PNF package onboarding progress

Ericsson

2019-02-12

ONAP onboarding procedure

onboarding

Onboarding package

Heat template
PNF csar
VNF csar

Create a VSP using onboarding package

VSP

Create a VSP manually via SDC UI

Create a Vendor License Model manually

VLM

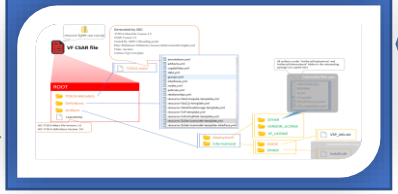
SDC adds additional info, including a link of the vendor license model based on input

Create a resource manually

SDC adds additional metadata including UUID, vendor license model, etc.

Create a resource from VSP

resource(VF)



SDC allows manually uploading any additional artifacts

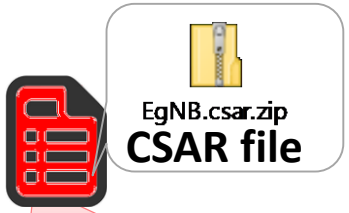
Add resource into service model

Service model

Run time

SDC API

PNF packaging with TOSCA-Metadata proposed for Dublin



EgNB.csar.zip
CSAR file

Note:
- This is an example of the package.
- The files listed in the folder is example only. And not all files are listed here.
- Folder / file name in blue is requested by SOL004.
- Folder / file name in black is example only.

TOSCA-Meta-Version: 1.0
CSAR-Version: 1.1
Created-By: Ericsson (Zu Qiang 2018-12-03)
Entry-Definitions: Definitions/MainServiceTemplate.yaml
Entry-Manifest: MainServiceTemplate.mf
Entry-Change-Log: Artifacts/ChangLog.txt
Entry-Tests: Artifacts/Tests
Entry-Certificate: Artifacts/MainServiceTemplate.cert
Entry-Licenses: Artifacts/License_term.txt

ROOT

- TOSCA-Metadata
  - TOSCA.meta
- Definitions
  - MainServiceTemplate.yaml

NF descriptor

- Artifacts
  - Images
  - Deployment
    - Configuration
    - Yang_module
      - yang-module.yang
    - Measurements
      - pm-dictionary.yaml
    - Events
      - ves-dictionary.yaml
    - …
  - Scripts
    - Install.csh
    - …
  - Tests
  - ChangeLog.txt
  - MainServiceTemplate.cert
  - License_term.txt
- MainServiceTemplate.mf

Not supported by ONAP Casablanca

No ONAP requirement yet

```
metadata:
pnf_product_name: gNB
pnf_provider_id: Ericsson
pnf_package_version:1.0
pnf_release_date_time:2018-12-03T08:44:00-05:00
non_mano_artifact_sets:
onap_ves_events :
source: Artifacts/Deployment/Events/VES_registration.yaml
```

SDC User Story

Priority	User Story	comments	Jiar ticket
1	US 1	Create VSP package from PNF onboarding csar - PNF onboarding csar based on ETSI SOL004 - Create VSP from onboarding csar including meta file, manifest, and PNF descriptor - Move the non MANO artifacts into the right folder - Add a original onboarding csar in VSP artifact folder - Display contents of VSP	SDC-1973 SDC-1974 SDC-1977 SDC-1978 SDC-1979 SDC-2113 SDC-2112
1	US 2	Create PNF internal csar (resource) by import VSP to add PNF - similar procedure as import VSP to create VF for VNF - PNFD mapping: ETSI SOL001 → AID internal modelling - Including the VSP csar artifact including the the original onboarding csar in artifact folder - Display contents	SDC-2108 SDC-1976
2	US 3	Service Handling - Create / distribute Service composing of one or more PNFs / VNFs	SDC-2109
3	US 4	Modify the manually PNF onboard procedure - Adding Meta and Manifest files in generated csar - Add non-MANO artifacts to generated csar	SDC-2110
2	US 5	Support PNF onboarding package security (starting with option 2)	SDC-1980
2	US 6	Create VSP for PNF Manually	SDC-2111
3	US 7	Support SOL004 VNF onboarding package and security	
3	US 8	Investigation on run time microservice impacts	

ONAP PNF onboarding procedure

onboarding

Onboarding PNF package

US1 US5 US7

Create a VSP using onboarding package

VSP

US6

Create a VSP manually via SDC UI

Create a Vendor License Model manually

VLM

SDC adds additional info, including a link of the vendor license model based on input

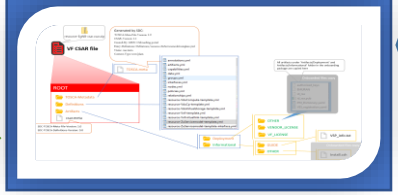
US4

Create a resource manually

SDC adds additional metadata including UUID, vendor license model, etc.

