ONAP
OPEN NETWORK AUTOMATION PLATFORM

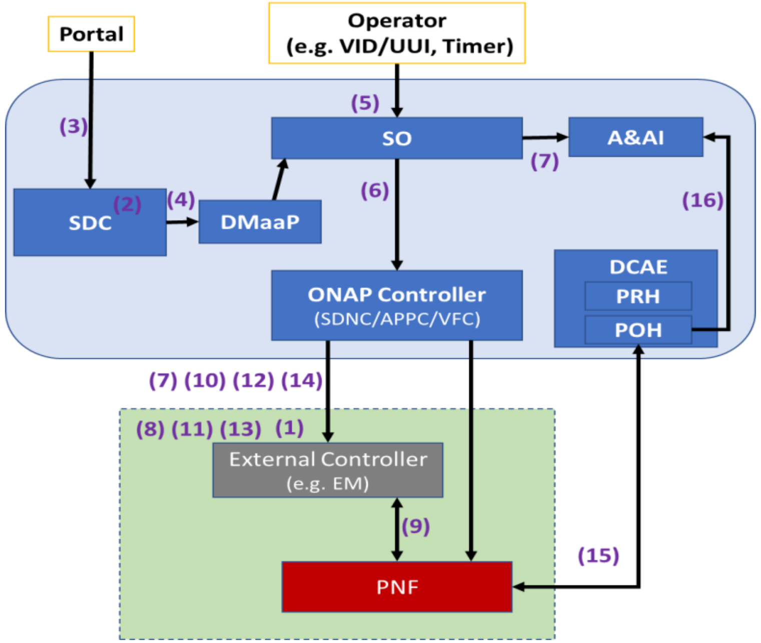
PNF software upgrade for Frankfurt release

Zu Qiang (Ericsson)

Yaoguang Wang (Huawei)

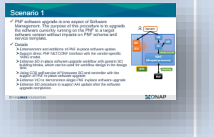
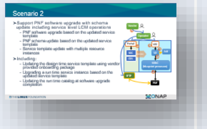
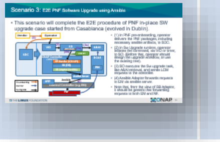
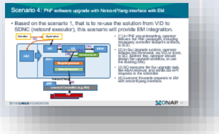
Current Development Status

- ✓VNF in place software upgrade is supported
  - ✓using Ansible and Chef
  - ✓with LCM API
  - ✓‘generic’ SO building blocks
- ✓PNF in place software upgrade is supported in Casablanca and updated in Dublin
  - ➢With the support of an EM
  - ➢Ansible protocol only
  - ➢Plan to use LCM API with existing SO building blocks
  - ➢Impacts on SDNC only (not E2E solution yet)



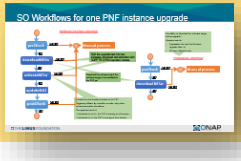
Wiki Home in R6： https://wiki.onap.org/display/DW/PNF+software+upgrade

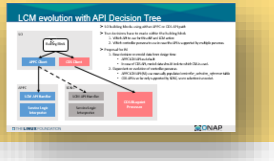
PNF software upgrade scenarios

Scenarios	Descriptions	Service level impacts	PNF software upgrade	Schema updates	Controller API	Protocols	EM	Proposed by	Target releases
1  Scenario 1	•Support direct PNF NETCONF interface with the vendor-specific YANG model. •Enhance SO in-place software upgrade workflow with generic SO building blocks, which can be used for workflow design in the design time. •Using CDS self-service API between SO and controller with the support of PNF in-place software upgrade •Enhance VID to demonstrate single PNF in-place software upgrade •Enhance SO procedure to support AAI update after the software upgrade completion.	No	one PNF instance	No	CDS self-service	Netconf	No	Ericsson	Frankfurt
2  Scenario 2	•Updating the design time service template using vendor provided onboarding package •Upgrading a run time service instance based on the updated service template •Updating the run time catalog at software upgrade completion	Yes	One or more PNF instances	Yes	CDS self-service	Netconf	No	Ericsson	Frankfurt +
3  Scenario 3: E2E PNF Software Upgrade using Ansible	•Enhancement and additions of PNF in-place software update •Using LCM API •Using Ansible protocol •With EM	No	one PNF instance	No	LCM API	Ansible	Yes	Huawei	Frankfurt
4  Scenario 4: PNF software upgrade with Netconf/Yang interface with EM	•NETCONF interface with EM •Using CDS self-service API	No	one PNF instance	No	CDS self-service	Netconf	Yes	Huawei	Frankfurt

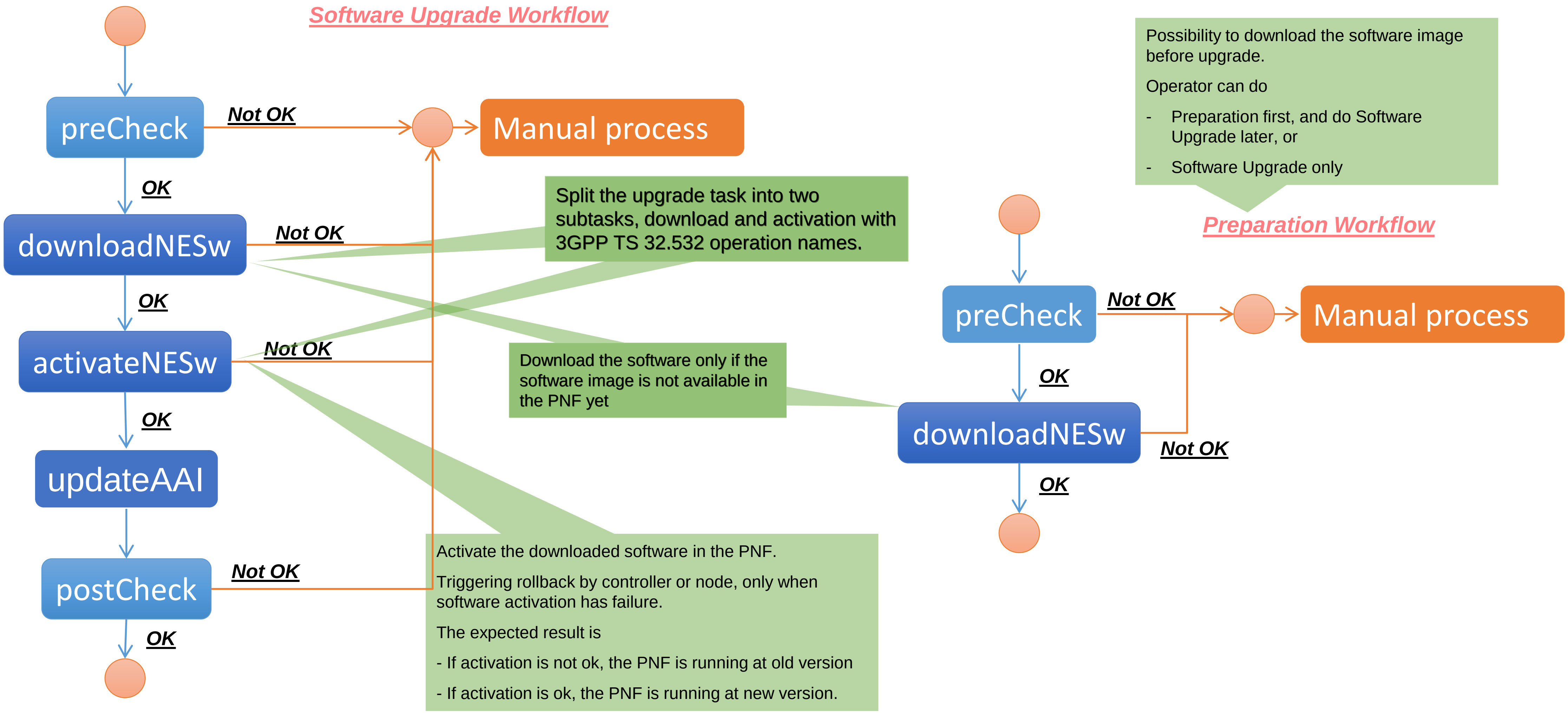
Common tasks for scenario 1, 3 & 4:

- Same SO work flows with generic workflow design and generic SO building blocks
- LCM evolution with API Decision Tree
- Same PNF upgrade UI in VID


SO Workflows for one PNF instance upgrade


LCM evolution with API Decision Tree

SO Workflows for one PNF instance upgrade



Split the upgrade task into two subtasks, download and activation with 3GPP TS 32.532 operation names.

