SOL003 Adapter

Use Case

To manage vendor-provided VNFs and VIMs, ONAP wants to leverage the vendor-supplied VNFMs. For that, ONAP SO in Dublin added a plug-in capability for SVNFMs and interfaces with the plugged SVNFMs through SOL003 API standards, along with SOL004 VNF packages which include SOL001 VNFD. The SOL003 Adapter in Dublin supported the SOL003 Create, Instantiate, Terminate and Delete operations with Granting, Subscription and Notification. In the Frankfurt release, the additional SOL003 operations will be supported as follows. For the SOL004 and SOL001 support, see the ETSI Package Management section (ETSI Package Management).

- Package Management for SVNFM
- Granting Enhancement with HPA
- Query
- Modify (TBD)
- Policy-based Scaling (Stretch goal)
- Security between the Adapter and VNFMs
- Additional operations will be determined

Feature Descriptions

Feature	Description
SOL003 VNFM Adapter NBI Enhancement	SOL003 VNFM Adapter exposes its NBI to any VNFM Adapter Client. - VNFM Adapter will be part of SO microservices, but the adapter can be invoked thru MSB. - It is a stepping stone for GNFC to leverage the VNFM Adapter - Supports additional NBI operations for the additional SOL003-based SBI - Support secured communication protocol and authentication and authorization via AAF
SOL003 Adapter Package Management Support	SOL003 Adapter package management support based on SOL003 APIs - Handle SOL003-based VNF package requests from SVNFMs - Query VNF packages from the ONAP-ETSI Catalog Manager - Provide VNF packages to SVNFMs
Granting Enhancement	Granting is enhanced to support HPA by leveraging OOF
Additional of SOL003 operations	Support of additional SOL003 operations, such as Grant enhancement, Query, Modify, Scale, Operation Status, FM, PM, Heal, VNF Indicator, Retry, Rollback, Failing, Cancelling, Resource Quota Available Notification - operation selections and priorities will be determined

Mapping between SOL001 VNFD and SDC AID DM	The VNFM Adapter needs to handle mapping between SOL001 VNFD and SDC AID DM • Not all VNFD needs to be transformed to the SDC AID DM • The Adapter will map the model as much as it needs • Mapping to the VF-Module is questionable at this time	
Policy-based Scaling (Stretch goal)	Policy-based Scaling with VNF Indicator and VES event handling • Currently, SOL001 specifications does not support the VNF indicator configuration. As a result, VNFD-based VNF indicator is not supported; i.e., auto scaling is not yet supported • Mapping between VF-Module and ScalingAspect+Data is under discussion. Several VNF vendors indicated that they don't want to handle the VF-Module level. • It is a stretch goal for Frankfurt - TBD	
Secured communication and authentication and authorization support by SOL003 Adapter	Secured communication and authentication and authorization support • HTTPS protocol • Authentication and Authorization support via AAF	

Epic and User Story

Epic	User Story	Description	In Frankfurt?	JIRA	Size
SOL003 Adapter Enhancement for Frankfurt		Epic: ETSI Alignment Support - SOL003 Adapter Enhancement for Frankfurt	Yes (partially)	**SO-2156** - Getting issue details... **STATUS**	
	SOL003 Adapter NBI Enhancement	Expose the Adapter NBI to any SOL003 Adapter client in ONAP	No	**SO-2407** - Getting issue details... **STATUS**	
	• Create (Create /Instantiate)	• Create (Create/Instantiate)	No	**SO-2408** - Getting issue details... **STATUS**	
	• Delete (Terminate /Delete)	• Delete (Terminate/Delete)	No	**SO-2409** - Getting issue details... **STATUS**	
	• Query	• Query	No	**SO-2410** - Getting issue details... **STATUS**	
	• Get Operation Status	• Get Operation Status	No	**SO-2411** - Getting issue details... **STATUS**	
	• Modify	• Modify	No	-	
	• Heal	• Heal VNF (triggered by SOL003 Adapter, not auto-healing)	No	-	
	• Scaling (Stretch goal)	• Scaling (Stretch goal)	No	-	

|					
	SOL003 Adapter Package Management by leveraging ONAP-ETSI Catalog Manager	Support VNF Package Management interfaces between the SOL003 Adapter and SVNFM (SVNFM SOL003 Adapter) by leveraging ONAP-ETSI Catalog Manager (SOL003 Adapter ONAP-ETSI Catalog Mgr) • ETSI Catalog Manager APIs have been registered in MSB, and its access endpoint is http://{{MSB_HOST}}:{{MSB_PORT}}/api/vnfpkgm/v1/***. 📎 etsicatalog_API_v1.yaml •	Yes	**SO-2412** - Getting issue details... **STATUS**	

	• SOL003 Package Management Swagger update	• SOL003 Package Management Swagger update
		  ○ SOL003 Package Management Swagger
		  ○ ETSI Catalog Manager Swagger
		  ○ SOL003 Package Management Notification Swagger
			Yes	**SO-2501** - Getting issue details...
				STATUS
				
				SO-2542 - Getting issue details...
				STATUS
				
				SO-2464 - Getting issue details...
				STATUS	

	• Query VNF Package	• Query VNF Package (SVNFM SOL003 Adapter & SOL003 Adapter ETSI Catalog Manager) : query information about one or multiple VNF Package	Yes	**SO-2413** - Getting issue details... **STATUS**	
	• Read VNFD	• Read VNFD of an onboarded VNF package : VNFD is returned (SVNFM SOL003 Adapter & SOL003 Adapter ETSI Catalog Manager) • ETSI Catalog Manager will return a zip file of the Definition directory. 📎 parse_vnfpacakge.txt •	Yes	**SO-2414** - Getting issue details... **STATUS**	

• Fetch VNF Package Content	• Fetch VNF Package (SVNFM SOL003 Adapter & SOL003 Adapter ETSI Catalog Manager) : fetch the content of an onboarded VNF package (VNF package file is returned) • Support of the package_contents partial is a stretch goal in Frankfurt. The ETSI Catalog Manager supports fetching package_contents partial by setting the HTTP_RANGE in the request headers when the SOL003 Adapter wants to obtain a single range of bytes from the VNF package file.	Yes	**SO-2415** - Getting issue details... **STATUS**	
• Fetch VNF Package Artifacts	• Fetch VNF Package Artifacts (SVNFM SOL003 Adapter & SOL003 Adapter ETSI Catalog Manager) : fetch an individual artifact contained in an onboarded VNF package	Yes	**SO-2416** - Getting issue details... **STATUS**	
• Subscribe	• Subscribe (SVNFM SOL003 Adapter & SOL003 Adapter ETSI Catalog Manager) : subscribe to notifications related to onboarding and/or changes of VNF package	Yes	**SO-2417** - Getting issue details... **STATUS**	
• Query Subscription Info	• Query Subscription Info (SVNFM SOL003 Adapter)	Yes	**SO-2418** - Getting issue details... **STATUS**	
• Terminate /Delete Subscription	• Terminate/Delete Subscription (SVNFM SOL003 Adapter)	Yes	**SO-2419** - Getting issue details... **STATUS**	
• Notify	• Notify (SVNFM SOL003 Adapter & ETSI Catalog Manager SOL003 Adapter) : notify about VNF package onboarding or change	Yes	**SO-2420** - Getting issue details... **STATUS**	

• SOL003 Adapter Package Management - Get VNF Packages from ONAP-ETSI Catalog Manager	• **This is obsolete, by merge the functions into the above SO-2412 story and sub-tasks.**
	• Support the following VNF Package management (SOL003 Adapter ONAP-ETSI Catalog Manager). ETSI Catalog Manager will return the original vendor package to the Adapter.
	  ◦ Query VNF Package (/api/catalog/v1/vnfpackages)
	  ◦ Read VNFD // get the VNF package and extract VNFD model (sample)
	  ◦ Fetch VNF package content
	  ◦ Fetch VNF package artifacts
	  ◦ subscribe
	  ◦ Notify
		No	**SO-2421** - Getting issue details...
			
			STATUS	

SOL003 Adapter Grant Enhancement that supports HPA by leveraging OOF	VNF Granting that supports HPA by leveraging OOF	Yes	**SO-2422** - Getting issue details... **STATUS**	

SOL003 Adapter Enhancement for VNF Query	To invoke VNF Query	No	**SO-1636** - Getting issue details... **STATUS**	
SOL003 Adapter Enhancement for VNF Operation Status	To invoke Operation Status	No	**SO-2423** - Getting issue details... **STATUS**	
SOL003 Adapter Enhancement for VNF Modify	To invoke VNF Modify	No	-	
SOL003 Adapter Enhancement for VNF Heal	To invoke VNF Heal	No	-	
SOL003 Adapter Enhancement for VNF Scaling (Stretch goal)	to support Policy-based VNF Scaling support; Interface to DCAE for VES event	No	-	
Secured communication support between SOL003 Adapter and SVNFM	Secured communication between SOL003 Adapter and SVNFM	Yes (minimum)	**SO-2424** - Getting issue details... **STATUS**	
Authentication and authorization support between betwee n SOL003 Adapter and SVNFM	Authentication and authorization support between between SOL003 Adapter and SVNFM	Yes (minimum)	**SO-2425** - Getting issue details... **STATUS**	
Preload using user_param (without UI changes)	Preload using user_param (without UI changes)	Yes	**SO-1635** - Getting issue details... **STATUS**	
SO BPPN Workflows & Java for SOL003 Operations	SO BPMN Workflows & Java for SOL003 Operations by leveraging the SOL003 Adapter NBIs	No	**SO-2466** - Getting issue details... **STATUS**	
• Enhance Create /Instantiate Workflows & Associated Java code	• Enhance Create/Instantiate Workflows & Associated Java code ○ Modify the existing workflow to separate Create and Instantiate tasks ○ Invoke the modified SOL003 Adapter NBIs	No	**SO-2467** - Getting issue details... **STATUS**	
• Enhance Terminate /Delete Workflows & Associated Java code	• Enhance Terminate/Delete Workflows & Associated Java code ○ Modify the existing workflow to separate Terminate and Delete tasks ○ Invoke the modified SOL003 Adapter NBIs	No	**SO-2468** - Getting issue details... **STATUS**	
• Create Query VNF Workflows & Associated Java code	• Create Query VNF Workflows & Associated Java code ○ Invoke the modified SOL003 Adapter Query VNF NBI	No	**SO-2469** - Getting issue details... **STATUS**	

