

VSP Compliance and Validation Check within SDC (Frankfurt) - Phase 2

Table of Contents:

- [Overview](#)
 - [Abstract](#)
 - [Rationale](#)
 - [Other Options Considered](#)
 - [Code Structure](#)
 - [Back-end Code Impacts](#)
 - [API Layer changes](#)
 - [Lib Layer changes](#)
 - [Internal library dependencies](#)
 - [External library dependencies](#)
 - [Front-end Code Impacts](#)
 - [Proposal](#)
- [E2E Strategy \(integration within SDC\)](#)
- [Impacts & Planning:](#)
 - <https://wiki.lfnetworking.org/display/LN/TOSCA+VNF+Validation+Planning>
 - [References](#)
 - [VTP New API changes for this feature](#)
 - [VSP Compliance Check within SDC - Integration Test Plan](#)
 - [VSP Compliance Check within SDC - Integration Test Result](#)
 - [CSIT Test Result](#)
 - [Demo](#)

Status	Choose One: <div>DRAFT</div>
Submitter	Rabi Abdel (Vodafone)
Contributors	China Mobile – Yan Yang: yangyanyj@chinamobile.com Huawei - Victor Gao: victor.gao@huawei.com Huawei - Kanag Manickam: kanagaraj.manickam@huawei.com Intel – Haibin Huang: haibin.huang@intel.com
Proposed Release	Frankfurt Release
JIRA Ticket(s)	<div>SDC-2482 - Getting issue details... <div>STATUS</div></div>

Overview

This is an expansion to a previous contribution made for Dublin Release ([VSP Compliance Check within SDC \(Dublin\) - Phase 1](#))



VSP Compliance and Verifica...heck – Phase II - v0.2 .pptx

Abstract

This proposal is to add dynamic checks within SDC to invoke VTP to run compliance and validation tests against VSP (VNF) prior accepting the VSP (VNF).

Rationale

Please refer to: [VSP Compliance Check within SDC \(Dublin\) - Phase 1](#)

Other Options Considered

Please refer to: [VSP Compliance Check within SDC \(Dublin\) - Phase 1](#)

Code Structure

ONAP code can be found in: <https://git.onap.org/>

This feature impacts SDC code in two areas: openecomp-be and openecomp-ui. "openecomp-be" provides APIs and functionality to support the SDC user interface. "openecomp-ui" contains the react-based user interface.

Back-end Code Impacts

The back-end code is broken down into four layers - api, backend, lib and tools.

API Layer changes

The "api" layer defines the client facing REST APIs that are consumed by the front end. This feature includes additions to the api for "external" testing interfaces. There are internal validations in SDC to validate CSAR and HEAT content. The classes and packages are named with "external" in the name since this functionality is implemented external to SDC in VTP and OPNFV. These front-end APIs provide an interface to access and perform these tests in components external to SDC.

The code in the API layer is organized by page / function on the user interface. This makes adding new functionality fairly isolated from the existing code. A new package called externaltesting-rest in openecomp-sdc-rest-webapp has been added for this feature. A Java interface which defines the REST endpoints and implementation of that interface were created. The implementation simply delegates to a "lib" layer function to perform the operations.

To enable the new client-facing REST APIs, the beans-services.xml file in openecomp-sdc-rest-webapp/onboarding-rest-war was modified. This modification causes the external testing manager (see lib changes below) to be instantiated during application startup.

Lib Layer changes

The "lib" layer provides the functionality used in api layer. This layer implements REST calls to external testing resources including VTP and OPNFV (when available).

Like the API layer, functionality is broken down into separate sub-projects within the "lib" layer. A new sub-project called openecomp-sdc-externaltesting-lib was created for this feature. Under this project are api classes and implementation classes in the openecomp-sdc-externaltesting-api and openecomp-sdc-externaltesting-impl directories respectively.

Internal library dependencies

This new openecomp-sdc-externaltesting-lib library has dependencies on other internal library level functionality. This library uses both the vendorsoftware product and versioning libraries to retrieve CSAR artifacts in order to extract metadata to enable external testing.

External library dependencies

No additional 3rd party open source libraries were added to implement this feature.

Front-end Code Impacts

To be published...

Proposal

From a high-level perspective, the proposal will allow VSP, when uploaded into SDC, to go through a set of checks to make sure it is compliant to the deployment environment and custom requirements.

Those checks can be done during the VSP validation stage by implementing plug-ins to SDC and introducing both a "certification repository" and a testing framework to be consulted. **Figure 1** shows the proposal in a high-level view.