Download the software only if the software image is not available in the PNF yet

Activate the downloaded software in the PNF.

Triggering rollback by controller or node, only when software activation has failure.

The expected result is

- If activation is not ok, the PNF is running at old version

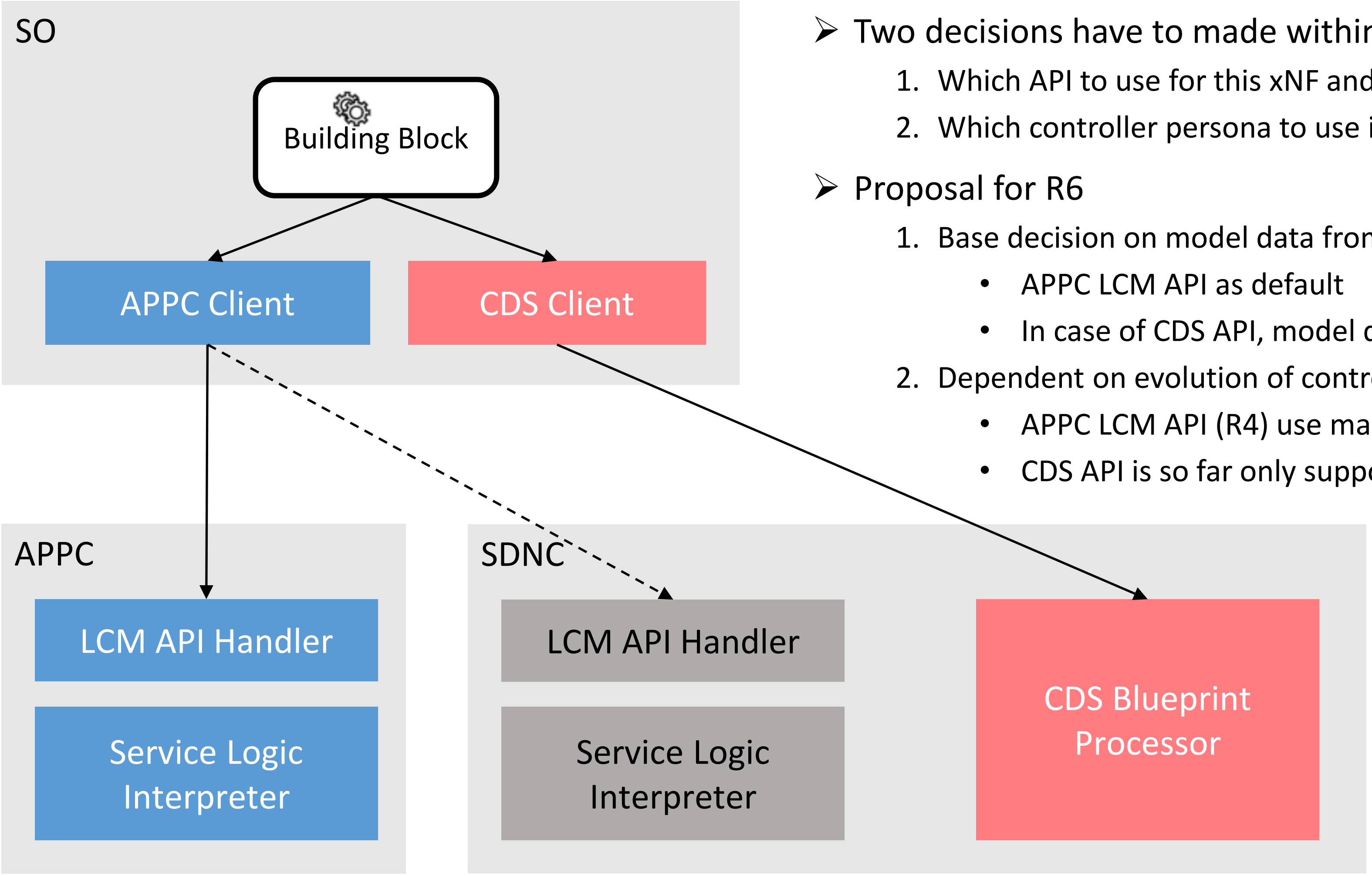
- If activation is ok, the PNF is running at new version.

Possibility to download the software image before upgrade.

Operator can do

- Preparation first, and do Software Upgrade later, or
- Software Upgrade only

LCM evolution with API Decision Tree



- SO building blocks using either APPC or CDS API path
- Two decisions have to made within the bulding block
  1. Which API to use for this xNF and LCM action
  2. Which controller persona to use in case the API is supported by multiple personas
- Proposal for R6
  1. Base decision on model data from design time
     - APPC LCM API as default
     - In case of CDS API, model data should indicte which CBA is used.
  2. Dependent on evolution of controller personas
     - APPC LCM API (R4) use manually populated *controller_selection_reference* table
     - CDS API is so far only supported by SDNC, so no selection is needed.

SDC PNF Model

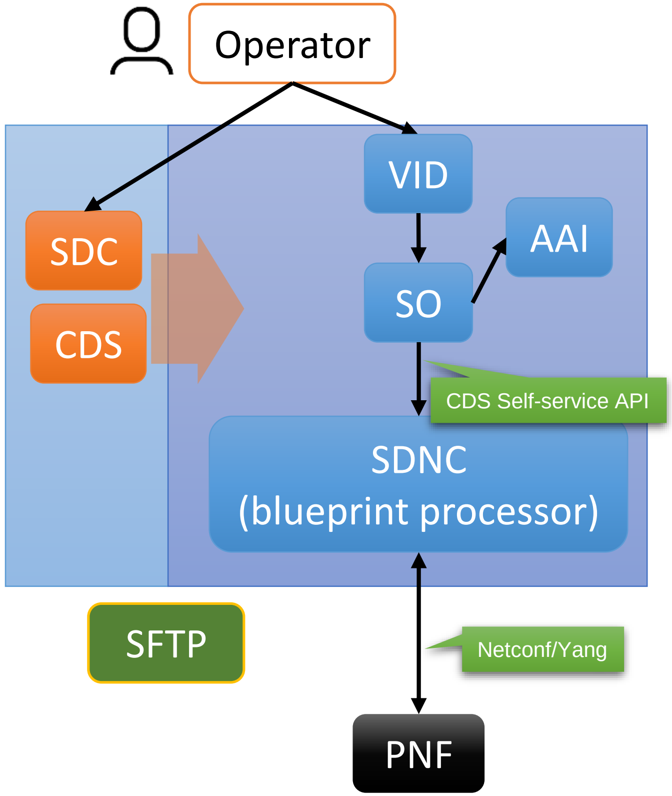
```
tosca_definitions_version: tosca_simple_yaml_1_1
metadata:
  invariantUUID: 47c4a735-1363-49d3-bce8-07f4f776fcf9
  UUID: d1505177-9b4e-49fa-b4b5-dc3d89628a86
  name: ZuPNF2
  description: PNF uploading
  type: PNF
  category: Application L4+
  subcategory: Application Server
  resourceVendor: Ericsson
  resourceVendorRelease: '123'
  resourceVendorModelNumber: ''
imports:
…
topology_template:
  inputs:
    nf_function:
      type: string
      required: false
    software_versions:
      type: list
      required: false
      entry_schema:
        type: string
    nf_role:
      type: string
      required: false
    nf_type:
      type: string
      required: false
    sdnc_model_name:
      type: string
    sdnc_model_version:
      type: string
    sdnc_artifact_name:
     type: string
    skip_post_instantiation_configuration:
      type: boolean
      default: true
      constraints:
      - valid_values: [true,false]
  node_templates:
    pnfExtCp-1:
      type: org.openecomp.resource.cp.v2.extCP
      metadata:
        invariantUUID: 7e82db71-1799-4b64-a029-4816a29741d3
        UUID: 6c84604c-21ed-4369-9948-1cd7f01f6201
        customizationUUID: 9f8729c7-5902-4e5e-8be0-54cadfcc8948
        version: '1.0'
        name: ExtCP2
        description: The AT&T Connection Point base type all other CP derive from
        type: CP
        category: Generic
        subcategory: Network Elements
        resourceVendor: ONAP (Tosca)
        resourceVendorRelease: 1.0.0.wd03
        resourceVendorModelNumber: ''
      properties:
        ip_requirements: [
             IpRequirements:
                ip_version: 4,
                assingment_method: dhcp
           ]
    capabilities:
        network.incoming.packets.rate:
        ....
  substitution_mappings:
    node_type: org.openecomp.resource.pnf.Zupnf2
    capabilities:
      pnfextcp1.forwarder:
      - pnfextcp1
      - forwarder
      ....
```

