

VSP Compliance Check within SDC (Dublin) - Phase 1

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](#)
- [Impacts](#)
- [References](#)
 - [VTP New API changes for this feature](#)
- [VSP Compliance Check within SDC - Integration Test Plan](#)
- [Demo](#)

Status	Choose One: <input type="button" value="DRAFT"/>
Submitter	<i>Rabi Abdel (Vodafone)</i>
Contributors	<i>Bryan Whittle (iconectiv)</i> <i>Kanagaraj Manickam (huawei)</i>
Proposed Release	<i>Dublin Release</i>
JIRA Ticket(s)	SDC-1956 - Getting issue details... <input type="button" value="STATUS"/>

Overview



Abstract

This proposal is to add dynamic checks within SDC to (1) Query a certification repository such as OVP to what certification a VNF has undergone and (2) invoke VTP to run custom Ad-hoc testing (checks) against VSP (VNF) prior accepting the VSP (VNF).

Rationale

1. Industry standard certification program repositories such as the OVP portal or third-party repositories can be checked in order to minimize or even remove the need for local testing. The first part of this proposal is to add the support within SDC to query a certification repository.
2. Currently, metadata values passed by the VSP package can contain values that are not compliant with either general ONAP requirements or specific requirements of the CSP installed deployment environment. That can lead to VNF deployment failure. Reducing the risk of such failure can be achieved by checking the metadata passed by the VSP against criteria in VTP or third-party testing frameworks. Examples of metadata passed by the VSP to be checked for compliance against a specific deployment environment are:
 - Compute flavor check as to whether it is supported by underlying NFV Infrastructure.
 - SR-IOV PCIe Pass through to a specific Network Interface Card as to whether that is available in the underlying NFV Infrastructure.

This second part of the proposal will add a capability within SDC to perform custom tests against a given VSP (VNF). This can be done by invoking VTP framework to run that tests and get the result back into SDC,

Other Options Considered

It has been considered that these compliant checks can be carried out offline but this will be restrictive in the following way:

- It is a big overhead to carry out those checks offline especially as VID deployments might vary from one instance of ONAP to another and hence the compliance criteria will change.
- Integrating it into SDC will allow queries to industry standard certification program repositories such as the OVP portal, as well as to third-party repositories, to perform certification checks.
- Integrating it into SDC will allow queries to VTP, as well as to third-party repositories, to perform deployment environment checks.
- Integrating it into SDC will allow for better integration with CI/CD tools and automate the checks, especially when rolling out VNF updates.

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

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 (1) has gone through the certifications process and the result is satisfactory. (2) 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.

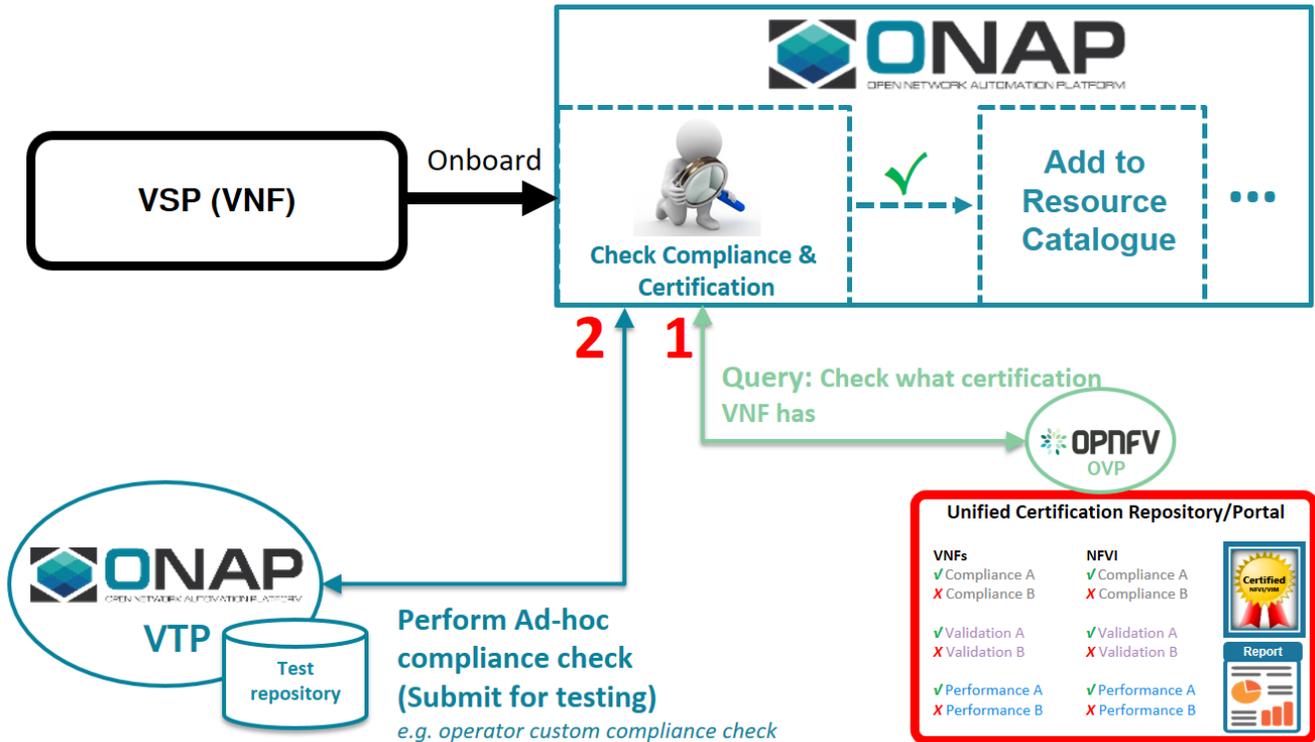


Figure 1: High-Level proposal.

As outlined in the Rationale section above, integrating these checks into SDC will allow queries to VTP and the OVP portal as well as to third-party repositories. Thus the APIs will be designed to interoperate SDC with that range of end-points. In order to meet the Dublin release schedule for the development of plug-ins and APIs, a registry is planned to be implemented as a discrete hosted component outside ONAP. Integration to VTP and OVP portal can be performed when those environments are ready.

Impacts

Project	Usage	Impact
SDC	<ul style="list-style-type: none"> User input for certifications query. User input for compliance checks. Extraction of VNF template attributes and values, both HEAT and TOSCA. Creation and sending queries to registry. Registry response processing. Display of results to user. Logging of query results. User continuation or termination of SDC processing. 	New functionality.

VNF Test Platform (VNFSDK)	<ul style="list-style-type: none"> List the available test cases for a given vnf Retrieve the Inputs and output details of given test case Retrieve the configuration /metadata for a given VNF Execute the given test cases. Retrieve the results of test case execution 	Provide the required NBI for supporting the requirements.
Documentation		<ul style="list-style-type: none"> User Guide https://wiki.onap.org/download/attachments/45305743/VSP%20Compliance%20Guide%20v0.3.docx?version=2&modificationDate=1556541627000&api=v2 Software documentation.

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

Demo

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