



Network Functions Virtualisation (NFV) Release 2; Protocols and Data Models; RESTful protocols specification for the Os-Ma-nfvo Reference Point

Disclaimer

The present document has been produced and approved by the Network Functions Virtualisation (NFV) ETSI Industry Specification Group (ISG) and represents the views of those members who participated in this ISG.
It does not necessarily represent the views of the entire ETSI membership.

Reference

RGS/NFV-SOL005ed281

Keywords

API, NFV, protocol

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

The present document can be downloaded from:

<http://www.etsi.org/standards-search>

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the prevailing version of an ETSI deliverable is the one made publicly available in PDF format at www.etsi.org/deliver.

Users of the present document should be aware that the document may be subject to revision or change of status.

Information on the current status of this and other ETSI documents is available at

<https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx>

If you find errors in the present document, please send your comment to one of the following services:

<https://portal.etsi.org/People/CommitteeSupportStaff.aspx>

Copyright Notification

No part may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm except as authorized by written permission of ETSI.

The content of the PDF version shall not be modified without the written authorization of ETSI.

The copyright and the foregoing restriction extend to reproduction in all media.

© ETSI 2020.

All rights reserved.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members.

3GPP™ and **LTE™** are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners.

oneM2M™ logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners.

GSM® and the GSM logo are trademarks registered and owned by the GSM Association.

Contents

Intellectual Property Rights	18
Foreword.....	18
Modal verbs terminology.....	18
1 Scope	19
2 References	19
2.1 Normative references	19
2.2 Informative references.....	20
3 Definition of terms, symbols and abbreviations.....	21
3.1 Terms.....	21
3.2 Symbols.....	21
3.3 Abbreviations	21
4 General Aspects.....	22
4.1 Overview	22
4.2 Void.....	23
4.3 Void.....	23
4.4 Common data types	23
4.4.1 Structured data types.....	23
4.4.1.1 Introduction.....	23
4.4.1.2 Void.....	23
4.4.1.3 Void.....	23
4.4.1.3a Void.....	23
4.4.1.4 Void.....	23
4.4.1.5 Type: NsInstanceSubscriptionFilter	23
4.4.1.6 Type: ResourceHandle	24
4.4.1.7 Void.....	24
4.4.2 Simple data types and enumerations	25
4.4.2.1 Introduction.....	25
4.4.2.2 Simple data types	25
4.4.2.3 Enumerations	25
4.5 Void.....	25
4.6 Void.....	25
4.7 Void.....	25
5 NSD Management interface	25
5.1 Description	25
5.1a API version.....	26
5.2 Resource structure and methods	26
5.3 Sequence diagrams (informative).....	28
5.3.1 Flow of the creation of an individual NS descriptor resource.....	28
5.3.2 Flow of the uploading of NSD archive content	29
5.3.3 Flow of the fetching of an onboarded NSD archive	30
5.3.4 Flow of the update of an individual NS descriptor resource.....	31
5.3.5 Flow of the deletion of an individual NS descriptor resource.....	32
5.3.6 Flow of the querying/reading of NS descriptor information.....	33
5.3.6a Flow of reading the NSD	34
5.3.6b Flow of fetching the manifest file of an onboarded NSD archive.....	34
5.3.7 Flow of the creation of an individual PNF descriptor resource	35
5.3.8 Flow of the uploading of PNFD archive	35
5.3.9 Flow of the fetching of an onboarded PNFD archive	36
5.3.10 Flow of the deletion of an individual PNF descriptor resource	37
5.3.11 Flow of the querying/reading of PNF descriptor information.....	37
5.3.11a Flow of reading the PNFD	39
5.3.11b Flow of fetching the manifest file of an onboarded PNFD archive	39
5.3.12 Flow of managing subscriptions	39
5.3.13 Flow of sending notifications.....	41

5.4	Resources	42
5.4.1	Introduction.....	42
5.4.1a	Resource: API versions.....	42
5.4.2	Resource: NS Descriptors	42
5.4.2.1	Description	42
5.4.2.2	Resource definition	42
5.4.2.3	Resource methods	43
5.4.2.3.1	POST	43
5.4.2.3.2	GET	43
5.4.2.3.3	PUT	44
5.4.2.3.4	PATCH.....	45
5.4.2.3.5	DELETE.....	45
5.4.3	Resource: Individual NS Descriptor	45
5.4.3.1	Description.....	45
5.4.3.2	Resource definition	45
5.4.3.3	Resource methods	45
5.4.3.3.1	POST	45
5.4.3.3.2	GET	45
5.4.3.3.3	PUT	46
5.4.3.3.4	PATCH.....	46
5.4.3.3.5	DELETE.....	47
5.4.4	Resource: NSD archive content	48
5.4.4.1	Description	48
5.4.4.2	Resource definition	48
5.4.4.3	Resource methods	49
5.4.4.3.1	POST	49
5.4.4.3.2	GET	49
5.4.4.3.3	PUT	50
5.4.4.3.4	PATCH.....	51
5.4.4.3.5	DELETE.....	51
5.4.4a	Resource: NSD	51
5.4.4a.1	Description.....	51
5.4.4a.2	Resource definition	52
5.4.4a.3	Resource methods	52
5.4.4a.3.1	POST	52
5.4.4a.3.2	GET	52
5.4.4a.3.3	PUT	54
5.4.4a.3.4	PATCH.....	54
5.4.4a.3.5	DELETE.....	54
5.4.4b	Resource: NSD archive manifest	54
5.4.4b.1	Description	54
5.4.4b.2	Resource definition	54
5.4.4b.3	Resource methods	54
5.4.4b.3.1	POST	54
5.4.4b.3.2	GET	54
5.4.4b.3.3	PUT	56
5.4.4b.3.4	PATCH.....	56
5.4.4b.3.5	DELETE.....	56
5.4.5	Resource: PNF Descriptors.....	56
5.4.5.1	Description	56
5.4.5.2	Resource definition	56
5.4.5.3	Resource methods	56
5.4.5.3.1	POST	56
5.4.5.3.2	GET	57
5.4.5.3.3	PUT	58
5.4.5.3.4	PATCH.....	58
5.4.5.3.5	DELETE.....	58
5.4.6	Resource: Individual PNF Descriptor	59
5.4.6.1	Description.....	59
5.4.6.2	Resource definition	59
5.4.6.3	Resource methods	59
5.4.6.3.1	POST	59

5.4.6.3.2	GET	59
5.4.6.3.3	PUT	60
5.4.6.3.4	PATCH.....	60
5.4.6.3.5	DELETE.....	60
5.4.7	Resource: PNFD archive content.....	61
5.4.7.1	Description.....	61
5.4.7.2	Resource definition	61
5.4.7.3	Resource methods	62
5.4.7.3.1	POST	62
5.4.7.3.2	GET	62
5.4.7.3.3	PUT	63
5.4.7.3.4	PATCH.....	64
5.4.7.3.5	DELETE.....	64
5.4.7a	Resource: PNFD	64
5.4.7a.1	Description.....	64
5.4.7a.2	Resource definition	65
5.4.7a.3	Resource methods	65
5.4.7a.3.1	POST	65
5.4.7a.3.2	GET	65
5.4.7a.3.3	PUT	66
5.4.7a.3.4	PATCH.....	67
5.4.7a.3.5	DELETE.....	67
5.4.7b	Resource: PNFD archive manifest.....	67
5.4.7b.1	Description.....	67
5.4.7b.2	Resource definition	67
5.4.7b.3	Resource methods	67
5.4.7b.3.1	POST	67
5.4.7b.3.2	GET	67
5.4.7b.3.3	PUT	69
5.4.7b.3.4	PATCH.....	69
5.4.7b.3.5	DELETE.....	69
5.4.8	Resource: Subscriptions.....	69
5.4.8.1	Description.....	69
5.4.8.2	Resource definition	69
5.4.8.3	Resource methods	69
5.4.8.3.1	POST	69
5.4.8.3.2	GET	70
5.4.8.3.3	PUT	71
5.4.8.3.4	PATCH.....	72
5.4.8.3.5	DELETE.....	72
5.4.9	Resource: Individual subscription.....	72
5.4.9.1	Description.....	72
5.4.9.2	Resource definition	72
5.4.9.3	Resource methods	72
5.4.9.3.1	POST	72
5.4.9.3.2	GET	72
5.4.9.3.3	PUT	73
5.4.9.3.4	PATCH.....	73
5.4.9.3.5	DELETE.....	73
5.4.10	Resource: Notification endpoint	73
5.4.10.1	Description.....	73
5.4.10.2	Resource definition	74
5.4.10.3	Resource methods	74
5.4.10.3.1	POST	74
5.4.10.3.2	GET	74
5.4.10.3.3	PUT	75
5.4.10.3.4	PATCH.....	75
5.4.10.3.5	DELETE.....	75
5.5	Data model	75
5.5.1	Introduction.....	75
5.5.2	Resource and notification data types	75
5.5.2.1	Type: NsdInfoModifications	75

5.5.2.2	Type: NsdInfo	76
5.5.2.3	Type: CreateNsdInfoRequest	77
5.5.2.4	Type: PnfdInfoModifications	77
5.5.2.5	Type: PnfdInfo	78
5.5.2.6	Type: CreatePnfdInfoRequest	79
5.5.2.7	Type: NsdmSubscriptionRequest	79
5.5.2.8	Type: NsdmSubscription	79
5.5.2.9	Type: NsdOnboardingNotification	79
5.5.2.10	Type: NsdOnboardingFailureNotification	80
5.5.2.11	Type: NsdChangeNotification	80
5.5.2.12	Type: NsdDeletionNotification	81
5.5.2.13	Type: PnfdOnboardingNotification	81
5.5.2.14	Type: PnfdOnboardingFailureNotification	82
5.5.2.15	Type: PnfdDeletionNotification	82
5.5.3	Referenced structured data types	83
5.5.3.1	Introduction	83
5.5.3.2	Type: NsdmNotificationsFilter	83
5.5.3.3	Type: NsdmLinks	84
5.5.3.4	Type: PnfdmLinks	85
5.5.3.5	Type: NsdArchiveArtifactInfo	85
5.5.3.6	Type: PnfdArchiveArtifactInfo	85
5.5.4	Referenced simple data types and enumerations	86
5.5.4.1	Introduction	86
5.5.4.2	Simple data types	86
5.5.4.3	Enumeration: NsdOperationalStateType	86
5.5.4.4	Enumeration: NsdUsageStateType	86
5.5.4.5	Enumeration: NsdOnboardingStateType	86
5.5.4.6	Enumeration: PnfdOnboardingStateType	86
5.5.4.7	Enumeration: PnfdUsageStateType	87
6	NS Lifecycle Management interface	87
6.1	Description	87
6.1a	API version	88
6.2	Resource structure and methods	88
6.3	Sequence diagrams (informative)	90
6.3.1	Flow of the creation of an NS instance resource	90
6.3.2	Flow of the deletion of an NS instance resource	90
6.3.3	Flow of NS lifecycle management operations triggered by task resources	91
6.3.4	Flow of the get operations status operation	93
6.3.5	Flow of managing subscriptions	94
6.3.6	Flow of sending notifications	95
6.3.7	Flow of retrying an NS lifecycle management operation	96
6.3.8	Flow of rolling back an NS lifecycle management operation	97
6.3.9	Flow of continuing an NS lifecycle management operation	99
6.3.10	Flow of failing an NS lifecycle management operation	100
6.3.11	Flow of cancelling an NS lifecycle management operation	101
6.4	Resources	102
6.4.1	Introduction	102
6.4.1.1	Overview	102
6.4.1.2	Task resources that trigger NS LCM operations	102
6.4.1a	Resource: API versions	103
6.4.2	Resource: NS Instances	103
6.4.2.1	Description	103
6.4.2.2	Resource definition	103
6.4.2.3	Resource methods	104
6.4.2.3.1	POST	104
6.4.2.3.2	GET	104
6.4.2.3.3	PUT	106
6.4.2.3.4	PATCH	106
6.4.2.3.5	DELETE	107
6.4.3	Resource: Individual NS Instance	107
6.4.3.1	Description	107

6.4.3.2	Resource definition	107
6.4.3.3	Resource methods	107
6.4.3.3.1	POST	107
6.4.3.3.2	GET	107
6.4.3.3.3	PUT	108
6.4.3.3.4	PATCH.....	108
6.4.3.3.5	DELETE.....	108
6.4.4	Resource: Instantiate NS task	109
6.4.4.1	Description	109
6.4.4.2	Resource definition	109
6.4.4.3	Resource methods	109
6.4.4.3.1	POST	109
6.4.4.3.2	GET	110
6.4.4.3.3	PUT	110
6.4.4.3.4	PATCH.....	110
6.4.4.3.5	DELETE.....	110
6.4.5	Resource: Scale NS task	111
6.4.5.1	Description	111
6.4.5.2	Resource definition	111
6.4.5.3	Resource methods	111
6.4.5.3.1	POST	111
6.4.5.3.2	GET	112
6.4.5.3.3	PUT	112
6.4.5.3.4	PATCH.....	112
6.4.5.3.5	DELETE.....	112
6.4.6	Resource: Update NS task	113
6.4.6.1	Description	113
6.4.6.2	Resource definition	113
6.4.6.3	Resource methods	113
6.4.6.3.1	POST	113
6.4.6.3.2	GET	114
6.4.6.3.3	PUT	114
6.4.6.3.4	PATCH.....	114
6.4.6.3.5	DELETE.....	114
6.4.7	Resource: Heal NS task	114
6.4.7.1	Description	114
6.4.7.2	Resource definition	115
6.4.7.3	Resource methods	115
6.4.7.3.1	POST	115
6.4.7.3.2	GET	116
6.4.7.3.3	PUT	116
6.4.7.3.4	PATCH.....	116
6.4.7.3.5	DELETE.....	116
6.4.8	Resource: Terminate NS task.....	116
6.4.8.1	Description	116
6.4.8.2	Resource definition	116
6.4.8.3	Resource methods	116
6.4.8.3.1	POST	116
6.4.8.3.2	GET	117
6.4.8.3.3	PUT	117
6.4.8.3.4	PATCH.....	117
6.4.8.3.5	DELETE.....	117
6.4.9	Resource: NS LCM operation occurrences.....	118
6.4.9.1	Description	118
6.4.9.2	Resource definition	118
6.4.9.3	Resource methods	118
6.4.9.3.1	POST	118
6.4.9.3.2	GET	118
6.4.9.3.3	PUT	119
6.4.9.3.4	PATCH.....	120
6.4.9.3.5	DELETE.....	120
6.4.10	Resource: Individual NS LCM operation occurrence.....	120

6.4.10.1	Description	120
6.4.10.2	Resource definition	120
6.4.10.3	Resource methods	120
6.4.10.3.1	POST	120
6.4.10.3.2	GET	120
6.4.10.3.3	PUT	121
6.4.10.3.4	PATCH	121
6.4.10.3.5	DELETE	121
6.4.11	Resource: Retry operation task	121
6.4.11.1	Description	121
6.4.11.2	Resource definition	121
6.4.11.3	Resource methods	122
6.4.11.3.1	POST	122
6.4.11.3.2	GET	123
6.4.11.3.3	PUT	123
6.4.11.3.4	PATCH	123
6.4.11.3.5	DELETE	123
6.4.12	Resource: Rollback operation task	123
6.4.12.1	Description	123
6.4.12.2	Resource definition	123
6.4.12.3	Resource methods	123
6.4.12.3.1	POST	123
6.4.12.3.2	GET	124
6.4.12.3.3	PUT	124
6.4.12.3.4	PATCH	125
6.4.12.3.5	DELETE	125
6.4.13	Resource: Continue operation task	125
6.4.13.1	Description	125
6.4.13.2	Resource definition	125
6.4.13.3	Resource methods	125
6.4.13.3.1	POST	125
6.4.13.3.2	GET	126
6.4.13.3.3	PUT	126
6.4.13.3.4	PATCH	126
6.4.13.3.5	DELETE	127
6.4.14	Resource: Fail operation task	127
6.4.14.1	Description	127
6.4.14.2	Resource definition	127
6.4.14.3	Resource methods	127
6.4.14.3.1	POST	127
6.4.14.3.2	GET	128
6.4.14.3.3	PUT	128
6.4.14.3.4	PATCH	128
6.4.14.3.5	DELETE	129
6.4.15	Resource: Cancel operation task	129
6.4.15.1	Description	129
6.4.15.2	Resource definition	129
6.4.15.3	Resource methods	129
6.4.15.3.1	POST	129
6.4.15.3.2	GET	130
6.4.15.3.3	PUT	130
6.4.15.3.4	PATCH	131
6.4.15.3.5	DELETE	131
6.4.16	Resource: Subscriptions	131
6.4.16.1	Description	131
6.4.16.2	Resource definition	131
6.4.16.3	Resource methods	131
6.4.16.3.1	POST	131
6.4.16.3.2	GET	132
6.4.16.3.3	PUT	133
6.4.16.3.4	PATCH	134
6.4.16.3.5	DELETE	134

6.4.17	Resource: Individual subscription.....	134
6.4.17.1	Description	134
6.4.17.2	Resource definition	134
6.4.17.3	Resource methods	134
6.4.17.3.1	POST	134
6.4.17.3.2	GET	134
6.4.17.3.3	PUT	135
6.4.17.3.4	PATCH.....	135
6.4.17.3.5	DELETE.....	135
6.4.18	Resource: Notification endpoint	136
6.4.18.1	Description	136
6.4.18.2	Resource definition	136
6.4.18.3	Resource methods	136
6.4.18.3.1	POST	136
6.4.18.3.2	GET	137
6.4.18.3.3	PUT	137
6.4.18.3.4	PATCH.....	137
6.4.18.3.5	DELETE.....	137
6.5	Data model	137
6.5.1	Introduction.....	137
6.5.2	Resource and notification data types	137
6.5.2.1	Introduction	137
6.5.2.2	Type: LccnSubscriptionRequest	138
6.5.2.3	Type: NsLcmOpOcc	138
6.5.2.4	Type: LccnSubscription	140
6.5.2.5	Type: NsLcmOperationOccurrenceNotification	140
6.5.2.6	Type: NsIdentifierCreationNotification	141
6.5.2.7	Type: NsIdentifierDeletionNotification	142
6.5.2.8	Type: NsChangeNotification.....	142
6.5.2.9	Type: CreateNsRequest.....	143
6.5.2.10	Type: NsInstance.....	144
6.5.2.11	Type: InstantiateNsRequest.....	145
6.5.2.12	Type: UpdateNsRequest.....	146
6.5.2.13	Type: HealNsRequest.....	148
6.5.2.14	Type: ScaleNsRequest	148
6.5.2.15	Type: TerminateNsRequest.....	148
6.5.2.16	Type: CancelMode.....	148
6.5.3	Referenced structured data types	149
6.5.3.1	Introduction	149
6.5.3.2	Type: AffectedVnf	149
6.5.3.3	Type: AffectedPnf	150
6.5.3.4	Type: AffectedVirtualLink.....	150
6.5.3.5	Type: AffectedVnffg.....	151
6.5.3.6	Type: AffectedNs	151
6.5.3.7	Type: AffectedSap	152
6.5.3.8	Type: LifecycleChangeNotificationsFilter.....	152
6.5.3.9	Type: LccnLinks	153
6.5.3.10	Type: SapData.....	153
6.5.3.11	Type: CpProtocolData.....	154
6.5.3.12	Type: IpOverEthernetAddressData	154
6.5.3.13	Type: PnfInfo	155
6.5.3.14	Type: AddPnfData	155
6.5.3.15	Type: ModifyPnfData	156
6.5.3.16	Type: PnfExtCpData	156
6.5.3.17	Type: PnfExtCpInfo	156
6.5.3.18	Type: IpOverEthernetAddressInfo	156
6.5.3.19	Type: VnfInstanceData	157
6.5.3.19a	Type: NestedNsInstanceData	157
6.5.3.20	Type: VnfLocationConstraint	158
6.5.3.21	Type: LocationConstraints	158
6.5.3.21a	Type: ParamsForNestedNs.....	158
6.5.3.22	Type: ParamsForVnf.....	158

6.5.3.23	Type: AffinityOrAntiAffinityRule	159
6.5.3.24	Type: InstantiateVnfData	160
6.5.3.25	Type: ChangeVnfFlavourData	160
6.5.3.26	Type: ExtVirtualLinkData	161
6.5.3.27	Type: ExtManagedVirtualLinkData	161
6.5.3.28	Type: ExtLinkPortData	162
6.5.3.29	Type: VnfExtCpData	162
6.5.3.30	Type: VnfExtCpConfig	162
6.5.3.31	Type: OperateVnfData	163
6.5.3.32	Type: ModifyVnfInfoData	163
6.5.3.33	Type: ChangeExtVnfConnectivityData	164
6.5.3.34	Type: AssocNewNsdVersionData	164
6.5.3.35	Type: MoveVnfInstanceData	165
6.5.3.36	Type: AddVnffgData	165
6.5.3.37	Type: UpdateVnffgData	165
6.5.3.38	Type: NfpData	166
6.5.3.39	Type: ChangeNsFlavourData	166
6.5.3.40	Type: NfpRule	166
6.5.3.41	Type: Mask	167
6.5.3.42	Type: PortRange	168
6.5.3.43	Type: HealNsData	168
6.5.3.44	Type: HealVnfData	168
6.5.3.45	Type: ScaleNsData	169
6.5.3.46	Type: ScaleNsByStepsData	169
6.5.3.47	Type: ScaleNsToLevelData	170
6.5.3.48	Type: NsScaleInfo	170
6.5.3.49	Type: ScaleVnfData	170
6.5.3.50	Type: ScaleToLevelData	171
6.5.3.51	Type: VnfScaleInfo	171
6.5.3.52	Type: ScaleByStepData	171
6.5.3.53	Type: NsVirtualLinkInfo	172
6.5.3.54	Type: VirtualStorageResourceInfo	172
6.5.3.55	Type: NsLinkPortInfo	172
6.5.3.56	Type: NsCpHandle	172
6.5.3.57	Type: VnfInstance	173
6.5.3.58	Type: CpProtocolInfo	176
6.5.3.59	Type: ExtManagedVirtualLinkInfo	176
6.5.3.60	Type: VnfcResourceInfo	177
6.5.3.61	Type: VnfVirtualLinkResourceInfo	177
6.5.3.62	Type: ExtVirtualLinkInfo	178
6.5.3.63	Type: ExtLinkPortInfo	178
6.5.3.64	Type: VnfLinkPortInfo	178
6.5.3.65	Type: VnffgInfo	179
6.5.3.66	Type: NfpInfo	180
6.5.3.67	Type: SapInfo	180
6.5.3.68	Type: NsMonitoringParameter	180
6.5.3.69	Type: VnfMonitoringParameter	181
6.5.3.70	Type: VnfExtCpInfo	181
6.5.3.71	Type: CpGroupInfo	181
6.5.3.72	Type: CpPairInfo	182
6.5.3.73	Type: ForwardingBehaviourInputParameters	182
6.5.4	Referenced simple data types and enumerations	183
6.5.4.1	Introduction	183
6.5.4.2	Simple data types	183
6.5.4.3	Enumeration: NsLcmOpType	183
6.5.4.4	Enumeration: NsLcmOperationStateType	183
6.5.4.5	Enumeration: NsComponentType	184
6.5.4.6	Enumeration: LcmOpNameForChangeNotificationType	184
6.5.4.7	Enumeration: LcmOpOccStatusForChangeNotificationType	184
6.5.4.8	Enumeration: OperationalStates	185
6.5.4.9	Enumeration: StopType	185
6.5.4.10	Enumeration: CancelModeType	185

6.6	Handling of errors during NS lifecycle management operations.....	186
6.6.1	Basic concepts (informative)	186
6.6.1.1	Motivation	186
6.6.1.2	Failure resolution strategies: Retry, Rollback and Continue	186
6.6.1.3	Error handling at NFVO and OSS/BSS	186
6.6.2	States and state transitions of an NS lifecycle management operation occurrence.....	188
6.6.2.1	General	188
6.6.2.2	States of an NS lifecycle management operation occurrence.....	188
6.6.2.3	Error handling operations that change the state of an NS lifecycle operation.....	191
6.6.3	Detailed flows	191
6.6.3.1	Immediate failure	191
6.6.3.2	Failure during actual NS LCM operation execution	192
6.6.3.3	LCM operation cancellation.....	193
7	NS Performance Management interface.....	194
7.1	Description	194
7.1a	API version.....	194
7.2	Resource structure and methods.....	194
7.3	Sequence diagrams (informative).....	195
7.3.1	Flow of creating a PM job	195
7.3.1a	Flow of updating the callback URI of a PM job	196
7.3.2	Flow of querying/reading PM jobs	197
7.3.3	Flow of deleting a PM job	198
7.3.4	Flow of obtaining performance reports.....	199
7.3.5	Flow of creating a threshold	200
7.3.5a	Flow of updating the callback URI of a threshold	200
7.3.6	Flow of querying/reading thresholds	201
7.3.7	Flow of deleting thresholds.....	202
7.3.8	Void	203
7.3.9	Flow of sending notifications.....	203
7.4	Resources	203
7.4.1	Introduction.....	203
7.4.1a	Resource: API versions.....	203
7.4.2	Resource: PM jobs	204
7.4.2.1	Description	204
7.4.2.2	Resource definition	204
7.4.2.3	Resource methods	204
7.4.2.3.1	POST	204
7.4.2.3.2	GET	205
7.4.2.3.3	PUT	207
7.4.2.3.4	PATCH.....	207
7.4.2.3.5	DELETE.....	207
7.4.3	Resource: Individual PM job	207
7.4.3.1	Description	207
7.4.3.2	Resource definition	207
7.4.3.3	Resource methods	207
7.4.3.3.1	POST	207
7.4.3.3.2	GET	207
7.4.3.3.3	PUT	208
7.4.3.3.4	PATCH.....	208
7.4.3.3.5	DELETE.....	209
7.4.4	Resource: Individual performance report	209
7.4.4.1	Description	209
7.4.4.2	Resource definition	210
7.4.4.3	Resource methods	210
7.4.4.3.1	POST	210
7.4.4.3.2	GET	210
7.4.4.3.3	PUT	210
7.4.4.3.4	PATCH.....	211
7.4.4.3.5	DELETE.....	211
7.4.5	Resource: Thresholds.....	211
7.4.5.1	Description	211

7.4.5.2	Resource definition	211
7.4.5.3	Resource methods	211
7.4.5.3.1	POST	211
7.4.5.3.2	GET	212
7.4.5.3.3	PUT	213
7.4.5.3.4	PATCH.....	213
7.4.5.3.5	DELETE.....	213
7.4.6	Resource: Individual threshold	214
7.4.6.1	Description	214
7.4.6.2	Resource definition	214
7.4.6.3	Resource methods	214
7.4.6.3.1	POST	214
7.4.6.3.2	GET	214
7.4.6.3.3	PUT	215
7.4.6.3.4	PATCH.....	215
7.4.6.3.5	DELETE.....	216
7.4.7	Void	216
7.4.8	Void	216
7.4.9	Resource: Notification endpoint	216
7.4.9.1	Description	216
7.4.9.2	Resource definition	216
7.4.9.3	Resource methods	217
7.4.9.3.1	POST	217
7.4.9.3.2	GET	217
7.4.9.3.3	PUT	218
7.4.9.3.4	PATCH.....	218
7.4.9.3.5	DELETE.....	218
7.5	Data Model.....	218
7.5.1	Introduction.....	218
7.5.2	Resource and notification data types	218
7.5.2.1	Introduction.....	218
7.5.2.2	Void.....	218
7.5.2.3	Void.....	218
7.5.2.4	Type: ThresholdCrossedNotification	218
7.5.2.5	Type: PerformanceInformationAvailableNotification	219
7.5.2.6	Type: CreatePmJobRequest	220
7.5.2.7	Type: PmJob	221
7.5.2.8	Type: CreateThresholdRequest	222
7.5.2.9	Type: Threshold	223
7.5.2.10	Type: PerformanceReport	224
7.5.2.11	Type: ThresholdModifications	224
7.5.2.12	Type: PmJobModifications	225
7.5.3	Referenced structured data types	225
7.5.3.1	Introduction.....	225
7.5.3.2	Void.....	225
7.5.3.3	Type: PmJobCriteria	225
7.5.3.4	Type: ThresholdCriteria	226
7.5.4	Referenced simple data types and enumerations	226
7.5.4.1	Introduction.....	226
7.5.4.2	Simple data types	226
7.5.4.3	Enumeration: CrossingDirectionType.....	227
8	NS Fault Management interface.....	227
8.1	Description	227
8.1a	API version.....	227
8.2	Resource structure and methods	227
8.3	Sequence diagrams (informative).....	228
8.3.1	Flow of the Get Alarm List operation	228
8.3.2	Flow of acknowledging alarm	229
8.3.3	Flow of managing subscriptions	230
8.3.4	Flow of sending notifications.....	231
8.4	Resources	232

