests to verify and validate the reference implementation. ("cookbook")
- Describes How to run the tests, analyze results. ("cookbook")
- Provide links to tools and documentation.
- Describes badging process (?)

## More Detailed View of RA-1, Ch 09 and 10



- b. Why
  - c. Who
  - d. Explain how to use and navigate docs
- 2. Requirements
  - a. Explanation of traceability
  - b. link to RA-1 Ch 02
  - c. Implementation Specs
    - i. <make sure that RA-1 Ch 02 has requirements that will link ("Traceability" column) to this section>
    - ii. Number of nodes, sizes, etc.
    - iii. Configuration info
    - iv. Software versions (eg libvirt version)
    - v. <may contain references to other chapters, such as RA-1 Ch 04>
- 3. Examples for how to implement
  - a. No SDN
    - i. High transaction volume that needs a network node (eg)
    - ii. Low transaction volume
    - iii. Conformance lab (eg, 1 jump host, 3 controllers, etc)
  - b. Yes SDN
    - i. Production environment
    - ii. Conformance lab
  - c. Specify hardware configurations
    - i. Cores, etc
    - ii. Storage volumes
    - iii. Underlay Network
- 4. Installer Selection
  - a. Explain the RI is provided as a example/guide if needed
  - b. "It is the choice of the end user how to install, what installer, etc.
  - c. As a guide, here is a **link** to available installers as maintained by OPNFV."
- 1. Introduction
  - a. What
  - b. Why
  - c. Who
  - d. Explain how to use and navigate docs
- 2. Verification
  - a. Make sure the manifests match the RA-1 configurations/specs
  - b. Look at an example, eg Profile (link)
  - c. Software versions, etc — this will refer back to RA-1 Ch 09
  - d. This is currently a manual process to "verify accuracy"
- 3. NFVi Validation
  - a. **OSTK Services.** Make sure the OSTK services are correctly configured
    - i. Describe the tests needed
    - ii. Functest implements the tests needed
    - iii. Reference to Current RC Ch 03 (3.3.4)
  - b. **OSTK API.** Make sure the OSTK APIs are matching the correct version / mVersion per RA-1 Ch 02/05
    - i. Describe the tests needed
    - ii. Functest implements the tests needed
    - iii. Reference to Current RC Ch 03 (3.3.4)
  - c. **NVFI Validation using Golden VNF.** Make sure the infrastructure conforms to RA-1 Ch 02 requirements.
    - i. Describe the tests needed
    - ii. Utilizes "Golden VNF" (eg, Clearwater IMS, VyOS vRouter and OpenAirInterface vEPC)
    - iii. Could use this "Golden VNF" to ensure NFVi meets min level of performance specified by RA-1 Ch 02 (TBD, pending Traceability Matrix PR and RM Cleanup PR).
- 4. OPNFV Badging
  - a. Badging for NFVi (link)
  - b. Badging for VNF (link)