Added in Dublin for VNF / PNF configuration using CDS self-service API

Proposal: reuse the existing porosities (using the existing porosities' value as API flag) to support PNF upgrade

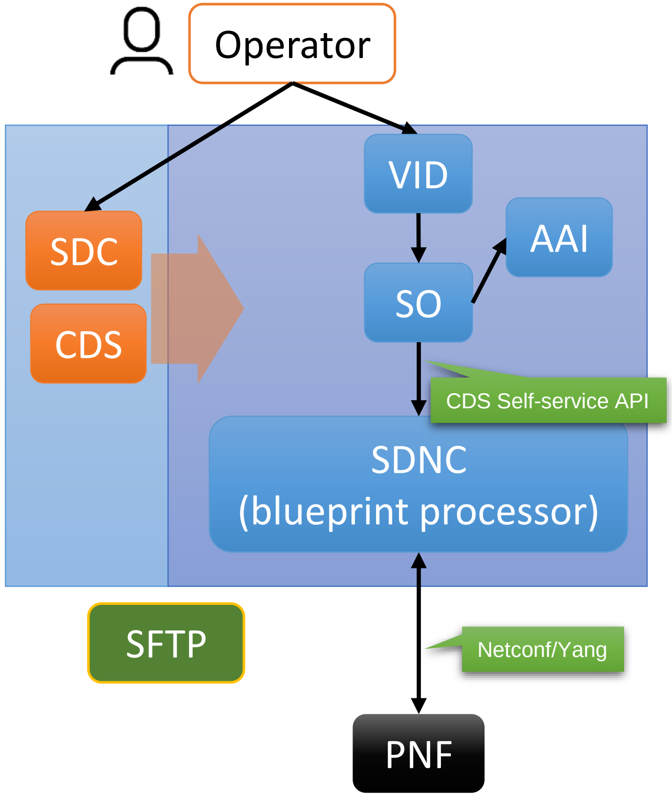
Scenario 1

- ✓ PNF software upgrade is one aspect of Software Management. The purpose of this procedure is to upgrade the software currently running on the PNF to a target software version without impacts on PNF schema and service template.
- ✓ Details
  - ➢ Enhancement and additions of PNF in-place software update.
  - ➢ Support direct PNF NETCONF interface with the vendor-specific YANG model.
  - ➢ Enhance SO in-place software upgrade workflow with generic SO building blocks, which can be used for workflow design in the design time.
  - ➢ Using CDS self-service API between SO and controller with the support of PNF in-place software upgrade
  - ➢ Enhance VID to demonstrate single PNF in-place software upgrade
  - ➢ Enhance SO procedure to support AAI update after the software upgrade completion.



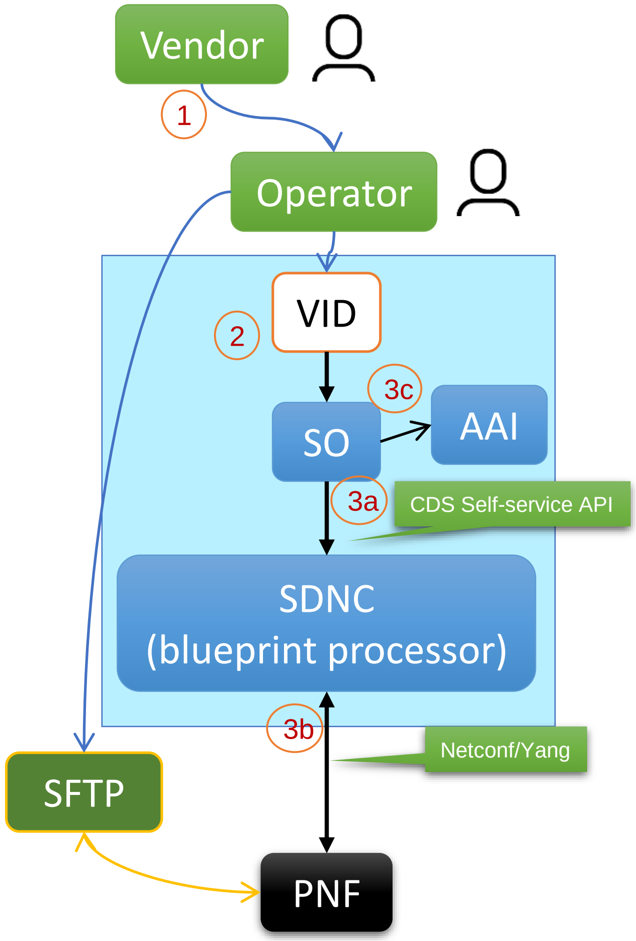
Pre-conditions (Scenario 1)

- ✓ ONAP is ready to use
- ✓ SO upgrade workflows are ready to use
- ✓ A SDC service template with one PNF resource is designed (including CBA association) and it is distributed to run time
- ✓ Service instantiation is completed, including PNF PnP.
  - ➢ A PNF instance is in operation with connectivity between PNF-ONAP, PNF-SFTP