8.4.1	Introduction.....	232
8.4.1a	Resource: API versions.....	232
8.4.2	Resource: Alarms.....	232
8.4.2.1	Description.....	232
8.4.2.2	Resource definition.....	232
8.4.2.3	Resource methods.....	232
8.4.2.3.1	POST.....	232
8.4.2.3.2	GET.....	232
8.4.2.3.3	PUT.....	234
8.4.2.3.4	PATCH.....	234
8.4.2.3.5	DELETE.....	234
8.4.3	Resource: Individual alarm.....	234
8.4.3.1	Description.....	234
8.4.3.2	Resource definition.....	234
8.4.3.3	Resource methods.....	234
8.4.3.3.1	POST.....	234
8.4.3.3.2	GET.....	234
8.4.3.3.3	PUT.....	235
8.4.3.3.4	PATCH.....	235
8.4.3.3.5	DELETE.....	236
8.4.4	Resource: Subscriptions.....	236
8.4.4.1	Description.....	236
8.4.4.2	Resource definition.....	236
8.4.4.3	Resource methods.....	237
8.4.4.3.1	POST.....	237
8.4.4.3.2	GET.....	238
8.4.4.3.3	PUT.....	239
8.4.4.3.4	PATCH.....	239
8.4.4.3.5	DELETE.....	239
8.4.5	Resource: Individual subscription.....	240
8.4.5.1	Description.....	240
8.4.5.2	Resource definition.....	240
8.4.5.3	Resource methods.....	240
8.4.5.3.1	POST.....	240
8.4.5.3.2	GET.....	240
8.4.5.3.3	PUT.....	241
8.4.5.3.4	PATCH.....	241
8.4.5.3.5	DELETE.....	241
8.4.6	Resource: Notification endpoint.....	241
8.4.6.1	Description.....	241
8.4.6.2	Resource definition.....	241
8.4.6.3	Resource methods.....	242
8.4.6.3.1	POST.....	242
8.4.6.3.2	GET.....	242
8.4.6.3.3	PUT.....	243
8.4.6.3.4	PATCH.....	243
8.4.6.3.5	DELETE.....	243
8.5	Data Model.....	243
8.5.1	Introduction.....	243
8.5.2	Resource and notification data types.....	243
8.5.2.1	Introduction.....	243
8.5.2.2	Type: FmSubscriptionRequest.....	243
8.5.2.3	Type: FmSubscription.....	244
8.5.2.4	Type: Alarm.....	244
8.5.2.5	Type: AlarmNotification.....	245
8.5.2.6	Type: AlarmClearedNotification.....	245
8.5.2.7	Type: AlarmListRebuiltNotification.....	246
8.5.2.8	Type: AlarmModifications.....	246
8.5.3	Referenced structured data types.....	247
8.5.3.1	Introduction.....	247
8.5.3.2	Type: FmNotificationsFilter.....	247
8.5.3.3	Type: FaultyResourceInfo.....	247

8.5.3.4	Type: FaultyComponentInfo	247
8.5.4	Referenced simple data types and enumerations	248
8.5.4.1	Introduction	248
8.5.4.2	Simple data types	248
8.5.4.3	Enumeration: PerceivedSeverityType	248
8.5.4.4	Enumeration: EventType	248
8.5.4.5	Enumeration: FaultyResourceType	249
9	VNF Package Management interface	249
9.1	Description	249
9.1a	API version	250
9.2	Resource structure and methods	250
9.3	Sequence diagrams (informative)	252
9.3.1	Flow of the creation of an individual VNF package resource	252
9.3.2	Flow of the uploading of VNF package content	252
9.3.2a	Flow of handling errors during upload of VNF package content	254
9.3.2b	Flow of fetching the VNF package manifest	256
9.3.3	Flow of querying/reading VNF package information	256
9.3.4	Flow of reading the VNFD of an on-boarded VNF package	257
9.3.4a	Flow of bulk-fetching VNF package artifacts that are not images	258
9.3.5	Flow of updating information of a VNF package	259
9.3.6	Flow of deleting a VNF package resource	260
9.3.7	Flow of fetching an on-boarded VNF package	261
9.3.8	Flow of fetching a VNF package artifact	262
9.3.9	Flow of managing subscriptions	262
9.3.10	Flow of sending notifications	264
9.4	Resources	264
9.4.1	Introduction	264
9.4.1a	Resource: API versions	265
9.4.2	Resource: VNF packages	265
9.4.2.1	Description	265
9.4.2.2	Resource definition	265
9.4.2.3	Resource methods	265
9.4.2.3.1	POST	265
9.4.2.3.2	GET	266
9.4.2.3.3	PUT	267
9.4.2.3.4	PATCH	267
9.4.2.3.5	DELETE	268
9.4.3	Resource: Individual VNF package	268
9.4.3.1	Description	268
9.4.3.2	Resource definition	268
9.4.3.3	Resource methods	268
9.4.3.3.1	POST	268
9.4.3.3.2	GET	268
9.4.3.3.3	PUT	269
9.4.3.3.4	PATCH	269
9.4.3.3.5	DELETE	270
9.4.4	Resource: VNFD in an individual VNF package	271
9.4.4.1	Description	271
9.4.4.2	Resource definition	271
9.4.4.3	Resource methods	271
9.4.4.3.1	POST	271
9.4.4.3.2	GET	272
9.4.4.3.3	PUT	274
9.4.4.3.4	PATCH	274
9.4.4.3.5	DELETE	274
9.4.4a	Resource: Access configuration for external artifacts	274
9.4.4a.1	Description	274
9.4.4a.2	Resource definition	274
9.4.4a.3	Resource methods	275
9.4.4a.3.1	POST	275
9.4.4a.3.2	GET	275

9.4.4a.3.3	PUT	275
9.4.4a.3.4	PATCH.....	276
9.4.4a.3.5	DELETE.....	277
9.4.4b	Resource: Manifest in an individual VNF package	277
9.4.4b.1	Description	277
9.4.4b.2	Resource definition	277
9.4.4b.3	Resource methods	277
9.4.4b.3.1	POST	277
9.4.4b.3.2	GET	277
9.4.4b.3.3	PUT	278
9.4.4b.3.4	PATCH.....	278
9.4.4b.3.5	DELETE.....	278
9.4.5	Resource: VNF package content.....	279
9.4.5.1	Description	279
9.4.5.2	Resource definition	279
9.4.5.3	Resource methods	279
9.4.5.3.1	POST	279
9.4.5.3.2	GET	279
9.4.5.3.3	PUT	280
9.4.5.3.4	PATCH.....	281
9.4.5.3.5	DELETE.....	281
9.4.5a	Resource: VNF package artifacts.....	282
9.4.5a.1	Description	282
9.4.5a.2	Resource definition	282
9.4.5a.3	Resource methods	282
9.4.5a.3.1	POST	282
9.4.5a.3.2	GET	282
9.4.5a.3.3	PUT	285
9.4.5a.3.4	PATCH.....	285
9.4.5a.3.5	DELETE.....	285
9.4.6	Resource: Upload VNF package from URI task.....	285
9.4.6.1	Description	285
9.4.6.2	Resource definition	285
9.4.6.3	Resource methods	285
9.4.6.3.1	POST	285
9.4.6.3.2	GET	286
9.4.6.3.3	PUT	286
9.4.6.3.4	PATCH.....	286
9.4.6.3.5	DELETE.....	286
9.4.7	Resource: Individual VNF package artifact.....	287
9.4.7.1	Description	287
9.4.7.2	Resource definition	287
9.4.7.3	Resource methods	287
9.4.7.3.1	POST	287
9.4.7.3.2	GET	287
9.4.7.3.3	PUT	289
9.4.7.3.4	PATCH.....	289
9.4.7.3.5	DELETE.....	289
9.4.8	Resource: Subscriptions.....	289
9.4.8.1	Description	289
9.4.8.2	Resource definition	289
9.4.8.3	Resource methods	290
9.4.8.3.1	POST	290
9.4.8.3.2	GET	291
9.4.8.3.3	PUT	292
9.4.8.3.4	PATCH.....	293
9.4.8.3.5	DELETE.....	293
9.4.9	Resource: Individual subscription.....	293
9.4.9.1	Description	293
9.4.9.2	Resource definition	293
9.4.9.3	Resource methods	293
9.4.9.3.1	POST	293

9.4.9.3.2	GET	293
9.4.9.3.3	PUT	294
9.4.9.3.4	PATCH.....	294
9.4.9.3.5	DELETE.....	294
9.4.10	Resource: Notification endpoint	295
9.4.10.1	Description	295
9.4.10.2	Resource definition	295
9.4.10.3	Resource methods	295
9.4.10.3.1	POST	295
9.4.10.3.2	GET	295
9.4.10.3.3	PUT	296
9.4.10.3.4	PATCH.....	296
9.4.10.3.5	DELETE.....	296
9.5	Data model	296
9.5.1	Introduction.....	296
9.5.2	Resource and notification data types	296
9.5.2.1	Introduction.....	296
9.5.2.2	Type: CreateVnfPkgInfoRequest	296
9.5.2.3	Type: VnfPkgInfoModifications	297
9.5.2.4	Type: UploadVnfPackageFromUriRequest	297
9.5.2.5	Type: VnfPkgInfo	298
9.5.2.6	Type: PkgmSubscriptionRequest	299
9.5.2.7	Type: PkgmSubscription	300
9.5.2.8	Type: VnfPackageOnboardingNotification.....	300
9.5.2.9	Type: VnfPackageChangeNotification.....	301
9.5.2.10	Type: ExternalArtifactsAccessConfig.....	301
9.5.3	Referenced structured data types	302
9.5.3.1	Introduction.....	302
9.5.3.2	Type: VnfPackageSoftwareImageInfo	302
9.5.3.3	Type: VnfPackageArtifactInfo	304
9.5.3.4	Type: PkgmNotificationsFilter.....	305
9.5.3.5	Type: PkgmLinks.....	306
9.5.3.6	Type: Checksum	306
9.5.4	Referenced simple data types and enumerations	306
9.5.4.1	Introduction.....	306
9.5.4.2	Simple data types	306
9.5.4.3	Enumeration: PackageOnboardingStateType.....	307
9.5.4.4	Enumeration: PackageOperationalStateType.....	307
9.5.4.5	Enumeration: PackageUsageStateType.....	307
9.5.4.6	Enumeration: PackageChangeType	307
Annex A (informative): Mapping operations to protocol elements.....		308
A.1	Overview	308
A.2	NSD Management interface	308
A.3	NS Lifecycle Management interface.....	309
A.4	NS Performance Management interface.....	309
A.5	NS Fault Management interface.....	310
A.6	VNF Package Management interface.....	310
Annex B (informative): State models.....		311
B.1	NSD state model.....	311
B.1.1	Introduction	311
B.1.2	State model.....	311
B.2	VNF package state model.....	312
B.2.1	Introduction	312
B.2.2	State model.....	312

Annex C (informative):	Complementary material for API utilization	315
Annex D (informative):	Bibliography	316
Annex E (informative):	Change History	317
History		326

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (<https://ipr.etsi.org/>).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

Foreword

This Group Specification (GS) has been produced by ETSI Industry Specification Group (ISG) Network Functions Virtualisation (NFV).

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the [ETSI Drafting Rules](#) (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

1 Scope

The present document specifies a set of RESTful protocol specifications and data models fulfilling the requirements specified in ETSI GS NFV-IFA 013 [3] for the interfaces used over the Os-Ma-nfvo reference point.

2 References

2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at <https://docbox.etsi.org/Reference/>.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long-term validity.

The following referenced documents are necessary for the application of the present document.

[1] IANA: "Assigned Internet Protocol Numbers".

NOTE: Available at <https://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml>.

[2] ETSI GS NFV-IFA 010: "Network Functions Virtualisation (NFV) Release 2; Management and Orchestration; Functional requirements specification".

[3] ETSI GS NFV-IFA 013: "Network Functions Virtualisation (NFV) Release 2; Management and Orchestration; Os-Ma-Nfvo reference point - Interface and Information Model Specification".

[4] ETSI GS NFV-SOL 003: "Network Functions Virtualisation (NFV) Release 2; Protocols and Data Models; RESTful protocols specification for the Or-Vnfm Reference Point".

[5] ETSI GS NFV-SOL 004: "Network Functions Virtualisation (NFV) Release 2; Protocols and Data Models; VNF Package and PNFD Archive specification".

[6] IEEE™ 802.1Q-2018: "IEEE Standard for Local and metropolitan area networks - Bridges and Bridged Networks".

[7] IETF RFC 791: "Internet Protocol".

NOTE: Available at <https://tools.ietf.org/html/rfc791>.

[8] IETF RFC 4776: "Dynamic Host Configuration Protocol (DHCPv4 and DHCPv6) Option for Civic Addresses Configuration Information".

NOTE: Available at <https://tools.ietf.org/html/rfc4776>.

[9] IETF RFC 5646: "Tags for Identifying Languages".

NOTE: Available at <https://tools.ietf.org/html/rfc5646>.

[10] IETF RFC 7233: "Hypertext Transfer Protocol (HTTP/1.1): Range Requests".

NOTE: Available at <https://tools.ietf.org/html/rfc7233>.

[11] IETF RFC 7396: "JSON Merge Patch".

NOTE: Available at <https://tools.ietf.org/html/rfc7396>.

[12] IETF RFC 8200: "Internet Protocol, Version 6 (IPv6) Specification".

NOTE: Available at <https://tools.ietf.org/html/rfc8200>.

[13] ISO 3166 (all parts): "Codes for the representation of names of countries and their subdivisions".

[14] Recommendation ITU-T X.733: "Information technology - Open Systems Interconnection - Systems Management: Alarm reporting function".

[15] ETSI GS NFV-IFA 027: "Network Functions Virtualisation (NFV) Release 2; Management and Orchestration; Performance Measurements Specification".

[16] ETSI GS NFV-SOL 013: "Network Functions Virtualisation (NFV) Release 2; Protocols and Data Models; Specification of common aspects for RESTful NFV MANO APIs".

[17] IETF RFC 7193: "The application/cms Media Type".

NOTE: Available at <https://tools.ietf.org/html/rfc7193>.

[18] ETSI GS NFV-SOL 007: "Network Functions Virtualisation (NFV) Release 2; Protocols and Data Models; Network Service Descriptor File Structure Specification".

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long-term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] ETSI GS NFV-SOL 001: "Network Functions Virtualisation (NFV) Release 2; Protocols and Data Models; NFV descriptors based on TOSCA specification".

[i.2] OpenStack®: "Disk and container formats for images".

NOTE 1: Available at <https://docs.openstack.org/glance/pike/user/formats.html>.

NOTE 2: The OpenStack® Word Mark and OpenStack Logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. ETSI is not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

[i.3] OpenAPI™ Specification.

NOTE 1: Available at <https://github.com/OAI/OpenAPI-Specification>.

NOTE 2: OpenAPI Specification and OpenAPI Initiative and their respective logos, are trademarks of the Linux Foundation.

[i.4] Void.

[i.5] Void.

[i.6] Void.

[i.7] ETSI GS NFV-SOL 015: "Network Functions Virtualisation (NFV); Protocols and Data Models; Specification of Patterns and Conventions for RESTful NFV-MANO APIs".

3 Definition of terms, symbols and abbreviations

3.1 Terms

Void.

3.2 Symbols

Void.

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

API	Application Programming Interface
BSS	Business Support System
CD-ROM	Compact Disk - Read-Only Memory
CIDR	Classless Inter-Domain Routing
CP	Connection Point
CPD	CP Descriptor
CSAR	Cloud Service Archive
DF	Deployment Flavour
DSCP	Differentiated Services Code Point
EM	Element Management
ETSI	European Telecommunications Standards Institute
FM	Fault Management
GS	Group Specification
GUI	Graphical User Interface
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	HTTP Secure
IANA	Internet Assigned Numbers Authority
ICMP	Internet Control Message Protocol
IETF	Internet Engineering Task Force
IFA	Interfaces and Architecture
IP	Internet Protocol
ISG	Industry Specific Group
ISO	International Organization for Standardization
ITU-T	International Telecommunication Union - Telecommunication standardization sector
JSON	Javascript Object Notation
LB	Load Balancer
LCCN	Lifecycle Change Notification
LCM	Lifecycle Management
MAC	Medium Access Control
MANO	Management and Orchestration
NFP	Network Forwarding Path
NFPD	NFP Descriptor
NFV	Network Functions Virtualisation
NFVI	Network Function Virtualisation Infrastructure
NFVO	NFV Orchestrator
NS	Network Service
NSD	Network Service Descriptor
OSS	Operation Support System
PKG	Package
PM	Performance Management
PNF	Physical Network Function
PNFD	Physical Network Function Descriptor
RAM	Random-Access Memory

RFC	Request For Comments
SAP	Service Access Point
SAPD	Service Access Point Descriptor
TCP	Transmission Control Protocol
TLS	Transport Layer Security
TOSCA	Topology and Orchestration Specification for Cloud Applications
UDP	User Datagram Protocol
URI	Uniform Resource Identifier
VDU	Virtualisation Deployment Unit
VIM	Virtualised Infrastructure Manager
VL	Virtual Link
VLAN	Virtual Local Area Network
VLD	VL Descriptor
VNF	Virtualised Network Function
VNFC	VNF Component
VNFD	VNF Descriptor
VNFFG	VNF Forwarding Graph
VNFFGD	VNFFG Descriptor
VNFM	VNF Manager
YAML	YAML Ain't Markup Language

4 General Aspects

4.1 Overview

The present document defines the protocol and data model for the following interfaces, in the form of RESTful Application Programming Interface (API) specifications:

- NSD Management interface (as produced by the NFVO towards the OSS/BSS)
- NS Lifecycle Management interface (as produced by the NFVO towards the OSS/BSS)
- NS Performance Management interface (as produced by the NFVO towards the OSS/BSS)
- NS Fault Management interface (as produced by the NFVO towards the OSS/BSS)
- VNF Package Management interface (as produced by the NFVO towards the OSS/BSS)

Table 4.1-1 lists the versions of the APIs defined in the present document.

Table 4.1-1: Versions of the APIs specified in the present document

API	API version
NSD Management interface	2.0.0
NS Lifecycle Management interface	1.4.0
NS Performance Management interface	2.1.0
NS Fault Management interface	1.2.0
VNF Package Management interface	2.1.0

The design of the protocol and data model for the above interfaces is based on the information model and requirements defined in ETSI GS NFV-IFA 013 [3]. In clause 4, general aspects are specified that apply to multiple APIs on the Os-Ma-nfvo reference point. In addition, the provisions in clauses 4, 5, 6, 8 and 9 of ETSI GS NFV-SOL 013 [16] define common aspects of RESTful NFV MANO APIs, and shall apply for all APIs defined in the present document.

In the subsequent clauses, the protocol and data model for the individual interfaces are specified. Per interface, the resource structure with associated HTTP methods is defined and applicable flows are provided. Further, the resources and the data model are specified in detail.

Annex A provides the mapping of the combination of resources and methods defined in the present document to the operations defined in ETSI GS NFV-IFA 013 [3].

Even though the various interfaces defined in the present document are related, implementations shall not assume a particular order of messages that arrive via different interfaces.

4.2 Void

4.3 Void

4.4 Common data types

4.4.1 Structured data types

4.4.1.1 Introduction

This clause defines data structures that are referenced from data structures in multiple interfaces. In addition, the structured data types defined in clause 7.1 of ETSI GS NFV-SOL 013 [16] shall apply.

4.4.1.2 Void

4.4.1.3 Void

4.4.1.3a Void

4.4.1.4 Void

4.4.1.5 Type: NsInstanceSubscriptionFilter

This type represents subscription filter criteria to match NS instances. It shall comply with the provisions defined in Table 4.4.1.5-1.

Table 4.4.1.5-1: Definition of the NsInstanceSubscriptionFilter data type

Attribute name	Data type	Cardinality	Description
nsdlds	Identifier	0..N	If present, match NS instances that were created based on an NSD identified by one of the nsdld values listed in this attribute. See note 1.
vnfdlds	Identifier	0..N	If present, match NS instances that contain VNF instances that were created based on a VNFD identified by one of the vnfdld values listed in this attribute. See note 1.
pnfdlds	Identifier	0..N	If present, match NS instances that contain PNFs that are represented by a PNFD identified by one of the pnfdld values listed in this attribute. See note 1.
nsInstancelds	Identifier	0..N	If present, match NS instances with an instance identifier listed in this attribute. See note 2.
nsInstanceNames	String	0..N	If present, match NS instances with an NS Instance Name listed in this attribute. See note 2.
NOTE 1: The attributes "nsdlds", "vnfdlds" and "pnfdlds" are alternatives to reference to NS instances that are created based on certain NSDs, or contain VNF instances that are based on certain VNFDs, or contain PNFs that are based on certain PNFDs in a filter. They should not be used together in the same filter instance, but one alternative should be chosen.			
NOTE 2: The attributes "nsInstancelds" and "nsInstanceNames" are alternatives to reference to particular NS Instances in a filter. They should not be used both in the same filter instance, but one alternative should be chosen.			

4.4.1.6 Type: ResourceHandle

This type represents the information that allows addressing a virtualised resource that is used by a VNF instance or by an NS instance. Information about the resource is available from the VIM. The ResourceHandle type shall comply with the provisions defined in Table 4.4.1.6-1.

Table 4.4.1.6-1: Definition of the ResourceHandle data type

Attribute name	Data type	Cardinality	Description
vimId	Identifier	0..1	Identifier of the VIM under whose control this resource is placed. This attribute shall be present if VNF-related resource management in direct mode is applicable. It shall also be present for resources that are part of an NS instance such as virtual link resources.
resourceProviderId	Identifier	0..1	Identifier of the entity responsible for the management of the resource. This attribute shall only be supported and present when VNF-related resource management in indirect mode is applicable. The identification scheme is outside the scope of the present document.
resourceId	IdentifierInVim	1	Identifier of the resource in the scope of the VIM or the resource provider.
vimLevelResourceType	String	0..1	Type of the resource in the scope of the VIM or the resource provider. See note.
NOTE: The value set of the "vimLevelResourceType" attribute is within the scope of the VIM or the resource provider and can be used as information that complements the ResourceHandle.			

4.4.1.7 Void

4.4.2 Simple data types and enumerations

4.4.2.1 Introduction

This clause defines simple data types and enumerations that can be referenced from data structures defined in multiple interfaces.

4.4.2.2 Simple data types

Table 4.4.2.2-1 defines simple data types for reference from data type definitions in the present document. In addition, the simple data types defined in clause 7.2.2 of ETSI GS NFV-SOL 013 [16] shall apply.

Table 4.4.2.2-1: Simple data types

Type name	Description
IdentifierInNs	An identifier that is unique with respect to an NS. Representation: string of variable length.
IdentifierInNsd	An identifier that is unique within an NS descriptor. Representation: string of variable length.
IdentifierInPnf	An Identifier that is unique within respect to a PNF. Representation: string of variable length.
IdentifierInVim	An identifier maintained by the VIM or other resource provider. It is expected to be unique within the VIM instance. Representation: string of variable length.

4.4.2.3 Enumerations

The enumerations defined in clause 7.2.3 of ETSI GS NFV-SOL 013 [16] shall apply to be available for referencing from data type definitions in the present document.

4.5 Void

4.6 Void

4.7 Void

5 NSD Management interface

5.1 Description

This interface allows the OSS/BSS to invoke management operations of NSDs towards the NFVO and to subscribe to notifications related to NSD management changes.

The operations provided through this interface are as follows:

- Create NSD Info
- Upload NSD archive
- Fetch NSD archive
- Update NSD Info
- Delete NSD

- Query NSD Info
- Read NSD
- Fetch NSD archive manifest
- Create PNFD Info
- Upload PNFD archive
- Fetch PNFD archive
- Update PNFD Info
- Delete PNFD
- Query PNFD Info
- Read PNFD
- Fetch PNFD archive manifest
- Subscribe
- Terminate Subscription
- Query Subscription Information
- Notify

This interface also enables to invoke error handling procedures (i.e. Retry, Rollback, Continue, Cancel, and Fail) on the actual NS lifecycle management operation occurrences, and API version information retrieval.

The state changes of an NSD are illustrated in clause B.2.

5.1a API version

For the NSD management interface version as specified in the present document, the MAJOR version field shall be 2, the MINOR version field shall be 0, and the PATCH version field shall be 0 (see clause 9.1 of ETSI GS NFV-SOL 013 [16] for a definition of the version fields). Consequently, the {apiMajorVersion} URI variable shall be set to "v2".

NOTE: In the present document, there were no changes to the clauses defining the NSD management interface that are visible at interface level compared to the previous version of the present document; hence, the MAJOR/MINOR/PATCH version fields are kept the same.

5.2 Resource structure and methods

All resource URIs of the API shall use the base URI specification defined in clause 4.1 of ETSI GS NFV-SOL 013 [16]. The string "nsd" shall be used to represent {apiName}. All resource URIs in the clauses below are defined relative to the above base URI.

Figure 5.2-1 shows the overall resource URI structure defined for the NSD management interface.

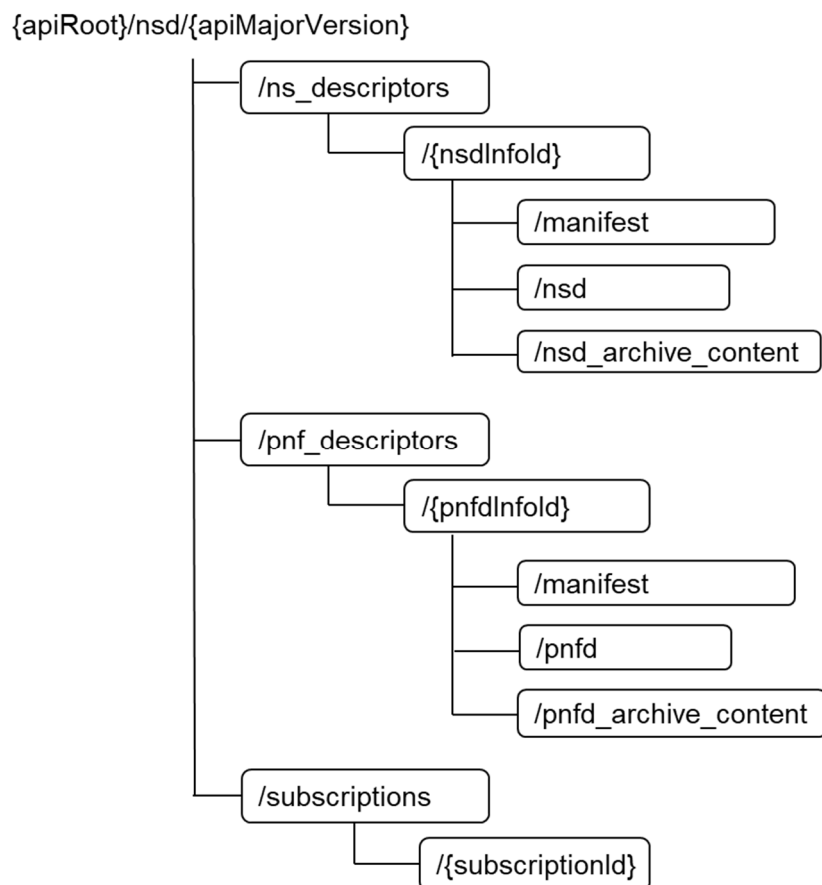


Figure 5.2-1: Resource URI structure of NSD Management Interface

Table 5.2-1 lists the individual resources defined, and the applicable HTTP methods.

The NFVO shall support responding to requests for all HTTP methods on the resources in Table 5.2-1 that are marked as "M" (mandatory) in the "Cat" column. The NFVO shall also support the "API versions" resources as specified in clause 9.3.2 of ETSI GS NFV-SOL 013 [16].

Table 5.2-1: Resources and methods overview of the NSD Management interface

Resource name	Resource URI	HTTP Method	Cat	Meaning
NS Descriptors	/ns_descriptors	GET	M	Query information about multiple NS descriptor resources.
		POST	M	Create a new NS descriptor resource.
Individual NS Descriptor	/ns_descriptors/{nsdInfold}	GET	M	Read information about an individual NS descriptor resource.
		PATCH	M	Modify the operational state and/or the user defined data of an individual NS descriptor resource.
		DELETE	M	Delete an individual NS descriptor resource.
NSD archive content	/ns_descriptors/{nsdInfold}/nsd_archive_content	GET	M	Fetch the content of an NS descriptor archive.
		PUT	M	Upload the content of an NS descriptor archive.
NSD of an individual NSD archive	/ns_descriptors/{nsdInfold}/nsd	GET	M	Read an onboarded NS descriptor.
Manifest of an individual NSD archive	/ns_descriptors/{nsdInfold}/manifest	GET	M	Fetch the manifest file of an onboarded NSD archive.