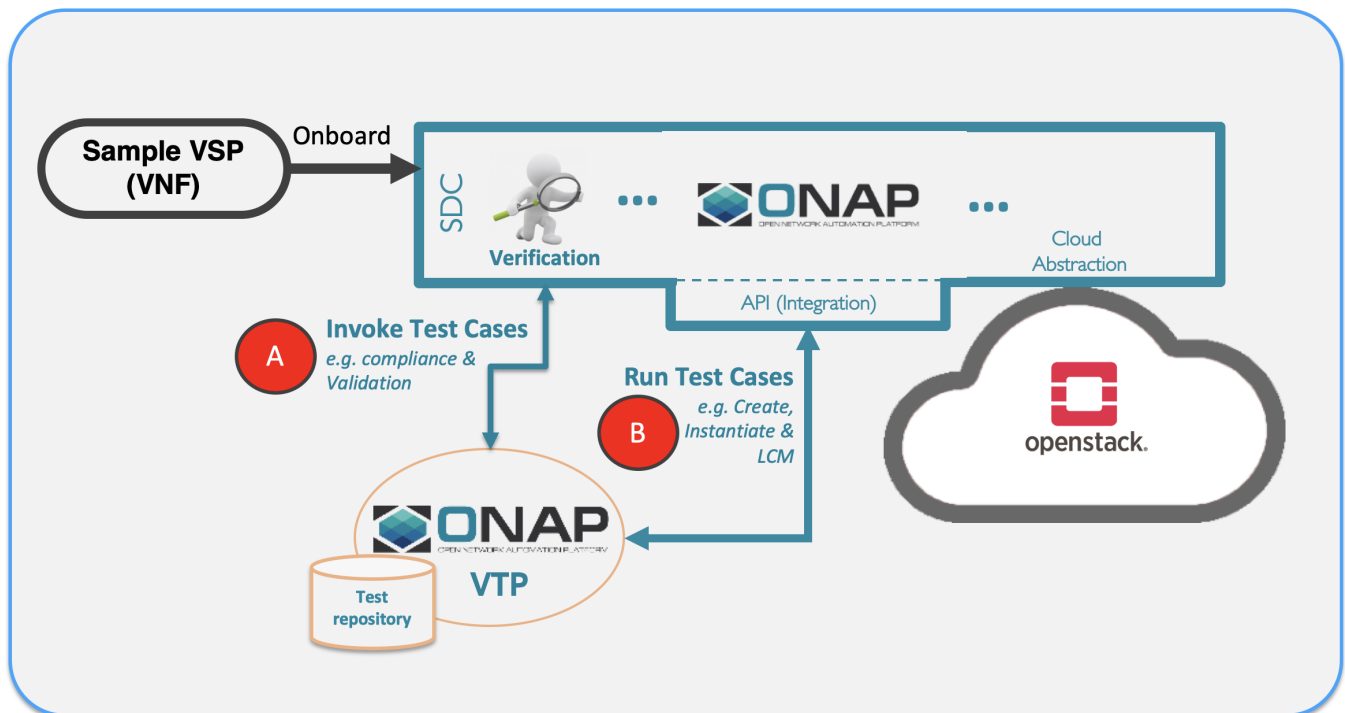
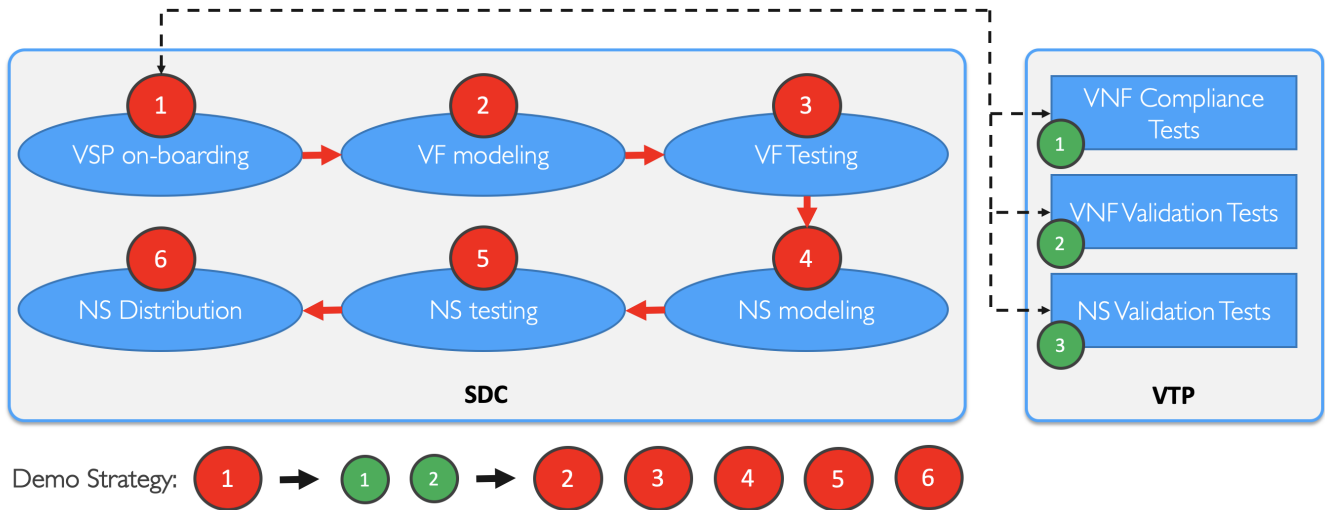


Figure 1: High-Level proposal.