			No	**SO-2470** - Getting issue details... STATUS	
• Create Get Operation Status Workflows & Associated Java code	• Create Get Operation Status Workflows & Associated Java code ◦ Invoke the modified SOL003 Adapter Get Operation Status NBI ◦ This operation could be used by the Instantiation and Termination of VNF; in this case, its workflow could be omitted	No	**SO-2470** - Getting issue details... STATUS	
CSIT enhancement for testing SOL003 Adapter package management enhancement	CSIT enhancement for testing SOL003 Adapter package management enhancement	No	**SO-2683** - Getting issue details... STATUS	
Documentation for SOL003 Adapter enhancement features	Documentation for SOL003 Adapter enhancement features	NO		

SOL003 VNFM Adapter Architecture for Frankfurt

The diagram depicts SOL003 VNFM Adapter Architecture.
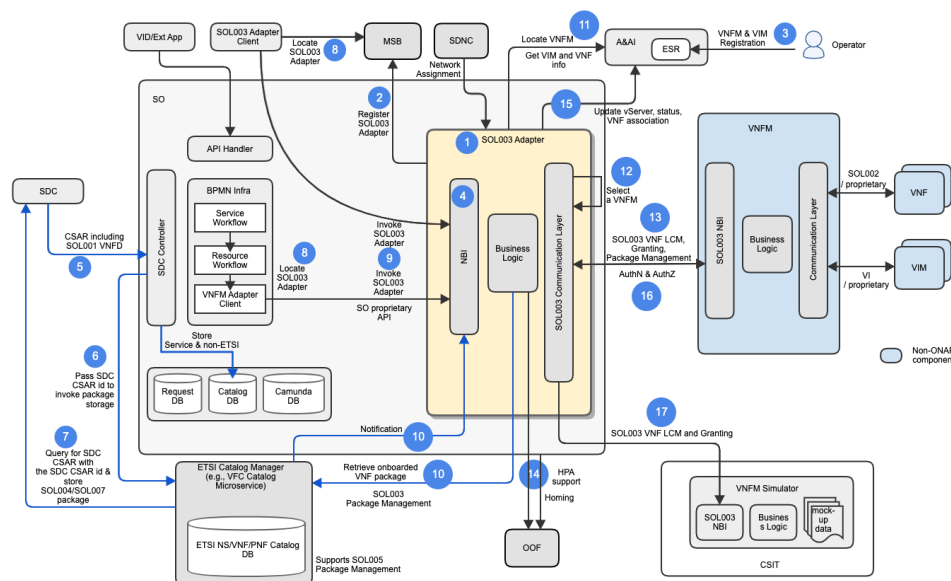
1. SOL003 Adapter continues to be an SO microservice component, and exposes its NBI to any SOL003 Adapter client in ONAP
2. SOL003 Adapter is registered to MSB.
3. Operator registers VNFM and VIM to ESR in AAI.
4. SOL003 Adapter exposes its NBI to any SOL003 Adapter client in ONAP (not for Frankfurt)
    a. Interfaces will be refactored to be generic to allow access by other ONAP components.
    b. The NBI will be enhanced for additional SOL003 operation support
5. SDC distributes SDC packages including the vendor original SOL004 (VNF and PNF) and SOL007 (NS) packages – SOL007 is not for Frankfurts
6. SO (SDC Controller) passes the SDC CSAR ID to ETSI Catalog Manager to invoke storage
7. ETSI Catalog Manager queries for SDC CSAR with the SDC CSAR id & store SOL004/SOL007 package.
8. SO (BPMN) and the SOL003 Adapter client locates SOL003 Adapter.
9. SO (BPMN) and the SOL003 Adapter client invokes SOL003 Adapter.
10. SOL003 Adapter retrieves VNF package from Catalog Manager.
11. SOL003 Adapter gets available VNFM locations (endpoints) and gets VIM and VNF Info.
12. SOL003 Adapter selects a VNFM, based on a VNFM locating mechanism.
13. SOL003 Adapter and SVNFM supports SOL003 VNF LCM, granting and package management operations.
14. SOL003 Adapter supports HPA-based Granting, leveraging OOF.
15. SOL003 Adapter updates vServer, status and VNF association in AAI
16. SOL003 Adapter and SVNFM support authentication and authorization (AAF, and vendor AA mechanism) – partially for Frankfurt.
17. For integration testing, the VNFM Simulator is used.

SOL003 Operations

The following SOL003 operations will be supported:

- Package Management
- Grant enhancement
- Query of VNF
- Scaling (stretch goal)
- Modify
- Operation Status

SOL003-based Operation Sequence Flows

- Create / Instantiate VNF
  - It is supported in the Dublin release, and it will be enhanced to leverage the ONAP-ETSI Catalog Manager.
- The following sequence flows depicts the SOL003 Create/Instantiate operations by leveraging the above architecture.

VID | SO_BPMN | SOL003_Adapter | ONAP_ETSI_Catalog_Mgr | AAI | OOF | VNFM

1 Create VNF

2 Create

3 Get Generic VNF using ID from create request

4 Response with Generic VNF

5 Get VNFMs

6 Response with VNFMs

7 Select VNFM using nf type from Generic VNF

8 Get package using Model ID from Generic VNF

9 Response with VNF Package

10 Send Create to SVNFM

11 Create the VNF ID

12 Response with VNF ID

13 Update Generic VNF with self link. Add relationship from Generic VNF to VNFM and Tenant

14 Subscribe for notifications related to the created VNF

15 Response for subscription

16 Send Instantiate to SVNFM

17 Start Instantiation

18 Response with Operation ID

19 Add Operation ID

20 Response with Job ID

21 Send Grant Request

alt [resource-level HPA]

22 Request for HPA homing information

23 Response with homing information

[SO cashes Homing Info at the Service level]

24 get HPA homing information from SO/SOL003 Adapter cache

25 Send Grant Response

26 Instantiate VNF on VIM

27 Send operation completed notification

28 Add vserver info, update orchestration status of Generic VNF

29 Get Job Status

30 Query Operation Status

31 Response with Operation Status

32 Response with Status

33 Create VNF is done

VID | SO_BPMN | SOL003_Adapter | ONAP_ETSI_Catalog_Mgr | AAI | OOF | VNFM

- Terminate/Delete VNF
  - It is supported in the Dublin release, and it will be enhanced to leverage the ONAP-ETSI Catalog Manager.
- The following sequence flows depicts the SOL003 Terminate/Delete operations by leveraging the above architecture.

<Diagram : tbd>

SOL003 Adapter - SVNFM - ETSI Catalog Manager Operations

The SOL003 Adapter supports the following SOL003-based operations.
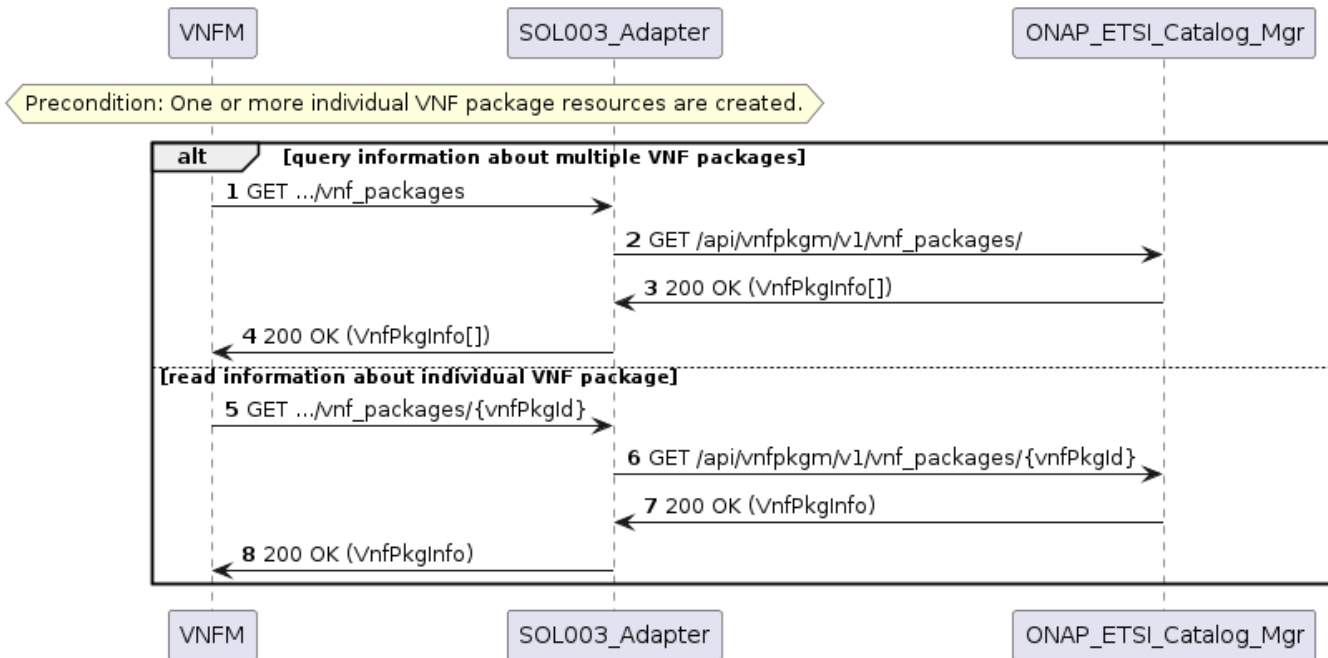
Operation Interfaces Requirements

API Action	Actor	Method	URI	Description
Query VNF Package	VNFM SOL003 Adapter	GET	`/vnf_packages, /vnf_packages/ {vnfPkgId}`	To query VNF packages or individual packages
Read VNFD	VNFM SOL003 Adapter	GET	`/vnf_packages/{vnfPkgId}/vnfd`	To read VNFD
Fetch VNF Package	VNFM SOL003 Adapter	GET	`/vnf_packages/{vnfPkgId} /package_content`	To fetch a VNF package
Fetch VNF Package Artifact	VNFM SOL003 Adapter	GET	`/vnf_packages/{vnfPkgId}/artifacts/ {artifactPath}`	To fetch a VNF package artifact
Create Subscription for Package Management	VNFM SOL003 Adapter	POST	/subscription (PkgmSubscriptionRequest)	To create a subscription for package management
Sending Notification for Package Management	SOL003 Adapter VNFM	POST	<<Client side URL>> (<<Notification>>)	To post a notification for package management
Grant Request	VNFM SOL003 Adapter	POST	/grants/ (GrantRequest)	To get VNF granting
Query VNF	SOL003 Adapter VNFM	GET	/vnf_instances /vnf_instances/{vnfInstanceId}	To get VNF instances or a VNF instance
Modify VNF	SOL003 Adapter VNFM	PATCH	/vnf_instances/{vnfInstanceId} (VnfInfoModificationRequest)	To modify VNF instances
Get Operation Status	SOL003 Adapter VNFM	GET	/vnf_lcm_op_occs	To get VNF operation status

Package Management

- **Query/Reading VNF package information (VNFM SOL003 Adapter)**
  - SOL003 Adapter Package - Query VNF Package

    - SOL003 Adapter provides the SOL003 Query/Reading VNF package services to the SVNFM.
    - SVNFM requests for multiple VNF packages or an individual VNF package for the given vnfPkgId to SOL003 Adapter
      - GET .../vnf_packages, GET .../vnf_packages/{vnfPkgId}
    - SOL003 Adapter queries for VNF packages or an individual VNF package to the ETSI Catalog Manager.
      - GET /api/vnfpkgm/v1/vnf_packages, GET /api/vnfpkgm/v1/vnf_packages/{vnfPkgId}
    - ETSI Catalog Manager returns VNF packages or an individual VNF package to the SOL003 Adapter
    - SOL003 Adapter returns the VnfPkgInfo[] or VnfPkgInfo to the SVNFM

    Precondition: one or more individual VNF package resources are created.