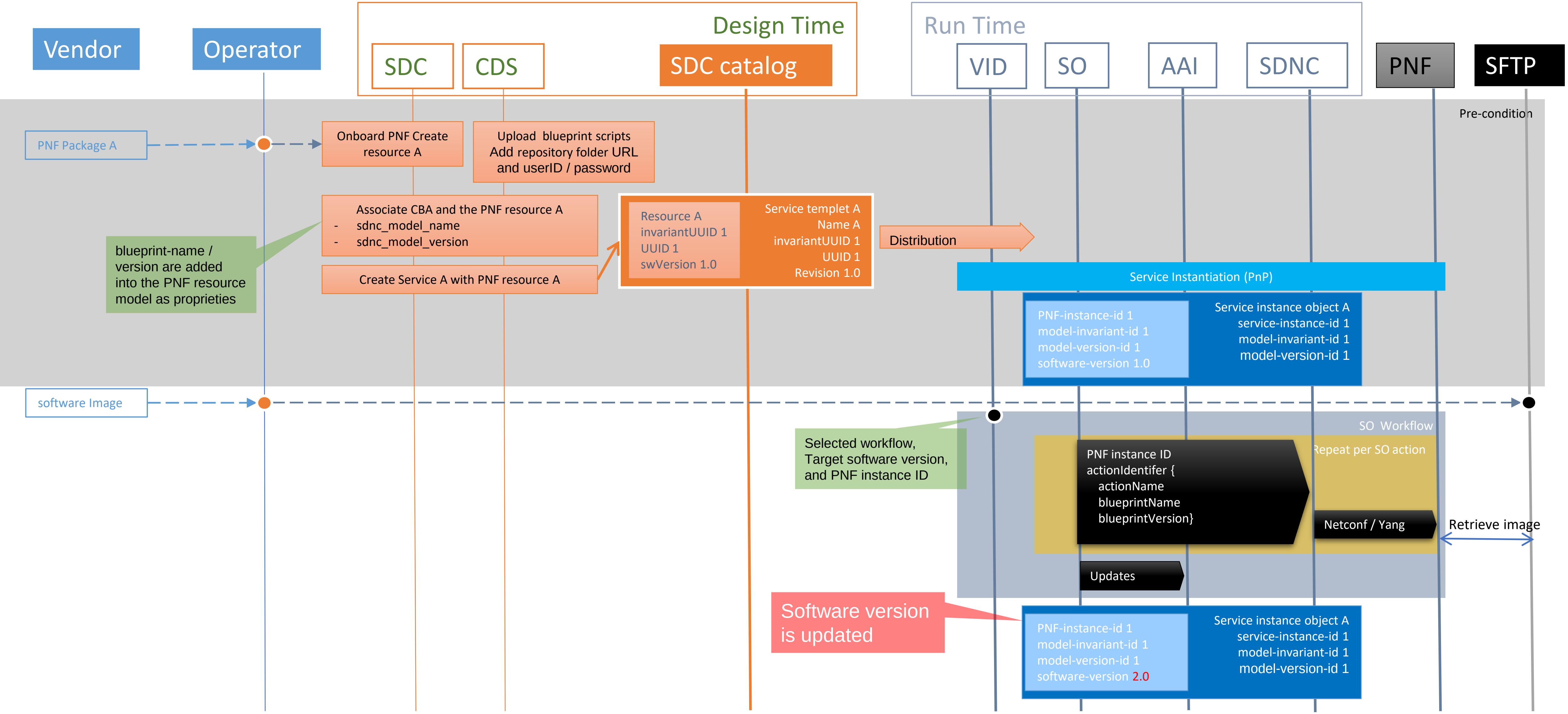


Upgrade one PNF instance (Scenario 1)

1. Vendor delivers the new software image to the operator and stored in the SFTP server
2. At the VID, operator
   - selects a work-flow, and a PNF instance,
   - provides the target software version, and
   - initiates the upgrade procedure
3. SO executes the workflow
   a) SO sends CDS request(s) with action-identifier {actionName, blueprintName, blueprintVersion} to the blueprint processor inside the controller using CDS self-service API
   b) Controller/blueprint processor executes the blueprint scripts including sending Netconf request(s) to the PNF instance
   c) SO updates the A&AI with the active software-version when the upgrade is completed



Upgrade one PNF instance (Scenario 1)



Vendor

Operator

Design Time

SDC

CDS

SDC catalog

Run Time

VID

SO

AAI

SDNC

PNF

SFTP

Pre-condition

PNF Package A

Onboard PNF Create resource A

Upload blueprint scripts
Add repository folder URL and userID / password

Associate CBA and the PNF resource A
- sdnc_model_name
- sdnc_model_version

blueprint-name / version are added into the PNF resource model as proprieties

Create Service A with PNF resource A

Resource A
invariantUUID 1
UUID 1
swVersion 1.0

Service templet A
Name A
invariantUUID 1
UUID 1
Revision 1.0

Distribution

Service Instantiation (PnP)

PNF-instance-id 1
model-invariant-id 1
model-version-id 1
software-version 1.0

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

software Image

SO Workflow

Selected workflow, Target software version, and PNF instance ID

Repeat per SO action

PNF instance ID
actionIdentifer {
  actionName
  blueprintName
  blueprintVersion}

Netconf / Yang

Retrieve image

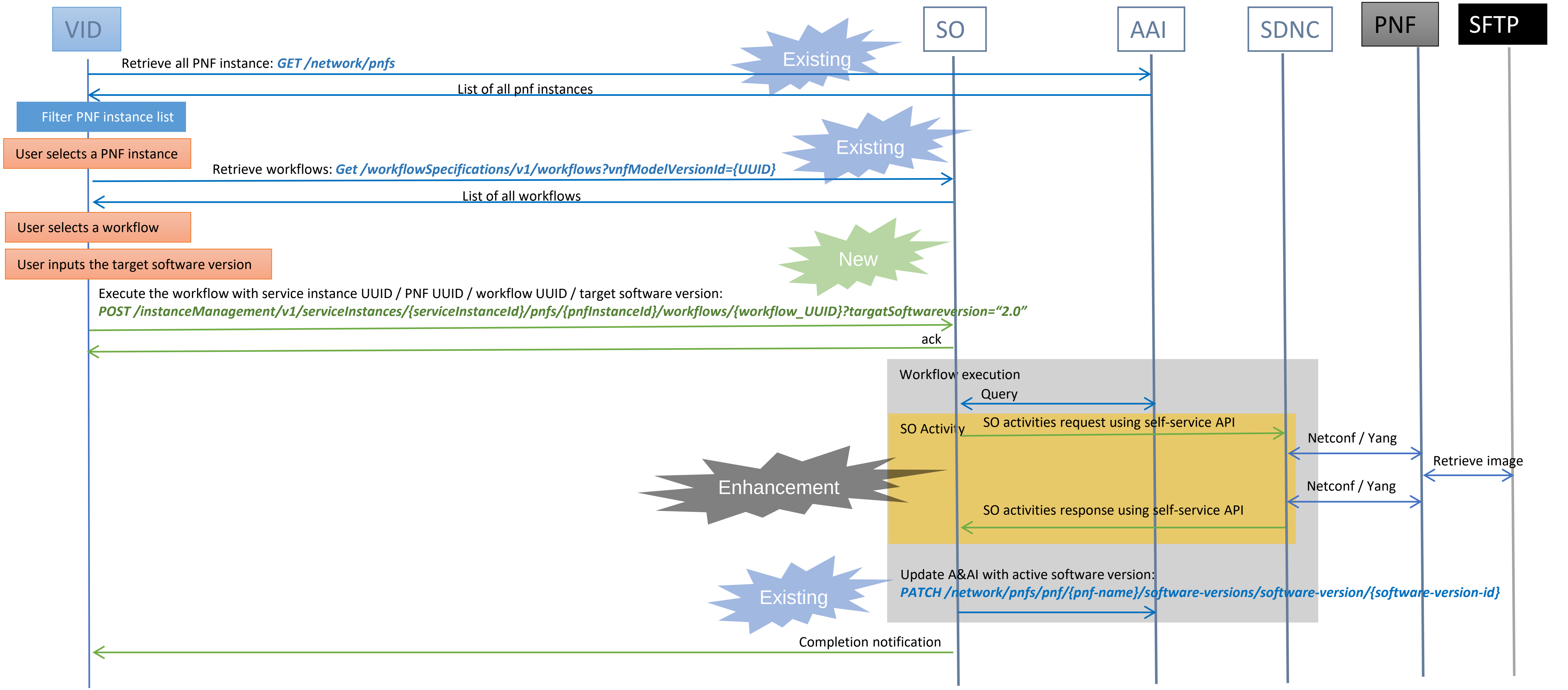
Updates

Software version is updated

PNF-instance-id 1
model-invariant-id 1
model-version-id 1
software-version 2.0

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

API impacts (Scenario 1)