Resource name	Resource URI	HTTP Method	Cat	Meaning
PNF Descriptors	/pnf_descriptors	GET	M	Query information about multiple PNF descriptor resources.
		POST	M	Create a new PNF descriptor resource.
Individual PNF Descriptor	/pnf_descriptors/{pnfdInfold}	GET	M	Read information about an individual PNF descriptor resource.
		PATCH	M	Modify the user defined data of an individual PNF descriptor resource.
		DELETE	M	Delete an individual PNF descriptor resource.
PNFD archive content	/pnf_descriptors/{pnfdInfold}/pnfd_archive_content	GET	M	Fetch the content of a PNF descriptor archive.
		PUT	M	Upload the content of a PNF descriptor archive.
PNFD of an individual PNFD archive	/pnf_descriptors/{pnfdInfold}/pnfd	GET	M	Read an onboarded PNF descriptor.
Manifest of an individual PNFD archive	/pnf_descriptors/{pnfdInfold}/manifest	GET	M	Fetch the manifest file of an onboarded PNFD archive.
Subscriptions	/subscriptions	POST	M	Subscribe to NS descriptor and PNF descriptor change notifications.
		GET	M	Query multiple subscriptions.
Individual subscription	/subscriptions/{subscriptionId}	GET	M	Read an individual subscription resource.
		DELETE	M	Terminate a subscription.
Notification endpoint	(provided by API consumer)	POST	See note	Notify about NS descriptor and PNF descriptor changes. See note.
		GET	See note	Test the notification endpoint. See note.
NOTE:	The NFVO shall support invoking the HTTP methods defined for the "Notification endpoint" resource exposed by the OSS/BSS. If the OSS/BSS supports invoking the POST method on the "Subscriptions" resource towards the NFVO, it shall also support responding to the HTTP requests defined for the "Notification endpoint" resource.			

5.3 Sequence diagrams (informative)

5.3.1 Flow of the creation of an individual NS descriptor resource

This clause describes the procedure for creating an individual NS descriptor resource.

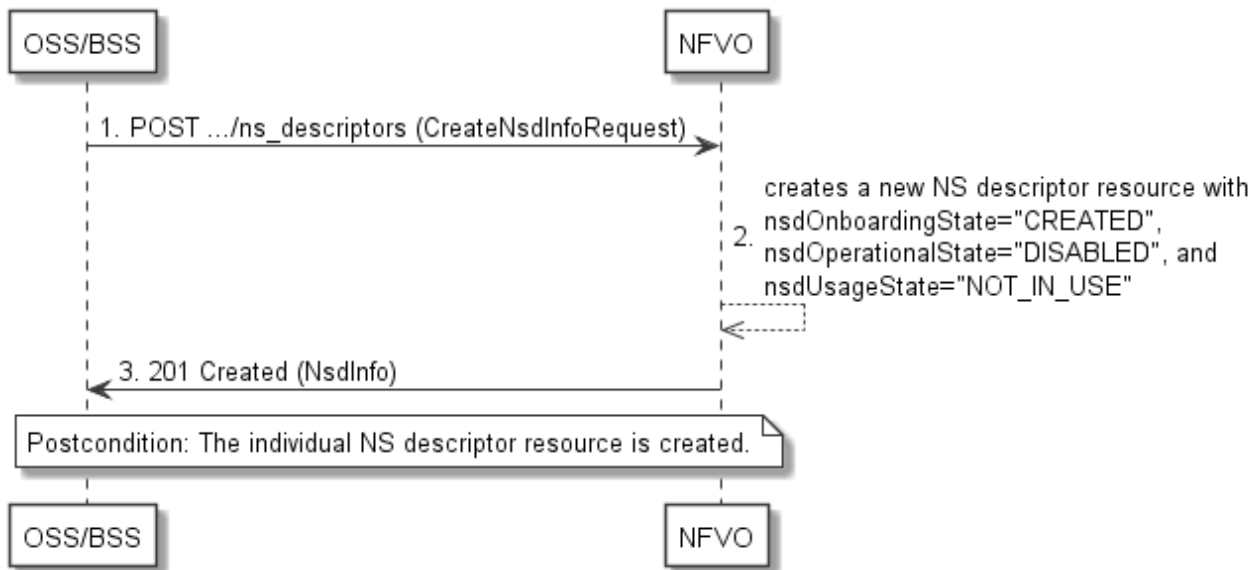


Figure 5.3.1-1: Flow of the creation of an individual NS descriptor resource

The procedure consists of the following steps as illustrated in Figure 5.3.1-1:

- 1) The OSS/BSS sends a POST request to the "NS descriptors" resource including in the payload body a data structure of type "CreateNsdInfoRequest".
- 2) The NFVO creates a new NS descriptor resource with nsdOnboardingState="CREATED", nsdOperationalState="DISABLED" and nsdUsageState="NOT_IN_USE".
- 3) The NFVO returns a 201 Created response containing a representation of the individual NS descriptor resource just created by the NFVO.

Postcondition: Upon successful completion, the individual NS descriptor resource has been created with nsdOnboardingState="CREATED", nsdOperationalState="DISABLED", and nsdUsageState="NOT_IN_USE".

5.3.2 Flow of the uploading of NSD archive content

This clause describes the procedure for the uploading of an NSD archive.

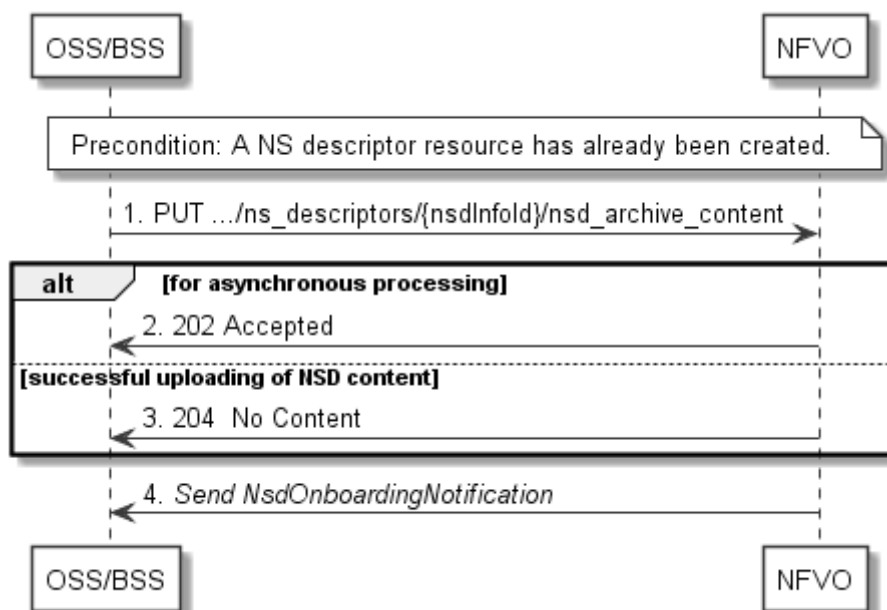


Figure 5.3.2-1: Flow of the uploading of NSD archive content

NOTE: Due to possible race conditions, the 202 or 204 response and the NsdOnBoardingNotification can arrive in any order at the OSS/BSS.

Precondition: An NS descriptor resource has been created and the value of "onboardingState" attribute equals to "CREATED", or the value of "onboardingState" equals to "ERROR" due to a previous failure of uploading or processing.

The procedure consists of the following steps as illustrated in Figure 5.3.2-1:

- 1) The OSS/BSS sends a PUT request to an "NSD archive content" resource using a "Content-Type" HTTP header as defined in clause 5.4.4.3.3 of the present document.
- 2) For asynchronous processing, the NFVO returns "202 Accepted".
- 3) Otherwise, the NFVO returns a "204 No Content" response to the OSS/BSS with an empty payload body for successful uploading of the NSD content.

After successful uploading and processing, the NFVO sends a NsdOnBoardingNotification to the OSS/BSS.

Error handling: In case of failure, appropriate error information is provided in the response, and/or the "onboardingState" is set to "ERROR".

5.3.3 Flow of the fetching of an onboarded NSD archive

This clause describes the procedure for fetching the content of an onboarded NSD archive.

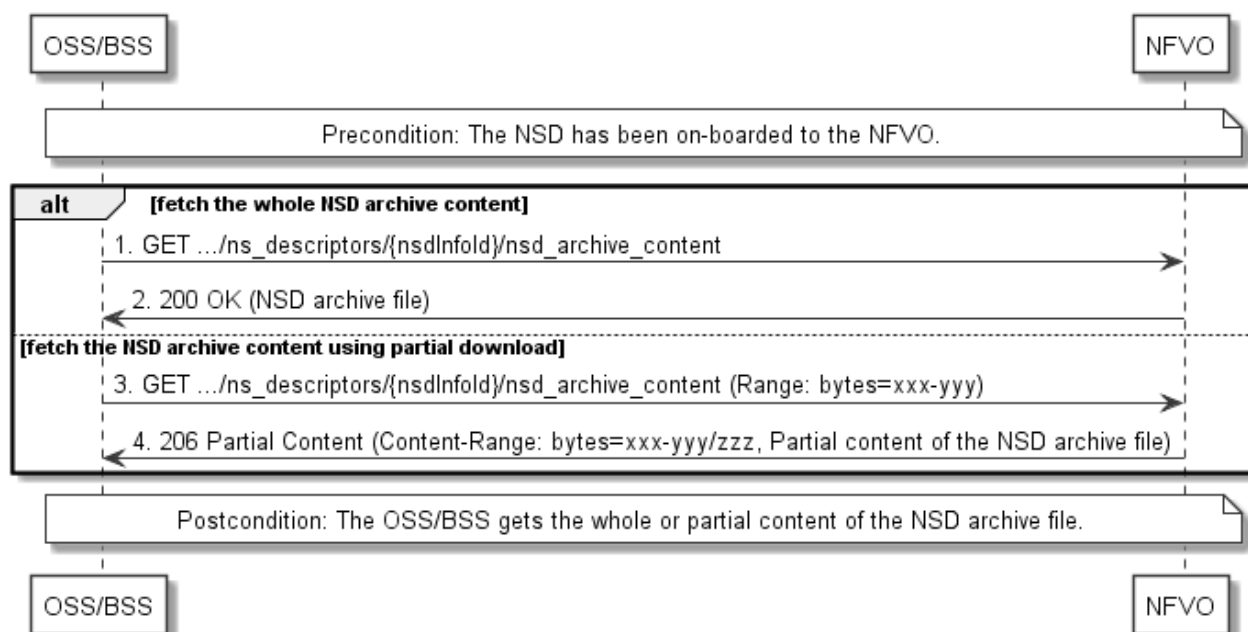


Figure 5.3.3-1: Flow of the fetching of an NSD archive

Precondition: The NSD is on-boarded to the NFVO.

Fetching an on-boarded NSD, as illustrated in Figure 5.3.3-1, consists of the following steps:

- 1) If fetching the whole NSD content, the OSS/BSS sends a GET request to the "NSD content" resource.
- 2) The NFVO returns a "200 OK" response, and includes a copy of the complete NSD archive in the payload body.

NOTE: The complete NSD archive is the zip file as onboarded in the Upload NSD archive operation.

- 3) If fetching the NSD content using partial download, the OSS/BSS sends a GET request to the "NSD content" resource, and includes a "Range" HTTP header indicating the partition of the NSD content that needs to be transferred.

- 4) The NFVO returns a "206 Partial Content" response with a payload body containing the partial content of the NSD, and a "Content-Range" HTTP header indicating the byte range enclosed in the payload and the complete length of the NSD.

Postcondition: Upon successful completion, the OSS/BSS gets the whole or partial content of the NSD.

Error handling: In case of failure, appropriate error information is provided in the response.

5.3.4 Flow of the update of an individual NS descriptor resource

This clause describes the procedure for the update of an NS descriptor resource. The Update NSD Info operation allows for the modification of the operational state and/or user defined data of an individual NS descriptor resource.

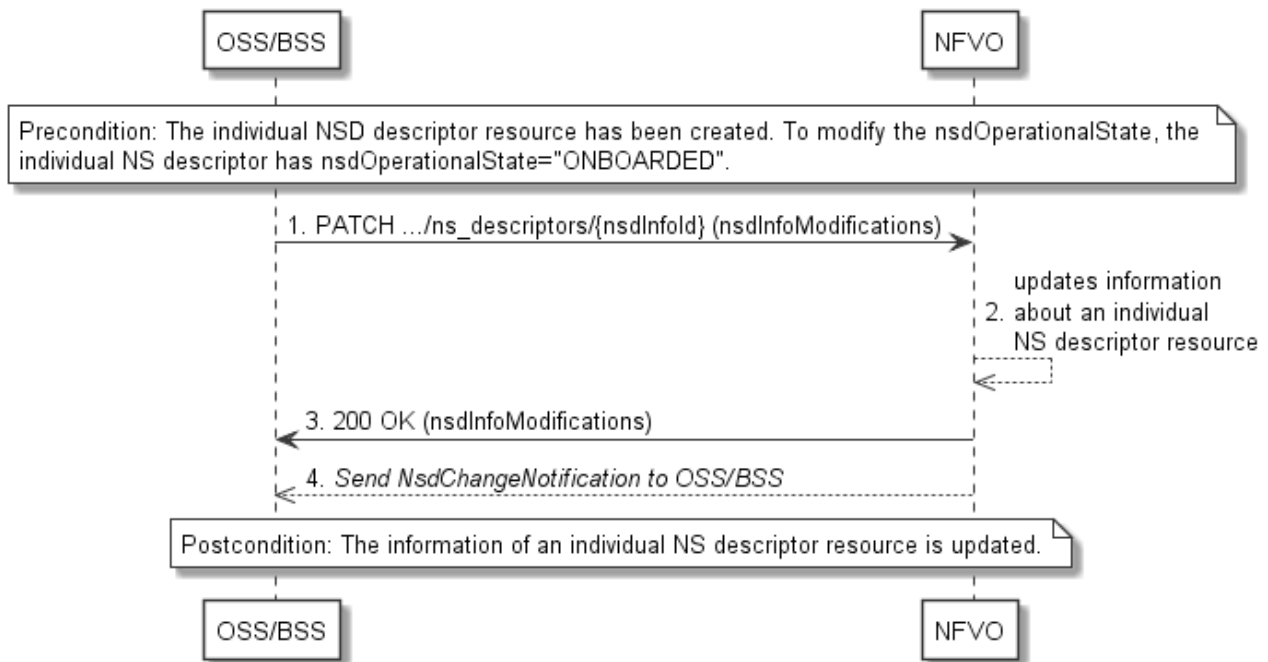


Figure 5.3.4-1: Flow of the update of an individual NS descriptor resource

NOTE: Due to possible race conditions, the 200 response and the NsdChangeNotification can arrive in any order at the OSS/BSS.

Precondition: The individual NS descriptor resource has been created. To modify the nsdOperationalState from "ENABLED" to "DISABLED" or vice-versa, the individual NS descriptor resource has nsdOnboardingState="ONBOARDED".

The procedure consists of the following steps as illustrated in Figure 5.3.4-1:

- 1) The OSS/BSS sends a PATCH request to the "Individual NS descriptor" resource.
- 2) The NFVO modifies the information associated with the individual NS descriptor resource.
- 3) The NFVO returns a "200 OK" response including the data structure of type "nsdInfoModifications" in the payload body.
- 4) When modifying the nsdOperationalState attribute, the NFVO sends to the OSS/BSS a NsdChangeNotification to indicate the state change of the individual NS descriptor resource.

Postcondition: Upon successful completion, the information about an individual NS descriptor resource has been updated.

5.3.5 Flow of the deletion of an individual NS descriptor resource

This clause describes the procedure for the deletion of an individual NS descriptor resource.

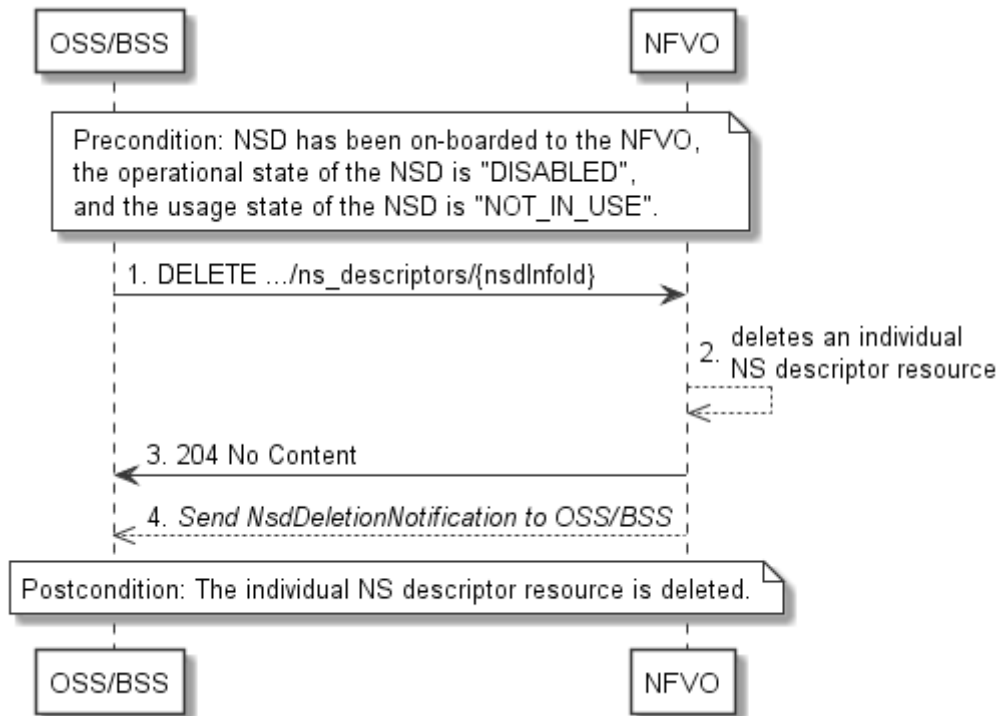


Figure 5.3.5-1: Flow of the deletion of an individual NS descriptor resource

NOTE: Due to possible race conditions, the 204 response and the NsdDeletionNotification can arrive in any order at the OSS/BSS.

Precondition: NSD has been on-boarded to the NFVO, the operational state of the NSD is equal to "DISABLED", and the usage state of the NSD is equal to "NOT_IN_USE".

The procedure consists of the following steps as illustrated in Figure 5.3.5-1:

- 1) The OSS/BSS sends a DELETE request to an "Individual NS descriptor" resource.
- 2) The NFVO deletes the individual NS descriptor resource.
- 3) The NFVO returns a "204 No Content" response to the OSS/BSS with an empty payload body.
- 4) The NFVO sends to the OSS/BSS a NsdDeletionNotification to indicate the deletion of the individual NS descriptor resource.

Postcondition: Upon successful completion, the individual NS descriptor resource has been deleted.

5.3.6 Flow of the querying/reading of NS descriptor information

This clause describes the procedure for querying information about multiple NS descriptor resources and reading information about an individual NS descriptor resource.

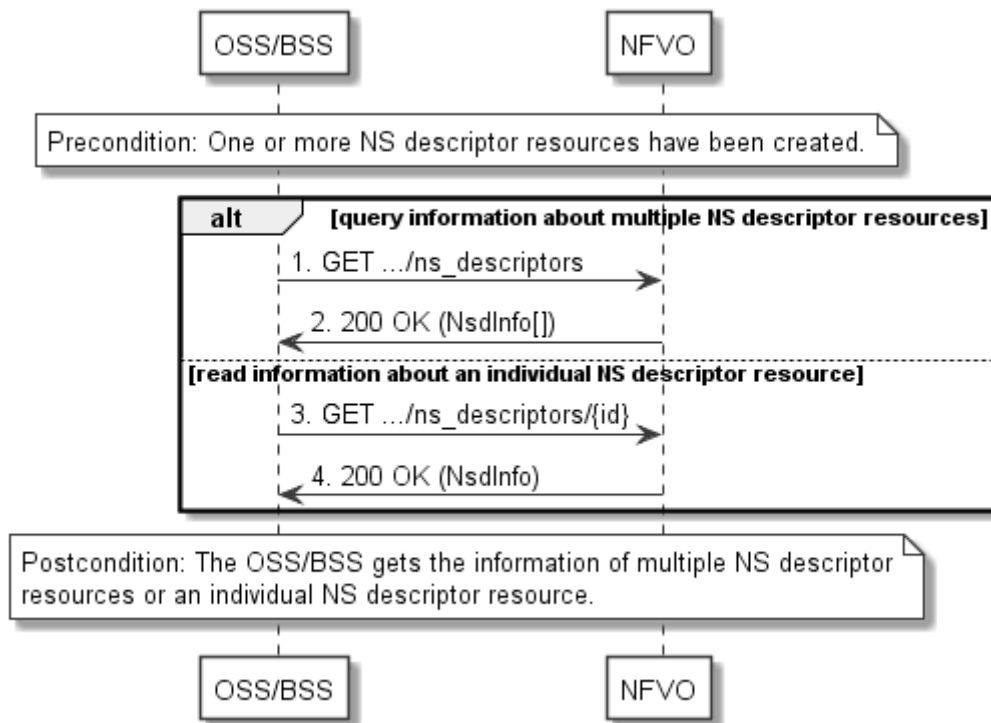


Figure 5.3.6-1: Flow of the querying/reading of NS descriptor resources

Precondition: One or more NS descriptor resources have been created.

The procedure consists of the following steps as illustrated in Figure 5.3.6-1:

- 1) If the OSS/BSS intends to query information about multiple NS descriptor resources, it sends a GET request to the "NS descriptors" resource.
- 2) The NFVO returns a "200 OK" response and includes in the payload body zero or more data structures of type "NsdInfo".
- 3) If the OSS/BSS intends to read information about an individual NS descriptor resource, the OSS/BSS sends a GET request to the "Individual NS descriptor" resource, addressed by the appropriate NsdInfo identifier in its resource URI.
- 4) The NFVO returns a "200 OK" response, and includes in the payload body a data structure of type "NsdInfo".

Postcondition: Upon successful completion, the OSS/BSS gets the information of multiple (i.e. zero or more) NS descriptor resources or an individual NS descriptor resource.

Error handling: In case of failure, appropriate error information is provided in the response.

5.3.6a Flow of reading the NSD

This clause describes the procedure for reading an onboarded NSD.

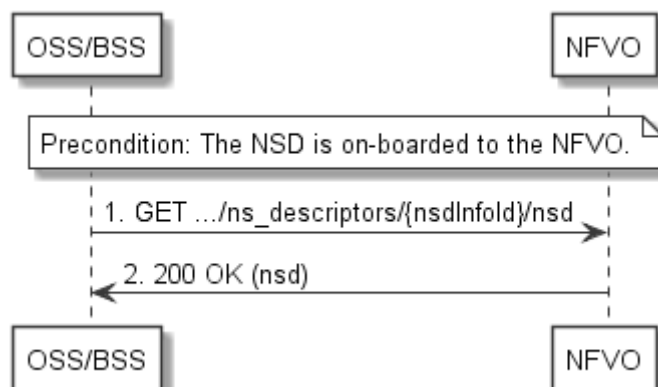


Figure 5.3.6a-1: Flow of reading an NSD

Precondition: The NSD is onboarded to the NFVO.

The procedure consists of the following steps as illustrated in Figure 5.3.6a-1:

- 1) The OSS/BSS sends a GET request to the "nsd" resource.
- 2) The NFVO returns a "200 OK" response and includes a copy of the NSD in the payload body.

5.3.6b Flow of fetching the manifest file of an onboarded NSD archive

This clause describes the procedure for fetching the manifest file of an onboarded NSD.

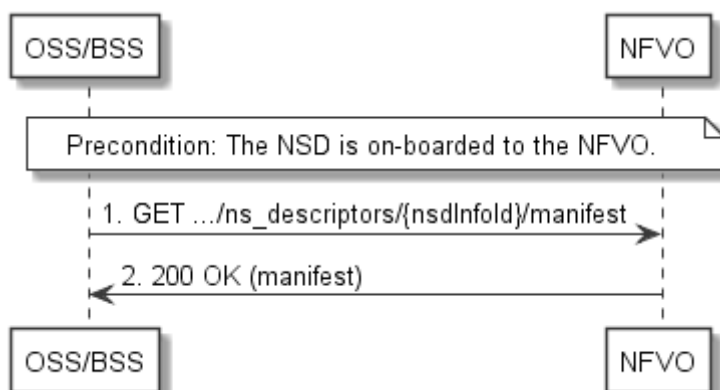


Figure 5.3.6b-1: Flow of fetching the manifest file of an onboarded NSD archive

Precondition: The NSD is onboarded to the NFVO.

The procedure consists of the following steps as illustrated in Figure 5.3.6b-1:

- 1) The OSS/BSS sends a GET request to the "manifest" resource.
- 2) The NFVO returns a "200 OK" response and includes a copy of the manifest file in the payload body.

5.3.7 Flow of the creation of an individual PNF descriptor resource

This clause describes the procedure for creating an individual PNF descriptor resource.

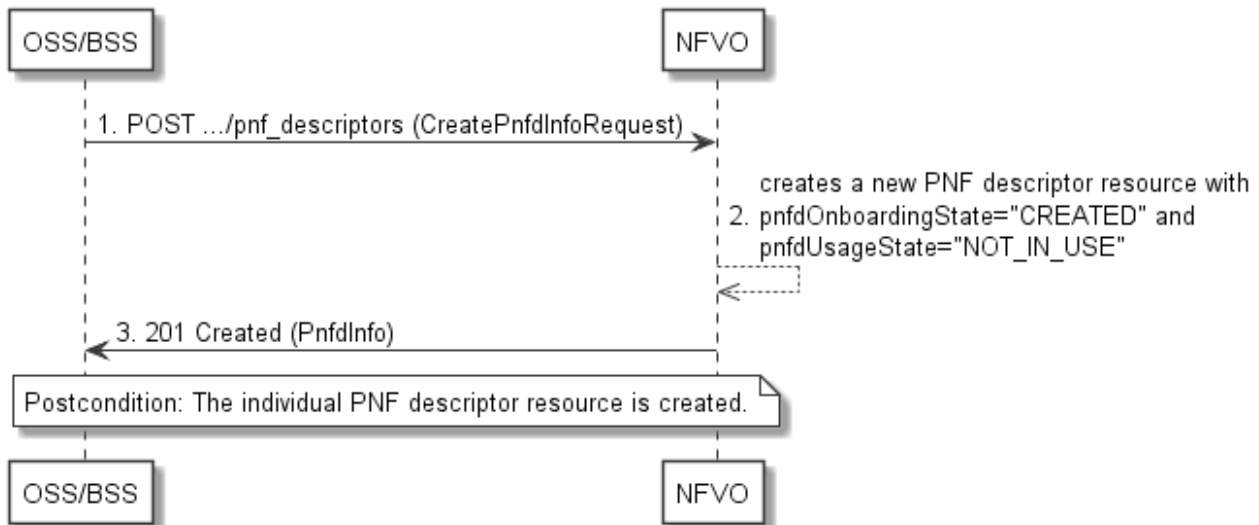


Figure 5.3.7-1: Flow of the creation of an individual PNF descriptor resource

The procedure consists of the following steps as illustrated in Figure 5.3.7-1:

- 1) The OSS/BSS sends a POST request to the "PNF descriptors" resource including in the payload body a data structure of type "CreatePnfdInfoRequest".
- 2) The NFVO creates a new PNF descriptor resource with `pnfdOnboardingState="CREATED"` and `pnfdUsageState="NOT_IN_USE"`.
- 3) The NFVO returns a 201 Created response containing a representation of the individual PNF descriptor resource just created by the NFVO.

Postcondition: Upon successful completion, the individual PNF descriptor resource has been created with `pnfdOnboardingState="CREATED"` and `pnfdUsageState="NOT_IN_USE"`.

5.3.8 Flow of the uploading of PNFD archive

This clause describes the procedure for the uploading of a PNFD archive.

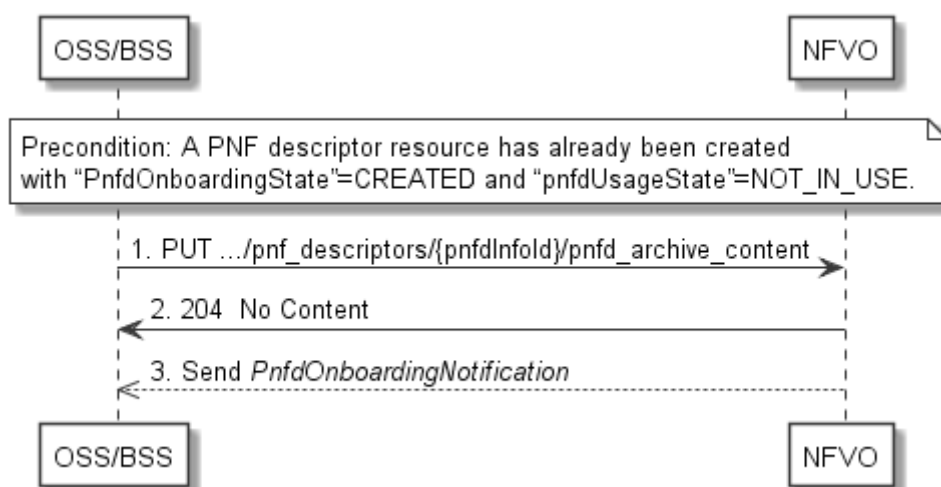


Figure 5.3.8-1: Flow of the uploading of PNFD archive content

Precondition: A PNF descriptor resource has been created and the value of the "pnfdOnboardingState" attribute equals to "CREATED" or the value of the "pnfdOnboardingState" attribute equals to "ERROR" due to a previous failure of uploading or processing.