    Postcondition: SVNFM receives VNF packages or an individual VNF package in form of VnfPkgInfo[] or VnfPkgInfo
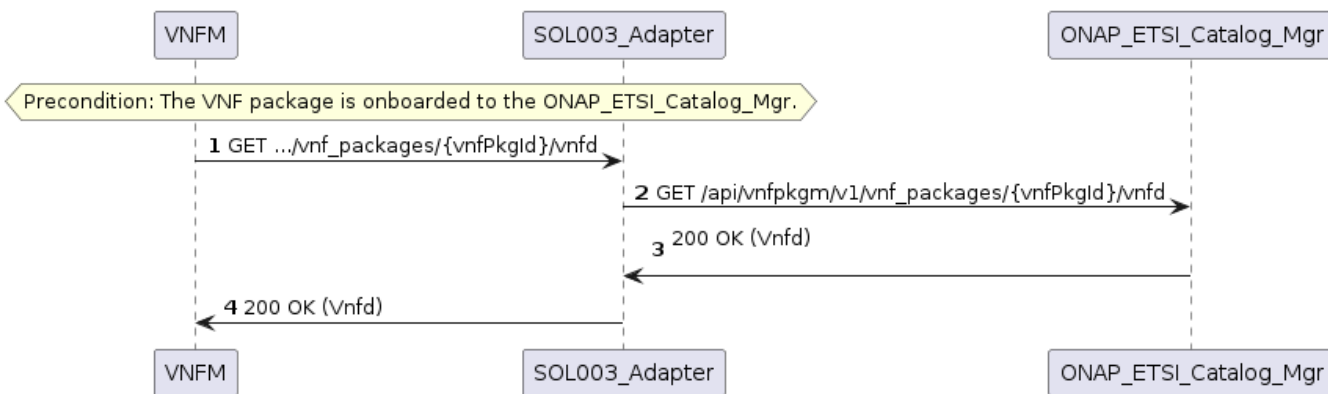
VNFM   SOL003_Adapter   ONAP_ETSI_Catalog_Mgr

Precondition: One or more individual VNF package resources are created.

alt   [query information about multiple VNF packages]

1 GET .../vnf_packages

2 GET /api/vnfpkgm/v1/vnf_packages/

3 200 OK (VnfPkgInfo[])

4 200 OK (VnfPkgInfo[])

[read information about individual VNF package]

5 GET .../vnf_packages/{vnfPkgId}

6 GET /api/vnfpkgm/v1/vnf_packages/{vnfPkgId}

7 200 OK (VnfPkgInfo)

8 200 OK (VnfPkgInfo)

VNFM   SOL003_Adapter   ONAP_ETSI_Catalog_Mgr

- **Reading VNFD of an on-boarded VNF package (VNFM SOL003 Adapter ETSI Catalog Manager)**
    - ○ SOL003 Adapter Package Management - Read VNFD
        - ■ SOL003 Adapter provides the SOL003 Reading VNFD service to SVNFM.
        - ■ SVNFM requests for a VNFD for the given vnfd to SOL003 Adapter
            - • GET .../vnf_packages/{vnfpkgid}/vnfd
        - ■ SOL003 Adapter requests for a VNFD for the given vnfd to ETSI Catalog Manager
            - • GET /api/vnfpkgm/v1/vnf_packages/{vnfPkgId}/vnfd
        - ■ ETSI Catalog Manager returns a zip file of VNF package Definition directory to SOL003 Adapter
        - ■ SOL003 Adapter returns a zip file of VNF package Definition directory to SVNFM

Precondition: The VNF package is onboarded to the ONAP-ETSI Catalog Manager

Postcondition: SVNFM receives a Vnf package definition zip file

VNFM   SOL003_Adapter   ONAP_ETSI_Catalog_Mgr

Precondition: The VNF package is onboarded to the ONAP_ETSI_Catalog_Mgr.

1 GET .../vnf_packages/{vnfPkgId}/vnfd

2 GET /api/vnfpkgm/v1/vnf_packages/{vnfPkgId}/vnfd

3 200 OK (Vnfd)

4 200 OK (Vnfd)

VNFM   SOL003_Adapter   ONAP_ETSI_Catalog_Mgr

- **Fetching an on-boarded VNF package (VNFM SOL003 Adapter)**
    - ○ SOL003 Adapter supports the package content fetch.
        - ■ Support of the package_contents partial is a stretch goal in Frankfurt. The ETSI Catalog Manager supports fetching package_contents partial by setting the HTTP_RANGE in the request headers when the SOL003 Adapter wants to obtain a single range of bytes from the VNF package file.
    - ○ SVNFM sends a GET request for the VNF package content resource to the SOL003 Adapter.
        - ■ GET .../vnf_packages/{vnfPkgId}/package_content
    - ○ SOL003 Adapter sends a GET request for the VNF package content resource to the ETSI Catalog Manager
        - ■ GET /api/vnfpkgm/v1/vnf_packages/{vnfPkgId}/package_content
    - ○ ETSI Catalog Manager returns a copy of the VNF package file in the payload body to SOL003 Adapter.
    - ○ SOL003 Adapter returns with a copy of the VNF package file in the payload body to SVNFM.

Precondition: the VNF package is onboarded to the ETSI Catalog Manager

Postcondition: the SVNFM gets the whole content of the VNF package
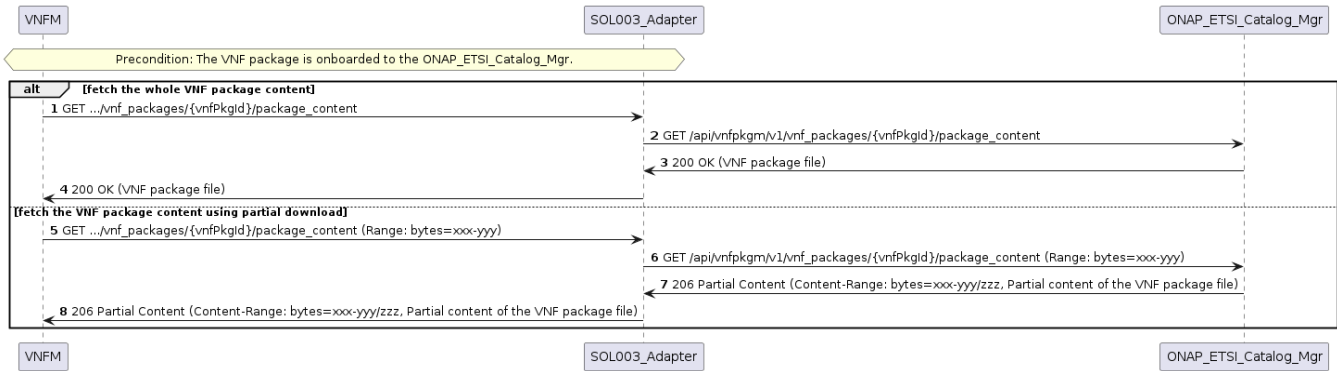
- **Fetching a VNF package artifact (VNFM SOL003 Adapter)**
  - ○ The SOL003 Adapter supports the whole content of the artifact only.
  - ○ The SVNFM sends a GET request for the Individual VNF package artifact to SOL003 Adapter.
    - GET .../vnf_packages/{vnfPkgId}/artifacts/{artifactPath}
  - ○ The SOL003 Adapter sends a GET request for the individual VNF package artifact to the ETSI Catalog Manager
    - GET /api/vnfpkgm/v1/vnf_packages/{vnfPkgId}/artifacts/{artifactPath}
  - ○ The ETSI Catalog Manager returns a copy of the applicable artifact files from the onboarded VNF package to the SOL003 Adapter.
  - ○ The SOL003 Adapter returns with a copy of the applicable artifact file from the onboarded VNF package to SVNFM.