VID

SO

AAI

SDNC

PNF

SFTP

Existing

Retrieve all PNF instance: ***GET /network/pnfs***

List of all pnf instances

Filter PNF instance list

User selects a PNF instance

Existing

Retrieve workflows: ***Get /workflowSpecifications/v1/workflows?vnfModelVersionId={UUID}***

List of all workflows

User selects a workflow

User inputs the target software version

New

Execute the workflow with service instance UUID / PNF UUID / workflow UUID / target software version:

POST /instanceManagement/v1/serviceInstances/{serviceInstanceId}/pnfs/{pnfInstanceId}/workflows/{workflow_UUID}?targatSoftwareversion="2.0"

ack

Workflow execution

Query

SO Activity

SO activities request using self-service API

Netconf / Yang

Retrieve image

Enhancement

Netconf / Yang

SO activities response using self-service API

Existing

Update A&AI with active software version:

PATCH /network/pnfs/pnf/{pnf-name}/software-versions/software-version/{software-version-id}

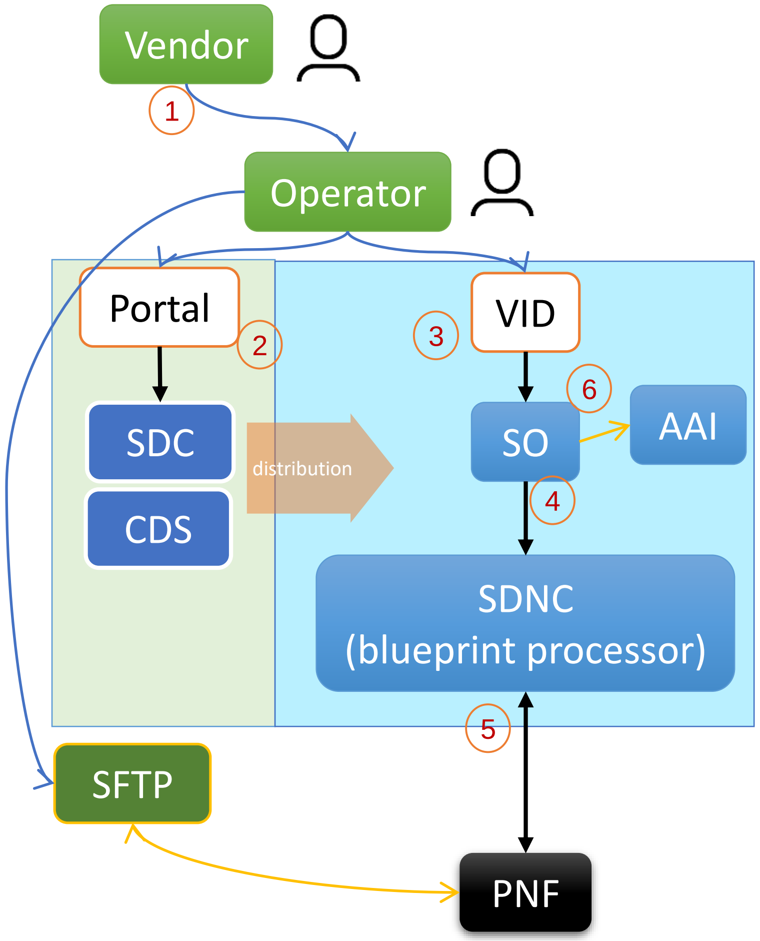
Completion notification

ONAP Impacts (Scenario 1)

Story	components
Support generic workflow design low	SDC/SO
Support creating of SO upgrade workflows for PNF upgrade, including Upgrade and Preparation small	SDC/SO
Create or modify SO activity building block for PNF upgrade, including downloadNESw, activateNESw, updateAAI, preCheck, postCheck	SDC/SO SO
Support PNF upgrade UI	VID VID
Update VID-SO API to execute the workflow of PNF software upgrade with target software version: POST /instanceManagement/v1/serviceInstances/{serviceInstanceId}/pnfs/{pnfInstanceId}/workflows/{workflow_UUID}?targatSoftwareversion=“2.0”	VID/SO SO
PNF and CBA association enhancement to support PNF upgrade	SDC/CDS
Support PNF upgrade with CDS self-service API SDC/CDS	SO/CCSDK
Implement updateAAI activity for A&AI updates with active software-version	SO SO
PNF simulator extension	Integration
integration / testing and demo	Integration
Documentation	VNFRQTS

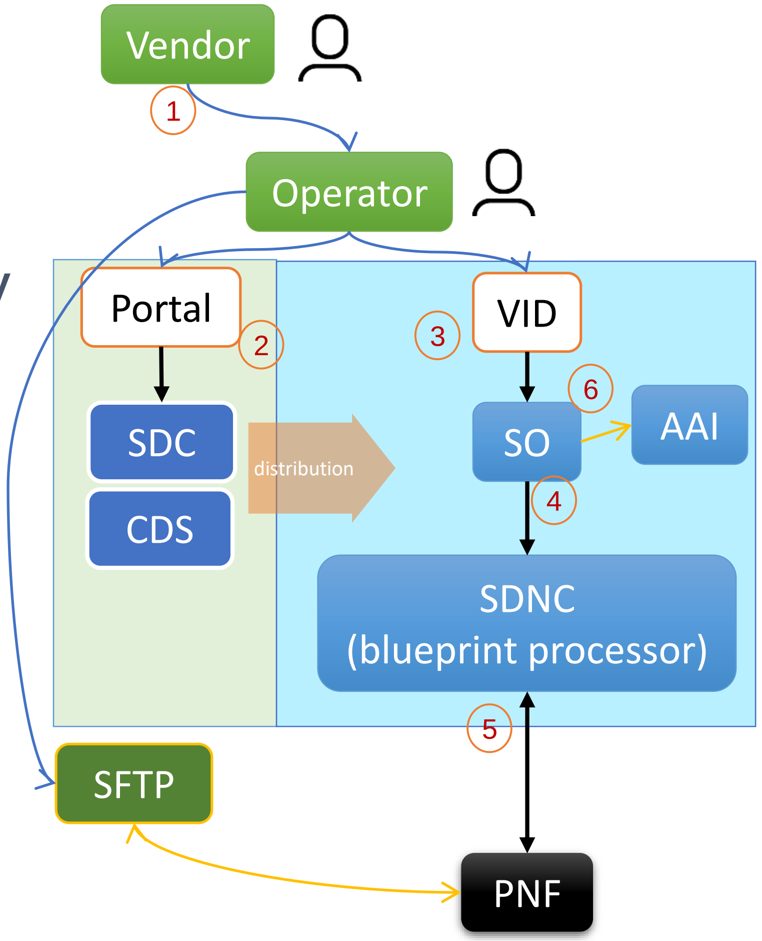
Scenario 2

- ➢Support PNF software upgrade with schema update including service level LCM operations
  - PNF software upgrade based on the updated service template
  - PNF schema update based on the updated service template
  - Service template update with multiple resource instances
- ➢Including:
  - Updating the design time service template using vendor provided onboarding package
  - Upgrading a run time service instance based on the updated service template
  - Updating the run time catalog at software upgrade completion