E2E Strategy (integration within SDC)



Impacts & Planning:

<https://wiki.lfnetworking.org/display/LN/TOSCA+VNF+Validation+Planning>

References

VTP New API changes for this feature

[VNF Test Platform \(VTP\)](#)

VSP Compliance Check within SDC - Integration Test Plan

https://wiki.onap.org/download/attachments/45305743/VSP%20Compliance%20Integration%20Test%20v0.9_Sprint5.docx?api=v2

VSP Compliance Check within SDC - Integration Test Result



VSP Compliance and Verific...egration Testing v0.5.docx

CSIT Test Result

Test Case ID	· TEST-CASE-1	
Test Case Name	Validation Result page	
Description	When there is no test case executed then the "Validation Result" page shown with default Message "No Test Performed"	
Release	Frankfurt	
Preconditions	<ul style="list-style-type: none">· Local SDC or ONAP deployment available and running;· A SDC VSP should be created;· VTP Server should be configured and enabled	
Testing Steps	#	Step
	1	Access SDC GUI as a Designer.
	2	Go to the Onboard Workspace.
	3	Click on VSP.
	4	Click on Validation Results
		Expected Result
		SDC Home screen is shown.
		Onboard workspace is shown.
		VSP overview page is shown
		Validation Result Page is shown
Status (Pass/Fail)	Validation Result Page: COMPLETE	
Testing Lab	Local SDC deployment	

Test Case ID	· TEST-CASE-2
---------------------	----------------------

Test Case Name	Execute Validation Test Case with Result																						
Description	Validation test case by loading the vendor provided corresponding CSAR(s) and required input(s) and result displaying in the Validation Result page																						
Release	Frankfurt																						
Preconditions	<ul style="list-style-type: none"> Local SDC or ONAP deployment available and running; A SDC VSP should be created; VTP Server should be configured and enabled 																						
Testing Steps	<table> <tr> <th>#</th><th>Step</th><th>Expected Result</th></tr> <tr> <td>1</td><td>Access SDC GUI as a Designer</td><td> <ul style="list-style-type: none"> SDC Home screen is shown. </td></tr> <tr> <td>2</td><td>Go to the Onboard Workspace</td><td> <ul style="list-style-type: none"> Onboard workspace is shown. </td></tr> <tr> <td>3</td><td>Click on VSP</td><td> <ul style="list-style-type: none"> VSP overview page is shown </td></tr> <tr> <td>4</td><td>Click on Validation</td><td> <ul style="list-style-type: none"> Validation Page is shown with All Tests </td></tr> <tr> <td>5</td><td>Click on vnf-compliance > csar-validate test</td><td> <ul style="list-style-type: none"> Test form is displayed with File type input </td></tr> <tr> <td>6</td><td>Attach a CSAR for testing and submit</td><td> <ul style="list-style-type: none"> After submitting form, Page redirects to Validation Result page with the result and Refresh button </td></tr> </table>		#	Step	Expected Result	1	Access SDC GUI as a Designer	<ul style="list-style-type: none"> SDC Home screen is shown. 	2	Go to the Onboard Workspace	<ul style="list-style-type: none"> Onboard workspace is shown. 	3	Click on VSP	<ul style="list-style-type: none"> VSP overview page is shown 	4	Click on Validation	<ul style="list-style-type: none"> Validation Page is shown with All Tests 	5	Click on vnf-compliance > csar-validate test	<ul style="list-style-type: none"> Test form is displayed with File type input 	6	Attach a CSAR for testing and submit	<ul style="list-style-type: none"> After submitting form, Page redirects to Validation Result page with the result and Refresh button
#	Step	Expected Result																					
1	Access SDC GUI as a Designer	<ul style="list-style-type: none"> SDC Home screen is shown. 																					
2	Go to the Onboard Workspace	<ul style="list-style-type: none"> Onboard workspace is shown. 																					
3	Click on VSP	<ul style="list-style-type: none"> VSP overview page is shown 																					
4	Click on Validation	<ul style="list-style-type: none"> Validation Page is shown with All Tests 																					
5	Click on vnf-compliance > csar-validate test	<ul style="list-style-type: none"> Test form is displayed with File type input 																					
6	Attach a CSAR for testing and submit	<ul style="list-style-type: none"> After submitting form, Page redirects to Validation Result page with the result and Refresh button 																					
Status (Pass /Fail)	Validation Result Page: COMPLETE Used package: vsp-vgw.csar CSIT build: https://jenkins.onap.org/view/sdc/job/sdc-master-csit-sanity-testng/2245/																						
Testing Lab	Local SDC deployment																						

Demo

[VSP Compliance Checks Demo VOICE NARRATEDv2 Tuesday, April 02, 2019 1.39.34 PM.mp4](#)