Precondition: The VNF package is onboarded to the ONAP-ETSI Catalog Manager

Postcondition: the SVNF gets the applicable artifact file.
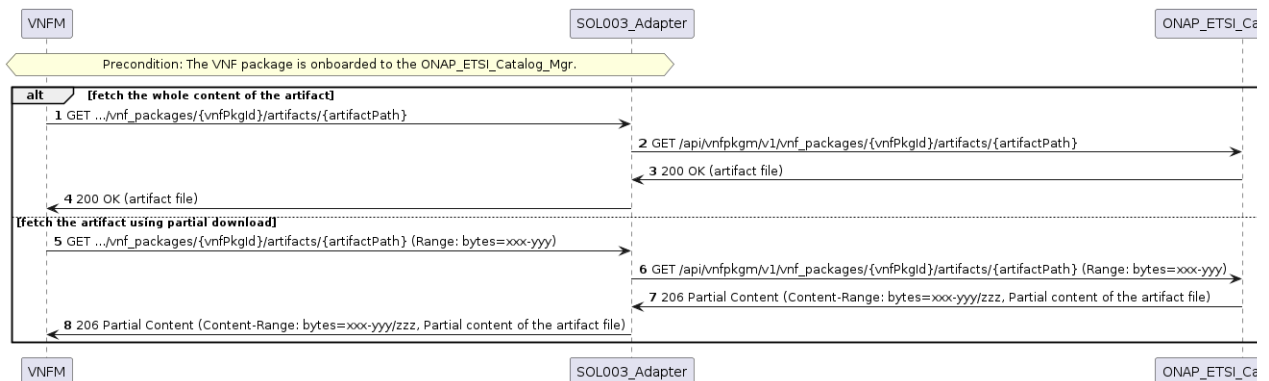
- **Creating subscriptions for the package management (VNFM SOL003 Adapter)**
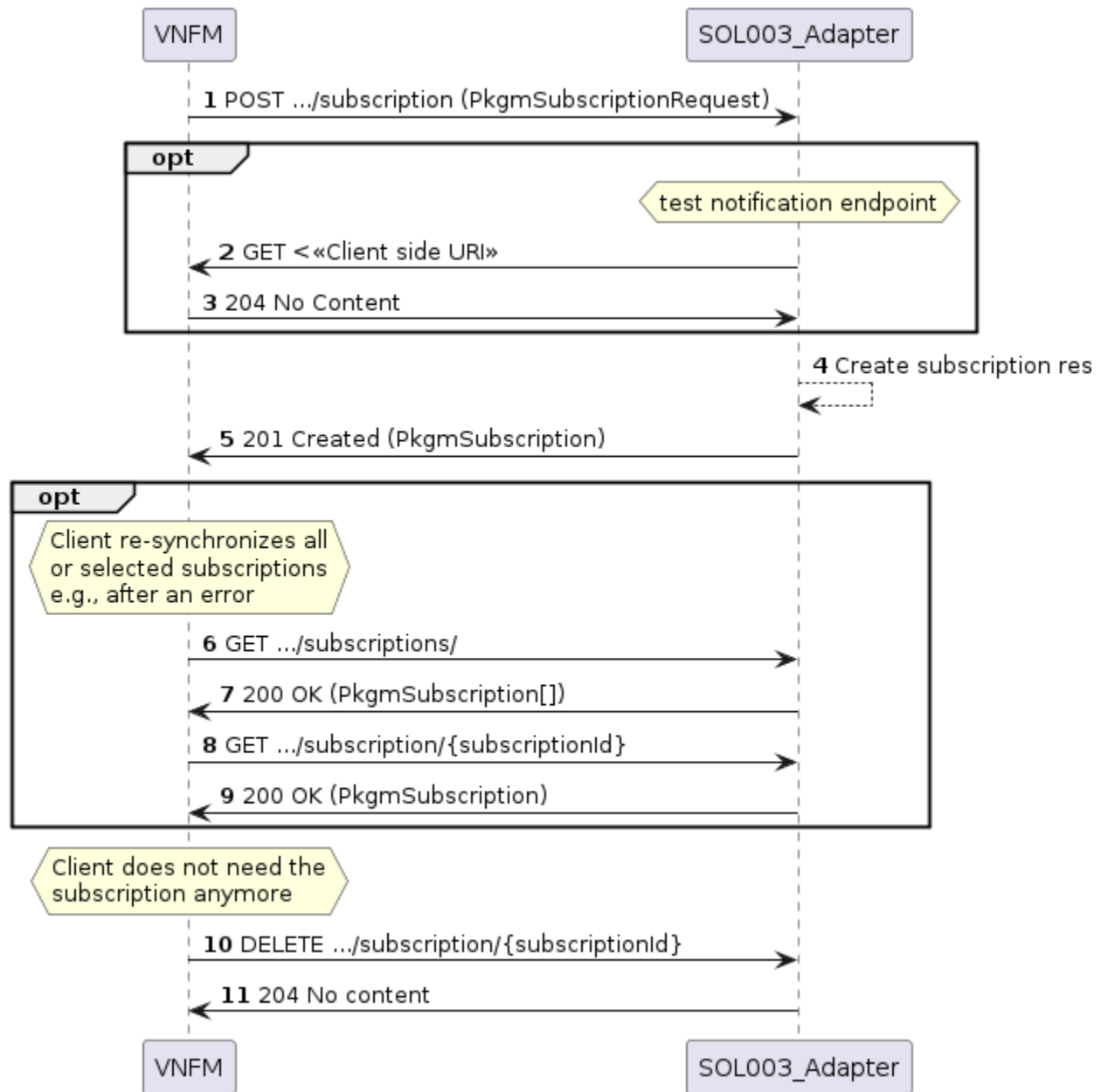 - ○ Subscribe (SVNFM SOL003 Adapter) : subscribe to notifications related to onboarding and/or changes of VNF package
  - ○ The SVNFM sends a POST request for the Subscriptions resource including PkgSubscriptionRequest to the SOL003 Adapter.
  - ○ The SOL003 Adapter returns with a data structure of PkgmSubscription to the SVNFM

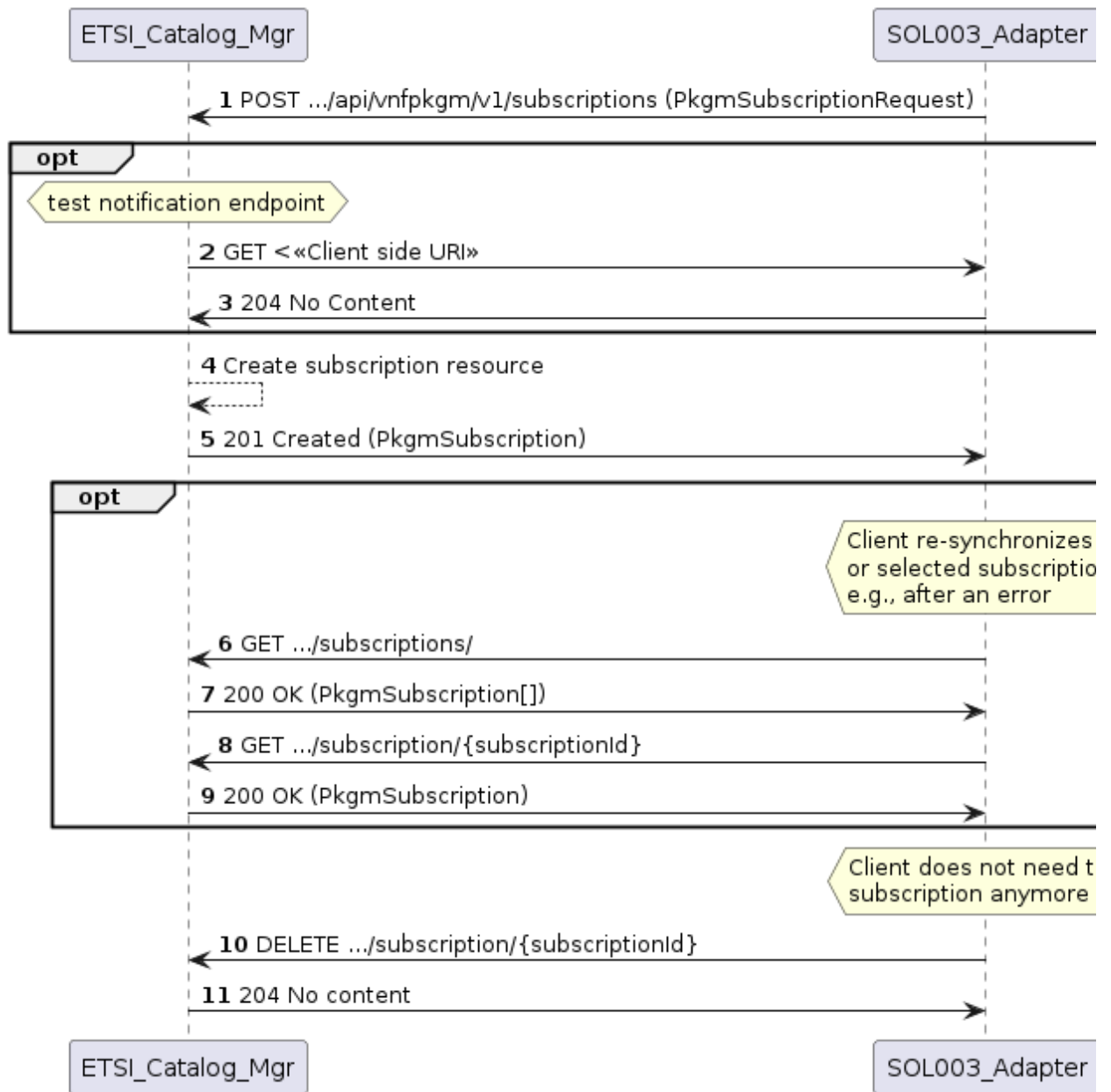Precondition: SVNFM does not have package subscription.

Postcondition: SOL003 Adapter create subscription resource in memory (persistency is to be determined)

SVNFM gets package subscription (PkgmSubscription)

<Sequence Diagram SVNFM SOL003 Adapter>

VNFM → SOL003_Adapter

1 POST .../subscription (PkgmSubscriptionRequest)

opt

test notification endpoint

2 GET <«Client side URI»

3 204 No Content

4 Create subscription res

5 201 Created (PkgmSubscription)

opt

Client re-synchronizes all or selected subscriptions e.g., after an error

6 GET .../subscriptions/

7 200 OK (PkgmSubscription[])

8 GET .../subscription/{subscriptionId}

9 200 OK (PkgmSubscription)

Client does not need the subscription anymore

10 DELETE .../subscription/{subscriptionId}

11 204 No content

<Sequence Diagram SOL003 Adapter ETSI_Catalog_Manager>

- **Query Subscription Info**
  - SVNFM re-synchronizes all or selected subscriptions after some errors in SVNFM or some exception handling.
  - SVNFM sends a Query Subscription Info request to the SOL003 Adapter.
    - GET .../subscriptions // for all subscriptions
  - SOL003 Adapter returns package subscription lists in the form of PkgmSubscription[] in the payload
  - or
  - SVNFM sends a Query Subscription Info request for a selected subscription to the SOL003 Adapter
    - GET .../subscriptions/{subscriptionId}
  - SOL003 Adapter returns a selected package subscription to SVNFM.