Create a resource from VSP

resource(VF)



SDC allows manually uploading any additional artifacts

US2

Add resource into service model

US3

Service model

US8

Run time

SDC API

SDC Requirements

SDC-1970 - Supporting PNF package onboarding OPEN

- SDC-1973 - Supporting PNF onboarding OPEN
- SDC-1974 - Supporting PNF manifest file OPEN
- SDC-1975 - Design time catalog to associate artifacts with PNF OPEN
- SDC-1976 - Supporting PNFD (SOL001) mapping to AID model OPEN
- SDC-1977 - Removing folder name dependence OPEN
- SDC-1978 - Removing the duplicate descriptor yaml file OPEN
- SDC-1979 - Allowing the dedicated artifact folder with Entry-point in TOSCA.meta OPEN
- SDC-1980 - Supporting packaging security OPEN
- SDC-2072 - Add new artifacts in VF CSAR for PNF on boarding if it is needed OPEN
- SDC-2108 - Import VSP and Create PNF internal csar OPEN
- SDC-2109 - Add manifest file and metadata file with new key words into service csar OPEN
- SDC-2110 - Add PNF manually (without using vsp) OPEN
- SDC-2111 - Manually PNF onboard procedure (create VSP) OPEN
- SDC-2112 - Add a copy of the onboarded package under artifact folder OPEN
- SDC-2113 - copy the on boarding artifacts into the right SDC artifact type OPEN

PNF Package manifest keywords

- Package requirements shall be same as VNF Package, except that the valid name in manifest file shall be:
  - pnf_provider_id
  - pnf_product_name
  - pnf_release_date_time
  - pnf_package_version
- Support Non-mano-artifact-sets in manifest file for both VNF and PNF
- Proposed public non-MANO artifact set identifiers which can be used in a PNF/VNF onboarding package:
  - **onap_ves_events**: contains VES registration files
  - **onap_pm_dictionary**: contains the PM dictionary files
  - **onap_yang_modules**: contains Yang module files for configurations
  - **onap_ansible_playbooks**: contains any ansible_playbooks
  - **onap_others**: contains any other non_MANO artifacts, e.g. informational documents

PNF Artifacts handling

Onboarding PNF package

Onboard based on SOL004 (TOSCA way)

Map the onboarding artifacts into the SDC artifact type

Create a VSP using onboarding package

VSP

Create a VSP manually via SDC UI

Create a Vendor License Model manually

VLM

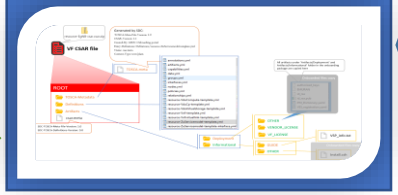
SDC adds additional info, including a link of the vendor license model based on input

Create a resource manually

Create a resource from VSP

SDC adds additional metadata including UUID, vendor license model, etc.

resource(VF)



