Support xNF Software Upgrade in association to schema updates

AJAY MAHIMKAR (mahimkar@research.att.com)
Łukasz Rajewski (Lukasz.Rajewski@orange.com)
CHRIS RAPPOSELLI-MANZO (cr2431@att.com)
Zu Qiang (Zu.Qiang@Ericsson.com)
Executive Summary - A schema update in relation to a xNF software upgrades is a routine for network upgrade to support new xNF features, improve efficiency or increase xNF capacity on the field, and to eliminate bugs. This use case provides to ONAP an advantage in orchestrating and managing the Life Cycle of a Network Services in-line with business and service objectives.

Business Impact - Deployment and orchestration of new services over CNFs, VNFs and PNFs in a model and software driven way simplifies the network management. Enables operators and service providers to manage the Life Cycle of a Network Service. Assuring continuity of operation of services is crucial for production and carrier grade environments. The actualization or upgrades of software and in consequence required changes in the service model is a natural part of service instance life cycle. Without the support of ONAP service update with schema change, service life cycle management by ONAP can be very difficult which can impact the quality and continuity of services.

Business Markets - All operators and service providers that are using ONAP for service and network function Life Cycle Management

Funding/Financial Impacts - Reduction in operations expense from using industry standard Interfaces.

Organization Mgmt, Sales Strategies - There is no additional organizational management or sales strategies for this requirement outside of a service providers "normal" ONAP deployment and its attendant organizational resources from a service provider.
xNF Software Upgrade without xNF artifacts updating in Release F

- VNF in-place software upgrade procedure on a single VNF instance without VNF artifacts updating (schema update)
- PNF in-place software upgrade procedure on a single PNF instance without PNF artifacts updating (schema update)
  - Scenario 1: Using direct Netconf/Yang interface with PNF
  - Scenario 3: Using Ansible protocol with EM
  - Scenario 4: Using Netconf/Yang interface with EM

The descriptor and artifacts information included in the resource and service template are named as schema in this project. The procedure of updating the resource/service template info, i.e. descriptor and artifacts, is named as schema update in this project.
Update one PNF instance without schema update

**Pre-condition**

- Vendor
- Operator

**Design Time**

- SDC catalog
- Upload blueprint scripts
- Add repository folder URL and userID / password
- Resource Instance name M
  - Resource Name A
  - invariantUUID 1
  - UUID 1
  - swVersion 1.0
  - Artifacts
- Service template
  - Name A
  - invariantUUID 1
  - UUID 1
  - Revision 1.0
- Onboard package & Create resource A
- Associate additional artifacts (e.g. CBA) with the resource A
- Create Service A with resource A

**Run Time**

- VID
- SO
- AAI
- SDNC
- PNF
- SFTP

**Service Instantiation (PnP)**

- Service instance object A
  - service-instance-id 1
  - model-invariant-id 1
  - model-version-id 1
  - software-version 1.0

**Selected workflow, Target software version, and PNF instance ID**

- Retrieve image
- Updates
- Software version is updated

**Repeat per SO action**

- Service instance object A
  - service-instance-id 1
  - model-invariant-id 1
  - model-version-id 1
  - software-version 2.0
1. A vendor shall provide
   - a new VNF / PNF package with updated artifacts, and
   - the new VNF/ PNF software image to the operator.

2. At receiving of the new package, the operator shall
   - onboard the new package and create a new resource template or update the existing resource template (PNF / VNF)
   - update the existing service template with the new or updated resource template
   - distribute the updated service template to run time.

3. At run time, the operator shall, based on the updated service template,
   - upgrade a service instance and its resource instances, and
   - update the AAI entry accordingly

Schema upgrade must be handled at service level for both design time and run time

Then ONAP can make use of the new artifacts provided by the vendors, after schema update
Service level LCM operation example with the same PNF resource name

**Vendor**

- SDC
- CDS

**Operator**

**Pre-condition**

<table>
<thead>
<tr>
<th>Vendor</th>
<th>Operator</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDC</td>
<td>CDS</td>
</tr>
</tbody>
</table>

**SDC catalog**

**Design Time**

1. **Step 1:** Provide new image and new package to the operator
   - PNF Package X
     - Descriptor
     - Artifacts
   - Software Image
   - New schema with updated software version and artifacts

2. **Step 2:** Update the resource template using the new onboarding package
   - Onboard package & create resource A
   - Upload blueprint scripts and parameters
   - Associate additional artifacts (e.g. CBA) with the resource A
   - Create Service A with one resource instance
   - Add additional service artifacts (e.g. wf)

3. **Step 3:** Update service template with the new revision of the same resource template
   - Onboard package & update resource A
   - Upload/update blueprint scripts and parameters
   - Update additional artifacts (e.g. CBA) with the resource A
   - Update Service template by replacing the existing resource A with a new revision
   - Update additional service artifacts (e.g. wf)

4. **Step 4:** Select an existing service instance, a target service template revision and a service level upgrade workflow
   - Resource Instance name M
     - Resource Name A
     - invariantUUID 1
     - UUID 1
     - swVersion 1.0
     - Artifacts
   - Service template
     - Name A
     - invariantUUID 1
     - UUID 1
     - Revision 1.0
   - Service instance object A
     - instance-id 1
     - model-invariant-id 1
     - model-version-id 1
     - software-version 1.0