Precondition: SVNFM lost package subscription information

Postcondition: SVNFM gets package subscription information
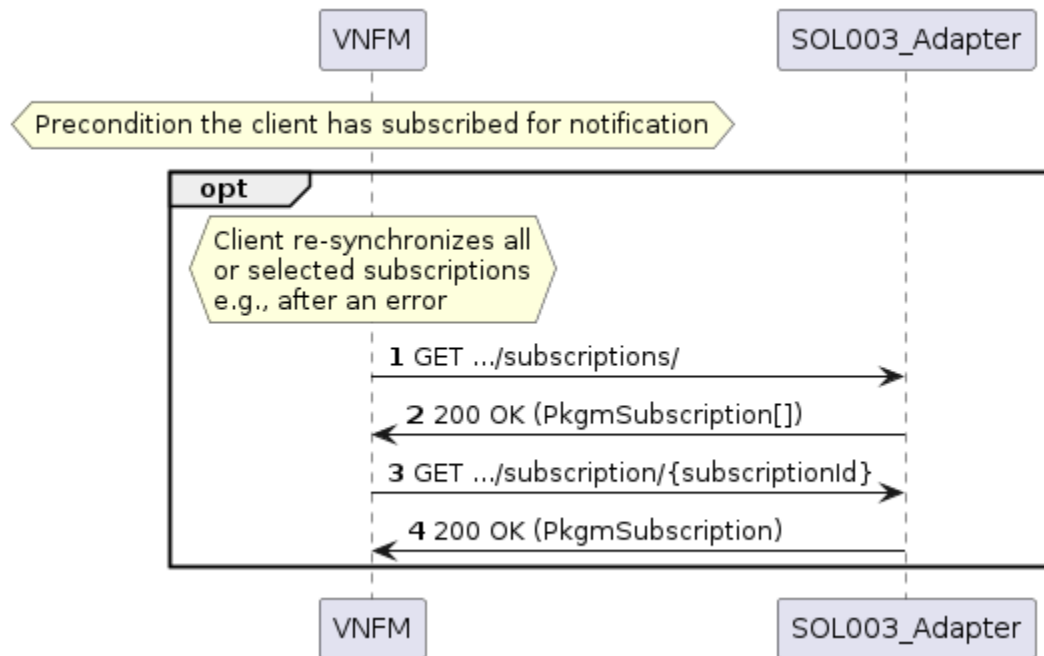
- **Delete Subscription**
  - SVNFM does not need the subscription anymore.
  - SVNFM sends a request for Terminate/Delete Subscription to the SOL003 Adapter.
  - SOL003 Adapter removes the SVNFM package subscription.
  - SOL003 Adapter sends the 204 No content response to SVNFM.

Precondition: SVNFM subscribed to the SOL003 Adapter for package notifications

Postcondition: SOL003 Adapter removed the SVNFM package subscription.
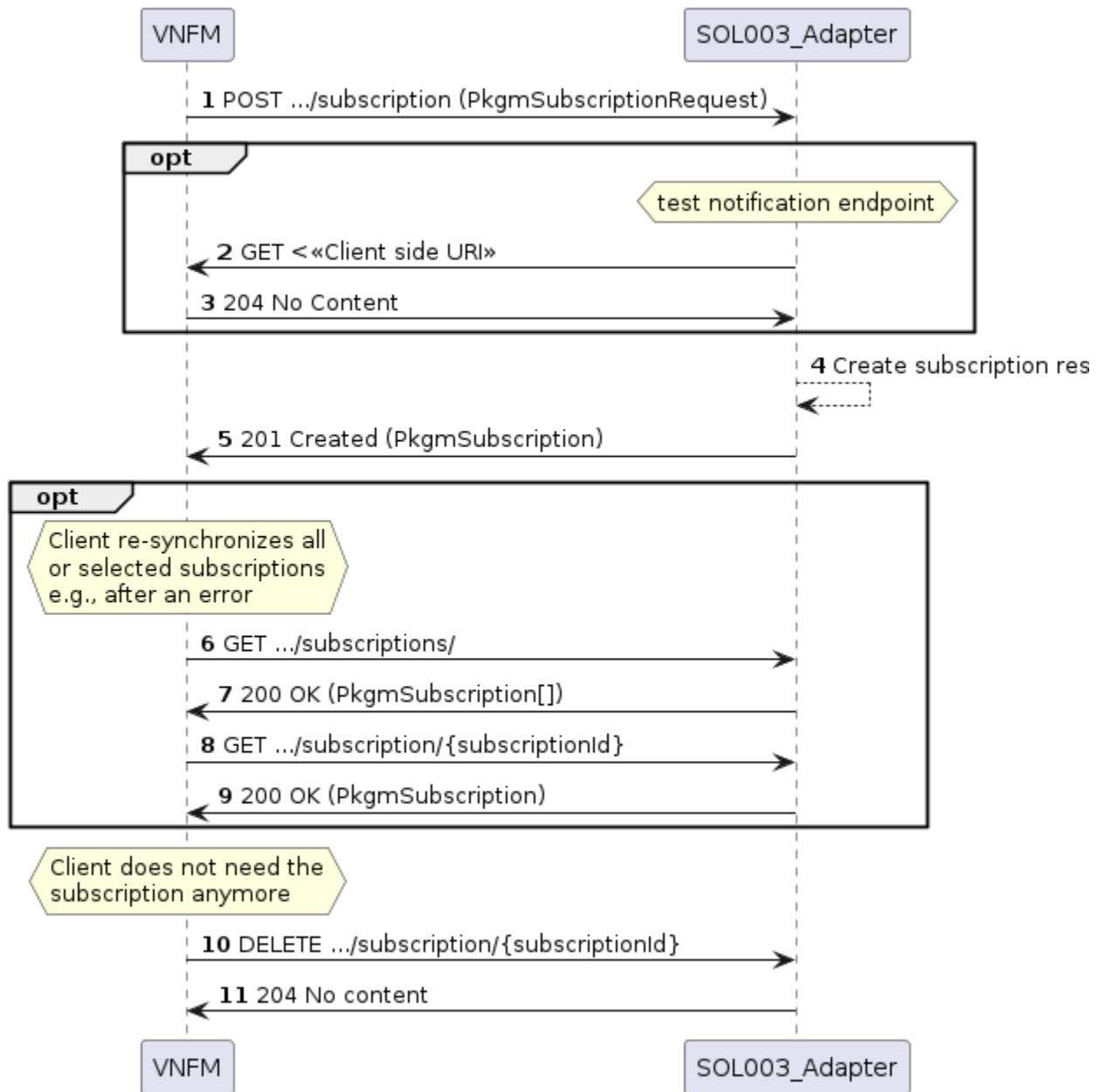
- **Sending notifications for the package management (SOL003 Adapter VNFM)**
  - If an event occurs that matches the filtering criteria, the SOL003 Adapter generates a notification that includes information about the event, and sends it to the registered SVNFM.
  - SOL003 Adapter receives a notification from the ONAP-ETSI Catalog Manager
  - SOL003 Adapter sends a Notify message about VNF package onboarding or change
    - POST <<Client side URL>> (<<Notification>>)
  - SVNFM acknowledges the successful delivery of notification.
    - 204 No content

Precondition: SVNFM has subscribed to the SOL003 Adapter previously

      SOL003 Adapter subscribed to the ETSI Catalog Manager previously

Postcondition: SVNFM receives a notification
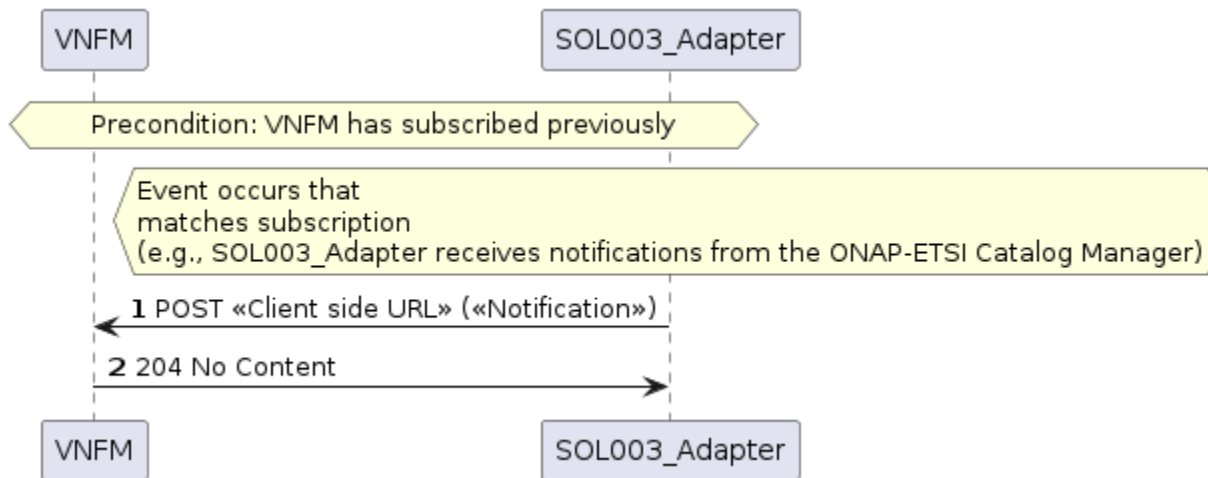
- Receiving notifications for the package management (ETSI Catalog Manager SOL003 Adapter)
  - The ETSI Catalog Manager team is working on the Notification API.
  - Once the notification api is settled, the subscribed SOL003 Adapter will receive package notifications from the ETSI Catalog Manager.
- <Sequence Diagram: TBD>

Grant Request with Synchronous Response with HPA (VNFM SOL003 Adapter)

The SOL003 Adapter acts as a NFVO and supports the Synchronous Grant operation. Hardware Platform capability requirements are downloaded as part of the VNFD data.

There are two methods to support HPA in SO. **In the Frankfurt release, only the Method 1 will be supported.**

- **Method 1: Use of Homing information that is made during the SO decompose processing.**
  - During SO decompose processing, SO calls OOF for collecting homing information for the service (which includes the VNFs).
  - Call the OOF APIs to perform the optimize service/VNF homing and placement
  - Use the existing homing workflows to pass hardware platform capability requirements to OOF
  - See the following diagram for the component interaction.