SDC allows manually uploading any additional artifacts

Add resource into service model

Service model

Retrieve an artifact based the SDC artifact type

Run time

SDC API

VNFSDK

TASK	VNF SDK S/W FUNCTION - DESCRIPTION	Release Priority
#1: MANIFEST FILE (VNF SDK) vs FILE CHECK (Test only)	Verifies the MANIFEST file (MainServiceTemplate.mf) and checks that the defined directories of the PNF package against the manifest file. for example the manifest file might say a files should exist: "*Measurements: source: Artifacts/Deployment/Measurements/PM_Dictionary.yaml*", the VNF SDK would check that the file PM_Dictionary.yaml exists in the actual PNF package. **ASSOCIATED DEVELOPMENT**: VNFSDK-340 - Supporting PNF manifest file **OPEN**	R4 HIGH
#2: TOSCA MetaFile LICENSE File Exists Check (VNF SDK) (Test Only)	VNF SDK may/will(?) check a License Check in the PNF package. TOSCA meta file points to a License. Just a check that the file exists ***no content check*** at all. Note: Related requirements standards from ETSI IFA011, SOL004 **ASSOCIATED DEVELOPMENT:** (Already Supported)	R4 HIGH
#3: TOSCA MetaFile CERTIFICATE Check (VNF SDK) (Test Only)	(Test only) CERTIFICATE check. In the PNF package it is expected that there will be MainServiceTemplate.cert. This is mentioned in the TOSCA MetaFile. For example, in the TOSCA MetaFile, it could be mentioned "*Entry-Certificate: Artifacts/resource-gnodeb-template.cert*". And VNF SDK would check to make sure that the resource-gnodeb-template.cert file exists in the mentioned directory, the Artifacts in this case. VNF SDK does not look inside this file. (Needs Investigation) SOL004 has option 1 (signing each artifact individually / individual digest) and option 2 (sign entire package). It would be nice if VNF SDK supported both Option 1 and Option 2. (Needs Investigation) **ASSOCIATED DEVELOPMENT:** VNFSDK-342 - Support packaging security **OPEN**	R4 HIGH
#4: SOL004 PNF TAGS	Check keywords. needs VNF SDK to check the PNF keywords. in the MainServiceTemplate.mf there are new tags, *pnf_product_name* and *pnf_provider_id*, *pnf_package_version*, *pnf_release_date_time* and *non_mano_artifact_sets*; and the NON ETSI MANO artifact tags public tags. These *public tags* are under the "non_mano_artifact_sets". This would be NEW development in VNF SDK. **ASSOCIATED DEVELOPMENT**: VNFSDK-339 - Support PNF CSAR structure based SOL004 **OPEN**	
#5: VALIDATION FOR META DATA CHECK (ETSI SOL004)	Following ETSI SOL004 Validation for Meta-Data file and Manufacturer file, this is the ***TOSCA.meta*** file that is part of the PNF Package. Both VNF SDK implementing only meta-data option, in the package there is a meta file. Check TOSCA.meta, while this file is not mandatory, when it is included that it follows the SOL004 standard (ETSI). We expect that "*TOSCA-Meta-Version*" and "*CSAR-Version*" and "*Created by*" are already supported, and new checks for "*Entry definition, Entry-manifest, Entry-change-log, Entry-tests, Entry-certificates*" would be new VNF SDK development work (needs to be verified). NOTE: SOL004: Option 1 (Supported in R4 Dublin): TOSCA.meta (exists) Meta-directory based, XML based approach. Option 2 (NOT support in R4 Dublin): CSAR without TOSCA.meta. Manifest (.mf) file that has everything (so the TOSCA.meta is redundant). Yaml-based approach. **ASSOCIATED DEVELOPMENT:** VNFSDK-339 - Support PNF CSAR structure based SOL004 **OPEN**	R4 HIGH
#6: PNF DESCRIPTOR	The descriptor. There is validation of the VNFD. PNF Descriptor: TOSCA descriptor, and validate the node type. Validation of TOSCA PNFD. Following TOSCA rules. Components required are there. (NEEDS INVESTIGATION) **ASSOCIATED DEVELOPMENT**: VNFSDK-341 - PNFD validation based on SOL001 **OPEN**	R4 HIGH
#7: PNF PACKAGE TESTING (Test Only)	Enhancement of Package Testing. A item to make sure that integration testing is performed and that VNF-SDK supports the functions as will be described in the Requirements work. Testing the package against the requirements (a user can enter a requirement#) VNF-RQTS project. It would be ideal if the PNF Package used by the VNF-SDK work is shared by the rest of the PNF preonboarding/onboarding development & integration. **ASSOCIATED DEVELOPMENT**: VNFSDK-343 - enhancement of the test on PNF package **OPEN**	R4 HIGH
	LOW PRIORITY / PUSHED TO R5 EL ALTO	
#F1: CREATE PACKAGE FUNCTION FOR PNF	The create package function creates the metadata files, and CSAR files. This needs to be modified to support SOL004. (NEEDS INVESTIGATION) [Low Priority]	R5 EL ALTO LOW PRI
#F2: TOSCA Metafile License Content Check	SDC license model check. Potential ARTIFACTS: Vendor license model & agrement, features. VNF can have >1 features, entitlement pool, license key pools, actual keys. [Low Priority] PUSH TO R5 EL ALTO.	R5 EL ALTO LOW PRI

VNFSDK impacts

- VNFSDK-337: Supporting PNF package onboarding
  - VNFSDK-338: Project scope to include PNF
  - VNFSDK-339: PNF CSAR structure based SOL004
  - VNFSDK-340: PNF manifest file
  - VNFSDK-341: PNFD validation based on SOL001
  - VNFSDK-342: Support packaging security
  - VNFSDK-343: Enhancement of the test on PNF package