The procedure consists of the following steps as illustrated in Figure 5.3.8-1:

- 1) The OSS/BSS sends a PUT request to a "PNFD archive content" resource using a "Content-Type" HTTP header as defined in clause 5.4.4.3.3 of the present document.
- 2) The NFVO returns a "204 No Content" response to the OSS/BSS with an empty payload body for successful uploading of the PNF content.
- 3) After successful uploading and processing, the NFVO sends a PnfdOnboardingNotification to the OSS/BSS.

Error handling: In case of failure, appropriate error information is provided in the response, and/or the "pnfdOnboardingState" is set to "ERROR".

5.3.9 Flow of the fetching of an onboarded PNF archive

This clause describes the procedure for fetching the content of an onboarded PNF archive.

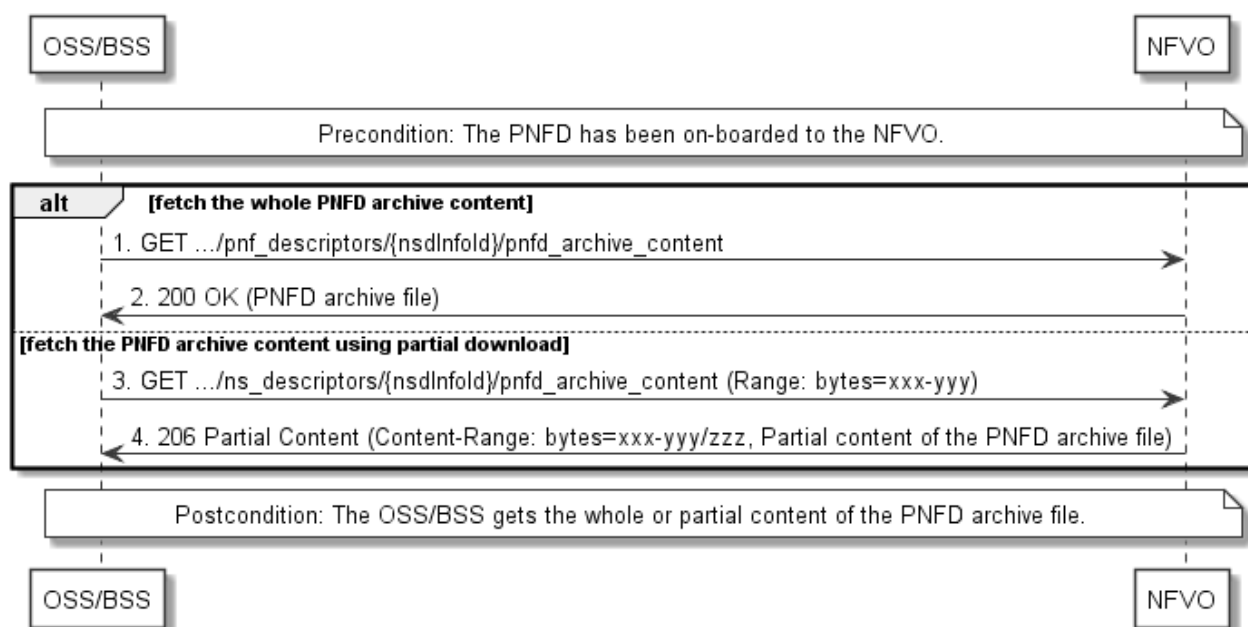


Figure 5.3.9-1: Flow of the fetching of PNF archive content

Precondition: The PNF has been on-boarded to the NFVO.

Fetching an on-boarded PNF, as illustrated in Figure 5.3.9-1, consists of the following steps:

- 1) The OSS/BSS sends a GET request to the "PNFD content" resource.
- 2) The NFVO returns a "200 OK" response and includes a copy of the complete PNF archive in the payload body.

NOTE: The complete PNF archive is the zip file as onboarded in the Upload PNF archive operation.

- 3) If fetching the PNF archive using partial download, the OSS/BSS sends a GET request to the "PNFD archive content" resource and includes a "Range" HTTP header indicating the partition of the PNF archive content that needs to be transferred.
- 4) The NFVO returns a "206 Partial Content" response with a payload body containing the partial content of the PNF archive, and a "Content-Range" HTTP header indicating the byte range enclosed in the payload and the complete length of the PNF archive content.

Postcondition: Upon successful completion, the OSS/BSS gets the content of the PNF.

Error handling: In case of failure, appropriate error information is provided in the response.

5.3.10 Flow of the deletion of an individual PNF descriptor resource

This clause describes the procedure for the deletion of an individual PNF descriptor resource.

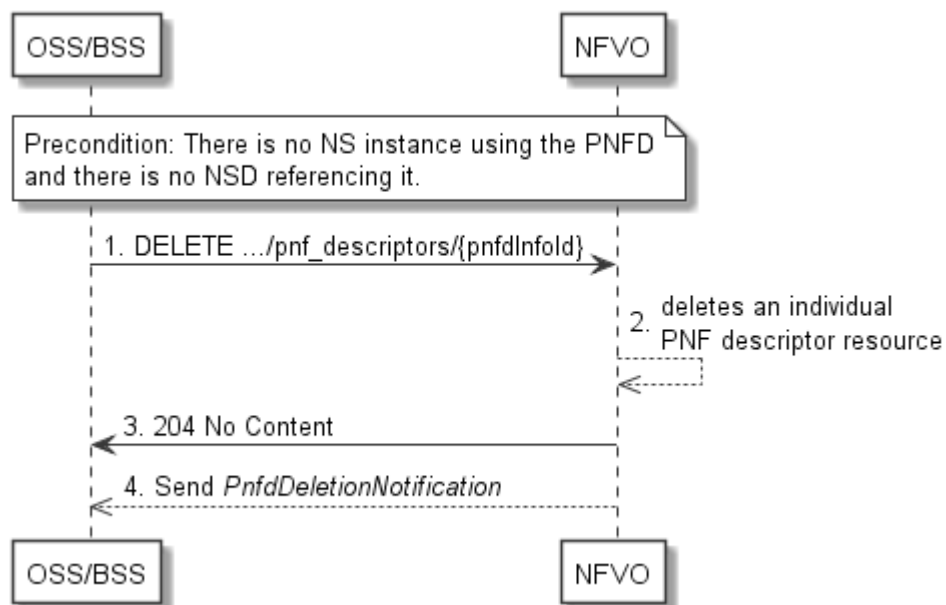


Figure 5.3.10-1: Flow of the deletion of an individual PNF descriptor resource

Precondition: NSD has been on-boarded to the NFVO and in the Disabled state.

The procedure consists of the following steps as illustrated in Figure 5.3.10-1:

- 1) The OSS/BSS sends a DELETE request to an "Individual PNF descriptor" resource.
- 2) The NFVO deletes the individual PNF descriptor resource.
- 3) The NFVO returns a "204 No Content" response to the OSS/BSS with an empty payload body.
- 4) The NFVO sends to the OSS/BSS a PnfdDeletionNotification to indicate the deletion of the individual PNF descriptor resource.

5.3.11 Flow of the querying/reading of PNF descriptor information

This clause describes the procedure for querying information about multiple PNF descriptor resources and reading information about an individual PNF descriptor resource.

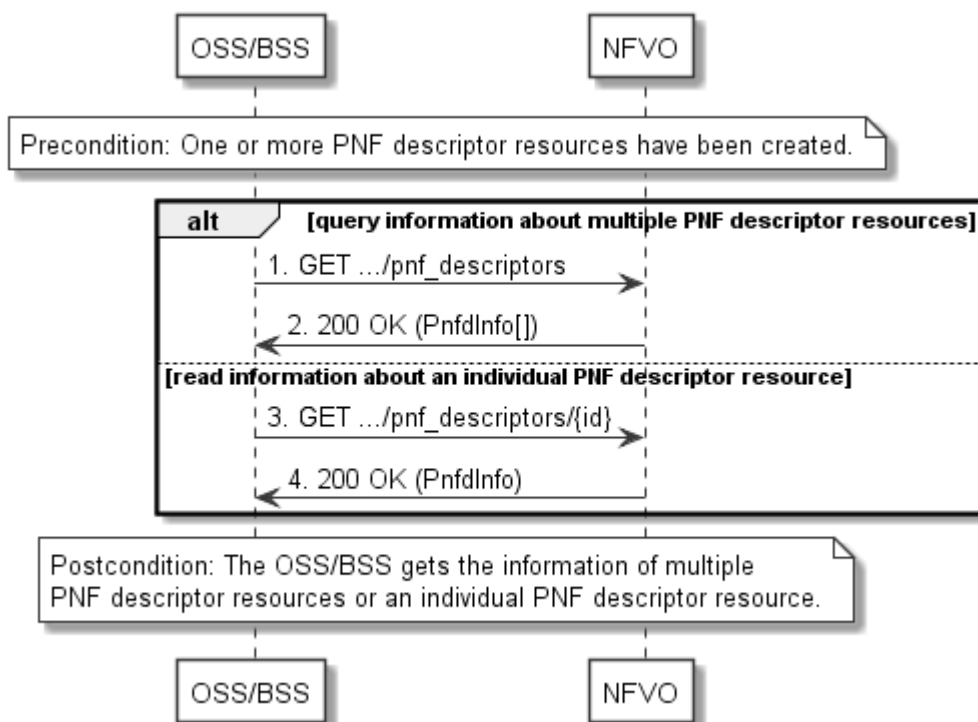


Figure 5.3.11-1: Flow of the querying/reading of PNF descriptor resources

Precondition: One or more PNF descriptor resources have been created.

The procedure consists of the following steps as illustrated in Figure 5.3.11-1:

- 1) If the OSS/BSS intends to query information about multiple PNF descriptor resources, it sends a GET request to the "PNF descriptors" resource.
- 2) The NFVO returns a "200 OK" response, and includes in the payload body zero or more data structures of type "PnfdInfo".
- 3) If the OSS/BSS intends to read information about an individual PNF descriptor resource, the OSS/BSS sends a GET request to the "Individual PNF descriptor" resource, addressed by the appropriate PnfdInfo identifier in its resource URI.
- 4) The NFVO returns a "200 OK" response, and includes in the payload body a data structure of type "PnfdInfo".

Postcondition: Upon successful completion, the OSS/BSS gets the information of multiple (i.e. zero or more) PNF descriptor resources or an individual PNF descriptor resource.

Error handling: In case of failure, appropriate error information is provided in the response.

5.3.11a Flow of reading the PNFD

This clause describes the procedure for reading an onboarded PNFD.

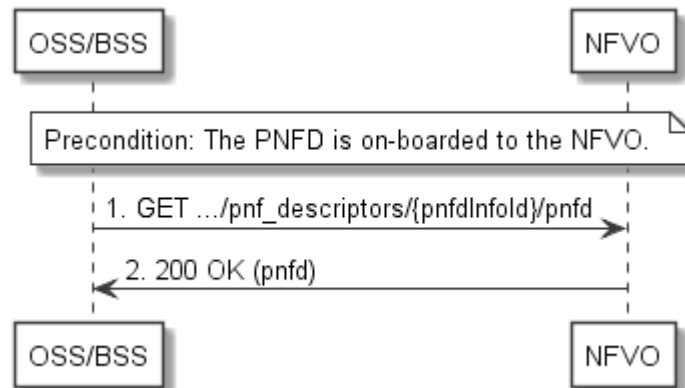


Figure 5.3.11a-1: Flow of reading a PNFD

Precondition: The PNFD is onboarded to the NFVO.

The procedure consists of the following steps as illustrated in Figure 5.3.11a-1:

- 1) The OSS/BSS sends a GET request to the "pnfd" resource.
- 2) The NFVO returns a "200 OK" response and includes a copy of the PNFD in the payload body.

5.3.11b Flow of fetching the manifest file of an onboarded PNFD archive

This clause describes the procedure for fetching the manifest file of an onboarded PNFD archive.

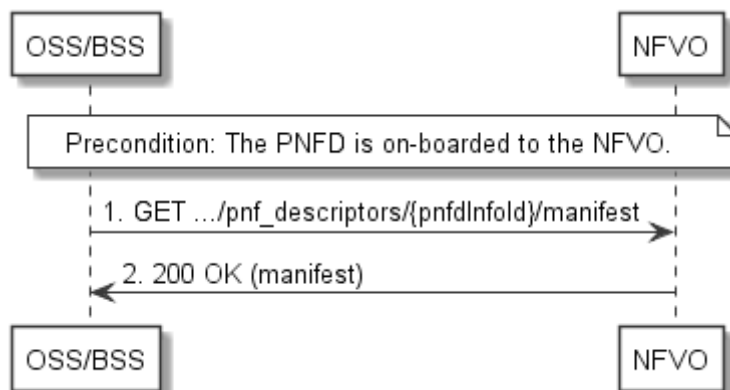


Figure 5.3.11b-1: Flow of fetching the manifest file of an onboarded PNFD archive

Precondition: The PNFD is onboarded to the NFVO.

The procedure consists of the following steps as illustrated in Figure 5.3.11b-1:

- 1) The OSS/BSS sends a GET request to the "manifest" resource.
- 2) The NFVO returns a "200 OK" response and includes the manifest file in the payload body.

5.3.12 Flow of managing subscriptions

This clause describes the procedure for creating, reading and terminating subscriptions to notifications related to NSD management.

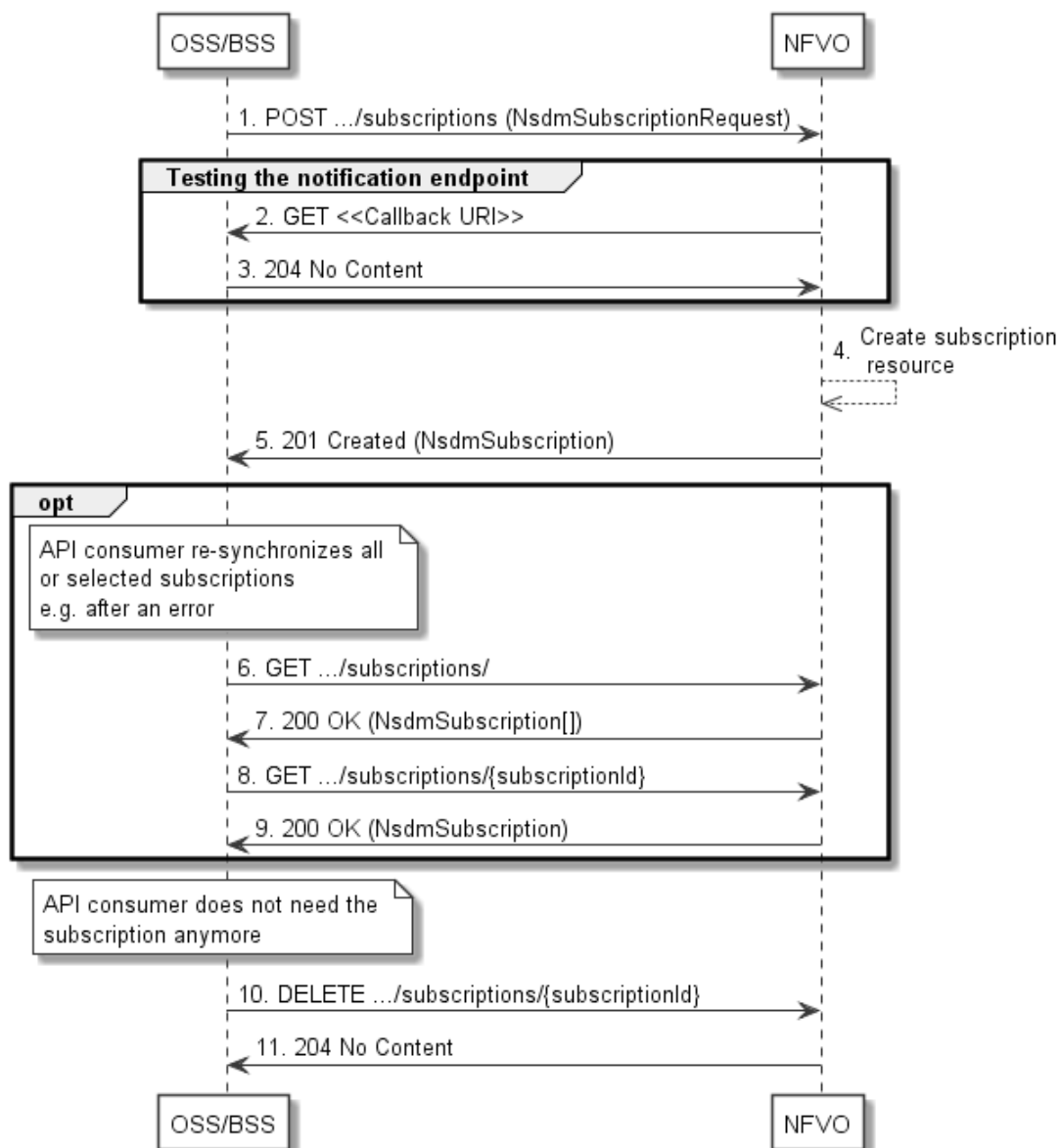


Figure 5.3.12-1: Flow of managing subscriptions

The procedure consists of the following steps as illustrated in Figure 5.3.12-1:

- 1) The OSS/BSS sends a POST request to the "Subscriptions" resource including in the payload body a data structure of type "NsdmSubscriptionRequest". This data structure contains filtering criteria and a callback URI to which the NFVO will subsequently send notifications about events that match the filter.
- 2) To test the notification endpoint that has been registered by the OSS/BSS as part of the subscription, the NFVO sends a GET request to the notification endpoint URI.
- 3) The OSS/BSS returns a "204 No Content" response to indicate success.
- 4) The NFVO creates a new subscription for notifications related to NS fault management, and a resource that represents this subscription.
- 5) The NFVO returns a "201 Created" response containing a data structure of type "NsdmSubscription", representing the subscription resource just created by the NFVO, and provides the URI of the newly-created resource in the "Location" HTTP header.
- 6) If desired, e.g. to recover from an error situation, the OSS/BSS can query information about its subscriptions by sending a GET request to the "Subscriptions" resource.

- 7) In that case, the NFVO returns a "200 OK" response that contains the list of representations of all existing subscriptions that were created by the NFVO.
- 8) If desired, e.g. to recover from an error situation, the OSS/BSS can read information about a particular subscription by sending a GET request to the resource representing that individual subscription.
- 9) In that case, the NFVO returns a "200 OK" response that contains a representation of that individual subscription.
- 10) When the OSS/BSS does not need the subscription anymore, it terminates the subscription by sending a DELETE request to the resource that represents the individual subscription.
- 11) The NFVO acknowledges the successful termination of the subscription by returning a "204 No Content" response.

Error handling: The NFVO rejects a subscription if the subscription information is not valid: endpoint cannot be reached, subscription information is malformed, etc.

5.3.13 Flow of sending notifications

This clause describes the procedure for sending notifications related to NSD management.

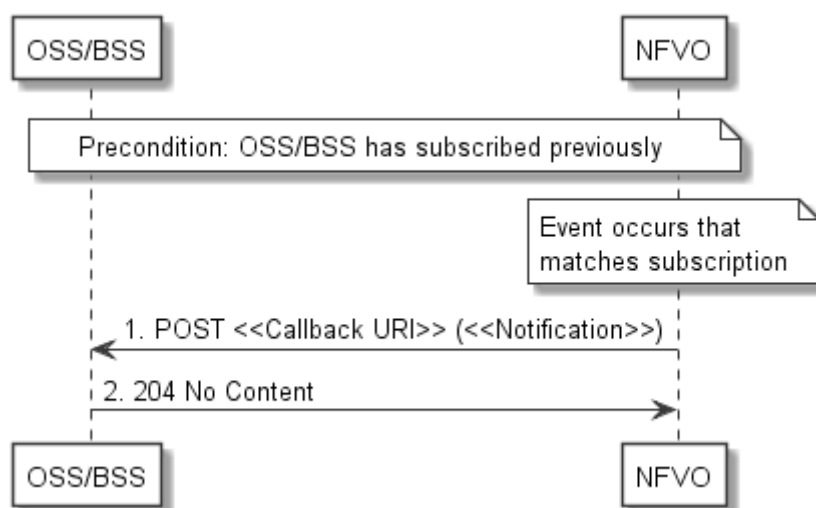


Figure 5.3.13-1: Flow of sending notifications

Precondition: The OSS/BSS has subscribed previously for notifications related to NSD management.

The procedure consists of the following steps as illustrated in Figure 5.3.13-1:

- 1) If an event occurs that matches the filtering criteria which are part of the subscription, the NFVO generates a notification that includes information about the event, and sends it in the body of a POST request to the URI which the NFVO has registered as part of the subscription request. The variable <<Notification>> in the flow is a placeholder for the different types of notifications that can be sent by this API (see clauses 5.5.2.9, 5.5.2.10, 5.5.2.11 and 5.5.2.12).
- 2) The OSS/BSS acknowledges the successful delivery of the notification by returning a "204 No Content" response.

Error handling: If the NFVO does not receive the "204 No Content" response from the OSS/BSS, it can retry sending the notification.

5.4 Resources

5.4.1 Introduction

This clause defines all the resource and methods provided by the NSD management interface.

The on-boarding of an NSD includes:

- 1) Creation of an individual NS descriptor resource.
- 2) Uploading the NSD content.
- 3) Validation of the NSD inside the NFVO.

In the present document, the NSD is referred to as "on-boarded" only after these three procedures are successfully accomplished.

NOTE: Annex B describes the state model of NSD in the NFVO. It includes the state models for two phases, i.e. onboarding phase of NSD and operational phase of NSD.

Further, the on-boarding of a PNFD includes:

- 1) Creation of an individual PNF descriptor resource.
- 2) Uploading the PNFD.
- 3) Processing the PNFD, including validation, inside the NFVO.

A PNFD is referred as "on-boarded" only after these three procedures are successfully accomplished.

5.4.1a Resource: API versions

The "API versions" resources as defined in clause 9.3.3 of ETSI GS NFV-SOL 013 [16] are part of the NSD management interface.

5.4.2 Resource: NS Descriptors

5.4.2.1 Description

This resource represents NS descriptors. It can be used to create an individual NS descriptor resource, and to query multiple NS descriptor resources.

5.4.2.2 Resource definition

The resource URI is:

{apiRoot}/nsd/{apiMajorVersion}/ns_descriptors

This resource shall support the resource URI variables defined in Table 5.4.2.2-1.

Table 5.4.2.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 5.1a.

5.4.2.3 Resource methods

5.4.2.3.1 POST

The POST method is used to create a new NS descriptor resource.

This method shall follow the provisions specified in the Tables 5.4.2.3.1-1 and 5.4.2.3.1-2 for URI query parameters, request and response data structures, and response codes.

Table 5.4.2.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
none supported		

Table 5.4.2.3.1-2: Details of the POST request/response on this resource

Request body	Data type	Cardinality	Description	
	CreateNsdInfoRequest	1	Parameters of creating an NS descriptor resource, as defined in clause 5.5.2.3.	
Response body	Data type	Cardinality	Response Codes	Description
	NsdInfo	1	201 Created	<p>Shall be returned when a new "Individual NS descriptor" resource and the associated NS descriptor identifier has been created successfully.</p> <p>The response body shall contain a representation of the created NS descriptor resource, as defined in clause 5.5.2.2.</p> <p>The HTTP response shall include a "Location" HTTP header that contains the resource URI of the created NS descriptor resource.</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

5.4.2.3.2 GET

The GET method queries information about multiple NS descriptor resources.

This method shall follow the provisions specified in the Tables 5.4.2.3.2-1 and 5.4.2.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 5.4.2.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
filter	0..1	<p>Attribute-based filtering expression according to clause 5.2 of ETSI GS NFV-SOL 013 [16].</p> <p>The NFVO shall support receiving this filtering parameter as part of the URI query string. The OSS/BSS may supply this parameter.</p> <p>All attribute names that appear in the NsdInfo and in data types referenced from it shall be supported by the NFVO in the filter expression.</p>
all_fields	0..1	Include all complex attributes in the response. See clause 5.3 of ETSI GS NFV-SOL 013 [16] for details. The NFVO shall support this parameter.
fields	0..1	Complex attributes to be included into the response. See clause 5.3 of ETSI GS NFV-SOL 013 [16] for details. The NFVO should support this parameter.

Name	Cardinality	Description
exclude_fields	0..1	Complex attributes to be excluded from the response. See clause 5.3 of ETSI GS NFV-SOL 013 [16] for details. The NFVO should support this parameter.
exclude_default	0..1	Indicates to exclude the following complex attributes from the response. See clause 5.3 of ETSI GS NFV-SOL 013 [16] for details. The NFVO shall support this parameter. The following attributes shall be excluded from the NsdInfo structure in the response body if this parameter is provided, or none of the parameters "all_fields", "fields", "exclude_fields", "exclude_default" are provided: <ul style="list-style-type: none"> - userDefinedData - onboardingFailureDetails
nextpage_opaque_marker	0..1	Marker to obtain the next page of a paged response. Shall be supported by the NFVO if the NFVO supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource.

Table 5.4.2.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	NsdInfo	0..N	200 OK	Shall be returned when information about zero or more NS descriptors has been queried successfully. The response body shall contain in an array the representations of zero or more NS descriptors, as defined in clause 5.5.2.2. If the NFVO supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource, inclusion of the Link HTTP header in this response shall follow the provisions in clause 5.4.2.3 of ETSI GS NFV-SOL 013 [16].
	ProblemDetails	1	400 Bad Request	Shall be returned upon the following error: Invalid attribute-based filtering expression. The response body shall contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.
	ProblemDetails	1	400 Bad Request	Shall be returned upon the following error: Invalid attribute selector. The response body shall contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.
	ProblemDetails	1	400 Bad Request	Shall be returned upon the following error: Response too big. If the NFVO supports alternative 1 (error) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource, this error response shall follow the provisions in clause 5.4.2.2 of ETSI GS NFV-SOL 013 [16].
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

5.4.2.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

5.4.2.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

5.4.2.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

5.4.3 Resource: Individual NS Descriptor

5.4.3.1 Description

This task resource represents an individual NS descriptor. The API consumer can use this resource to modify, delete and read the information of the individual NS descriptor.

5.4.3.2 Resource definition

The resource URI is:

{apiRoot}/nsd/{apiMajorVersion}/ns_descriptors/{nsdInfoId}

This resource shall support the resource URI variables defined in Table 5.4.3.2-1.

Table 5.4.3.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 5.1a.
nsdInfoId	Identifier of the individual NS descriptor resource. See note.
NOTE:	This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request creating a new NS descriptor resource. It can also be retrieved from the "id" attribute in the payload body of that response.

5.4.3.3 Resource methods

5.4.3.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

5.4.3.3.2 GET

The GET method reads information about an individual NS descriptor.

This method shall follow the provisions specified in the Tables 5.4.3.3.2-1 and 5.4.3.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 5.4.3.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
None supported		

Table 5.4.3.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	NsdInfo	1	200 OK	Shall be returned when information about the individual NS descriptor has been read successfully. The response body shall contain a representation of the individual NS descriptor, as defined in clause 5.5.2.2.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

5.4.3.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

5.4.3.3.4 PATCH

The PATCH method modifies the operational state and/or user defined data of an individual NS descriptor resource.

This method can be used to:

- 1) Enable a previously disabled individual NS descriptor resource, allowing again its use for instantiation of new network service with this descriptor. The usage state (i.e. "IN_USE/NOT_IN_USE") shall not change as a result.
- 2) Disable a previously enabled individual NS descriptor resource, preventing any further use for instantiation of new network service(s) with this descriptor. The usage state (i.e. "IN_USE/NOT_IN_USE") shall not change as a result.
- 3) Modify the user defined data of an individual NS descriptor resource.

This method shall follow the provisions specified in the Tables 5.4.3.3.4-1 and 5.4.3.3.4-2 for URI query parameters, request and response data structures, and response codes.