  Postcondition: SVNFM receives a granting
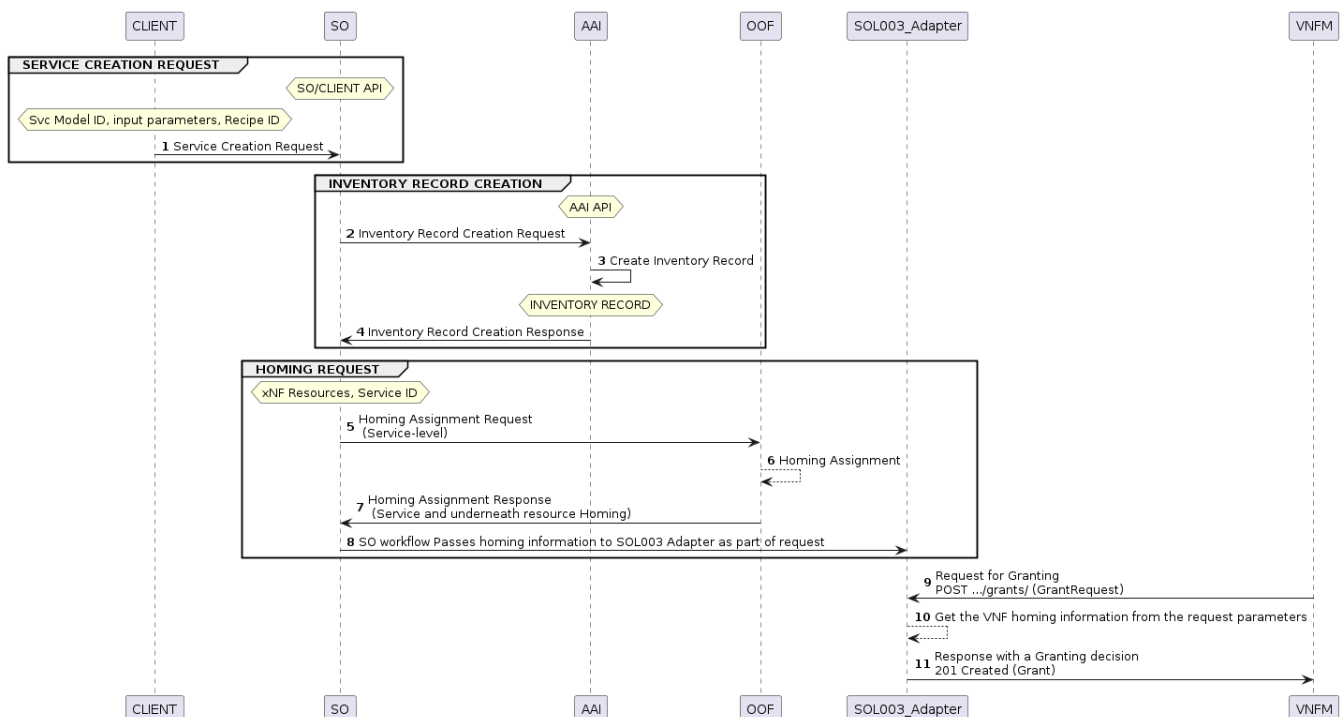
- **Method 2: If the above processing does not work for the vendor VNF HPA, OOF will be called from the SOL003 Adapter per VNF. The following describes the interactions between SOL003 Adapter and OOF.**
    - **Note: current, OOF does NOT support the VNF Level Homing request. OOF API enhancement is under discussion. Otherwise, the method 1 (Service Level Homing) is used for Frankfurt - Out of Frankfurt Release scope**
    - VNFM Adapter sends out homing requests to OOF (OSDF) containing resource info
    - OOF (OSDF) pulls all the related homing constraints from Policy
    - OOF (HAS) checks AAI database to pull region (flavor) information
    - OOF (HAS) communicates with Multi-Cloud to check cloud capacity (vims which fulfill the requirements)
    - OOF (OSDF) returns homing allocation solution to VNFM Adapter
    - OOF collects information as following:
    - Service and Resource Info, from: AAI
    - HPA Flavors/Capabilities/Capacity Info, from: AAI
    - Policy Models (Homing, PCI) from: Policy
    - Infrastructure Metrics Info (capacity), from: MultiCloud
    - Cloud Agnostic Intent Info, from: MultiCloud
    - PCI configuration data (not yet a part of SDC model)
- **SOL003 Adapter Homing Request to OOF**
    - <describe the homing request API and contract here>
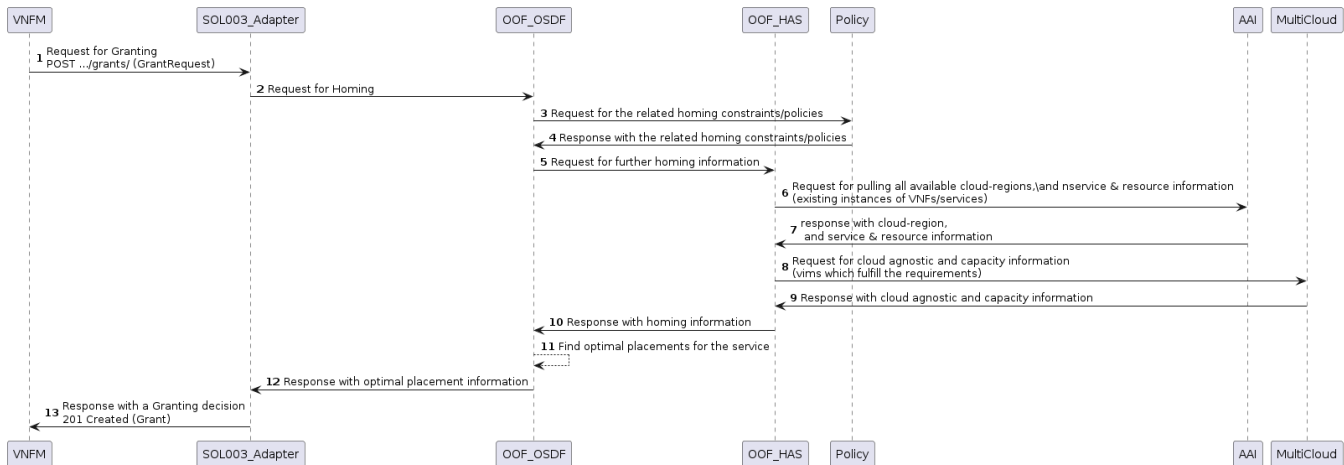
GrantRequest Structure

Attribute name	Data type	Cardinality	Description
vnfInstanceId	Identifier	1	Identifier of the VNF instance which this grant request is related to. Shall also be provided for VNFs that not yet exist but are planned to exist in the future, i.e. if the grant is requested for InstantiateVNF.
vnfLcmOpOccId	Identifier	1	The identifier of the VNF lifecycle management operation occurrence associated to the GrantRequest.
vnfdId	Identifier	1	Identifier of the VNFD that defines the VNF for which the LCM operation is to be granted.
flavourId	Identifier	0..1	Identifier of the VNF deployment flavour of the VNFD that defines the VNF for which the LCM operation is to be granted. Shall be provided when instantiating the VNF or changing the deployment flavour of the VNF instance.
operation	GrantedLcmOperationType	1	The lifecycle management operation for which granting is requested. See note 1.
isAutomaticInvocation	Boolean	1	Set to true if this VNF LCM operation occurrence has been triggered by an automated procedure inside the VNFM (i.e. ScaleVnf / ScaleVnfToLevel triggered by auto-scale, or HealVnf triggered by auto- heal). Set to false otherwise.
instantiationLevelId	Identifier	0..1	If operation=INSTANTIATE, the identifier of the instantiation level may be provided as an alternative way to define the resources to be added. This attribute shall only be used if operation=INSTANTIATE. See note 2.
addResources	ResourceDefinition	0..N	List of resource definitions in the VNFD for resources to be added by the LCM operation which is related to this grant request, with one entry per resource. See note 2.
tempResources	ResourceDefinition	0..N	List of resource definitions in the VNFD for resources to be temporarily instantiated during the runtime of the LCM operation which is related to this grant request, with one entry per resource. See note 3.

remove Resourc es	ResourceD efinition	0.. N	Provides the definitions of resources to be removed by the LCM operation which is related to this grant request, with one entry per resource.
updateR esource s	ResourceD efinition	0.. N	Provides the definitions of resources to be modified by the LCM operation which is related to this grant request, with one entry per resource.
placeme ntConstr aints	Placement Constraint	0.. N	Placement constraints that the VNFM may send to the NFVO in order to influence the resource placement decision. If sent, the NFVO shall take the constraints into consideration when making resource placement decisions, and shall reject the grant if they cannot be honoured. See note 4 and note 5.
vimCons traints	VimConstr aint	0.. N	Used by the VNFM to require that multiple resources are managed through the same VIM connection. If sent, the NFVO shall take the constraints into consideration when making VIM selection decisions, and shall reject the grant if they cannot be honoured. This attribute shall be supported if VNF- related Resource Management in direct mode is applicable. The applicability and further details of this attribute for indirect mode are left for future specification.
addition alParam s	KeyValueP airs	0.. 1	Additional parameters passed by the VNFM, specific to the VNF and the LCM operation.
_links	Structure (inlined)	1	Links to resources related to this request.
>vnfLcm OpOcc	Link	1	Related lifecycle management operation occurrence.
>vnfInst ance	Link	1	Related VNF instance.

NOTE 1: NOTE 2: NOTE 3:

NOTE 4: NOTE 5:

The VNF LCM operations CreateVnfIdentifier, DeleteVnfIdentifier, QueryVnf and ModifyVnfInformation can be executed by the VNFM without requesting granting.
If the granting request is for InstantiateVNF, either instantiationLevel or addResources shall be present.
The NFVO will assume that the VNFM will be responsible to both allocate and release the temporary resource during the runtime of the LCM operation. This means, the resource can be allocated and consumed after the "start" notification for the LCM operation is sent by the VNFM, and the resource will be released before the "result" notification of the VNF LCM operation is sent by the VNFM.
The affinity/anti-affinity rules defined in the VNFD , and the placement constraints in the GrantVnfLifecycleOperation as defined in this clause should be conflict-free. In case of conflicts, the placement constraints in the GrantVnfLifecycleOperation shall take precedence.
Passing constraints allows the VNFM or the lifecycle management scripts to influence resource placement decisions by the NFVO to ensure VNF properties such as performance or fault tolerance.