VNF requirements impacts

- VNFRQTS-506: Supporting PNF package onboarding
  - VNFRQTS-507: Project scope to include PNF
  - VNFRQTS-508: Add PNFD requirements
  - Section 5.1.6:
    - VNFRQTS-499: PNF onboarding CSAR package structure based SOL004
    - VNFRQTS-497: Adding package security requirements
  - Section 7.2:
    - Clarifications on the documentation requirements
    - VNFRQTS-505: PNF onboarding package artifacts
    - VNFRQTS-498: Adding VES Event Registration requirement to PNF package
    - Clarifications on artifacts structure requirements
    - VNFRQTS-496: supporting Ansible protocol in PNF

ONAP VNF Descrriptor(5.1.9)

R number	Description	Comments
R-35854	The VNF Descriptor (VNFD) provided by VNF vendor **MUST** comply with TOSCA/YAML based Service template for VNF descriptor specified in ETSI NFV-SOL001.	Shall applicable to PNFD
R-65486	The VNFD **MUST** comply with ETSI GS NFV-SOL001 document endorsing the above mentioned NFV Profile and maintaining the gaps with the requirements specified in ETSI GS NFV-IFA011 standard.	Shall applicable to PNFD
R-17852	The VNFD **MAY** include TOSCA/YAML definitions that are not part of NFV Profile. If provided, these definitions MUST comply with TOSCA Simple Profile in YAML v.1.2.	Shall applicable to PNFD
R-46527	A VNFD is a deployment template which describes a VNF in terms of deployment and operational behavior requirements. … including topology, deployment aspect, and VNF lifecycle management (LCM) operations	PNF LCM is not defined yet
R-15837	The major TOSCA Types specified in ETSI NFV-SOL001 standard draft	
R-54356 R-54876	VNF Data Types	CpProtocolData AddressData L2AddressData L3AddressData LocationInformation CivicAddressElement
R-67895	VNF Capability Types	VirtualLinkable
R-95321	VNF Relationship Types	VirtualLinksTo
R-32155	VNF Interface Types	
		PNF Node Types: PNF, PnfExtCp, Cp
		PBF Policy Types

ONAP VNF CSAR Package (5.1.6)

R number	Description	Comments
R-51347	The VNF package **MUST** be arranged as a CSAR archive as specified in TOSCA Simple Profile in YAML 1.2.	Shall applicable to PNF package
R-87234	The VNF package provided by a VNF vendor **MAY** be either with TOSCA-Metadata directory (CSAR Option 1) or without TOSCA-Metadata directory (CSAR Option 2) as specified in ETSI GS NFV-SOL004. On-boarding entity (ONAP SDC) must support both options. **Note:** SDC supports only the CSAR Option 1 in Casablanca. The Option 2 will be considered in future ONAP releases,	Shall applicable to PNF package
R-10087	The VNF package **MUST** contain all standard artifacts as specified in ETSI GS NFV-SOL004 including Manifest file, VNFD (or Main TOSCA/YAML based Service Template) and other optional artifacts. CSAR Manifest file as per SOL004 - for example ROOT\ **MainServiceTemplate.mf**	Shall applicable to PNF package
R-01123	The VNF package Manifest file **MUST** contain: VNF package meta-data, a list of all artifacts (both internal and external) entry's including their respected URI's, an algorithm to calculate a digest and a digest result calculated on the content of each artifacts, as specified in ETSI GS NFV-SOL004. The VNF Package MUST include VNF Identification Data to uniquely identify the resource for a given VNF provider. The identification data must include: an identifier for the VNF, the name of the VNF as was given by the VNF provider, VNF description, VNF provider, and version.	Shall applicable to PNF package With new valid names/values - pnf_provider_id - pnf_product_name - pnf_release_date_time - pnf_package_version
R-21322	The VNF provider **MUST** provide their testing scripts to support testing as specified in ETSI NFV-SOL004 - Testing directory in CSAR	Should applicable to PNF package
R-26885	The VNF provider **MUST** provide the binaries and images needed to instantiate the VNF (VNF and VNFC images) either as: • Local artifact in CSAR: ROOT\Artifacts\ **VNF_Image.bin** • externally referred (by URI) artifact in Manifest file (also may be referred by VNF Descriptor) Note: Currently, ONAP doesn't have the capability of Image management, we upload the image into VIM/VNFM manually.	May applicable to PNF package Not supported with current release
R-40820	The VNF provider MUST enumerate all of the open source licenses their VNF(s) incorporate. CSAR License directory as per ETSI SOL004. for example ROOT\Licenses\ **License_term.txt**	May applicable to PNF package
R-xxxxx	**VNF Package Authenticity**	May applicable to PNF package