5. **Step 5:** Execute the workflow on the service instance
   - SD workflow execution on service instance
   - Instance ID & actionIdentifier
   - Netconf/Yang
   - Updates
   - Retrieve image

6. **Step 6:** Update the management service instance object
   - PNF-instance-id 1
     - model-invariant-id 1
     - model-version-id 1
     - software-version 2.0
   - Service instance object A
     - instance-id 1
     - model-invariant-id 1
     - model-version-id 1
     - software-version 2.0

**Run Time**

- VID
- SO
- AAI
- SDNC
- PNF SFTP
Onboarding procedure

Onboarding Heat / xNF csar package

Create a VSP manually via SDC UI

Create a VSP from a new onboarding package

Update a VSP from a new onboarding package

Create a resource from VSP

Update a resource from a new version VSP

SDC adding additional metadata automatically, including UUID, vendor license model, etc.

Create a service model by adding resource models

Update a service model with updated resource models

Create a Vendor License Model manually

VSP

VLM

adding additional info automatically, including a link of the vendor license model based on input

Resource (VF)

Run time

Distribution

manually updating a resource

manually creating a resource

updating any descriptor proprieties

updating / adding / removing any artifacts

The Linux Foundation

ONAP
SO Service level upgrade workflow

- **Service Level Preparation**
  - ✓ Creating resource template instance upgrade list by comparing the service templates
  - ✓ Select a resource level health check workflow based on the resource type
  - ✓ Execute the selected resource level health check workflow on all resource instances within the service

- **Service Level Upgrade**
  - ✓ Select a resource level upgrade workflow based on the resource type
  - ✓ Execute the selected resource level upgrade workflow on each upgrading resource instances
  - ✓ Update the software version, model-invariant-id, and model-version-id of the resource template in the A&AI entry at end of each Resource level upgrade workflow

- **Service Level Update**
  - ✓ Update the model-version-id of the service template in the A&AI entry

- **Service Level postCheck**
  - ✓ Select a resource level health check workflow based on the resource type
  - ✓ Execute the selected resource level health check workflow on all resource instances
SO Service level upgrade workflow

Note: the service level workflow is network function type independent. When upgrade one resource instance, the subsequent resource level upgrade workflow is selected based on the network function type.

Example of resource level upgrade workflow

Repeat on each to-be-upgraded resource instance
## API Impacts

### Consumed existing A&AI APIs (No impacts on AAIE-1)

**Retrieve all services instance:**
GET /business/customers/customer/{global-customer-id}/service-subscriptions/service-subscription/{service-type}/service-instances

**Update A&AI with new resource model version:**
PATCH /network/pnfs/pnf/{pnf-name}
PATCH /network/generic-vnfs/generic-vnf/{vnf-id}

**Update A&AI with new service model version:**
PATCH /business/customers/customer/{global-customer-id}/service-subscriptions/service-subscription/{service-type}/service-instances/service-instance/{service-instance-id}

### New SO APIs (SO-E-01 extensions)

**Service model retrieving API**
Get /serviceSpecifications/v1/serviceModels?serviceModelInvariantId={model-invariant-id}

**Service level workflow retrieving API**
Get /workflowSpecifications/v1/workflows?serviceModelVersionId={UUID}

**Service level workflow execution API**
POST /instanceManagement/v1/serviceInstances/{serviceInstanceId}/workflows/{workflow_UUID} ? targetVersion=serviceModelVersionId
## ONAP Impacts

<table>
<thead>
<tr>
<th>ONAP components</th>
<th>Story</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SDC</strong></td>
<td>• PNF software version onboarding (done in R6)</td>
</tr>
<tr>
<td></td>
<td>• Update a resource template from a new onboarding package</td>
</tr>
<tr>
<td><strong>SO</strong></td>
<td>• New building blocks to enable service level LCM operation.</td>
</tr>
<tr>
<td></td>
<td>• Generic service level upgrade workflow</td>
</tr>
<tr>
<td></td>
<td>• Service model retrieving API</td>
</tr>
<tr>
<td></td>
<td>• Service level workflow retrieving API</td>
</tr>
<tr>
<td></td>
<td>• Service level workflow execution API</td>
</tr>
<tr>
<td><strong>Documentation</strong></td>
<td>• Defining the new service level software upgrade procedure</td>
</tr>
<tr>
<td><strong>Integration</strong></td>
<td>• test cases on resource template updating</td>
</tr>
<tr>
<td></td>
<td>• test cases on new SO APIs</td>
</tr>
<tr>
<td></td>
<td>• test cases on service level workflow execution</td>
</tr>
<tr>
<td></td>
<td>• test cases on e2e service level upgrade procedure (with PNF simulator)</td>
</tr>
<tr>
<td><strong>Modeling</strong></td>
<td>No impacts</td>
</tr>
<tr>
<td><strong>New components</strong></td>
<td>Not Applicable</td>
</tr>
</tbody>
</table>
Thank You!