Grant Structure

Attribute name	Data type	Cardi nal ity	Description
id	Identifier	1	Identifier of the grant.
vnfInstanc eId	Identifier	1	Identifier of the related VNF instance.
vnfLcmOp OccId	Identifier	1	Identifier of the related VNF lifecycle management operation occurrence.
vimConne ctions	VimConnec tionInfo	0.. N	Provides information regarding VIM connections that are approved to be used by the VNFM to allocate resources, and provides parameters of these VIM connections. The VNFM shall update the " vimConnectionInfo" attribute of the "VnfInstance" structure by adding unknown entries received in this attribute. This attribute is not intended for the modification of VimConnection entries passed earlier; for that, the VnfInfoModificationRequest structure shall be used. This attribute shall only be supported when VNF-related Resource Management in direct mode is applicable. In direct mode, this parameter shall be absent if the VIM information was configured to the VNFM in another way, present otherwise. See note 1.
zones	ZoneInfo	0.. N	Identifies resource zones where the resources are approved to be allocated by the VNFM.
zoneGrou ps	ZoneGroup Info	0.. N	Information about groups of resource zones that are related and that the NFVO has chosen to fulfil a zoneGroup constraint in the GrantVnfLifecycleOperation request. This information confirms that the NFVO has honoured the zoneGroup constraints that were passed as part of "placementConstraints" in the GrantRequest.
computeR eservation Id	IdentifierIn Vim	0.. 1	Information that identifies a reservation applicable to the compute resource requirements of the corresponding grant request. See note 2.

networkRe servationI d	IdentifierIn Vim	0.. 1	Information that identifies a reservation applicable to the network resource requirements of the corresponding grant request. See note 2.
storageRe servationI d	IdentifierIn Vim	0.. 1	Information that identifies a reservation applicable to the storage resource requirements of the corresponding grant request. See note 2.
addResou rces	GrantInfo	0.. N	List of resources that are approved to be added, with one entry per resource.
tempReso urces	GrantInfo	0.. N	List of resources that are approved to be temporarily instantiated during the runtime of the lifecycle operation, with one entry per resource.
removeRe sources	GrantInfo	0.. N	List of resources that are approved to be removed, with one entry per resource.
updateRe sources	GrantInfo	0.. N	List of resources that are approved to be modified, with one entry per resource.
vimAssets	Structure (inlined)	0.. 1	Information about assets for the VNF that are managed by the NFVO in the VIM, such as software images and virtualised compute resource flavours. This attribute is not intended for the modification of vimAssets entries passed earlier. See note 3.
>compute Resource Flavours	VimCompu teResource Flavour	0.. N	Mappings between virtual compute descriptors defined in the VNFD and compute resource flavours managed in the VIM.
>softwareI mages	VimSoftwar eImage	0.. N	Mappings between software images defined in the VNFD and software images managed in the VIM.
extVirtualL inks	ExtVirtualLi nkData	0.. N	Information about external VLs to connect the VNF to. See note 5.
extManag edVirtualLi nks	ExtManage dVirtualLin kData	0.. N	Information about internal VLs that are managed by other entities than the VNFM. See note 4 and note 5.
additional Params	KeyValueP airs	0.. 1	Additional parameters passed by the NFVO, specific to the VNF and the LCM operation.
_links	Structure (inlined)	1	Links to resources related to this resource.
>self	Link	1	URI of this resource.
>vnfLcmO pOcc	Link	1	Related VNF lifecycle management operation occurrence.
>vnfInstan ce	Link	1	Related VNF instance.

NOTE 1: NOTE 2:

NOTE 3: NOTE 4:

NOTE 5:

This interface allows to signal the use of multiple VIMs per VNF. However, due to the partial support of this feature in the present release, it is recommended in the present document that the number of entries in the "vims" attribute in the Grant is not greater than 1.
At least one of (computeReservationId, networkReservationId, storageReservationId) shall be present when policy is GRANT_RESERVE_SINGLE and an applicable reservation exists. None of these shall be present otherwise.

Modification of VIM assets during the lifetime of a VNF instance is not necessary, since it is expected that all applicable assets have been on boarded into the VIM before the VNF is instantiated.
The indication of externally-managed internal VLs is needed in case networks have been pre- configured for use with certain VNFs, for instance to ensure that these networks have certain properties such as security or acceleration features, or to address particular network topologies. The present document assumes that externally-managed internal VLs are managed by the NFVO and created towards the VIM.

External and/or externally-managed internal VLs can be passed in VNF lifecycle management operation requests such as InstantiateVnf or ChangeVnfFlavor, and/or in the grant response. The NFVO may choose to override in the grant response external and/or externally-managed VL instances that have been passed previously in the associated VNF lifecycle management request, if the lifecycle management request has originated from the NFVO itself.

Specifying HPA Capability Requirements using SOL001 VNFD

The VNFD is used to describe VNF-specific HPA capability requirements that will be matched against the capabilities of the underlying hardware infrastructure resources.
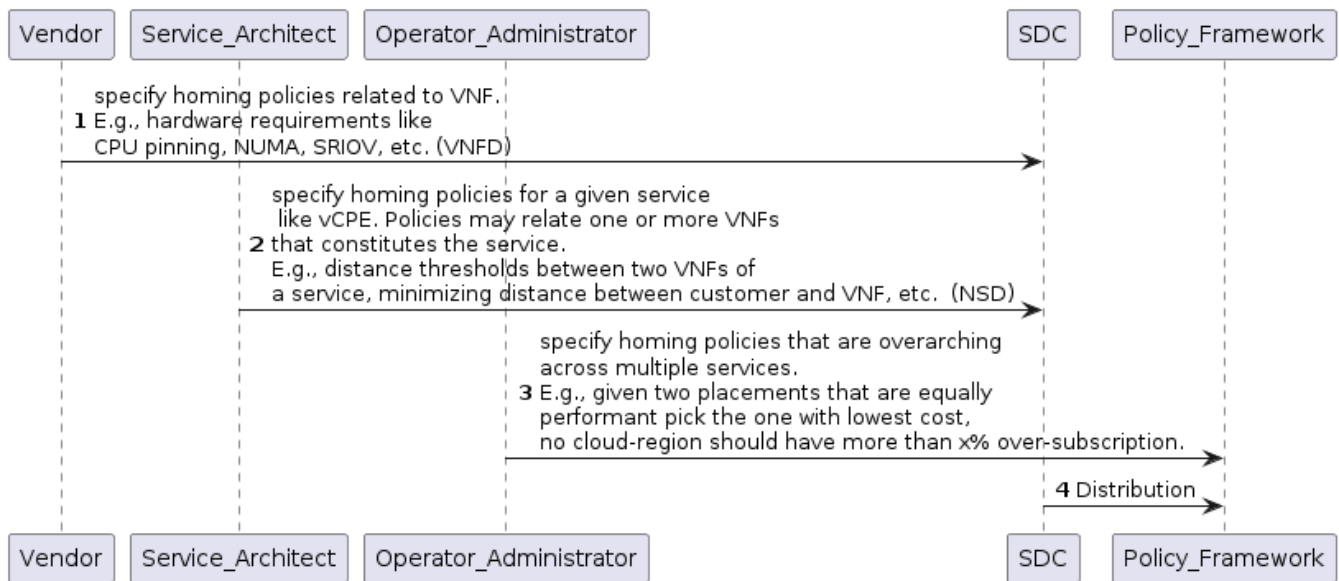
For HPA requirements, see Specifying HPA Capability Requirements using TOSCA-based VNF Descriptors.

<TBD>

ONAP Component Interactions for HPA

Homing policies may come from vendor, service architect and ONAP Operator/Administrator.


<describe further>

Query VNF (SOL003 Adapter_VNFM)

- SOL003 Adapter sends a query request for VNF instance(s) - multiple or selected VNF instance
  - GET .../vnf_instances   // multiple VNF instances
  - GET .../vnf_instances/{VnfInstnaceId}   // selected VNF instance
- SVNFM returns VNF Instance(s) in form of VnfInstance[] or VnfInstance
  - 200 OK with VnfInstance[]
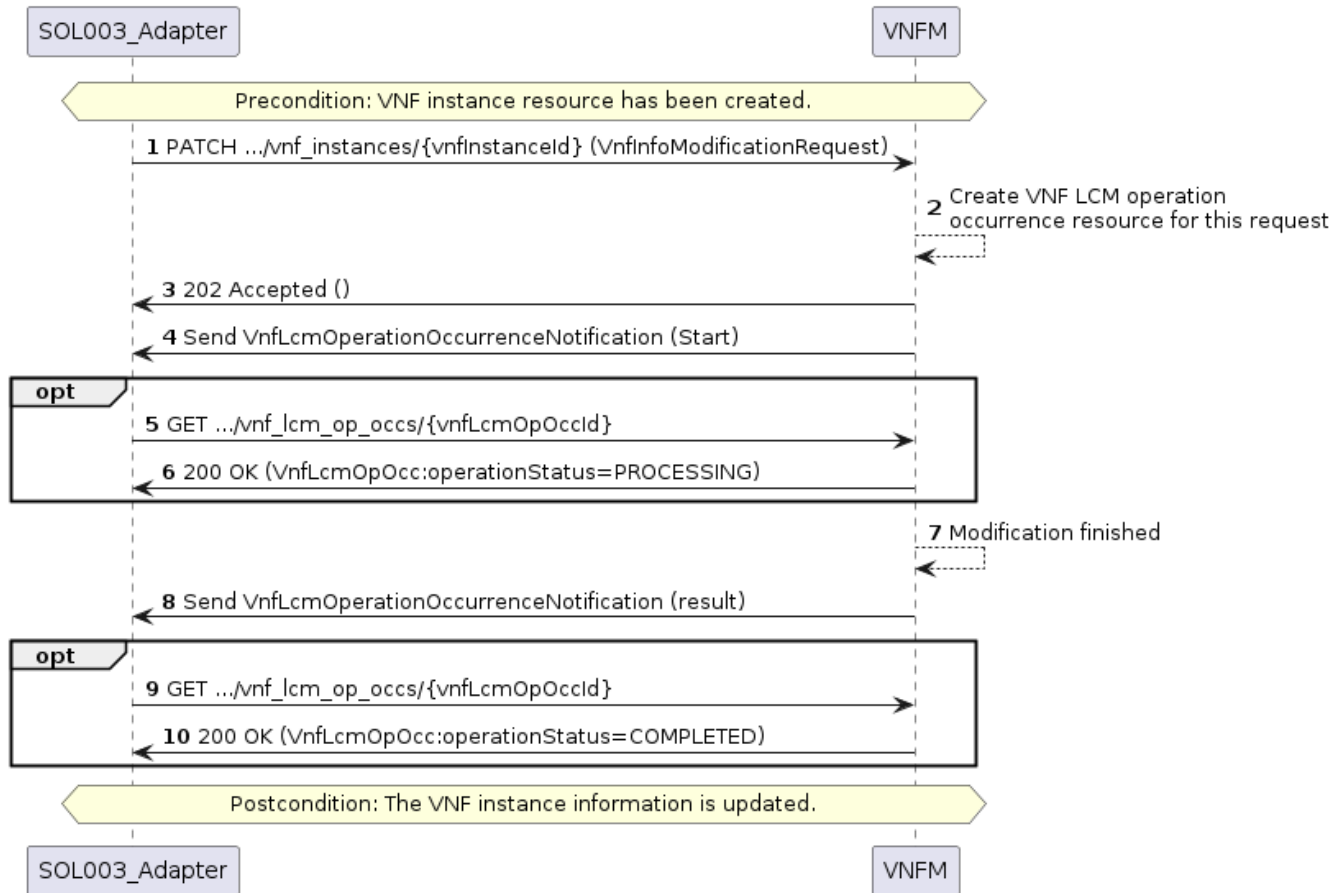 - 200 OK with VnfInstance