Table 5.4.3.3.4-1: URI query parameters supported by the PATCH method on this resource

Name	Cardinality	Description
None supported		

Table 5.4.3.3.4-2: Details of the PATCH request/response on this resource

Request body	Data type	Cardinality	Description	
	NsdInfoModifications	1	Parameters for the modification of an individual NS descriptor resource, as defined in clause 5.5.2.1. The Content-Type header shall be set to "application/merge-patch+json" according to IETF RFC 7396 [11].	
Response body	Data type	Cardinality	Response Codes	Description
	NsdInfoModifications	1	200 OK	Shall be returned when the operation has been accepted and completed successfully. The response body shall contain attribute modifications for an 'Individual NS Descriptor' resource (see clause 5.5.2.1).
	ProblemDetails	1	409 Conflict	Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the "Individual NS Descriptor" resource. Typically, this is due to an operational state mismatch, i.e. enable an already enabled or disable an already disabled individual NS descriptor resource, or the "nsdOnboardingState" is not ONBOARDED. The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.
	ProblemDetails	0..1	412 Precondition failed	Shall be returned upon the following error: A precondition given in an HTTP request header is not fulfilled. Typically, this is due to an ETag mismatch, indicating that the resource was modified by another entity. The response body should contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.
ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.	

5.4.3.3.5 DELETE

The DELETE method deletes an individual NS descriptor resource.

An individual NS descriptor resource can only be deleted when there is no NS instance using it (i.e. usageState = NOT_IN_USE) and has been disabled already (i.e. operationalState = DISABLED). Otherwise, the DELETE method shall fail.

This method shall follow the provisions specified in the Tables 5.4.3.3.5-1 and 5.4.3.3.5-2 for URI query parameters, request and response data structures, and response codes.

Table 5.4.3.3.5-1: URI query parameters supported by the DELETE method on this resource

Name	Cardinality	Description
None supported		

Table 5.4.3.3.5-2: Details of the DELETE request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	n/a		204 No Content	Shall be returned when the operation has completed successfully. The response body shall be empty.
	ProblemDetails	1	409 Conflict	Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource. Typically, this is due to the fact the NS descriptor resource is in the enabled operational state (i.e. operationalState = ENABLED) or there are running NS instances using the concerned individual NS descriptor resource (i.e. usageState = IN_USE). The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

5.4.4 Resource: NSD archive content

5.4.4.1 Description

This resource represents the content of the individual NSD archive. The API consumer can use this resource to upload and download the content of the NSD.

5.4.4.2 Resource definition

The resource URI is:

{apiRoot}/nsd/{apiMajorVersion}/ns_descriptors/{nsdInfoId}/nsd_content

This resource shall support the resource URI variables defined in Table 5.4.4.2-1.

Table 5.4.4.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 5.1a.
nsdInfoId	Identifier of the individual NS descriptor. See note.
NOTE:	This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request creating a new NS descriptor resource. It can also be retrieved from the "id" attribute in the payload body of that response.

5.4.4.3 Resource methods

5.4.4.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

5.4.4.3.2 GET

The GET method fetches the content of the NSD archive.

The NSD archive is implemented as a single zip file.

The content of the NSD archive is provided as onboarded, i.e. depending on the security option used, the CSAR or the CSAR wrapped in a ZIP archive together with an external signature is returned, as defined in clause 5.1 of ETSI GS NFV-SOL 007 [18].

NOTE: Information about the applicable security option can be obtained by evaluating the "archiveSecurityOption" attribute in the "nsdInfo" structure.

This method shall follow the provisions specified in the Tables 5.4.4.3.2-1 and 5.4.4.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 5.4.4.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
none supported		

Table 5.4.4.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			<p>The request may contain a "Range" HTTP header to obtain single range of bytes from the NSD archive. This can be used to continue an aborted transmission.</p> <p>If the Range header is present in the request and the NFVO does not support responding to range requests with a 206 response, it shall return a 200 OK response instead as defined below.</p>
Response body	Data type	Cardinality	Response Codes	Description
	n/a	1	200 OK	<p>Shall be returned when the content of the NSD has been read successfully.</p> <p>The payload body shall contain a copy of the ZIP file that contains the NSD file structure.</p> <p>The "Content-Type" HTTP header shall be set to "application/zip".</p>
	n/a	1	206 Partial Content	<p>If the NFVO supports range requests, this response shall be returned when a single consecutive byte range from the content of the NSD file has been read successfully according to the request.</p> <p>The response body shall contain the requested part of the NSD archive.</p> <p>The "Content-Range" HTTP header shall be provided according to IETF RFC 7233 [10].</p> <p>The "Content-Type" HTTP header shall be set as defined above for the "200 OK" response.</p>
	ProblemDetails	1	409 Conflict	<p>Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource.</p> <p>Typically, this is due to the fact "nsdOnboardingState" has a value different from ONBOARDED.</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</p>
	ProblemDetails	0..1	416 Range Not Satisfiable	<p>The byte range passed in the "Range" header did not match any available byte range in the NSD file (e.g. "access after end of file").</p> <p>The response body may contain a ProblemDetails structure.</p>
ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.	

5.4.4.3.3 PUT

The PUT method is used to upload the content of an NSD archive.

The NSD to be uploaded is implemented as a single ZIP file as defined in clause 5.4.4.3.2.

The "Content-Type" HTTP header in the PUT request shall be set to "application/zip".

This method shall follow the provisions specified in the Tables 5.4.4.3.3-1 and 5.4.4.3.3-2 for URI query parameters, request and response data structures, and response codes.

Table 5.4.4.3.3-1: URI query parameters supported by the PUT method on this resource

Name	Cardinality	Description
none supported		

Table 5.4.4.3.3-2: Details of the PUT request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a	1	The payload body contains a ZIP file that represents the NSD archive, as specified above. The request shall set the "Content-Type" HTTP header to "application/zip".	
Response body	Data type	Cardinality	Response Codes	Description
	n/a	1	202 Accepted	Shall be returned when the NSD archive has been accepted for uploading, but the processing has not been completed. It is expected to take some time for processing (asynchronous mode). The response body shall be empty. See note.
	n/a	1	204 No Content	The NSD content successfully uploaded and validated (synchronous mode). The response body shall be empty.
	ProblemDetails	1	409 Conflict	Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource. Typically, this is due to the fact that the "nsdOnboardingState" attribute has a value other than "CREATED" or "ERROR". The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.
ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error "response code" as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.	
NOTE: The API consumer can track the uploading progress by receiving the "NsdOnBoardingNotification" and "NsdOnBoardingFailureNotification" from the NFVO or by reading the status of the individual NS descriptor resource using the GET method.				

5.4.4.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

5.4.4.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

5.4.4a Resource: NSD

5.4.4a.1 Description

This resource represents the NSD contained in an on-boarded NSD archive. The API consumer can use this resource to obtain the content of the NSD.

5.4.4a.2 Resource definition

The resource URI is:

{apiRoot}/nsd/{apiMajorVersion}/ns_descriptors/{nsdInfoId}/nsd

This resource shall support the resource URI variables defined in Table 5.4.4a.2-1.

Table 5.4.4a.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 5.1a.
nsdInfoId	Identifier of the individual NS descriptor. See note.
NOTE:	This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request creating a new NS descriptor resource. It can also be retrieved from the "id" attribute in the payload body of that response.

5.4.4a.3 Resource methods

5.4.4a.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

5.4.4a.3.2 GET

The GET method reads the content of the NSD within an NSD archive.

The NSD can be implemented as a single file or as a collection of multiple files. If the NSD is implemented in the form of multiple files, a ZIP file embedding these files shall be returned. If the NSD is implemented as a single file, either that file or a ZIP file embedding that file shall be returned.

The selection of the format is controlled by the "Accept" HTTP header passed in the GET request:

- If the "Accept" header contains only "text/plain" and the NSD is implemented as a single file, the file shall be returned; otherwise, an error message shall be returned.
- If the "Accept" header contains only "application/zip", the single file or the multiple files that make up the NSD shall be returned embedded in a ZIP file.
- If the "Accept" header contains both "text/plain" and "application/zip", it is up to the NFVO to choose the format to return for a single-file NSD; for a multi-file NSD, a ZIP file shall be returned.

The default format of the ZIP file shall comply with the CSAR format as specified in ETSI GS NFV-SOL 007 [18] where only the YAML files representing the NSD, and information necessary to navigate the ZIP file and to identify the file that is the entry point for parsing the NSD and (if requested) further security information are included, and other artifacts referenced from the YAML files are excluded. This means that the content of the ZIP archive shall contain the following files from the NSD archive:

- TOSCA.meta (if available in the NSD archive).
- The main TOSCA definitions YAML file (either as referenced from TOSCA.meta or available as a file with the extension ".yaml" or ".yml" from the root of the archive).
- Every component of the NSD referenced (recursively) from the main TOSCA definitions YAML file.

NOTE 1: For a NSD based on TOSCA, it includes all the imported type definition files as indicated in the top level service template and in any of the lower level service template if it has any as described in ETSI GS NFV-SOL 001 [i.1].

NOTE 2: For a NSD based on YANG, it includes the file as indicated by the "yang_definitions" keyname in the metadata section of the main yaml file as described in ETSI GS NFV-SOL 007 [18].

- The related security information, if the "include_signatures" URI parameter is provided, as follows:
 - the manifest file;
 - the singleton certificate file in the root of the NSD archive (if available in the NSD archive);
 - the signing certificates of the individual files included in the ZIP archive (if available in the NSD archive);
 - the signatures of the individual files (if available in the NSD archive).

This method shall follow the provisions specified in the Tables 5.4.4a.3.2-1 and 5.4.4a.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 5.4.4a.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
include_signatures	0..1	<p>If this parameter is provided, the NFVO shall include in the ZIP file the security information as specified above.</p> <p>This URI query parameter is a flag, i.e. it shall have no value.</p> <p>The NFVO shall support this parameter.</p>

Table 5.4.4a.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a		The request shall contain the appropriate entries in the "Accept" HTTP header as defined above.	
Response body	Data type	Cardinality	Response Codes	Description
	n/a	1	200 OK	<p>Shall be returned when the content of the NSD has been read successfully.</p> <p>The payload body shall contain a copy of the file representing the NSD or a ZIP file that contains the file or multiple files representing the NSD, as specified above.</p> <p>The "Content-Type" HTTP header shall be set according to the format of the returned file. It shall be set to "text/plain" for a YAML file.</p>
	ProblemDetails	0..1	406 Not Acceptable	<p>If the "Accept" header does not contain at least one name of a content type for which the NFVO can provide a representation of the NSD, the NFVO shall respond with this response code.</p> <p>The "ProblemDetails" structure may be included with the "detail" attribute providing more information about the error.</p>
	ProblemDetails	1	409 Conflict	<p>Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource.</p> <p>Typically, this is due to the fact that "nsdOnboardingState" has a value different from "ONBOARDED".</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</p>
ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.	

5.4.4a.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

5.4.4a.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

5.4.4a.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

5.4.4b Resource: NSD archive manifest

5.4.4b.1 Description

This resource represents the manifest file contained in an on-boarded NSD archive. The API consumer can use this resource to obtain the content of the manifest file.

5.4.4b.2 Resource definition

The resource URI is:

{apiRoot}/nsd/{apiMajorVersion}/ns_descriptors/{nsdInfoId}/manifest

This resource shall support the resource URI variables defined in Table 5.4.4b.2-1.

Table 5.4.4b.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 5.1a.
nsdInfoId	Identifier of the individual NS descriptor. See note.
NOTE:	This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request creating a new NS descriptor resource. It can also be retrieved from the "id" attribute in the payload body of that response.

5.4.4b.3 Resource methods

5.4.4b.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

5.4.4b.3.2 GET

The GET method reads the content of the manifest file within an NSD archive.

This method shall follow the provisions specified in the Tables 5.4.4b.3.2-1 and 5.4.4b.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 5.4.4b.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
include_signatures	0..1	<p>If this parameter is provided, the NFVO shall return the manifest and related security information (signature and certificate) either in a single text file if the signature and certificate are included in the manifest file, or in a zip file containing the manifest and the certificate file, if this is provided as a separate file in the NSD archive.</p> <p>If this parameter is not given, the NFVO shall provide only a copy of the manifest file, as onboarded. If the security information is included in the onboarded manifest, it shall also be included in the returned copy.</p> <p>This URI query parameter is a flag, i.e. it shall have no value.</p> <p>The NFVO shall support this parameter.</p>

Table 5.4.4b.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	n/a	1	200 OK	<p>Shall be returned when the content of the manifest file has been read successfully.</p> <p>If the "include_signatures" URI query parameter was absent in the request, or if the manifest file has all security-related information embedded (i.e. there is no separate certificate file), the payload body shall contain a copy of the manifest file of the NSD archive, and the "Content-Type" HTTP header shall be set to "text/plain".</p> <p>If the "include_signatures" URI query parameter was present in the related request and the manifest file does not have all the security-related information embedded (i.e. there is a separate certificate file), the "Content-Type" HTTP header shall be set to "application/zip" and the payload body shall contain a ZIP archive which includes:</p> <ul style="list-style-type: none"> - a copy of the manifest file of the NSD archive; - a copy of the related individual certificate file.
	ProblemDetails	1	409 Conflict	<p>Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource.</p> <p>Typically, this is due to the fact that "nsdOnboardingState" has a value different from "ONBOARDED".</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

5.4.4b.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

5.4.4b.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

5.4.4b.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

5.4.5 Resource: PNF Descriptors

5.4.5.1 Description

This resource represents PNF descriptors and it can be used to create an individual PNF descriptor resource, and to query PNF descriptor resources.

5.4.5.2 Resource definition

The resource URI is:

{apiRoot}/nsd/{apiMajorVersion}/pnf_descriptors

This resource shall support the resource URI variables defined in Table 5.4.5.2-1.

Table 5.4.5.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 5.1a.

5.4.5.3 Resource methods

5.4.5.3.1 POST

The POST method is used to create a new PNF descriptor resource.

This method shall follow the provisions specified in the Tables 5.4.5.3.1-1 and 5.4.5.3.1-2 for URI query parameters, request and response data structures, and response codes.

Table 5.4.5.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
none supported		

Table 5.4.5.3.1-2: Details of the POST request/response on this resource

Request body	Data type	Cardinality	Description	
	CreatePnfdInfoRequest	1	Parameters of creating a PNF descriptor resource, as defined in clause 5.5.2.6.	
Response body	Data type	Cardinality	Response Codes	Description
	PnfdInfo	1	201 Created	<p>Shall be returned when a new "Individual PNF descriptor" resource and the associated PNF descriptor identifier has been created successfully.</p> <p>The response body shall contain a representation of the created PNF descriptor resource, as defined in clause 5.5.2.5.</p> <p>The HTTP response shall include a "Location" HTTP header that contains the resource URI of the created PNF descriptor resource.</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

5.4.5.3.2 GET

The GET method queries information about multiple PNF descriptor resources.

This method shall follow the provisions specified in the Tables 5.4.5.3.2-1 and 5.4.5.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 5.4.5.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
filter	0..1	<p>Attribute-based filtering expression according to clause 5.2 of ETSI GS NFV-SOL 013 [16].</p> <p>The NFVO shall support receiving this filtering parameter as part of the URI query string. The OSS/BSS may supply this parameter.</p> <p>All attribute names that appear in the PnfdInfo and in data types referenced from it shall be supported by the NFVO in the filter expression.</p>
all_fields	0..1	Include all complex attributes in the response. See clause 5.3 of ETSI GS NFV-SOL 013 [16] for details. The NFVO shall support this parameter.
fields	0..1	Complex attributes to be included into the response. See clause 5.3 of ETSI GS NFV-SOL 013 [16] for details. The NFVO should support this parameter.
exclude_fields	0..1	Complex attributes to be excluded from the response. See clause 5.3 of ETSI GS NFV-SOL 013 [16] for details. The NFVO should support this parameter.
exclude_default	0..1	<p>Indicates to exclude the following complex attributes from the response. See clause 5.3 of ETSI GS NFV-SOL 013 [16] for details. The NFVO shall support this parameter.</p> <p>The following attributes shall be excluded from the PnfdInfo structure in the response body if this parameter is provided, or none of the parameters "all_fields", "fields", "exclude_fields", "exclude_default" are provided:</p> <ul style="list-style-type: none"> - userDefinedData - onboardingFailureDetails
nextpage_opaque_marker	0..1	Marker to obtain the next page of a paged response. Shall be supported by the NFVO if the NFVO supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource.

Table 5.4.5.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
n/a				
Response body	Data type	Cardinality	Response Codes	Description
	PnfdInfo	0..N	200 OK	<p>Shall be returned when information about zero or more PNF descriptors has been queried successfully.</p> <p>The response body shall contain in an array the representations of zero or more PNF descriptors, as defined in clause 5.5.2.5.</p> <p>If the NFVO supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource, inclusion of the Link HTTP header in this response shall follow the provisions in clause 5.4.2.3 of ETSI GS NFV-SOL 013 [16].</p>
	ProblemDetails	1	400 Bad Request	<p>Shall be returned upon the following error: Invalid attribute-based filtering expression.</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.</p>
	ProblemDetails	1	400 Bad Request	<p>Shall be returned upon the following error: Invalid attribute selector.</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.</p>
	ProblemDetails	1	400 Bad Request	<p>Shall be returned upon the following error: Response too big.</p> <p>If the NFVO supports alternative 1 (error) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource, this error response shall follow the provisions in clause 5.4.2.2 of ETSI GS NFV-SOL 013 [16].</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

5.4.5.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

5.4.5.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

5.4.5.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

5.4.6 Resource: Individual PNF Descriptor

5.4.6.1 Description

This resource represents an individual PNF descriptor. The API consumer can use this resource to modify, delete and read the information of the individual PNF descriptor resource.

5.4.6.2 Resource definition

The resource URI is:

{apiRoot}/nsd/{apiMajorVersion}/pnf_descriptors/{pnfdInfoId}

This resource shall support the resource URI variables defined in Table 5.4.6.2-1.

Table 5.4.6.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 5.1a.
pnfdInfoId	Identifier of the individual PNF descriptor resource. See note.
NOTE:	This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request creating a new PNF descriptor resource. It can also be retrieved from the "id" attribute in the payload body of that response.

5.4.6.3 Resource methods

5.4.6.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

5.4.6.3.2 GET

The GET method reads information about an individual PNF descriptor.

This method shall follow the provisions specified in the Tables 5.4.6.3.2-1 and 5.4.6.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 5.4.6.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
None supported		

Table 5.4.6.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	PnfdInfo	1	200 OK	Shall be returned when information about the individual PNF descriptor has been read successfully. The response body shall contain a representation of the individual PNF descriptor, as defined in clause 5.5.2.5.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

5.4.6.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

5.4.6.3.4 PATCH

The PATCH method modifies the user defined data of an individual PNF descriptor resource.

This method shall follow the provisions specified in the Tables 5.4.6.3.4-1 and 5.4.6.3.4-2 for URI query parameters, request and response data structures, and response codes.

Table 5.4.6.3.4-1: URI query parameters supported by the PATCH method on this resource

Name	Cardinality	Description
None supported		

Table 5.4.6.3.4-2: Details of the PATCH request/response on this resource

Request body	Data type	Cardinality	Description	
	PnfdInfoModifications	1	Parameters for the modification of an individual PNF descriptor resource, as defined in clause 5.5.2.4. The Content-Type header shall be set to "application/merge-patch+json" according to IETF RFC 7396 [11].	
Response body	Data type	Cardinality	Response Codes	Description
	PnfdInfoModifications	1	200 OK	Shall be returned when the operation has been accepted and completed successfully. The response body shall contain attribute modifications for an 'Individual PNF Descriptor' resource (see clause 5.5.2.4).
	ProblemDetails	0..1	412 Precondition failed	Shall be returned upon the following error: A precondition given in an HTTP request header is not fulfilled. Typically, this is due to an ETag mismatch, indicating that the resource was modified by another entity. The response body should contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.
ProblemDetails	See clause 6.4 of [16]		4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

5.4.6.3.5 DELETE

The DELETE method deletes an individual PNF descriptor resource.

An individual PNF descriptor resource can only be deleted when there is no NS instance using it or there is NSD referencing it.

To delete all PNF versions identified by a particular value of the "pnfdInvariantId" attribute, the procedure is to first use the GET method with filter "pnfdInvariantId" towards the PNF descriptors resource to find all versions of the PNF. Then, the API consumer uses the DELETE method described in this clause to delete each PNF version individually.

This method shall follow the provisions specified in the Tables 5.4.6.3.5-1 and 5.4.6.3.5-2 for URI query parameters, request and response data structures, and response codes.

Table 5.4.6.3.5-1: URI query parameters supported by the DELETE method on this resource

Name	Cardinality	Description
none supported		

Table 5.4.6.3.5-2: Details of the DELETE request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	n/a		204 No Content	Shall be returned when the operation has completed successfully. The response body shall be empty.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

5.4.7 Resource: PNFD archive content

5.4.7.1 Description

This resource represents the content of the individual PNF archive, i.e. PNFD content. The API consumer can use this resource to upload and download the content of the PNFD.

5.4.7.2 Resource definition

The resource URI is:

{apiRoot}/nsd/{apiMajorVersion}/pnf_descriptors/{pnfdInfoId}/pnfd_content

This resource shall support the resource URI variables defined in Table 5.4.7.2-1.

Table 5.4.7.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 5.1a.
pnfdInfoId	Identifier of the individual PNF descriptor. See note.
NOTE:	This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request creating a new PNF descriptor resource. It can also be retrieved from the "id" attribute in the payload body of that response.

5.4.7.3 Resource methods

5.4.7.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

5.4.7.3.2 GET

The GET method fetches the content of the PNFD archive.

The content of the PNFD archive is provided as onboarded, i.e. depending on the security option used, the CSAR or the CSAR wrapped in a ZIP archive together with an external signature is returned, as defined in clause 5.1 of ETSI GS NFV-SOL 004 [5].

NOTE: Information about the applicable security option can be obtained by evaluating the "archiveSecurityOption" attribute in the "pnfdInfo" structure.

This method shall follow the provisions specified in the Tables 5.4.7.3.2-1 and 5.4.7.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 5.4.7.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
none supported		

Table 5.4.7.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			<p>The request may contain a "Range" HTTP header to obtain single range of bytes from the PNFD archive. This can be used to continue an aborted transmission.</p> <p>If the NFVO does not support range requests, the NFVO shall ignore the "Range" header, process the GET request, and return the whole PNFD archive with a 200 OK response (rather than returning a 4xx error status code).</p>
Response body	Data type	Cardinality	Response Codes	Description
	n/a	1	200 OK	<p>Shall be returned when the content of the PNFD archive has been read successfully.</p> <p>The payload body shall contain a copy of the PNFD archive.</p> <p>The "Content-Type" HTTP header shall be set to "application/zip".</p>
	n/a	1	206 Partial Content	<p>If the NFVO supports range requests, this response shall be returned when a single consecutive byte range from the content of the PNFD archive has been read successfully according to the request.</p> <p>The response body shall contain the requested part of the PNFD archive.</p> <p>The "Content-Range" HTTP header shall be provided according to IETF RFC 7233 [10].</p> <p>The "Content-Type" HTTP header shall be set as defined above for the "200 OK" response.</p>
	ProblemDetails	1	409 Conflict	<p>Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource.</p> <p>Typically, this is due to the fact pnfOnboardingState has a value different from ONBOARDED.</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</p>
	ProblemDetails	0..1	416 Range Not Satisfiable	<p>Shall be returned upon the following error: The byte range passed in the "Range" header did not match any available byte range in the PNFD archive (e.g. "access after end of file").</p> <p>The response body may contain a ProblemDetails structure.</p>
ProblemDetails	See clause 6.4 of [16]	4xx/5xx	<p>In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.</p>	

5.4.7.3.3 PUT

The PUT method is used to upload the content of a PNFD archive.

This method shall follow the provisions specified in the Tables 5.4.7.3.3-1 and 5.4.7.3.3-2 for URI query parameters, request and response data structures, and response codes.

Table 5.4.7.3.3-1: URI query parameters supported by the PUT method on this resource

Name	Cardinality	Description
none supported		

Table 5.4.7.3.3-2: Details of the PUT request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a	1	<p>The payload body contains a ZIP file representing the PNFD archive.</p> <p>The request shall set the "Content-Type" HTTP header to "application/zip".</p>	
Response body	Data type	Cardinality	Response Codes	Description
	n/a		202 Accepted	<p>Shall be returned when the PNFD archive has been accepted for uploading, but the processing has not been completed. It is expected to take some time for processing (asynchronous mode).</p> <p>The response body shall be empty. See note.</p>
	n/a	1	204 No Content	<p>Shall be returned when the PNFD archive content has been uploaded and validated successfully.</p> <p>The response body shall be empty.</p>
	ProblemDetails	1	409 Conflict	<p>Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource.</p> <p>Typically, this is due to the fact that the "pnfdOnboardingState" attribute has a value other than "CREATED" or "ERROR".</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</p>
ProblemDetails	See clause 6.4 of [16]	4xx/5xx	<p>In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.</p>	
<p>NOTE: The API consumer can track the uploading progress by receiving the "PnfdOnBoardingNotification" and "PnfdOnBoardingFailureNotification" from the NFVO or by reading the status of the individual PNF descriptor resource using the GET method.</p>				

5.4.7.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

5.4.7.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

5.4.7a Resource: PNFD

5.4.7a.1 Description

This resource represents the PNFD contained in an on-boarded PNFD archive. The API consumer can use this resource to obtain the content of the PNFD.

5.4.7a.2 Resource definition

The resource URI is:

{apiRoot}/nsd/{apiMajorVersion}/pnf_descriptors/{pnfdInfoId}/pnfd

This resource shall support the resource URI variables defined in Table 5.4.7a.2-1.

Table 5.4.7a.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 5.1a.
pnfdInfoId	Identifier of the individual PNF descriptor. See note.
NOTE:	This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request creating a new PNF descriptor resource. It can also be retrieved from the "id" attribute in the payload body of that response.

5.4.7a.3 Resource methods

5.4.7a.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

5.4.7a.3.2 GET

The GET method reads the content of the PNFD within a PNFD archive.

The PNFD can be implemented as a single file or as a collection of multiple files. If the PNFD is implemented in the form of multiple files, a ZIP file embedding these files shall be returned. If the PNFD is implemented as a single file, either that file or a ZIP file embedding that file shall be returned.

The selection of the format is controlled by the "Accept" HTTP header passed in the GET request:

- If the "Accept" header contains only "text/plain" and the PNFD is implemented as a single file, the file shall be returned; otherwise, an error message shall be returned.
- If the "Accept" header contains only "application/zip", the single file or the multiple files that make up the PNFD shall be returned embedded in a ZIP file.
- If the "Accept" header contains both "text/plain" and "application/zip", it is up to the NFVO to choose the format to return for a single-file PNFD; for a multi-file PNFD, a ZIP file shall be returned.

The default format of the ZIP file shall be the one specified in ETSI GS NFV-SOL 004 [5] where only the YAML files representing the PNFD, and information necessary to navigate the ZIP file and to identify the file that is the entry point for parsing the PNFD and (if requested) further security information are included. This means that the content of the ZIP archive shall contain the following files from the PNFD archive:

- TOSCA.meta (if available in the PNFD archive);
- the main service template (either as referenced from TOSCA.meta or available as a file with the extension ".yaml" or ".yml" from the root of the archive);
- every component of the PNFD referenced (recursively) from the main service template;
- the related security information, if the "include_signatures" URI parameter is provided, as follows:
 - the manifest file;
 - the singleton certificate file in the root of the PNFD archive (if available in the PNFD archive);

- the signing certificates of the individual files included in the ZIP archive (if available in the PNFD archive);
- the signatures of the individual files (if available in the PNFD archive).

This method shall follow the provisions specified in the Tables 5.4.7a.3.2-1 and 5.4.7a.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 5.4.7a.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
include_signatures	0..1	<p>If this parameter is provided, the NFVO shall include in the ZIP file the security information as specified above.</p> <p>This URI query parameter is a flag, i.e. it shall have no value.</p> <p>The NFVO shall support this parameter.</p>

Table 5.4.7a.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a		The request shall contain the appropriate entries in the "Accept" HTTP header as defined above.	
Response body	Data type	Cardinality	Response Codes	Description
	n/a	1	200 OK	<p>Shall be returned when the content of the PNFD has been read successfully.</p> <p>The payload body shall contain a copy of the file representing the PNFD or a ZIP file that contains the file or multiple files representing the PNFD, as specified above.</p> <p>The "Content-Type" HTTP header shall be set according to the format of the returned file. It shall be set to "text/plain" for a YAML file.</p>
	ProblemDetails	0..1	406 Not Acceptable	<p>If the "Accept" header does not contain at least one name of a content type for which the NFVO can provide a representation of the PNFD, the NFVO shall respond with this response code.</p> <p>The "ProblemDetails" structure may be included with the "detail" attribute providing more information about the error.</p>
	ProblemDetails	1	409 Conflict	<p>Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource.</p> <p>Typically, this is due to the fact that "pnfdOnboardingState" has a value different from "ONBOARDED".</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</p>
ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.	