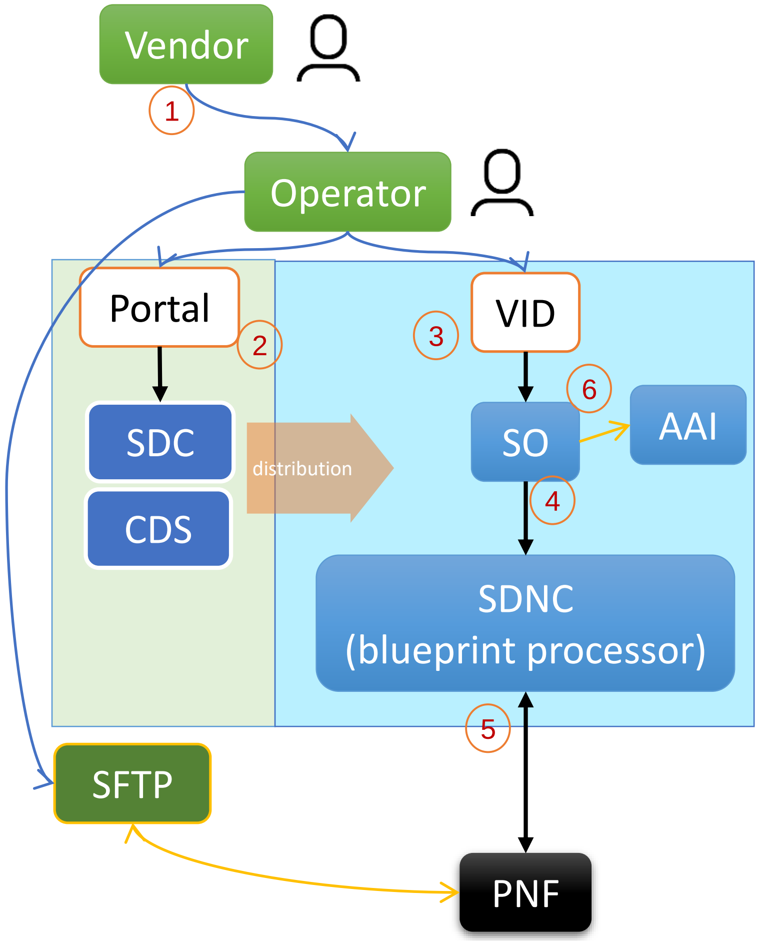
Schema Update

- Vendor delivers
  a) PNF box with pre-installed software, and
  b) PNF package which describes the functions supported by the PNF box
- At PNF software upgrade, a new software version is running in the PNF box. The new software version may introduce new functions. Without schema updates, the ONAP is not aware of any of the new functions, e.g. new PM counters, new CM Yang model, new Alarms, etc.
- In order to update the ONAP with the new functions introduced by the upgraded PNF software,
  - a vendor shall provide a new PNF package with updated artifacts and the new PNF software image to the operator.
  - the operator shall update the resource / service template based on the received PNF onboarding package, and upgrade the PNF software and update the run time catalog



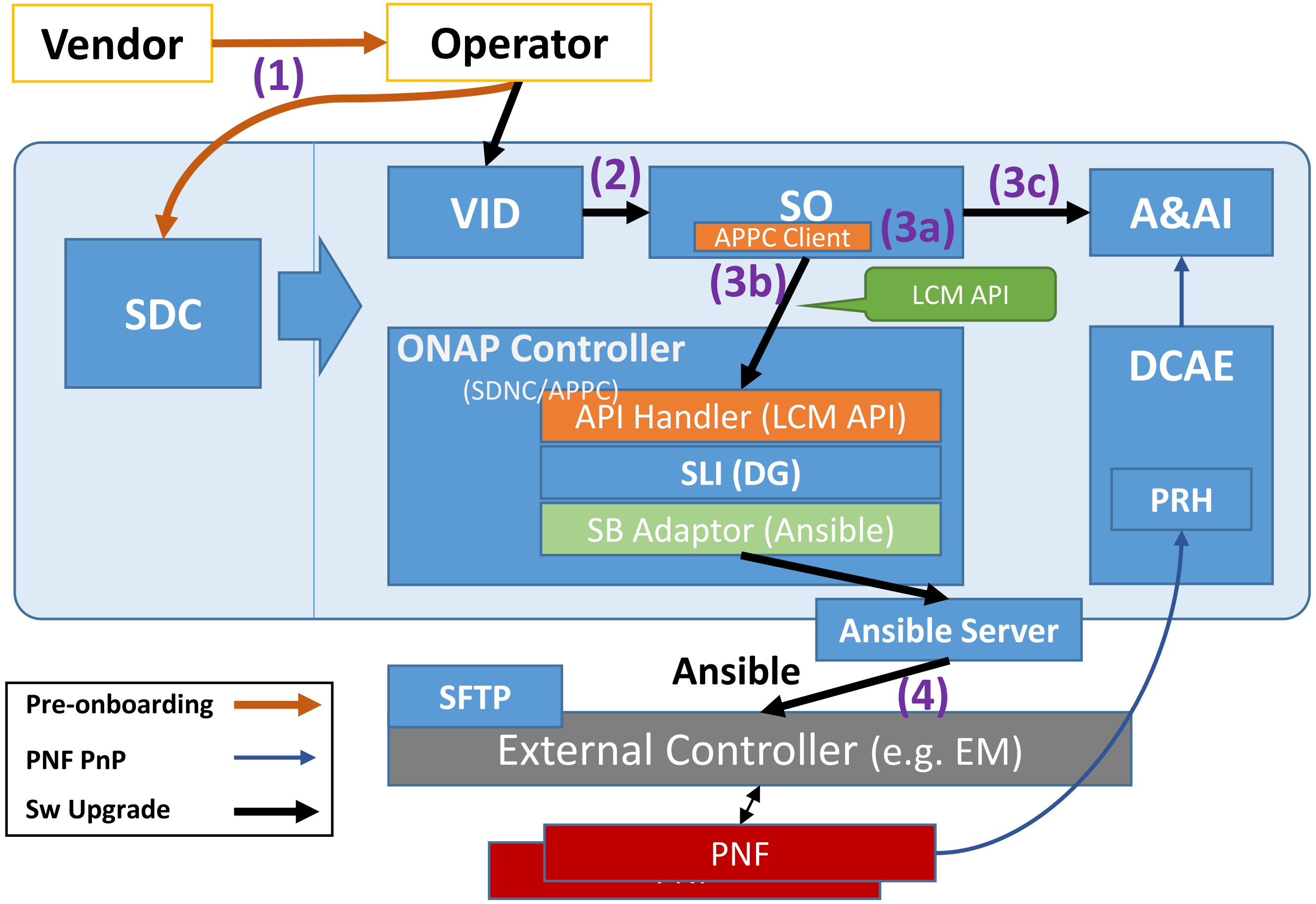
Key issues to support Scenario 2

- Enable software version (Modelling/SDC/AAI):
  - Onboarding the software version information using vendor provided onboarding package
  - Supporting software version in internal model
- Store the upgrade path info at design time (SDC/SO/VID):
  - Resource upgrade path
  - Service upgrade options
- Run the upgrade at service level (SO): Executing the workflow at service level
- Update the run time catalog (AAI):
  - Support updating resource model ID/version
  - Support updating service model ID/version