Precondition: VNF instance is created

Postcondition: SOL003 Adapter gets VnfInstance[] or VnfInstance
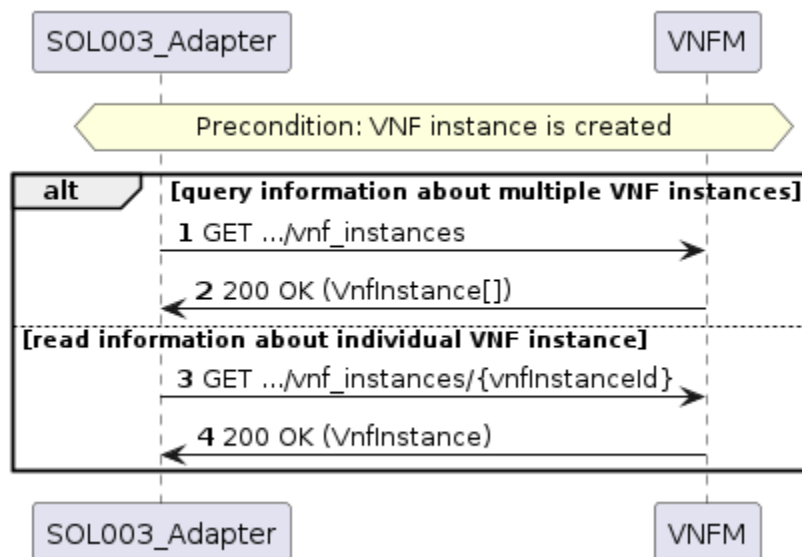
Modify VNF (SOL003 Adapter VNFM)

The SOL003 Adapter requests the Modify VNF to the VNFM.

Note: it is out of Frankfurt scope.

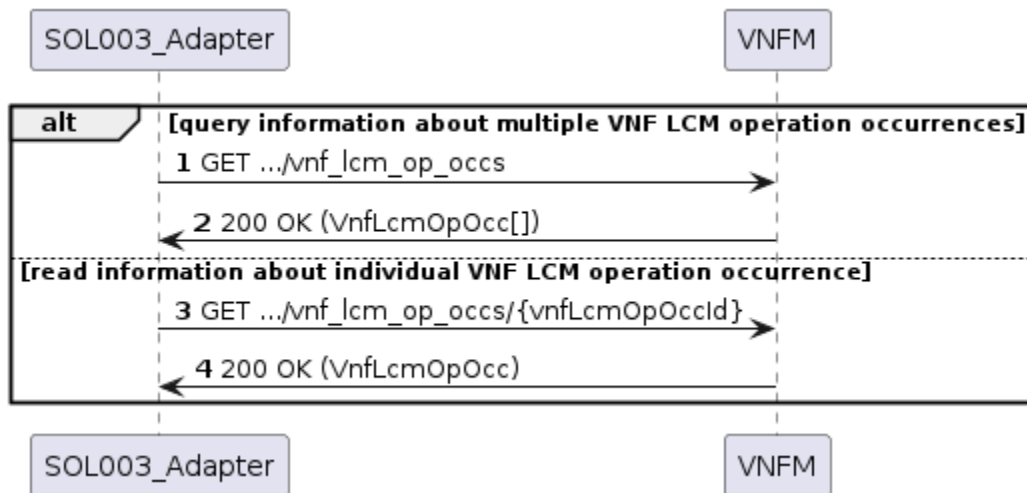## Get Operation Status (SOL003 Adapter VNFM)

The SOL003 Adapter requests the Get Operation Status operation. The following diagram depicts a sequence for obtaining the status of a VNF lifecycle management operation occurrence.

SOL003 Adapter supports VNF Operation Status

- SOL003 Adapter sends a request for VNF LCM operation occurrence(s) to the SVNFM
  - GET .../vnf_lcm_op_occs   // multiple
  - GET .../vnf_lcm_op_occs/{vnfLcmOpOccId}   // single
- VNFM returns VNF LCM operation occurrence(s) to the SOL003 Adapter
  - 200 OK (VnfLcmOpOcc[])
  - 200 OK (VnfLcmOpOcc)
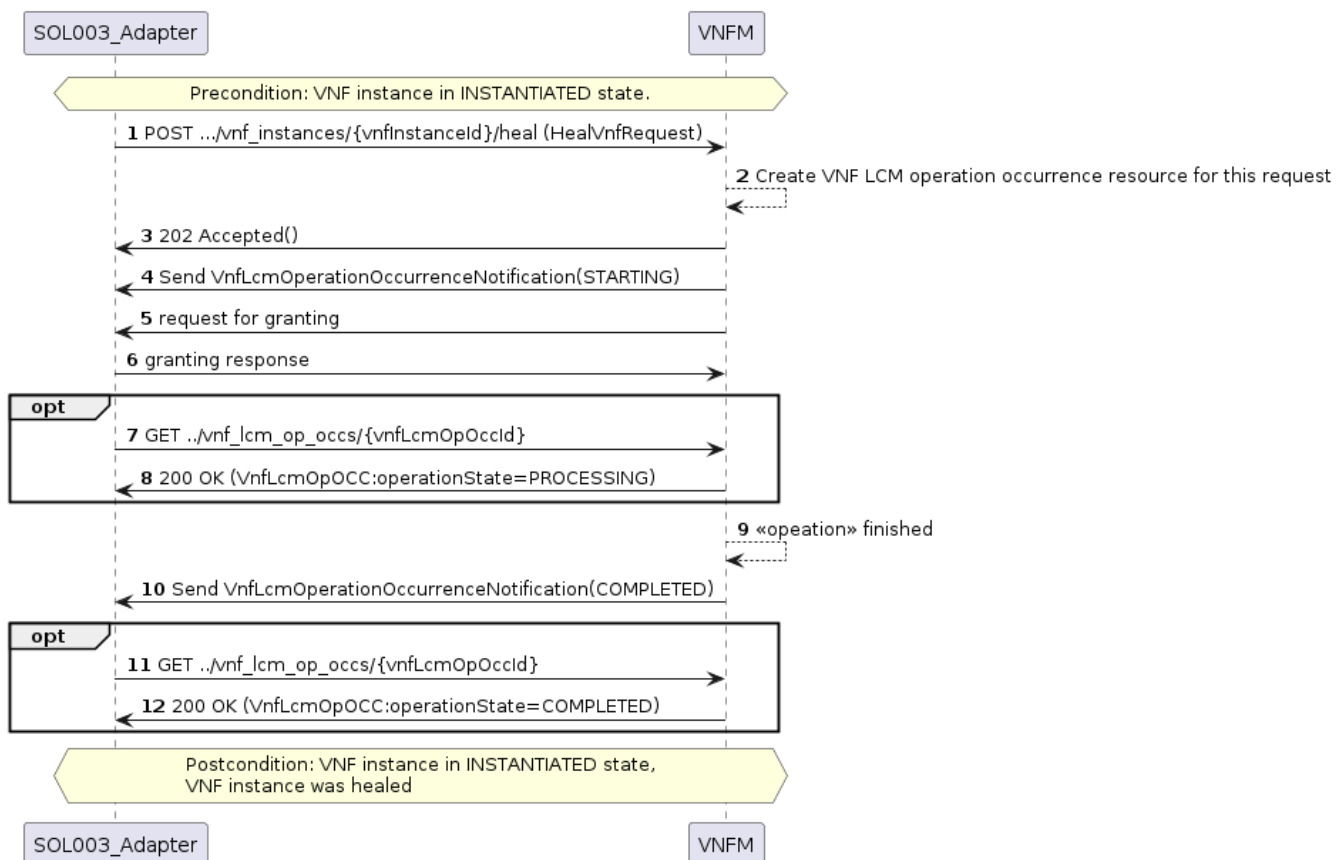

Precondition: SVNFM performed LCM operations

Postcondition: SOL003 Adapter receives VNF LCM operation occurrence(s)

Heal VNF (SOL003 Adapter VNFM)

The SOL003 Adapter requests the Heal VNF operation. The following diagram depicts a sequence for the Heal VNF operation.

Note: this is out of Frankfurt scope.

SO Workflows and Java Code (client) Enhancement

SO BPMN Workflows and associated Java code for SOL003 operations will be enhanced and created by leveraging the SOL003 Adapter NBIs.

Enhance Create/Instantiate Workflows & Associated Java Code

- Modify the existing workflow to separate Create and Instantiate tasks
- Invoke the modified SOL003 Adapter NBIs

Precondition: The generic VNF has been added in AAI. The VNF package has been distributed from SDC. The VNFM and VIM have been defined in AAI. VNFM simulator deployed as VNFM. SDNC preload completed through SDNC access site. Add the ETSI "Create_VNF" and "Terminate_VNF" building blocks to the "building_block_detail" table in MariaDB's "catalogdb". Edit the "orchestration_status_state_transition_directive" table in MariaDB's "catalogdb" to allow a service with "operationStatus" set to "CREATED" to allow a building block with a "TARGET_ACTION" of "ACTIVATE" to "CONTINUE".

Postcondition: Create and instantiate requests were correctly sent to the VNFM, the grant request from the VNFM was handled and reply sent to the VNFM and AAI was updated in accordance with the notifications received from the VNFM as a result of the VNF being instantiated.

Enhance Terminate/Delete Workflows & Associated Java Code

- Modify the existing workflow to separate Terminate and Delete tasks
- Invoke the modified SOL003 Adapter NBIs

Precondition: The VNF has been created via the VNFM adapter

Postcondition: Terminate and delete requests were correctly sent to the VNFM, the grant request from the VNFM was handled and reply sent to the VNFM and AAI was updated in accordance with the notifications received from the VNFM as a result of the VNF being terminated.

Create Query VNF Workflows & Associated Java Code

- Create a new workflow for Query VNF and associated Java code
- Invoke the modified SOL003 Adapter Query VNF NBI

Precondition: VNF is created previously

Postcondition: Query VNF workflow get VNF info

Create Get Operation Status Workflows & Associated Java code

- Create a new Get Operation Status Workflow & Associated Java code
- Invoke the modified SOL003 Adapter Get Operation Status NBI
- This operation could be used by the Instantiation and Termination of VNF; in this case, its workflow could be omitted

Precondition: VNF instantiation or terminate is started

Postcondition: the workflow or the SOL003 Adapter client gets the VnF operation status.

Authentication and Authorization for the SOL003 Adapter and the VNFM