5.4.7a.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

5.4.7a.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

5.4.7a.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

5.4.7b Resource: PNFD archive manifest

5.4.7b.1 Description

This resource represents the manifest file contained in an on-boarded PNFD archive. The API consumer can use this resource to obtain the content of the manifest file.

5.4.7b.2 Resource definition

The resource URI is:

{apiRoot}/nsd/{apiMajorVersion}/pnf_descriptors/{pnfdInfoId}/manifest

This resource shall support the resource URI variables defined in Table 5.4.7b.2-1.

Table 5.4.7b.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 5.1a.
pnfdInfoId	Identifier of the individual PNF descriptor. See note.
NOTE:	This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request creating a new PNF descriptor resource. It can also be retrieved from the "id" attribute in the payload body of that response.

5.4.7b.3 Resource methods

5.4.7b.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

5.4.7b.3.2 GET

The GET method reads the content of the manifest file within a PNFD archive.

This method shall follow the provisions specified in the Tables 5.4.7b.3.2-1 and 5.4.7b.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 5.4.7b.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
include_signatures	0..1	<p>If this parameter is provided, the NFVO shall return the manifest and related security information (signature and certificate) either in a single text file if the signature and certificate are included in the manifest file, or in a zip file containing the manifest and the certificate file, if this is provided as a separate file in the PNFD archive.</p> <p>If this parameter is not given, the NFVO shall provide only a copy of the manifest file, as onboarded. If the security information is included in the onboarded manifest, it shall also be included in the returned copy.</p> <p>This URI query parameter is a flag, i.e. it shall have no value.</p> <p>The NFVO shall support this parameter.</p>

Table 5.4.7b.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	n/a	1	200 OK	<p>Shall be returned when the content of the manifest file has been read successfully.</p> <p>If the "include_signatures" URI query parameter was absent in the request, or if the manifest file has all security-related information embedded (i.e. there is no separate certificate file), the payload body shall contain a copy of the manifest file of the PNFD archive, and the "Content-Type" HTTP header shall be set to "text/plain".</p> <p>If the "include_signatures" URI query parameter was present in the related request and the manifest file does not have all the security-related information embedded (i.e. there is a separate certificate file), the "Content-Type" HTTP header shall be set to "application/zip" and the payload body shall contain a ZIP archive which includes:</p> <ul style="list-style-type: none"> - a copy of the manifest file of the PNFD archive; - a copy of the related individual certificate file.
	ProblemDetails	1	409 Conflict	<p>Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource.</p> <p>Typically, this is due to the fact that "pnfdOnboardingState" has a value different from "ONBOARDED".</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

5.4.7b.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

5.4.7b.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

5.4.7b.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

5.4.8 Resource: Subscriptions

5.4.8.1 Description

This resource represents subscriptions. The API consumer can use this resource to subscribe to notifications related to NSD management and to query its subscriptions.

5.4.8.2 Resource definition

The resource URI is:

{apiRoot}/nsd/{apiMajorVersion}/subscriptions

This resource shall support the resource URI variables defined in Table 5.4.8.2-1.

Table 5.4.8.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 5.1a.

5.4.8.3 Resource methods

5.4.8.3.1 POST

The POST method creates a new subscription.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in the Tables 5.4.8.3.1-1 and 5.4.8.3.1-2.

Creation of two subscription resources with the same callback URI and the same filter can result in performance degradation and will provide duplicates of notifications to the OSS, and might make sense only in very rare use cases. Consequently, the NFVO may either allow creating a subscription resource if another subscription resource with the same filter and callback URI already exists (in which case it shall return the "201 Created" response code), or may decide to not create a duplicate subscription resource (in which case it shall return a "303 See Other" response code referencing the existing subscription resource with the same filter and callback URI).

Table 5.4.8.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Remarks
n/a		

Table 5.4.8.3.1-2: Details of the POST request/response on this resource

Request body	Data type	Cardinality	Remarks	
	NsdmSubscriptionRequest	1	Details of the subscription to be created, as defined in clause 5.5.2.7.	
Response body	Data type	Cardinality	Response Codes	Remarks
	NsdmSubscription	1	201 Created	<p>Shall be returned when the subscription has been created successfully.</p> <p>A representation of the created "Individual subscription" resource shall be returned in the response body, as defined in clause 5.5.2.8.</p> <p>The HTTP response shall include a "Location:" HTTP header that points to the created "Individual subscription" resource.</p>
	n/a		303 See Other	<p>Shall be returned when a subscription with the same callback URI and the same filter already exists and the policy of the NFVO is to not create redundant subscriptions.</p> <p>The HTTP response shall include a "Location" HTTP header that contains the resource URI of the existing "Individual subscription" resource.</p> <p>The response body shall be empty.</p>
	ProblemDetails	1	422 Unprocessable Entity	<p>Shall be returned upon the following error: The content type of the payload body is supported and the payload body of a request contains syntactically correct data but the data cannot be processed.</p> <p>The general cause for this error and its handling is specified in clause 6.4 of ETSI GS NFV-SOL 013 [16], including rules for the presence of the response body.</p> <p>Specifically in case of this resource, the response code 422 shall also be returned if the NFVO has tested the Notification endpoint as described in clause 5.4.10.3.2 and the test has failed.</p> <p>In this case, the "detail" attribute in the "ProblemDetails" structure shall convey more information about the error.</p>
ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.	

5.4.8.3.2 GET

The GET method queries the list of active subscriptions of the functional block that invokes the method. It can be used e.g. for resynchronization after error situations.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in the Tables 5.4.8.3.2-1 and 5.4.8.3.2-2.

Table 5.4.8.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Remarks
filter	0..1	Attribute filtering expression according to clause 5.2 of ETSI GS NFV-SOL 013 [16]. The NFVO shall support receiving this parameter as part of the URI query string. The OSS/BSS may supply this parameter. All attribute names that appear in the NsdmSubscription and in data types referenced from it shall be supported by the NFVO in the filter expression.
nextpage_opaque_marker	0..1	Marker to obtain the next page of a paged response. Shall be supported by the NFVO if the NFVO supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource.

Table 5.4.8.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Remarks	
	n/a			
Response body	Data type	Cardinality	Response Codes	Remarks
	NsdmSubscription	0..N	200 OK	Shall be returned when the list of subscriptions has been queried successfully. The response body shall contain in an array the representations of all active subscriptions of the functional block that invokes the method, i.e. zero or more representations of NSD management subscriptions as defined in clause 5.5.2.8. If the "filter" URI parameter was supplied in the request, the data in the response body shall have been transformed according to the rules specified in clause 5.2.2 of ETSI GS NFV-SOL 013 [16]. If the NFVO supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource, inclusion of the Link HTTP header in this response shall follow the provisions in clause 5.4.2.3 of ETSI GS NFV-SOL 013 [16].
	ProblemDetails		400 Bad Request	Shall be returned upon the following error: Invalid attribute-based filtering expression. The response body shall contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.
	ProblemDetails	1	400 Bad Request	Shall be returned upon the following error: Response too big. If the NFVO supports alternative 1 (error) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource, this error response shall follow the provisions in clause 5.4.2.2 of ETSI GS NFV-SOL 013 [16].
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

5.4.8.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

5.4.8.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

5.4.8.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

5.4.9 Resource: Individual subscription

5.4.9.1 Description

This resource represents an individual subscription. It can be used by the API consumer to read and to terminate a subscription to notifications related to NSD management.

5.4.9.2 Resource definition

The resource URI is:

{apiRoot}/nsd/{apiMajorVersion}/subscriptions/{subscriptionId}

This resource shall support the resource URI variables defined in Table 5.4.9.2-1.

Table 5.4.9.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 5.1a.
subscriptionId	Identifier of this subscription. See note.
NOTE: This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request creating a new subscription resource. It can also be retrieved from the "id" attribute in the payload body of that response.	

5.4.9.3 Resource methods

5.4.9.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

5.4.9.3.2 GET

The GET method retrieves information about a subscription by reading an individual subscription resource.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in the Tables 5.4.9.3.2-1 and 5.4.9.3.2-2.

Table 5.4.9.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Remarks
n/a		

Table 5.4.9.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Remarks	
	n/a			
Response body	Data type	Cardinality	Response Codes	Remarks
	NsdmSubscription	1	200 OK	Shall be returned when information about an individual subscription has been read successfully. The response body shall contain a representation of the subscription resource.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

5.4.9.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

5.4.9.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

5.4.9.3.5 DELETE

The DELETE method terminates an individual subscription.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in the Tables 5.4.9.3.5-1 and 5.4.9.3.5-2.

Table 5.4.9.3.5-1: URI query parameters supported by the DELETE method on this resource

Name	Cardinality	Remarks
n/a		

Table 5.4.9.3.5-2: Details of the DELETE request/response on this resource

Request body	Data type	Cardinality	Remarks	
	n/a			
Response body	Data type	Cardinality	Response Codes	Remarks
	n/a		204 No Content	Shall be returned when the "Individual subscription" resource has been deleted successfully. The response body shall be empty.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

5.4.10 Resource: Notification endpoint

5.4.10.1 Description

This resource represents a notification endpoint. The API producer can use this resource to send notifications to a subscribed API consumer, which has provided the URI of this resource during the subscription process.

5.4.10.2 Resource definition

The resource URI is provided by the API consumer when creating the subscription.

This resource shall support the resource URI variables defined in Table 5.4.10.2-1.

Table 5.4.10.2-1: Resource URI variables for this resource

Name	Definition
n/a	

5.4.10.3 Resource methods

5.4.10.3.1 POST

The POST method delivers a notification from the API producer to the API consumer.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in the Tables 5.4.10.3.1-1 and 5.4.10.3.1-2.

Table 5.4.10.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Remarks
n/a		

Each notification request body shall include exactly one of the alternatives defined in Table 5.4.10.3.1-2.

Table 5.4.10.3.1-2: Details of the POST request/response on this resource

Request body	Data type	Cardinality	Remarks	
	NsdOnBoardingNotification	1	A notification about the successful on-boarding of an NSD.	
	NsdOnBoardingFailureNotification	1	A notification about the failure of on-boarding an NSD.	
	NsdChangeNotification	1	A notification about the state change of an on-boarded NSD.	
	NsdDeletionNotification	1	A notification about the deletion of an on-boarded NSD.	
	PnfdOnBoardingNotification	1	A notification about the successful on-boarding of a PNFD.	
	PnfdOnBoardingFailureNotification	1	A notification about the failure of on-boarding a PNFD.	
	PnfdDeletionNotification	1	A notification about the deletion of an on-boarded PNFD.	
Response body	Data type	Cardinality	Response Codes	Remarks
	n/a		204 No Content	Shall be returned when the notification has been delivered successfully.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

5.4.10.3.2 GET

The GET method allows the API producer to test the notification endpoint that is provided by the API consumer, e.g. during subscription.

This method shall follow the provisions specified in the Tables 5.4.10.3.2-1 and 5.4.10.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 5.4.10.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
none supported		

Table 5.4.10.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	n/a		204 No Content	<p>Shall be returned when the notification endpoint has been tested successfully.</p> <p>The response body shall be empty.</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

5.4.10.3.3 PUT

This method is not supported. When this method is requested on this resource, the OSS/BSS shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

5.4.10.3.4 PATCH

This method is not supported. When this method is requested on this resource, the OSS/BSS shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

5.4.10.3.5 DELETE

This method is not supported. When this method is requested on this resource, the OSS/BSS shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

5.5 Data model

5.5.1 Introduction

This clause defines the request and response data structures of the NSD Lifecycle management interface.

5.5.2 Resource and notification data types

5.5.2.1 Type: NsdInfoModifications

This type represents attribute modifications for an individual NS descriptor resource based on the "NsdInfo" data type. The attributes of "NsdInfo" that can be modified are included in the "NsdInfoModifications" data type.

The "NsdInfoModifications" data type shall comply with the provisions defined in Table 5.5.2.1-1.

Table 5.5.2.1-1: Definition of the NsdInfoModifications data type

Attribute name	Data type	Cardinality	Description
nsdOperationalState	NsdOperationalStateType	0..1	New value of the "nsdOperationalState" attribute in "NsdInfo" data type. See note. Permitted values: ENABLED DISABLED
userDefinedData	KeyValuePairs	0..1	Modifications of the "userDefinedData" attribute in "NsdInfo" data type. See note. If present, these modifications shall be applied according to the rules of JSON Merge Patch (see IETF RFC 7396 [11]).
NOTE: At least one of the attributes - nsdOperationalState and userDefinedData - shall be present.			

5.5.2.2 Type: NsdInfo

This type represents a response for the query NSD operation. It shall comply with the provisions defined in Table 5.5.2.2-1.

Table 5.5.2.2-1: Definition of the NsdInfo data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of the onboarded individual NS descriptor resource. This identifier is allocated by the NFVO.
nsdId	Identifier	0..1	This identifier, which is allocated by the NSD designer, identifies the NSD in a globally unique way. It is copied from the NSD content and shall be present after the NSD content is on-boarded.
nsdName	String	0..1	Name of the onboarded NSD. This information is copied from the NSD content and shall be present after the NSD content is on-boarded.
nsdVersion	Version	0..1	Version of the on-boarded NSD. This information is copied from the NSD content and shall be present after the NSD content is on-boarded.
nsdDesigner	String	0..1	Designer of the on-boarded NSD. This information is copied from the NSD content and shall be present after the NSD content is on-boarded.
nsdInvariantId	Identifier	0..1	This identifier, which is allocated by the NSD designer, identifies an NSD in a version independent manner. This information is copied from the NSD content and shall be present after the NSD content is on-boarded.
vnfPkgIds	Identifier	0..N	Identifies the VNF package for the VNFD referenced by the on-boarded NS descriptor resource. See note 1.
pnfdInfolds	Identifier	0..N	Identifies the PnfdInfo element for the PNFD referenced by the on-boarded NS descriptor resource.
nestedNsdInfolds	Identifier	0..N	Identifies the NsdInfo element for the nested NSD referenced by the on-boarded NS descriptor resource. See note 1.
archiveSecurityOption	Enum (inlined)		Signals the security option used by the NSD archive as defined in clause 5.1 of ETSI GS NFV-SOL 007 [18]. Valid values: OPTION_1, OPTION_2
signingCertificate	String	0..1	The singleton signing certificate if it is included as a file in the NSD archive.
artifacts	NsdArchiveArtifactInfo	0..N	Information about NSD archive artifacts contained in the NSD archive. This attribute shall not be present before the NSD archive content is on-boarded. Otherwise, this attribute shall be present if the NSD archive contains artifacts.

Attribute name	Data type	Cardinality	Description
nsdOnboardingState	NsdOnboardingState Type	1	Onboarding state of the individual NS descriptor resource. See note 4.
onboardingFailureDetails	ProblemDetails	0..1	Failure details of current onboarding procedure. See clause 6.3 of ETSI GS NFV-SOL 013 [16] for the details of "ProblemDetails" structure. It shall be present when the "nsdOnboardingState" attribute is "ERROR", i.e. the uploading or processing has failed in NFVO, unless it has been requested to be excluded via an attribute selector.
nsdOperationalState	NsdOperationalState Type	1	Operational state of the individual NS descriptor resource. This attribute can be modified with the PATCH method. See note 2 and note 4.
nsdUsageState	NsdUsageState Type	1	Usage state of the individual NS descriptor resource. See note 3 and note 4.
userDefinedData	KeyValuePairs	0..1	User defined data for the individual NS descriptor resource. This attribute can be modified with the PATCH method.
_links	Structure (inlined)	1	Links to resources related to this resource.
>self	Link	1	URI of this resource.
>nsd_content	Link	1	Link to the NSD content resource.
NOTE 1: At least one of the attributes - vnfPkgId and nestedNsdInfo shall be present, after the NSD is on-boarded.			
NOTE 2: If the value of the nsdOnboardingState attribute is not equal to "ONBOARDED", the value of the nsdOperationalState attribute shall be equal to "DISABLED".			
NOTE 3: If the value of the nsdOnboardingState attribute is not equal to "ONBOARDED", the value of the nsdUsageState attribute shall be equal to "NOT_IN_USE".			
NOTE 4: State changes of an NSD are illustrated in clause B.2.			

5.5.2.3 Type: CreateNsdInfoRequest

This type creates a completely new NS descriptor resource. It shall comply with the provisions defined in Table 5.5.2.3-1.

Table 5.5.2.3-1: Definition of the CreateNsdInfoRequest data type

Attribute name	Data type	Cardinality	Description
userDefinedData	KeyValuePairs	0..1	User-defined data for the NS descriptor resource to be created. It shall be present when the user defined data is set for the individual NS descriptor resource to be created.

5.5.2.4 Type: PnfdInfoModifications

This type represents attribute modifications for an individual PNF descriptor resource based on the "PnfdInfo" data type. The attributes of "PnfdInfo" that can be modified are included in the "PnfdInfoModifications" data type.

The "PnfdInfoModifications" data type shall comply with the provisions defined in Table 5.5.2.4-1.

Table 5.5.2.4-1: Definition of the PnfdInfoModifications data type

Attribute name	Data type	Cardinality	Description
userDefinedData	KeyValuePairs	1	Modifications of the "userDefinedData" attribute in "PnfdInfo" data type. If present, these modifications shall be applied according to the rules of JSON Merge Patch (see IETF RFC 7396 [11]).

5.5.2.5 Type: PnfdInfo

This type represents a response for the query PNFD operation. It shall comply with the provisions defined in Table 5.5.2.5-1.

Table 5.5.2.5-1: Definition of the PnfdInfo data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of the onboarded individual PNF descriptor resource. This identifier is allocated by the NFVO.
pnfdId	Identifier	0..1	This identifier, which is managed by the PNFD designer, identifies the PNFD in a globally unique way. It is copied from the PNFD content and shall be present after the PNFD content is on-boarded.
pnfdName	String	0..1	Name of the onboarded PNFD. This information is copied from the PNFD content and shall be present after the PNFD content is on-boarded.
pnfdVersion	Version	0..1	Version of the onboarded PNFD. This information is copied from the PNFD content and shall be present after the PNFD content is on-boarded.
pnfdProvider	String	0..1	Provider of the onboarded PNFD. This information is copied from the PNFD content and shall be present after the PNFD content is on-boarded.
pnfdInvariantId	Identifier	0..1	Identifies a PNFD in a version independent manner. This attribute is invariant across versions of PNFD.
archiveSecurityOption	Enum (inlined)	0..1	Signals the security option used by the PNFD archive as defined in clause 5.1 of ETSI GS NFV-SOL 004 [5]. Valid values: OPTION_1, OPTION_2
signingCertificate	String	0..1	The singleton signing certificate if it is included as a file in the PNFD archive.
artifacts	PnfdArchiveArtifactInfo	0..N	Information about PNFD archive artifacts contained in the PNFD archive. This attribute shall not be present before the PNFD archive content is on-boarded. Otherwise, this attribute shall be present if the PNFD archive contains artifacts.
pnfdOnboardingState	PnfdOnboardingStateType	1	Onboarding state of the individual PNF descriptor resource.
onboardingFailureDetails	ProblemDetails	0..1	Failure details of current onboarding procedure. See clause 6.3 of ETSI GS NFV-SOL 013 [16] for the details of "ProblemDetails" structure. It shall be present when the "pnfdOnboardingState" attribute is "ERROR", i.e. the uploading or processing has failed in the NFVO, unless it has been requested to be excluded via an attribute selector.
pnfdUsageState	PnfdUsageStateType	1	Usage state of the individual PNF descriptor resource.
userDefinedData	KeyValuePairs	0..1	User defined data for the individual PNF descriptor resource. This attribute can be modified with the PATCH method.
_links	Structure (inlined)	1	Links to resources related to this resource.
>self	Link	1	URI of this resource.
>pnfd_content	Link	1	Link to the PNFD Content resource.

5.5.2.6 Type: CreatePnfdInfoRequest

This type creates a new PNF descriptor resource. It shall comply with the provisions defined in Table 5.5.2.6-1.

Table 5.5.2.6-1: Definition of the CreatePnfdInfoRequest data type

Attribute name	Data type	Cardinality	Description
userDefinedData	KeyValuePairs	0..1	User-defined data for the PNF descriptor resource to be created. It shall be present when the user defined data is set for the individual PNF descriptor resource to be created.

5.5.2.7 Type: NsdmSubscriptionRequest

This type represents a subscription request related to notifications about NSD management. It shall comply with the provisions defined in Table 5.5.2.7-1.

Table 5.5.2.7-1: Definition of the NsdmSubscriptionRequest data type

Attribute name	Data type	Cardinality	Description
filter	NsdmNotificationsFilter	0..1	Filter settings for this subscription, to define the subset of all notifications this subscription relates to. A particular notification is sent to the subscriber if the filter matches, or if there is no filter.
callbackUri	Uri	1	The URI of the endpoint to send the notification to.
authentication	SubscriptionAuthentication	0..1	Authentication parameters to configure the use of Authorization when sending notifications corresponding to this subscription, as defined in clause 8.3.4 of ETSI GS NFV-SOL 013 [16]. This attribute shall only be present if the subscriber requires authorization of notifications.

5.5.2.8 Type: NsdmSubscription

This type represents a subscription related to notifications about NSD management. It shall comply with the provisions defined in Table 5.5.2.8-1.

Table 5.5.2.8-1: Definition of the NsdmSubscription data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this subscription resource.
filter	NsdmNotificationsFilter	0..1	Filter settings for this subscription, to define the subset of all notifications this subscription relates to. A particular notification is sent to the subscriber if the filter matches, or if there is no filter.
callbackUri	Uri	1	The URI of the endpoint to send the notification to.
_links	Structure (inlined)	1	Links to resources related to this resource.
>self	Link	1	URI of this resource.

5.5.2.9 Type: NsdOnboardingNotification

This type represents an NSD management notification, which informs the receiver of the successful on-boarding of an NSD. It shall comply with the provisions defined in Table 5.5.2.9-1. The support of this notification is mandatory. The notification shall be triggered by the NFVO when the "nsdOnboardingState" attribute of a new NSD has changed to "ONBOARDED".

Table 5.5.2.9-1: Definition of the NsdOnboardingNotification data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this notification. If a notification is sent multiple times due to multiple subscriptions, the "id" attribute of all these notifications shall have the same value.
notificationType	String	1	Discriminator for the different notification types. Shall be set to "NsdOnboardingNotification" for this notification type.
subscriptionId	Identifier	1	Identifier of the subscription that this notification relates to.
timeStamp	DateTime	1	Date-time of the generation of the notification.
nsdInfold	Identifier	1	Identifier of the NSD information object. This identifier is allocated by the NFVO.
nsdId	Identifier	1	This identifier, which is managed by the service provider, identifies the NSD in a globally unique way. It is copied from the on-boarded NSD.
_links	NsdmLinks	1	Links to resources related to this notification.

5.5.2.10 Type: NsdOnboardingFailureNotification

This type represents an NSD management notification, which informs the receiver of the failure of on-boarding an NSD. It shall comply with the provisions defined in Table 5.5.2.10-1. The support of this notification is mandatory. The notification shall be triggered by the NFVO when the on-boarding of an NSD has failed.

Table 5.5.2.10-1: Definition of the NsdOnboardingFailureNotification data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this notification. If a notification is sent multiple times due to multiple subscriptions, the "id" attribute of all these notifications shall have the same value.
notificationType	String	1	Discriminator for the different notification types. Shall be set to "NsdOnboardingFailureNotification" for this notification type.
subscriptionId	Identifier	1	Identifier of the subscription that this notification relates to.
timeStamp	DateTime	1	Date-time of the generation of the notification.
nsdInfold	Identifier	1	Identifier of the NSD information object. This identifier is allocated by the NFVO.
nsdId	Identifier	0..1	This identifier, which is managed by the service provider, identifies the NSD in a globally unique way.
onboardingFailureDetails	ProblemDetails	1	Failure details of current onboarding procedure. See clause 6.3 of ETSI GS NFV-SOL 013 [16] for the details of "ProblemDetails" structure.
_links	NsdmLinks	1	Links to resources related to this notification.

5.5.2.11 Type: NsdChangeNotification

This type represents an NSD management notification, which informs the receiver of a change of the "nsdOperationalState" attribute of an on-boarded NSD. Changes in the value of the "nsdUsageState" and "nsdOnboardingState" attributes are not reported. The notification shall comply with the provisions defined in Table 5.5.2.11-1. The support of this notification is mandatory. The notification shall be triggered by the NFVO when the value of the "nsdOperationalState" attribute has changed, and the "nsdOperationalState" attribute has the value "ONBOARDED".

Table 5.5.2.11-1: Definition of the NsdChangeNotification data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this notification. If a notification is sent multiple times due to multiple subscriptions, the "id" attribute of all these notifications shall have the same value.
notificationType	String	1	Discriminator for the different notification types. Shall be set to "NsdChangeNotification" for this notification type.
subscriptionId	Identifier	1	Identifier of the subscription that this notification relates to.
timeStamp	DateTime	1	Date-time of the generation of the notification.
nsdInfold	Identifier	1	Identifier of the NSD information object. This identifier is allocated by the NFVO.
nsdId	Identifier	1	This identifier, which is managed by the service provider, identifies the NSD in a globally unique way. It is copied from the on-boarded NSD.
nsdOperationalState	NsdOperationalState Type	1	New operational state of the on-boarded NSD.
_links	NsdmLinks	1	Links to resources related to this notification.

5.5.2.12 Type: NsdDeletionNotification

This type represents an NSD management notification, which informs the receiver of the deletion of an on-boarded NSD. The notification shall comply with the provisions defined in Table 5.5.2.12-1. The support of this notification is mandatory. The notification shall be triggered by the NFVO when it has deleted an on-boarded NSD.

Table 5.5.2.12-1: Definition of the NsdDeletionNotification data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this notification. If a notification is sent multiple times due to multiple subscriptions, the "id" attribute of all these notifications shall have the same value.
notificationType	String	1	Discriminator for the different notification types. Shall be set to "NsdDeletionNotification" for this notification type.
subscriptionId	Identifier	1	Identifier of the subscription that this notification relates to.
timeStamp	DateTime	1	Date-time of the generation of the notification.
nsdInfold	Identifier	1	Identifier of the NSD information object. This identifier is allocated by the NFVO.
nsdId	Identifier	1	This identifier, which is managed by the service provider, identifies the NSD in a globally unique way. It is copied from the on-boarded NSD.
_links	NsdmLinks	1	Links to resources related to this notification.

5.5.2.13 Type: PnfdOnboardingNotification

This type represents a PNFD management notification, which informs the receiver of the successful on-boarding of a PNFD. It shall comply with the provisions defined in Table 5.5.2.13-1. The support of this notification is mandatory.

The notification is triggered when a new PNFD is on-boarded.

Table 5.5.2.13-1: Definition of the PnfdOnboardingNotification data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this notification. If a notification is sent multiple times due to multiple subscriptions, the "id" attribute of all these notifications shall have the same value.
notificationType	String	1	Discriminator for the different notification types. Shall be set to "PnfdOnboardingNotification" for this notification type.
subscriptionId	Identifier	1	Identifier of the subscription that this notification relates to.
timeStamp	DateTime	1	Date-time of the generation of the notification.
pnfdInfoId	Identifier	1	Identifier of the PNFD information object. This identifier is allocated by the NFVO.
pnfdId	Identifier	1	This identifier, which is managed by the service provider, identifies the PNFD in a globally unique way. It is copied from the on-boarded PNFD.
_links	PnfdmLinks	1	Links to resources related to this notification.

5.5.2.14 Type: PnfdOnboardingFailureNotification

This type represents a PNFD management notification, which informs the receiver of the failure of on-boarding a PNFD. It shall comply with the provisions defined in Table 5.5.2.14-1. The support of this notification is mandatory.

The notification is triggered when the on-boarding of a PNFD fails.

Table 5.5.2.14-1: Definition of the PnfdOnboardingFailureNotification data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this notification. If a notification is sent multiple times due to multiple subscriptions, the "id" attribute of all these notifications shall have the same value.
notificationType	String	1	Discriminator for the different notification types. Shall be set to "PnfdOnboardingFailureNotification" for this notification type.
subscriptionId	Identifier	1	Identifier of the subscription that this notification relates to.
timeStamp	DateTime	1	Date-time of the generation of the notification.
pnfdInfoId	Identifier	1	Identifier of the PNFD information object. This identifier is allocated by the NFVO.
pnfdId	Identifier	0..1	This identifier, which is managed by the service provider, identifies the PNFD in a globally unique way.
onboardingFailureDetails	ProblemDetails	1	Failure details of current onboarding procedure. See clause 6.3 of ETSI GS NFV-SOL 013 [16] for the details of "ProblemDetails" structure.
_links	PnfdmLinks	1	Links to resources related to this notification.