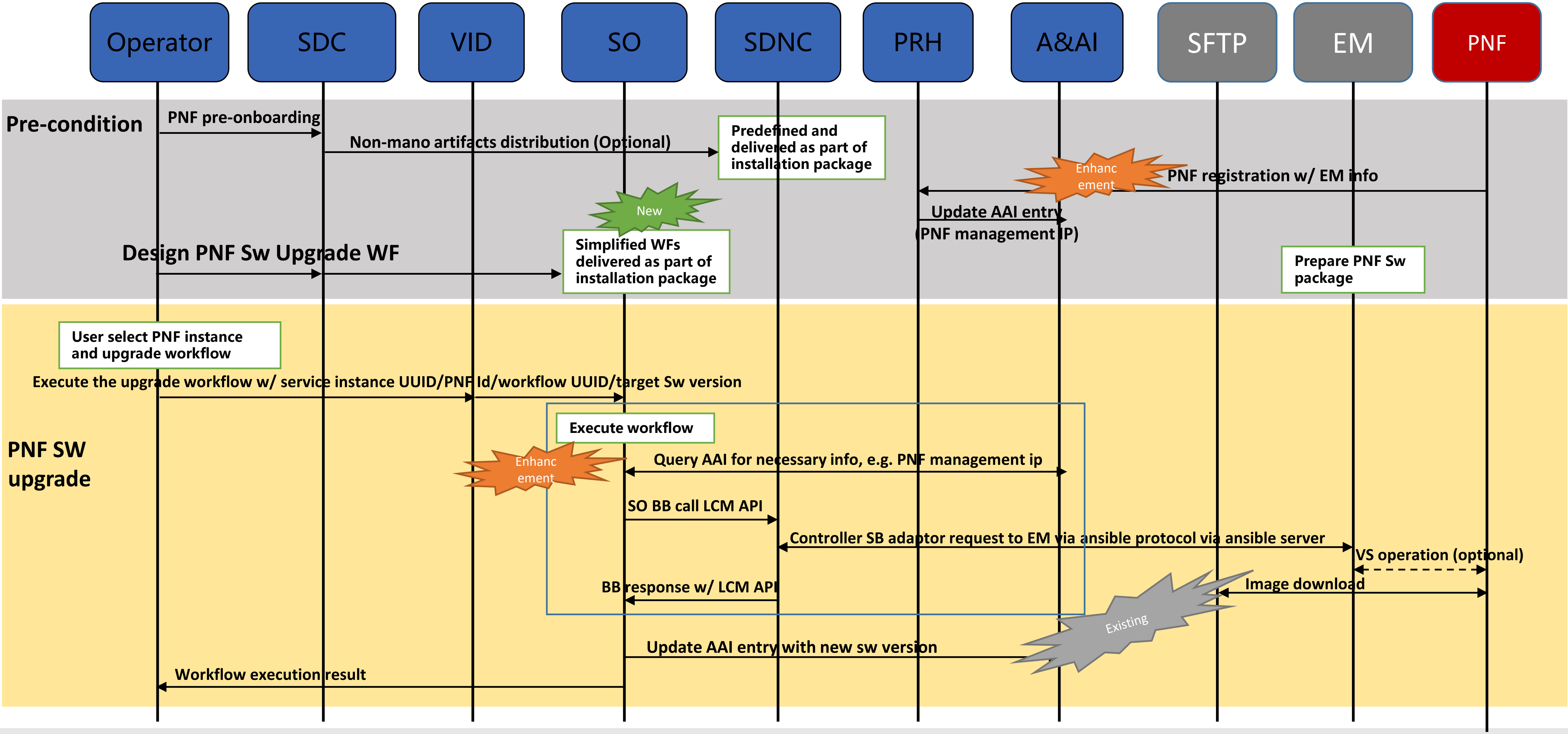
Scenario 3: E2E PNF Software Upgrade using Ansible

- This scenario will complete the E2E procedure of PNF in-place SW upgrade case started from Casablanca (evolved in Dublin).



- (1) In PNF pre-onboarding, operator delivers the PNF packages, including necessary ansible artifacts, to SDC.
- (2) In Sw Upgrade runtime, operator initiates the command, via VID or timer, to SO. (Before that, operator should design the upgrade workflow, or use the existing one).
- (3) SO executes the Sw upgrade task, like A&AI retrieval, and sends LCM requests to the controller.
- (4) Ansible Adaptor forwards requests to EM via ansible server.
- Note that, from the view of SB Adaptor, it should be generic that forwarding requests to both EM and NF.

Sequence Diagram (Scenario3)



Operator

SDC

VID

SO

SDNC

PRH

A&AI

SFTP

EM

PNF

Pre-condition

PNF pre-onboarding

Non-mano artifacts distribution (Optional)

Predefined and delivered as part of installation package

Enhancement

PNF registration w/ EM info

New

Update AAI entry (PNF management IP)

Design PNF Sw Upgrade WF

Simplified WFs delivered as part of installation package

Prepare PNF Sw package

User select PNF instance and upgrade workflow

Execute the upgrade workflow w/ service instance UUID/PNF Id/workflow UUID/target Sw version

PNF SW upgrade

Execute workflow

Enhancement

Query AAI for necessary info, e.g. PNF management ip

SO BB call LCM API

Controller SB adaptor request to EM via ansible protocol via ansible server

VS operation (optional)