PNF on-boarding requirements (7.2)

R number	Description	Comments
R-77707	The **xNF provider MUST** include a Manifest File that contains a list of **all** the components in the **xNF package**	OK. Overlapped with R-10087 in section 5.1.6.3
R-66070	The xNF Package **MUST** include xNF Identification Data to uniquely identify the resource for a given xNF provider. The identification data must include: an identifier for the xNF, the name of the xNF as was given by the xNF provider, xNF description, xNF provider, and version.	Part of the descriptor
R-98617	The xNF provider **MUST** provide information regarding any dependency (e.g., affinity, anti-affinity) with other xNFs and resources.	Part of the descriptor
R-22346	The VNF package MUST provide VES Event Registration for all VES events provided by that xNF.	VES event Registration Should be applicable to PNF
R-89571	The xNF **MUST** support and provide artifacts for configuration management using at least one of the following technologies; a) Netconf/YANG, b) Chef, or c) Ansible.	
R-30278	The xNF provider **MUST** provide a Resource/Device YANG model as a foundation for creating the YANG model for configuration. This will include xNF attributes/parameters and valid values/attributes configurable by policy.	
R-27711	The xNF provider **MUST** provide an XML file that contains a list of xNF error codes, descriptions of the error, and possible causes/corrective action	Not the proposed FM dictionary
R-74763	The xNF provider **MUST** provide an artifact per xNF that contains all of the xNF Event Records supported. The artifact should include reference to the specific release of the xNF Event Stream Common Event Data Model document it is based on. (e.g., VES Event Listener)	VES event Listener
R-35851	The xNF Package **MUST** include xNF topology that describes basic network and application connectivity internal and external to the xNF including Link type, KPIs, Bandwidth, latency, jitter, QoS (if applicable) for each interface.	Part of the descriptor?
R-26881	The xNF provider **MUST** provide the binaries and images needed to instantiate the xNF (xNF and VNFC images).	Not supported by Casablanca
R-96634	The xNF provider **MUST** describe scaling capabilities to manage scaling characteristics of the xNF.	Not supported by PNF
R-04298	The xNF provider **MUST** provide their testing scripts to support testing.	Testing Requirements.
R-58775	The xNF provider **MUST** provide software components that can be packaged with/near the xNF, if needed, to simulate any functions or systems that connect to the xNF system under test. This component is necessary only if the existing testing environment does not have the necessary simulators.	
R-85653	The xNF **MUST** provide metrics (e.g., number of sessions, number of subscribers, number of seats, etc.) to ONAP for tracking every license.	Only if Licensing is needed
R-40827	The xNF provider **MUST** enumerate all of the open source licenses their xNF(s) incorporate.	
R-85991	The xNF provider **MUST** provide a universal license key per xNF to be used as needed by services (i.e., not tied to a VM instance) as the recommended solution. The xNF provider may provide pools of Unique xNF License Keys, where there is a unique key for each xNF instance as an alternate solution. Licensing issues should be resolved without interrupting in-service xNFs.	
R-47849	The xNF provider **MUST** support the metadata about licenses (and their applicable entitlements) as defined in this document for xNF software, and any license keys required to authorize use of the xNF software. This metadata will be used to facilitate onboarding the xNF into the ONAP environment and automating processes for putting the licenses into use and managing the full lifecycle of the licenses.	

Impacts

- Proposed changes to ONAP
  - Update VNFSDK tool to include PNF
    - VNFSDK-337: Supporting PNF package onboarding
  - Update VNF requirement documentation to include PNF package
    - VNFRQTS-506: Supporting PNF package onboarding
  - Update SDC to adopt PNF package requirements
    - SDC-1970: Support PNF package onboarding
- Proposed changes to ETSI NFV
  - Adding PNF package requirements to SOL004
    - Extend SOL004 WI proposed: Specification of PNF Package file structure
    - Adding PNF Package Support in SOL004 CR
  - Update PNFD in SOL001


ONAP
OPEN NETWORK AUTOMATION PLATFORM