5.5.2.15 Type: PnfdDeletionNotification

This type represents a PNFD management notification, which informs the receiver of the deletion of an on-boarded PNFD. The notification shall comply with the provisions defined in Table 5.5.2.15-1. The support of this notification is mandatory.

The notification is triggered when an on-boarded PNFD is deleted.

Table 5.5.2.15-1: Definition of the PnfdDeletionNotification data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this notification. If a notification is sent multiple times due to multiple subscriptions, the "id" attribute of all these notifications shall have the same value.
notificationType	String	1	Discriminator for the different notification types. Shall be set to "PnfdDeletionNotification" for this notification type.
subscriptionId	Identifier	1	Identifier of the subscription that this notification relates to.
timeStamp	DateTime	1	Date-time of the generation of the notification.
pnfdInfoId	Identifier	1	Identifier of the PNFD information object. This identifier is allocated by the NFVO.
pnfdId	Identifier	1	This identifier, which is managed by the service provider, identifies the PNFD in a globally unique way. It is copied from the on-boarded PNFD.
_links	PnfdmLinks	1	Links to resources related to this notification.

5.5.3 Referenced structured data types

5.5.3.1 Introduction

This clause defines data structures that can be referenced from data structures defined in the previous clauses, but can neither be resource representations nor bound to any subscribe/notify mechanism.

5.5.3.2 Type: NsdmNotificationsFilter

This type represents a subscription filter related to notifications about NSD management. It shall comply with the provisions defined in Table 5.5.3.2-1.

At a particular nesting level in the filter structure, the following applies: All attributes shall match in order for the filter to match (logical "and" between different filter attributes). If an attribute is an array, the attribute shall match if at least one of the values in the array matches (logical "or" between the values of one filter attribute).

Table 5.5.3.2-1: Definition of the NsdmNotificationsFilter data type

Attribute name	Data type	Cardinality	Description
notificationTypes	Enum (inlined)	0..N	Match particular notification types. Permitted values: NsdOnBoardingNotification NsdOnboardingFailureNotification NsdChangeNotification NsdDeletionNotification PnfdOnBoardingNotification PnfdOnBoardingFailureNotification PnfdDeletionNotification See note 1.
nsdInfold	Identifier	0..N	Match the NsdInfo identifier which is allocated by the NFVO. See note 2.
nsdId	Identifier	0..N	Match the NSD identifier, which is allocated by the NSD designer. See note 2.
nsdName	String	0..N	Match the name of the onboarded NSD.
nsdVersion	Version	0..N	Match the NSD version listed as part of this attribute.
nsdDesigner	String	0..N	Match the NSD designer of the on-boarded NSD.
nsdInvariantId	Identifier	0..N	Match the NSD invariant identifier which is allocated by the NSD designer and identifies an NSD in a version independent manner.
vnfPkgIds	Identifier	0..N	Match VNF packages with a package identifier listed in the attribute.
pnfdInfolDs	Identifier	0..N	Match the PnfdInfo identifier for the PNFD referenced by the on-boarded NSD. See note 3.
nestedNsdInfolDs	Identifier	0..N	Match the NsdInfo identifier for the nested NSD referenced by the on-boarded NSD.
nsdOnboardingState	NsdOnboardingStateType	0..N	Match particular on-boarding state of the NSD.
nsdOperationalState	NsdOperationalStateType	0..N	Match particular operational state of the on-boarded NSD.
nsdUsageState	NsdUsageStateType	0..N	Match particular usage state of the on-boarded NSD.
pnfdId	Identifier	0..N	Match the PNFD identifier which is copied from the PNFD content. See note 3.
pnfdName	String	0..N	Match the name of the onboarded PNFD.
pnfdVersion	Version	0..N	Match the PNFD designer of the on-boarded PNFD.
pnfdProvider	String	0..N	Match the provider of the on-boarded PNFD.
pnfdInvariantId	Identifier	0..N	Match the PNFD in a version independent manner.
pnfdOnboardingState	PnfdOnboardingStateType	0..N	Match particular onboarding state of the PNFD.
pnfdUsageState	PnfdUsageStateType	0..N	Match the usage state of the individual PNF descriptor resource.
NOTE 1: The permitted values of the "notificationTypes" attribute are spelled exactly as the names of the notification types to facilitate automated code generation systems.			
NOTE 2: The attributes "nsdId" and "nsdInfold" are alternatives to reference to a particular NSD in a filter. They should not be used both in the same filter instance, but one alternative should be chosen.			
NOTE 3: The attributes "pnfdId" and "pnfdInfold" are alternatives to reference to a particular PNFD in a filter. They should not be used both in the same filter instance, but one alternative should be chosen.			

5.5.3.3 Type: NsdmLinks

This type represents the links to resources that an NSD management notification can contain. It shall comply with the provisions defined in Table 5.5.3.3-1.

Table 5.5.3.3-1: Definition of the NsdmLinks data type

Attribute name	Data type	Cardinality	Description
nsdInfo	NotificationLink	1	Link to the resource representing the NSD to which the notified change applies, i.e. the individual NS descriptor resource that represents the NSD.
subscription	NotificationLink	1	Link to the related subscription.

5.5.3.4 Type: PnfdmLinks

This type represents the links to resources that a PNFD management notification can contain. It shall comply with the provisions defined in Table 5.5.3.4-1.

Table 5.5.3.4-1: Definition of the PnfdmLinks data type

Attribute name	Data type	Cardinality	Description
pnfdInfo	NotificationLink	1	Link to the resource representing the PNFD to which the notified change applies, i.e. the individual PNF descriptor resource that represents the PNFD.
subscription	NotificationLink	1	Link to the related subscription.

5.5.3.5 Type: NsdArchiveArtifactInfo

This type represents an artifact contained in an NSD archive. It shall comply with provisions defined in Table 5.5.3.5-1.

Table 5.5.3.5-1: Definition of the NsdArchiveArtifactInfo data type

Attribute name	Data type	Cardinality	Description
artifactPath	String	1	Path in the NSD archive, which identifies the artifact and also allows to access a copy of the artifact.
checksum	Checksum	1	Checksum of the artifact file.
metadata	KeyValuePairs	0..1	The metadata of the artifact that are available in the NSD Archive, such as Content type, size, creation date, etc.

5.5.3.6 Type: PnfdArchiveArtifactInfo

This type represents an artifact contained in a PNFD archive. It shall comply with provisions defined in Table 5.5.3.6-1.

Table 5.5.3.6-1: Definition of the PnfdArchiveArtifactInfo data type

Attribute name	Data type	Cardinality	Description
artifactPath	String	1	Path in the PNFD archive, which identifies the artifact and also allows to access a copy of the artifact.
checksum	Checksum	1	Checksum of the artifact file.
nonManoArtifactSetId	String	0..1	Non-MANO artifact set identifier of the non-MANO artifact set to which the artifact belongs, as defined in clause 4.3.7 of ETSI GS NFV-SOL 004 [5]. Shall be provided if the artifact is a non-MANO artifact, and shall be omitted otherwise.
metadata	KeyValuePairs	0..1	The metadata of the artifact that are available in the PNFD Archive, such as Content type, size, creation date, etc.

5.5.4 Referenced simple data types and enumerations

5.5.4.1 Introduction

This clause defines simple data types and enumerations that can be referenced from data structures defined in the previous clauses.

5.5.4.2 Simple data types

No particular simple data types are defined for this interface, in addition to those defined in clause 4.4.2.

5.5.4.3 Enumeration: NsdOperationalStateType

The enumeration NsdOperationalStateType shall comply with the provisions defined in Table 5.5.4.3-1. It indicates the operational state of the resource.

Table 5.5.4.3-1: Enumeration NsdOperationalStateType

Enumeration value	Description
ENABLED	The operational state of the resource is enabled.
DISABLED	The operational state of the resource is disabled.

5.5.4.4 Enumeration: NsdUsageStateType

The enumeration NsdUsageStateType shall comply with the provisions defined in Table 5.5.4.4-1. It indicates the usage state of the resource.

Table 5.5.4.4-1: Enumeration NsdUsageStateType

Enumeration value	Description
IN_USE	The resource is in use.
NOT_IN_USE	The resource is not-in-use.

5.5.4.5 Enumeration: NsdOnboardingStateType

The enumeration NsdOnboardingStateType shall comply with the provisions defined in Table 5.5.4.5-1. It indicates the onboarding state of the NSD.

Table 5.5.4.5-1: Enumeration NsdOnboardingStateType

Enumeration value	Description
CREATED	The NSD information object has been created.
UPLOADING	The associated NSD content is being uploaded.
PROCESSING	The associated NSD content is being processed, e.g. validation.
ONBOARDED	The associated NSD content has been on-boarded successfully.
ERROR	There was an error during upload or processing of the NSD content.

5.5.4.6 Enumeration: PnfdOnboardingStateType

The enumeration PnfdOnboardingStateType shall comply with the provisions defined in Table 5.5.4.6-1. It indicates the onboarding state of the individual PNF descriptor resource.

Table 5.5.4.6-1: Enumeration PnfdOnboardingStateType

Enumeration value	Description
CREATED	The PNF descriptor resource has been created.
UPLOADING	The associated PNFD content is being uploaded.
PROCESSING	The associated PNFD content is being processed, e.g. validation.
ONBOARDED	The associated PNFD content has been on-boarded successfully.
ERROR	There was an error during upload or processing of the associated PNFD content.

5.5.4.7 Enumeration: PnfdUsageStateType

The enumeration PnfdUsageStateType shall comply with the provisions defined in Table 5.5.4.7-1. It indicates the usage state of the resource.

Table 5.5.4.7-1: Enumeration PnfdUsageStateType

Enumeration value	Description
IN_USE	The resource is in use.
NOT_IN_USE	The resource is not-in-use.

6 NS Lifecycle Management interface

6.1 Description

This interface allows the OSS/BSS to invoke NS lifecycle management operations of NS instances towards the NFVO, and to subscribe to notifications regarding NS lifecycle changes provided by the NFVO.

The operations provided through this interface are as follows:

- Create NS Identifier
- Instantiate NS
- Scale NS
- Update NS
- Query NS
- Terminate NS
- Delete NS Identifier
- Heal NS
- Get Operation Status
- Subscribe
- Query Subscription Information
- Notify
- Terminate Subscription

This interface also enables to invoke error handling procedures (Retry, Rollback, Continue, Cancel, Fail) on the actual NS lifecycle management operation occurrences, and API version information retrieval.

6.1a API version

For the NS lifecycle management interface version as specified in the present document, the MAJOR version field shall be 1, the MINOR version field shall be 4, and the PATCH version field shall be 0 (see clause 9.1 of ETSI GS NFV-SOL 013 [16] for a definition of the version fields). Consequently, the {apiMajorVersion} URI variable shall be set to "v1".

6.2 Resource structure and methods

All resource URIs of the API shall use the base URI specification defined in clause 4.1 of ETSI GS NFV-SOL 013 [16]. The string "nslcm" shall be used to represent {apiName}. All resource URIs in the clauses below are defined relative to the above base URI.

Figure 6.2-1 shows the overall resource URI structure defined for the NS lifecycle management interface.

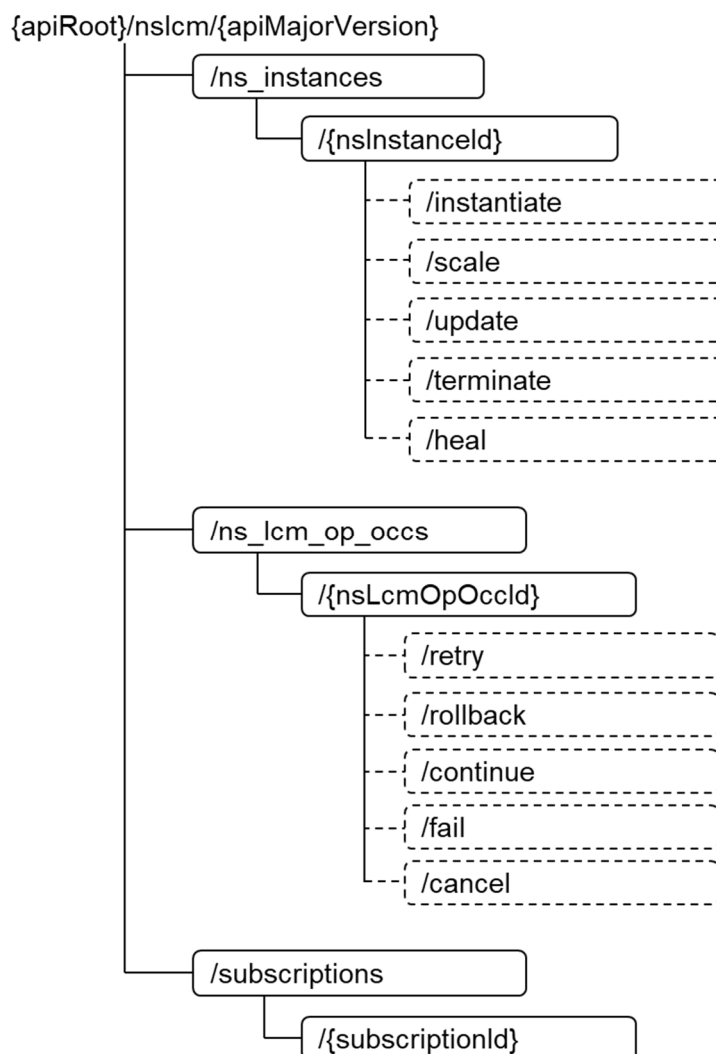


Figure 6.2-1: Resource URI structure of NS Lifecycle Management Interface

Table 6.2-1 lists the individual resources defined, and the applicable HTTP methods.

The NFVO shall support responding to requests for all HTTP methods on the resources in Table 6.2-1 that are marked as "M" (mandatory) in the "Cat" column. The NFVO shall also support the "API versions" resources as specified in clause 9.3.2 of ETSI GS NFV-SOL 013 [16].

Table 6.2-1: Resources and methods overview of the NS Lifecycle Management interface

Resource name	Resource URI	HTTP Method	Cat	Meaning
NS instances	/ns_instances	GET	M	Query multiple NS instances.
		POST	M	Create a new "Individual NS instance" resource.
Individual NS instance	/ns_instances/{nsInstancelId}	GET	M	Read an "Individual NS instance" resource.
		DELETE	M	Delete an "Individual NS instance" resource.
Instantiate NS task	/ns_instances/{nsInstancelId}/instantiate	POST	M	Instantiate an NS.
Scale NS task	/ns_instances/{nsInstancelId}/scale	POST	M	Scale an NS instance.
Update NS task	/ns_instances/{nsInstancelId}/update	POST	M	Updates an NS instance.
Terminate NS task	/ns_instances/{nsInstancelId}/terminate	POST	M	Terminate an NS instance.
Heal NS task	/ns_instances/{nsInstancelId}/heal	POST	M	Heal an NS instance.
NS LCM operation occurrences	/ns_lcm_op_occs	GET	M	Query information about multiple NS lifecycle management operation occurrences.
Individual NS LCM operation occurrence	/ns_lcm_op_occs/{nsLcmOpOcclId}	GET	M	Read information about an individual NS lifecycle management operation occurrence.
Retry operation task	/ns_lcm_op_occs/{nsLcmOpOcclId}/retry	POST	M	Retry an NS lifecycle management operation occurrence.
Rollback operation task	/ns_lcm_op_occs/{nsLcmOpOcclId}/rollback	POST	M	Rollback an NS lifecycle management operation occurrence.
Continue operation task	/ns_lcm_op_occs/{nsLcmOpOcclId}/continue	POST	M	Continue an NS lifecycle management operation occurrence.
Fail operation task	/ns_lcm_op_occs/{nsLcmOpOcclId}/fail	POST	M	Mark an NS lifecycle management operation occurrence as failed.
Cancel operation task	/ns_lcm_op_occs/{nsLcmOpOcclId}/cancel	POST	M	Cancel an NS lifecycle management operation occurrence.
Subscriptions	/subscriptions	POST	M	Subscribe to NS lifecycle change notifications.
		GET	M	Query multiple subscriptions.
Individual subscription	/subscriptions/{subscriptionId}	GET	M	Read an "Individual subscription" resource.
		DELETE	M	Terminate a subscription.
Notification endpoint	(provided by API consumer)	POST	See note	Notify about NS lifecycle change. See note.
		GET	See note	Test the notification endpoint. See note.
NOTE:	The NFVO shall support invoking the HTTP methods defined for the "Notification endpoint" resource exposed by the OSS/BSS. If the OSS/BSS supports invoking the POST method on the "Subscriptions" resource towards the NFVO, it shall also support responding to the HTTP requests defined for the "Notification endpoint" resource.			

Table 6.4.1.2-1 specifies the preconditions and postconditions applicable to the different NS lifecycle management operations triggered by task resources.

6.3 Sequence diagrams (informative)

6.3.1 Flow of the creation of an NS instance resource

This clause describes the procedure for the creation of an "Individual NS instance" resource.

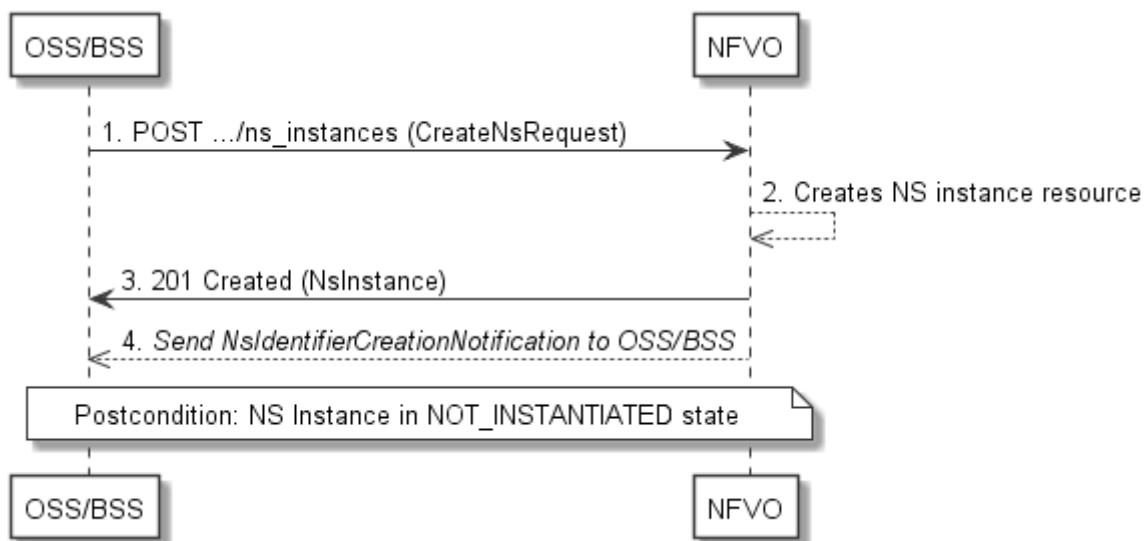


Figure 6.3.1-1: Flow of the creation of an NS instance resource

NOTE: Due to possible race conditions, the 201 response and the NsIdentifierCreationNotification can arrive in any order at the OSS/BSS.

The procedure consists of the following steps as illustrated in Figure 6.3.1-1:

- 1) The OSS/BSS sends a POST request to the "NS Instances" resource including in the payload body a data structure of type "CreateNsRequest".
- 2) The NFVO creates a new "Individual NS instance" resource in NOT_INSTANTIATED state, and the associated NS instance identifier.
- 3) The NFVO returns a 201 Created response containing a representation of the "Individual NS instance" resource just created by the NFVO, and provides the URI of the newly-created resource in the "Location" HTTP header.
- 4) The NFVO sends a NsIdentifierCreationNotification (see clause 6.3.6) to the OSS/BSS to indicate the creation of the "Individual NS instance" resource and the associated NS instance identifier.

Postcondition: Upon successful completion, a new "Individual NS instance" resource has been created in "NOT_INSTANTIATED" state.

6.3.2 Flow of the deletion of an NS instance resource

This clause describes the procedure for the deletion of an "Individual NS instance" resource.

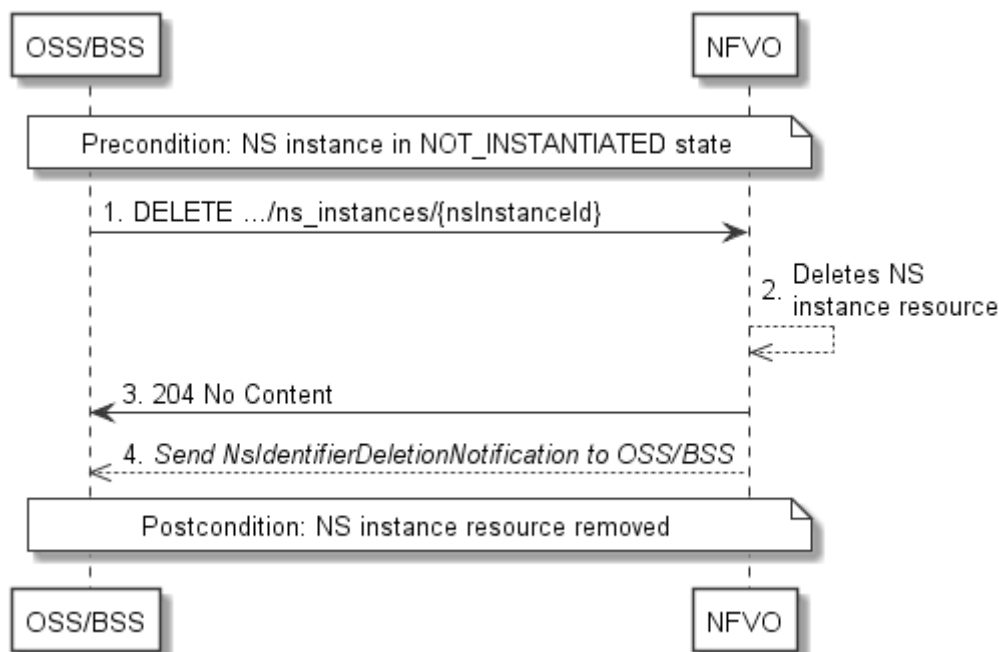


Figure 6.3.2-1: Flow of the deletion of an NS instance resource

NOTE: Due to possible race conditions, the 204 response and the NsIdentifierDeletionNotification can arrive in any order at the OSS/BSS.

Precondition: The resource representing the NS instance to be deleted needs to be in NOT_INSTANTIATED state.

The procedure consists of the following steps as illustrated in Figure 6.3.2-1:

- 1) The OSS/BSS sends a DELETE request to the "Individual NS Instance" resource.
- 2) The NFVO deletes the "Individual NS instance" resource and the associated NS instance identifier.
- 3) The NFVO returns a "204 No Content" response with an empty payload body.
- 4) The NFVO sends to the OSS/BSS an NS identifier deletion notification to indicate the deletion of the "Individual NS instance" resource and the associated NS instance identifier.

Postcondition: The resource representing the NS instance has been removed from the list of NS instance resources.

Error Handling: If the "Individual NS instance" is not in NOT_INSTANTIATED state, the NFVO rejects the deletion request.

6.3.3 Flow of NS lifecycle management operations triggered by task resources

This clause describes the general sequence for NS Lifecycle Management operations that operate on an NS instance resource and are triggered by task resources. The flows for these operations are very similar. The differences between the individual operations are covered in Table 6.4.1.2-1.

This flow is applicable to the following operations:

- Instantiate NS
- Scale NS
- Update NS
- Heal NS

- Terminate NS

Figure 6.3.3-1 illustrates the general lifecycle management flow. Placeholders in this flow allow for differentiating between the operations and are marked with double angular brackets "<<...>>".

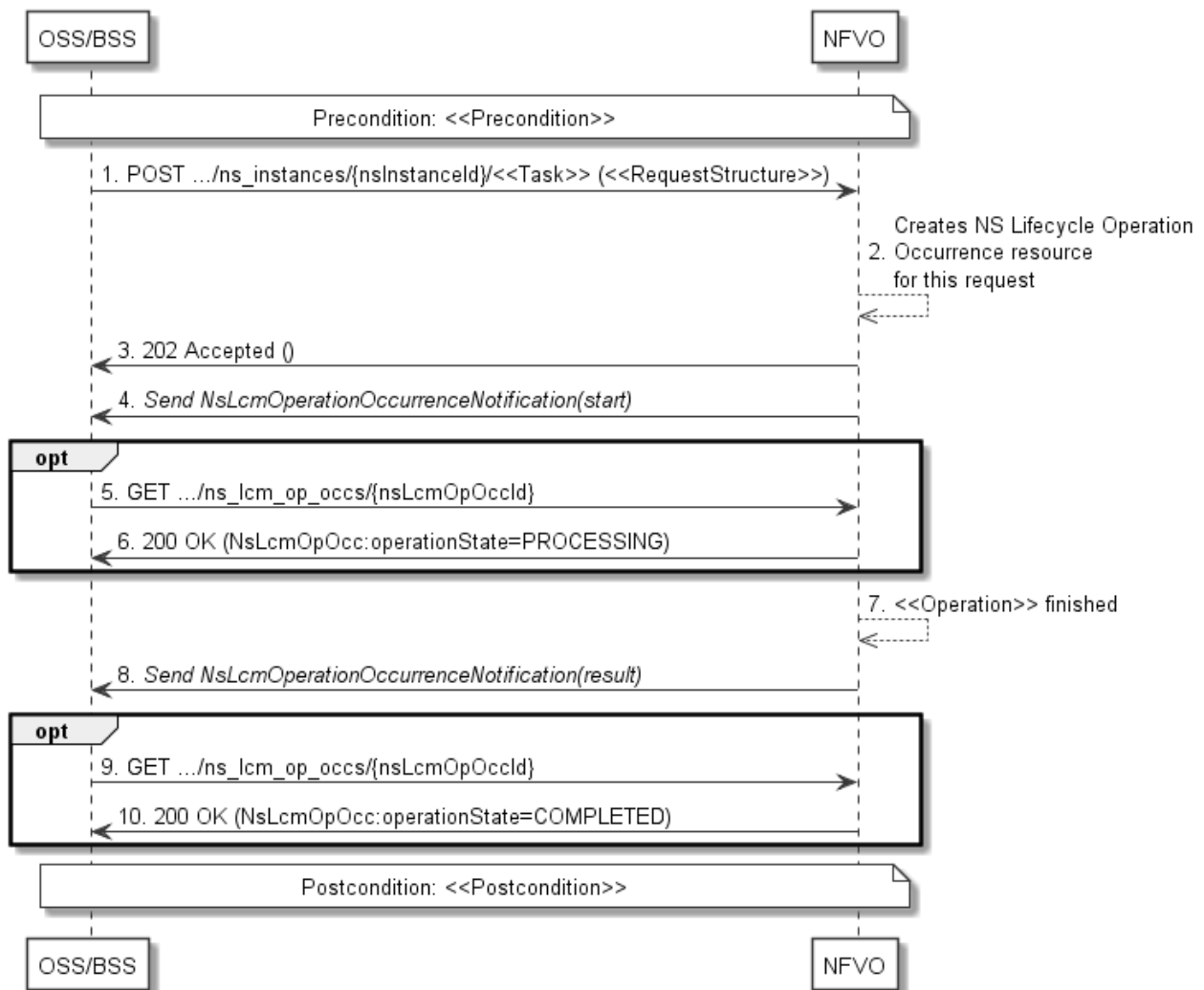


Figure 6.3.3-1: Flow of NS lifecycle operations triggered by task resources

NOTE: Due to possible race conditions, the 202 response and the "start" NsLcmOperationOccurrenceNotification can arrive in any order at the OSS/BSS.

Precondition: The precondition depends on the actual operation and is described by the template parameter <<Precondition>>. Table 6.4.1.2-1 specifies the applicable precondition.

An NS lifecycle operation, as illustrated in Figure 6.3.3-1, consists of the following steps:

- 1) The OSS/BSS sends a POST request to the <<Task>> resource that represents the lifecycle operation to be executed on the NS instance, and includes in the payload body a data structure of type <<RequestStructure>>. The name <<Task>> of the task resource and the <<RequestStructure>> depend on the operation and are described in Table 6.4.1.2-1.
- 2) The NFVO creates a new "Individual NS Lifecycle Operation Occurrence" resource for the request.
- 3) The NFVO returns a "202 Accepted" response with an empty payload body and a "Location" HTTP header that points to the new "Individual NS Lifecycle Operation Occurrence" resource, i.e. it includes the URI of that resource which is ".../ns_lcm_op_occs/{nsLcmOpOccId}".
- 4) The NFVO sends to the OSS/BSS an NS lifecycle management operation occurrence notification (see clause 6.3.6) to indicate the start of the lifecycle management operation occurrence. See note.