Image download

BB response w/ LCM API

Existing

Update AAI entry with new sw version

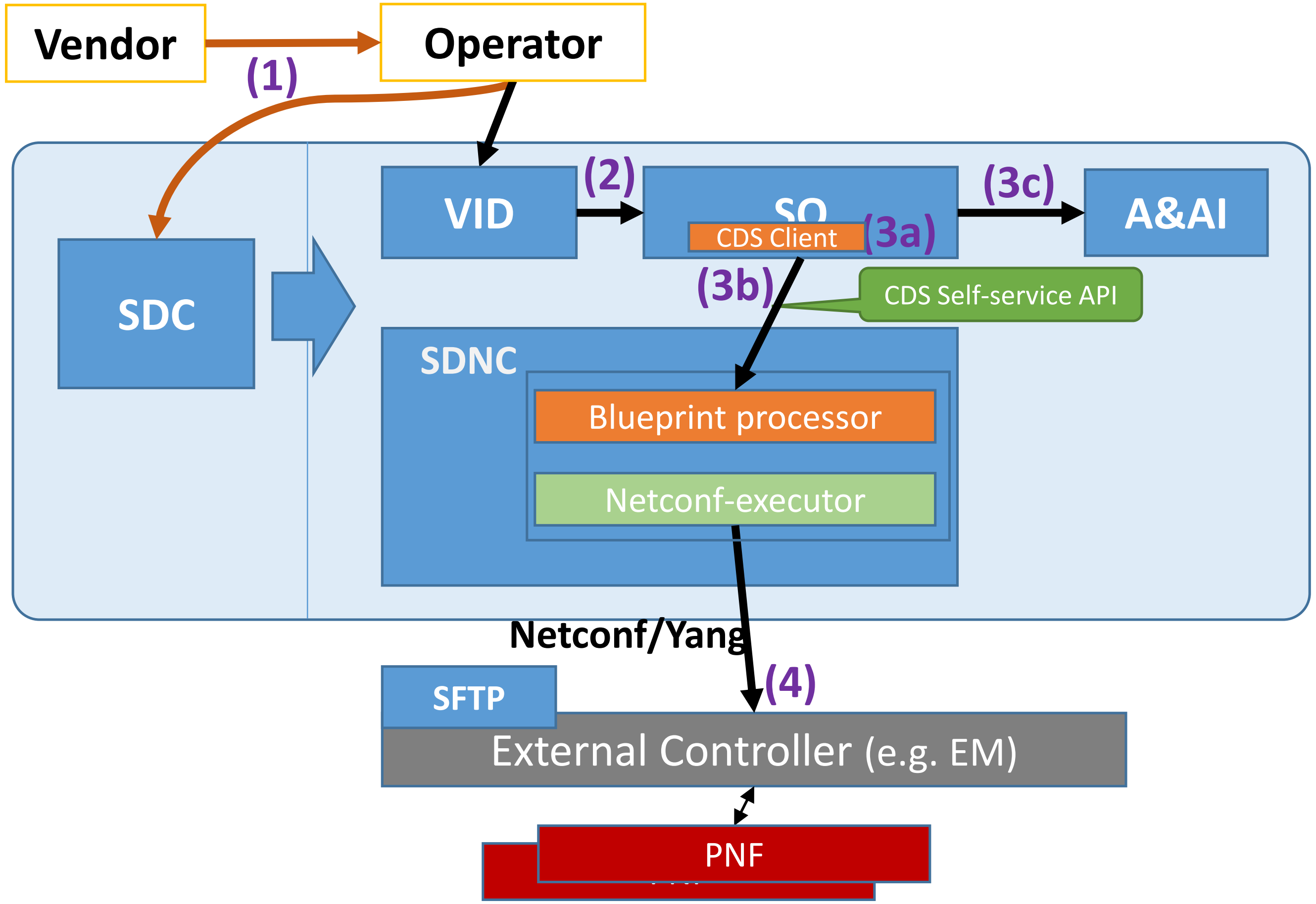
Workflow execution result

Impacts

Story	Components	Notes (ref)
Support generic PNF workflow design	SDC/SO	Reuse the solution from Scenario 1
Align with SO building block for PNF Upgrade procedure, including downloadNESw, activateNESw, update AAI, preCheck, postCheck	SDC/SO	Reuse the solution from Scenario 1
Support LCM API for downloadNESw and activateNESw actions	SDNC/CCSDK	
Provide ansible playbooks for downloadNESw and activateNESw	SDNC/CCSDK	
Enhance PNF registration with PNF management ip address	DCAE/PRH/VNFRQTS	
Support two different controller path in SO	SO	
Implement updateAAI activity for A&AI updated with active software version	SO	Reuse the solution from Scenario 1
Documentation	VNFRQTS	
Integration and testing	Integration	

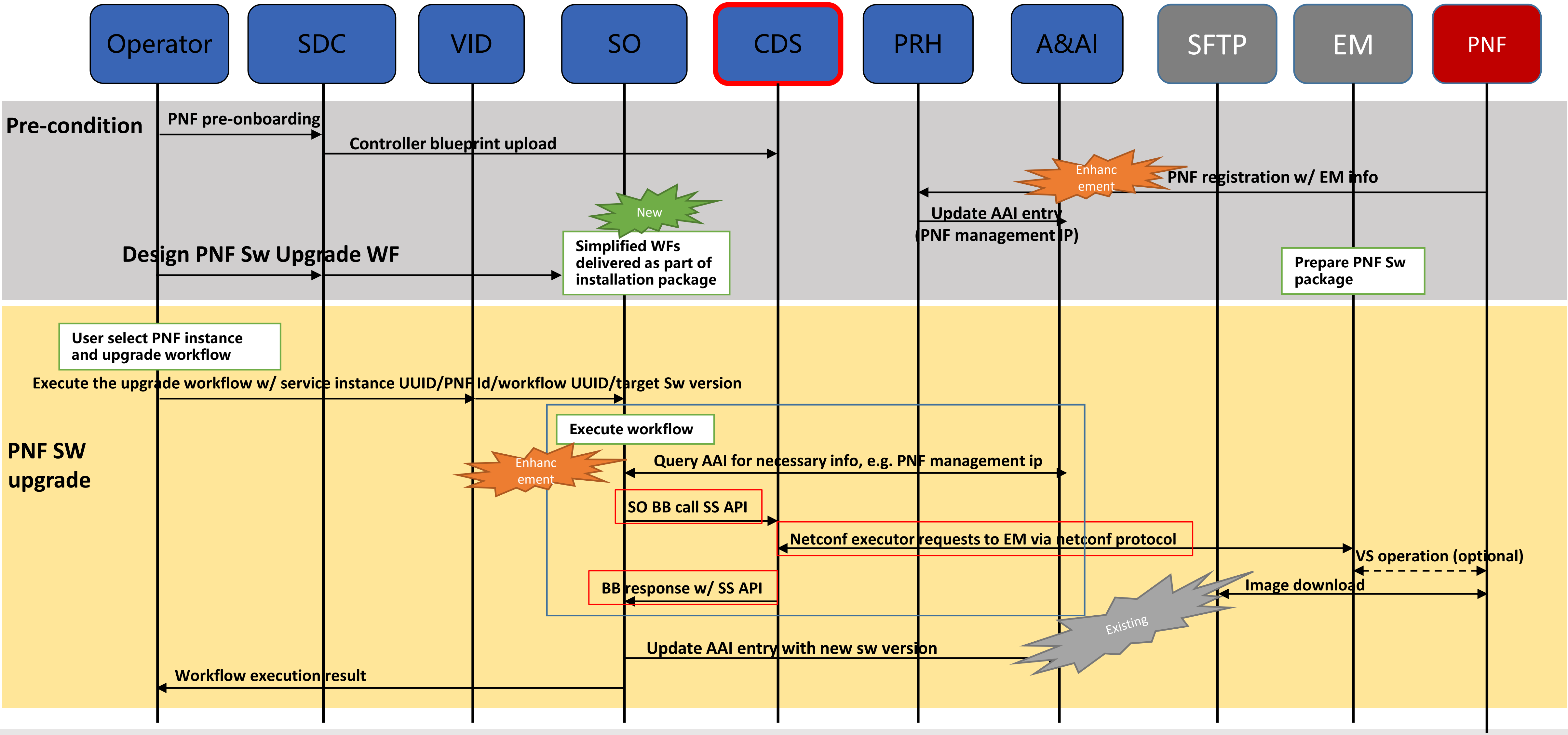
Scenario 4: PNF software upgrade with Netconf/Yang interface with EM

- Based on the scenario 1, that is to re-use the solution from VID to SDNC (netconf executor), this scenario will provide EM integration.



- (1) In PNF pre-onboarding, operator delivers the PNF packages, including necessary controller blueprint artifacts, to SDC.
- (2) In Sw Upgrade runtime, operator initiates the command, via VID or timer, to SO. (Before that, operator should design the upgrade workflow, or use the existing one).
- (3) SO executes the Sw upgrade task, like A&AI retrieval, and sends SS-API requests to the controller.
- (4) Executor forwards requests to EM with netconf/yang interface.

Sequence Diagram (Scenario4)


Operator
SDC
VID
SO
CDS
PRH
A&AI
SFTP
EM
PNF
Pre-condition
PNF pre-onboarding
Controller blueprint upload
Enhanc ement
PNF registration w/ EM info
New
Update AAI entry (PNF management IP)
Design PNF Sw Upgrade WF
Simplified WFs delivered as part of installation package
Prepare PNF Sw package
User select PNF instance and upgrade workflow
Execute the upgrade workflow w/ service instance UUID/PNF Id/workflow UUID/target Sw version
PNF SW upgrade
Execute workflow
Enhanc ement
Query AAI for necessary info, e.g. PNF management ip
SO BB call SS API
Netconf executor requests to EM via netconf protocol
VS operation (optional)
BB response w/ SS API
Image download
Existing
Update AAI entry with new sw version
Workflow execution result

Impacts

Story	Components	Notes (ref)
Support generic PNF workflow design	SDC/SO	Reuse the solution from Scenario 1/3
Create or modify SO activity building block for PNF upgrade, including downloadNESw, activateNESw, updateAAI, preCheck, postCheck	SDC/SO	Reuse the solution from Scenario 1
Update VID-SO API to execute the workflow of PNF software upgrade with target software version	VID/SO	Reuse the solution from Scenario 1
PNF/EM and CBA association enhancement to support PNF upgrade	SDC/CDS	
Support PNF upgrade with CDS self-service API	SO/CCSDK	Reuse the solution from Scenario 1
Implement updateAAI activity for A&AI updated with active software version	SO	Reuse the solution from Scenario 1
Enhance PNF registration with PNF management ip address	DCAE/PRH/VNFRQTS	Reuse the solution from Scenario 3
Support netconf interface with EM	EM	
Documentation	VNFRQTS	
Integration and testing	Integration	

ONAP

OPEN NETWORK AUTOMATION PLATFORM