- 5) If desired, the NFVO can poll the "Individual NS Lifecycle Operation Occurrence" resource to obtain information about the ongoing operation by sending a GET request to the resource that represents the NS Lifecycle Operation Occurrence.
- 6) In the response to that request, the NFVO returns to the OSS/BSS information of the operation, such as the operation status, by providing in the payload body a data structure of type "NsLcmOpOcc".
- 7) The NFVO has finished the operation <<Operation>>.
- 8) The NFVO sends an NS lifecycle management operation occurrence notification (see clause 6.3.6) to indicate the completion of the lifecycle management operation occurrence with the success state "COMPLETED".
- 9) If desired, the OSS/BSS can send a new GET request to the "Individual NS Lifecycle Operation Occurrence" resource.
- 10) In the response to that request, the NFVO returns to the OSS/BSS information about the result of the operation, by providing in the payload body a data structure of type "NsLcmOpOcc".

Postcondition: The postcondition depends on the actual operation and is described by the template parameter <<Postcondition>>. Table 6.4.1.2-1 specifies the applicable postcondition.

Error handling: If the NS lifecycle management operation fails, error information is provided in the notification message that reports the erroneous completion of the procedure, and is also available in the resource that represents the actual NS lifecycle management operation occurrence related to this NS lifecycle management operation.

Table 6.4.1.2-1 defines how the flow described above is parameterized for the different NS lifecycle management operations.

6.3.4 Flow of the get operations status operation

This clause describes a sequence for obtaining the status of an NS lifecycle management operation occurrence.

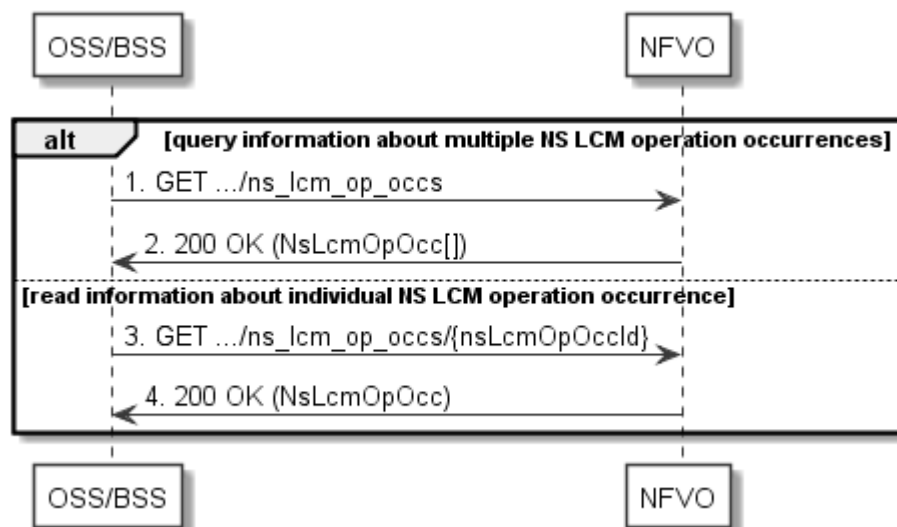


Figure 6.3.4-1: Flow of get NS lifecycle operation status

Obtaining the NS lifecycle operation status, as illustrated in Figure 6.3.4-1, consists of the following steps:

- 1) If the OSS/BSS intends to query all NS lifecycle management operation occurrences, it sends a GET request to the "NS LCM operation occurrences" resource.
- 2) The NFVO returns a "200 OK" response to the OSS/BSS, and includes zero or more data structures of type "NsLcmOpOcc" in the payload body.
- 3) If the OSS/BSS intends to read information about a particular NS LCM operation occurrence, it sends a GET request to the "Individual NS LCM operation occurrence" resource, addressed by the appropriate NS LCM operation occurrence identifier in its resource URI.

- 4) The NFVO returns a "200 OK" response to the OSS/BSS, and includes one data structure of type "NsLcmOpOcc" in the payload body.

Error Handling: In case of failure, appropriate error information is provided in the response.

6.3.5 Flow of managing subscriptions

This clause describes the procedure for creating, querying/reading, and terminating subscriptions to notifications related to NS lifecycle management.

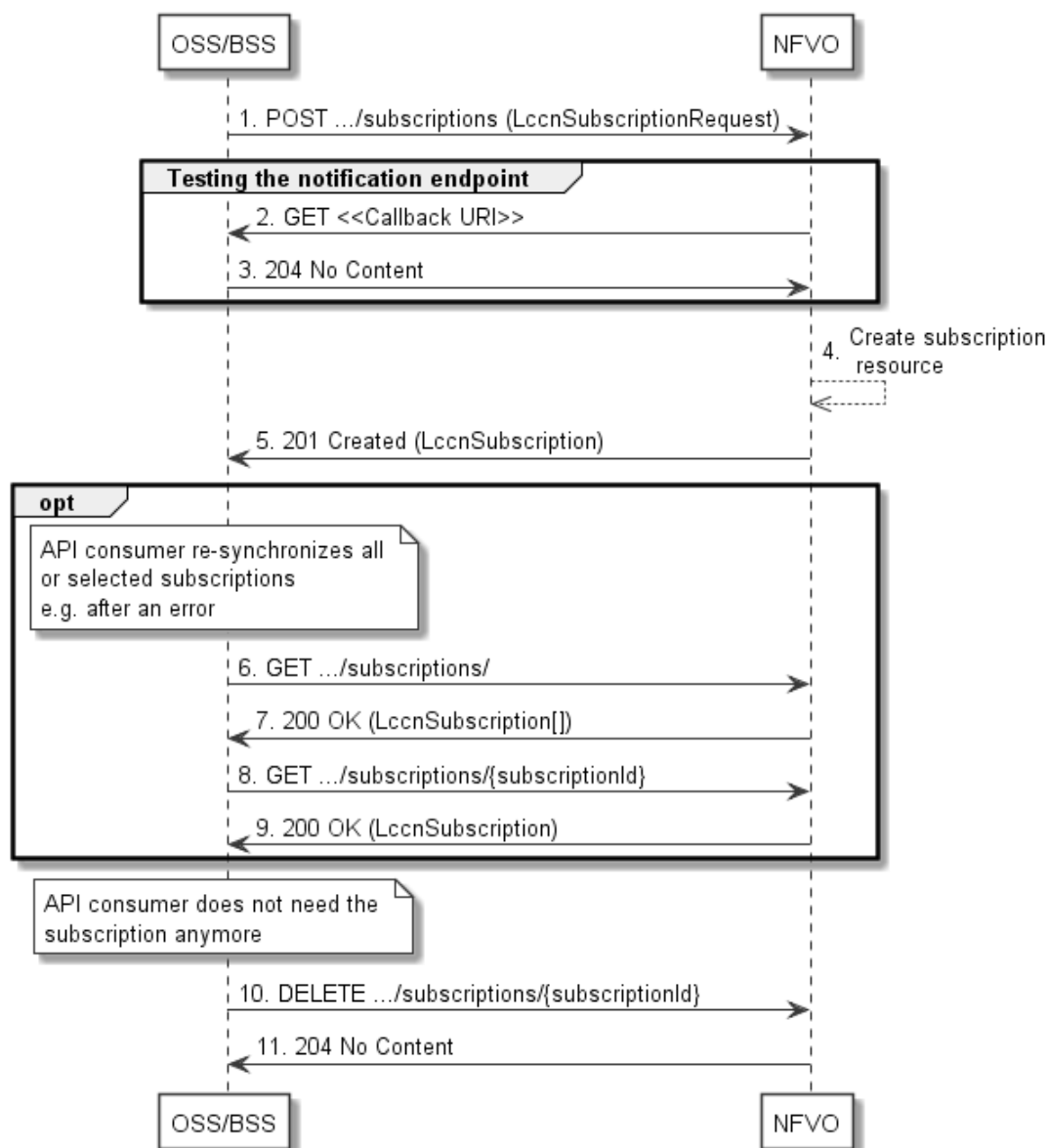


Figure 6.3.5-1: Flow of managing subscriptions

The procedure consists of the following steps as illustrated in Figure 6.3.5-1:

- 1) The OSS/BSS sends a POST request to the "Subscriptions" resource including in the payload body a data structure of type "LccnSubscriptionRequest". That data structure contains filtering criteria and a callback URI to which the NFVO will subsequently send notifications about events that match the filter.
- 2) To test the notification endpoint that has been registered by the OSS/BSS as part of the subscription, the NFVO sends a GET request to the notification endpoint URI.
- 3) The OSS/BSS returns a "204 No Content" response to indicate success.

- 4) The NFVO creates a new subscription to notifications related to NS lifecycle changes, and an "Individual subscription" resource that represents this subscription.
- 5) The NFVO returns a "201 Created" response containing a data structure of type "LccnSubscription" representing the "Individual subscription" resource just created by the NFVO, and provides the URI of the newly-created resource in the "Location" HTTP header.
- 6) If desired, e.g. to recover from an error situation, the OSS/BSS can query information about its subscriptions by sending a GET request to the resource representing the subscriptions.
- 7) In that case, the NFVO returns a "200 OK" response that contains zero or more representations of all existing subscriptions that were created by the OSS/BSS.
- 8) If desired, e.g. to recover from an error situation, the OSS/BSS can read information about a particular subscription by sending a GET request to the resource representing that individual subscription.
- 9) In that case, the NFVO returns a "200 OK" response that contains a representation of that individual subscription.
- 10) If the OSS/BSS does not need the subscription anymore, it terminates the subscription by sending a DELETE request to the resource that represents the individual subscription to remove.
- 11) The OSS/BSS acknowledges the successful termination of the subscription by returning a "204 No Content" response.

Error handling: The NFVO rejects a subscription if the subscription information is not valid: endpoint cannot be reached, subscription information is malformed, etc.

6.3.6 Flow of sending notifications

This clause describes the procedure for sending notifications related to NS lifecycle management.

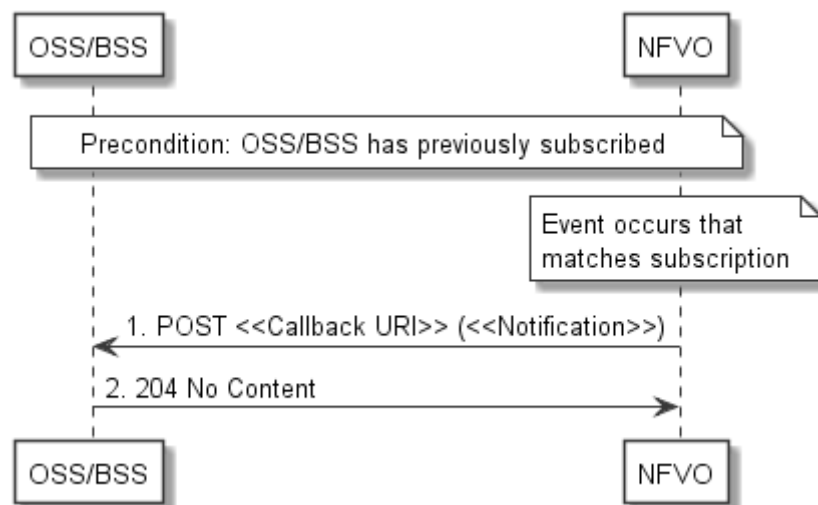


Figure 6.3.6-1: Flow of sending notifications

Precondition: The OSS/BSS has subscribed previously to notifications related to NS lifecycle management.

The procedure consists of the following steps as illustrated in Figure 6.3.6-1:

- 1) If an event occurs that matches the filtering criteria which are part of the subscription, the NFVO generates a notification that includes information about the event, and sends it in the body of a POST request to the URI which the OSS/BSS has registered as part of the subscription request. The variable <<Notification>> in the flow is a placeholder for the different types of notifications that can be sent by this API (see clauses 6.5.2.5 through 6.5.2.8).
- 2) The OSS/BSS acknowledges the successful delivery of the notification by returning a "204 No Content" response.

Error handling: If the NFVO does not receive the "204 No Content" response from the OSS/BSS, it can retry sending the notification.

6.3.7 Flow of retrying an NS lifecycle management operation

This clause describes a sequence for retrying an NS lifecycle management operation occurrence that is represented by an "Individual NS LCM operation occurrence" resource. Retry is used if an operation is in FAILED_TEMP state, and there is reason to believe that the operation will eventually succeed when retried, for instance because obstacle that led to an error during the execution of the LCM operation have been removed by an automated procedure, or by manual intervention. The "retry" operation is also called "idempotent retry" because it is possible to invoke retry multiple times, without side effects.

A comprehensive description of the handling of NS lifecycle management errors is provided in clause 6.6.

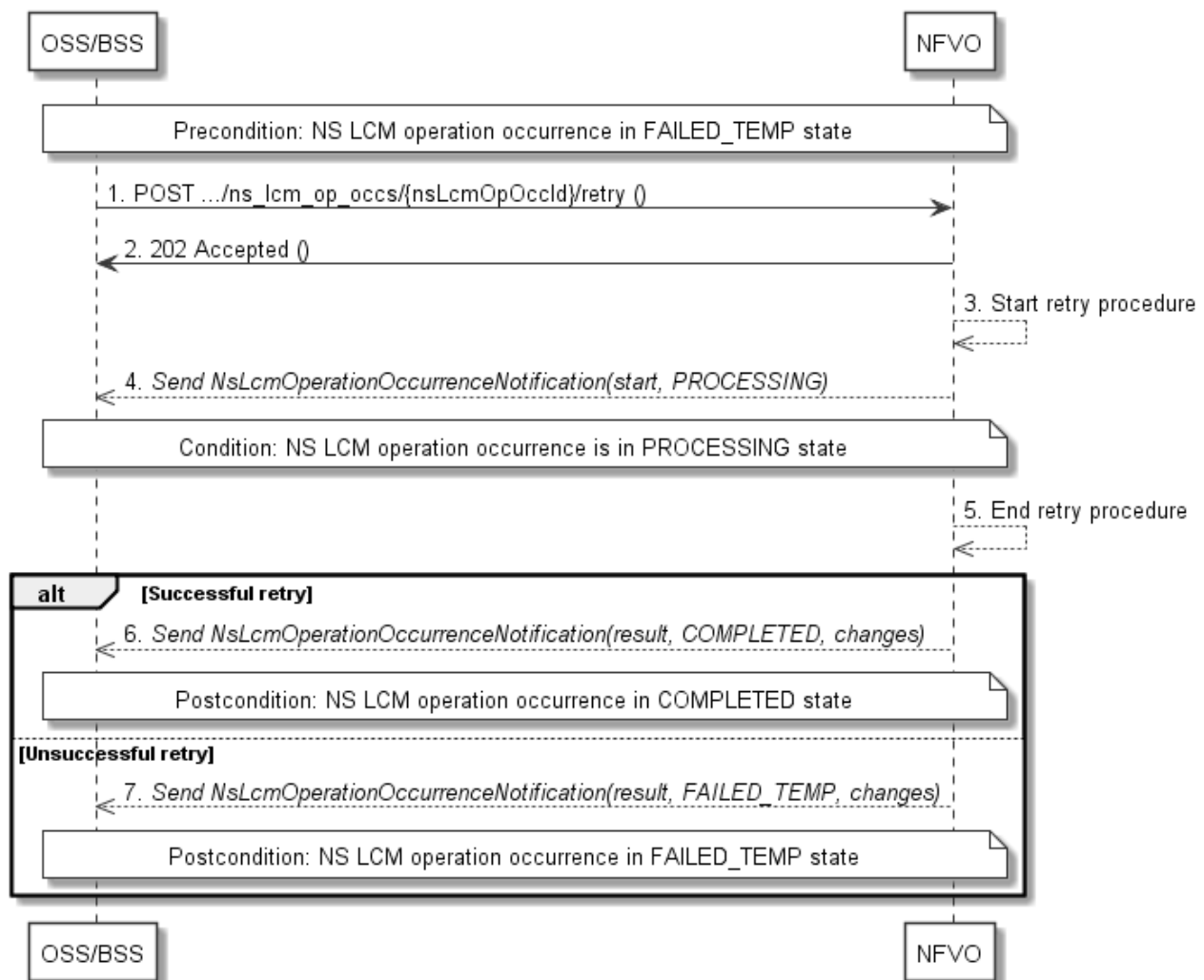


Figure 6.3.7-1: Flow of retrying an NS lifecycle management operation

NOTE: Due to possible race conditions, the 202 response and the "PROCESSING" NsLcmOperationOccurrenceNotification can arrive in any order at the OSS/BSS.

Precondition: The NS lifecycle operation occurrence is in FAILED_TEMP state.

Retrying an NS lifecycle operation, as illustrated in Figure 6.3.7-1, consists of the following steps:

- 1) The OSS/BSS sends a POST request with an empty body to the "Retry operation task" resource of the NS LCM operation occurrence that is to be retried.
- 2) The NFVO returns a "202 Accepted" response.

- 3) The NFVO starts the retry procedure.
- 4) The NFVO sends an NS lifecycle management operation occurrence notification of type "start" to indicate that the NS LCM operation occurrence enters the "PROCESSING" state.
- 5) The NFVO finishes the retry procedure.
- 6) On successful retry, the NFVO sends an NS lifecycle management operation occurrence notification (see clause 6.3.6) to indicate successful completion of the operation, and inform the OSS/BSS about the changes on the NS components (e.g. VNFs, VLs).
- 7) On unsuccessful retry, the NFVO sends an NS lifecycle management operation occurrence notification (see clause 6.3.6) to indicate an intermediate error (retry failed) of the operation, and inform the OSS/BSS about the changes on the NS components (e.g. VNFs, VLs).

Postcondition: The NS lifecycle operation occurrence resource is in one of the following states: FAILED_TEMP, COMPLETED. COMPLETED is a terminal state (see clause 6.6.2.2).

Error handling: The operation is rejected in case the "Individual NS LCM operation occurrence" resource is in any other state than FAILED_TEMP, or in case Retry is not supported by for the particular NS LCM operation for the particular NS.

6.3.8 Flow of rolling back an NS lifecycle management operation

This clause describes a sequence for rolling back an NS lifecycle management operation occurrence that is represented by an "Individual NS LCM operation occurrence" resource. Rollback can be used for example if an operation is in FAILED_TEMP state, and there is no reason to believe that retrying the operation will eventually succeed.

A comprehensive description of the handling of NS lifecycle management errors is provided in clause 6.6.

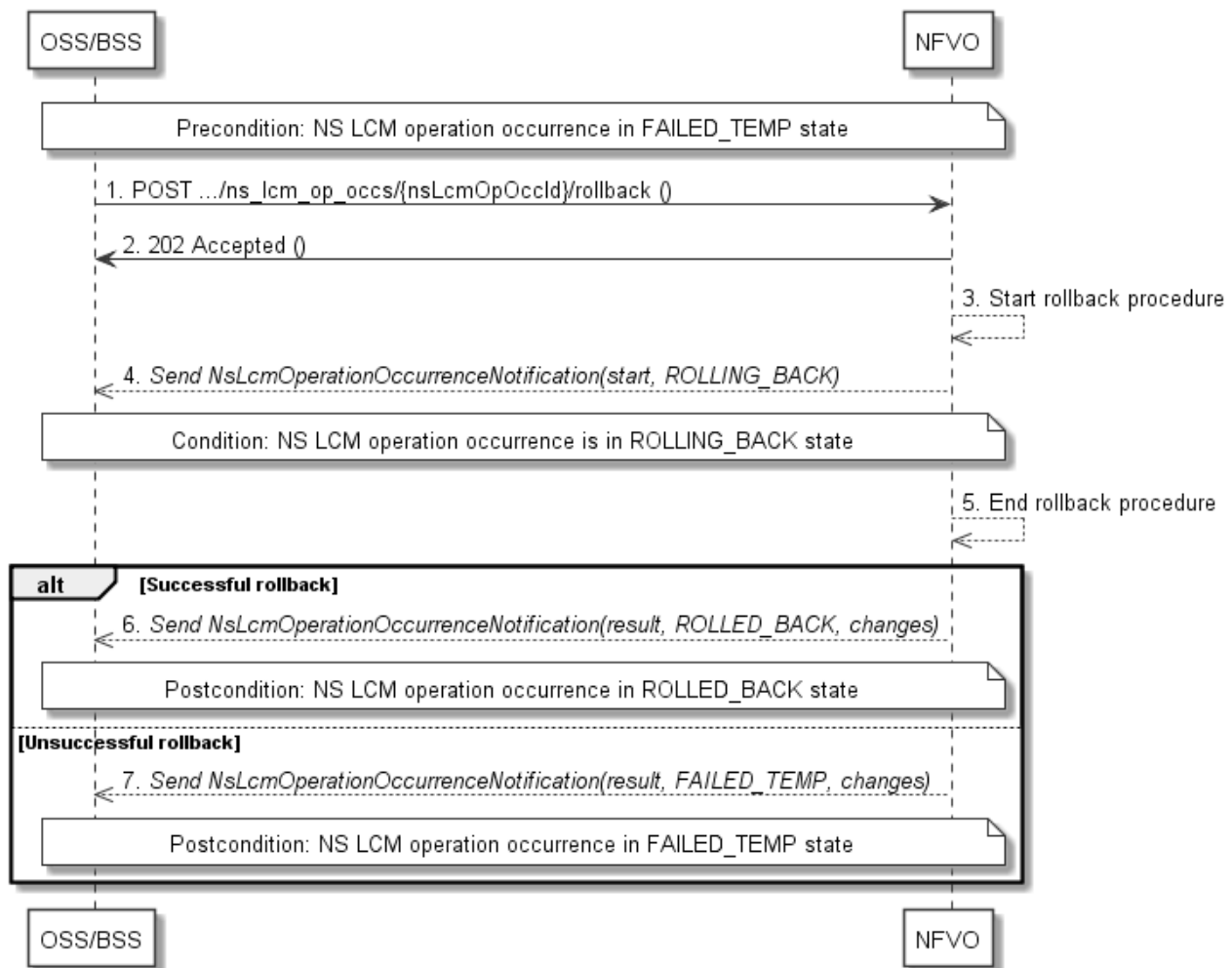


Figure 6.3.8-1: Flow of rolling back an NS lifecycle management operation

NOTE: Due to possible race conditions, the 202 response and the "ROLLING_BACK" NsLcmOperationOccurrenceNotification can arrive in any order at the OSS/BSS.

Precondition: The NS lifecycle operation occurrence is in FAILED_TEMP state.

Initiating the rollback of an NS lifecycle management operation, as illustrated in Figure 6.3.8-1, consists of the following steps:

- 1) The OSS/BSS sends a POST request with an empty body to the "Rollback operation task" resource of the NS LCM operation occurrence that is to be rolled back.
- 2) The NFVO returns a "202 Accepted" response.
- 3) The NFVO starts the rollback procedure.
- 4) The NFVO sends an NS lifecycle management operation occurrence notification of type "start" to indicate that the NS LCM operation occurrence enters the "ROLLING_BACK" state.
- 5) The NFVO finishes the rollback procedure.
- 6) On successful rollback, the NFVO sends an NS lifecycle management operation occurrence notification (see clause 6.3.6) to indicate successful completion of the operation, and inform the OSS/BSS about the changes on the NS components (e.g. VNFs, VLs).
- 7) On unsuccessful retry, the NFVO sends an NS lifecycle management operation occurrence notification (see clause 6.3.6) to indicate an intermediate error (rollback failed) of the operation, and inform the OSS/BSS about the changes on the NS components (e.g. VNFs, VLs).

Postcondition: The NS lifecycle operation occurrence resource is in one of the following states: FAILED_TEMP, ROLLED_BACK. ROLLED_BACK is a terminal state (see clause 6.6.2.2).

Error handling: The operation is rejected in case the NS lifecycle operation occurrence resource is in any other state than FAILED_TEMP, or in case Rollback is not supported for the particular NS LCM operation for the particular NS.

6.3.9 Flow of continuing an NS lifecycle management operation

This clause describes a sequence for continuing an NS lifecycle management operation occurrence that is represented by an "Individual NS LCM operation occurrence" resource. Continue is used if an operation is in FAILED_TEMP state, and there is reason to believe that the current operation can continue despite the error. The error can be fixed later, typically after current NS lifecycle management operation finishes.

A comprehensive description of the handling of NS lifecycle management errors is provided in clause 6.6.

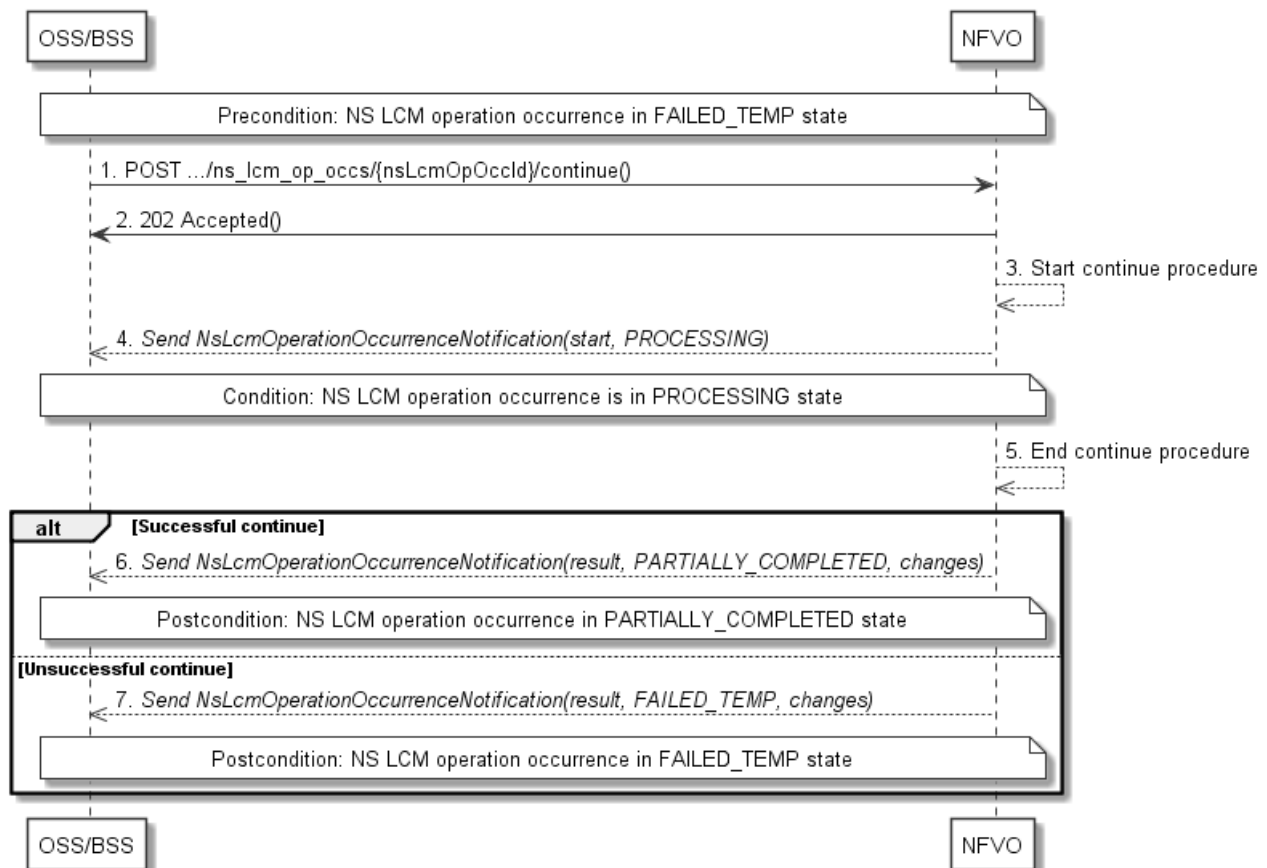


Figure 6.3.9-1: Flow of continuing an NS lifecycle management operation

NOTE: Due to possible race conditions, the 202 response and the "PROCESSING" NsLcmOperationOccurrenceNotification can arrive in any order at the OSS/BSS.

Precondition: The NS lifecycle operation occurrence is in FAILED_TEMP state.

Continuing an NS lifecycle operation, as illustrated in Figure 6.3.9-1, consists of the following steps:

- 1) The OSS/BSS sends a POST request with an empty body to the "Continue operation task" resource of the NS LCM operation occurrence that is to be retried.
- 2) The NFVO returns a "202 Accepted" response.
- 3) The NFVO starts the continue procedure.
- 4) The NFVO sends an NS lifecycle management operation occurrence notification of type "start" to indicate that the NS LCM operation occurrence enters the "PROCESSING" state.

- 5) The NFVO finishes the continue procedure.
- 6) On successful continue, the NFVO sends an NS lifecycle management operation occurrence notification (see clause 6.3.6) to indicate successful completion of the operation, and inform the OSS/BSS about the changes on the NS components (e.g. VNFs, VLs).
- 7) On unsuccessful continue, the NFVO sends an NS lifecycle management operation occurrence notification (see clause 6.3.6) to indicate an intermediate error (continue failed) of the operation, and inform the OSS/BSS about the changes on the NS components (e.g. VNFs, VLs).

Postcondition: The NS lifecycle operation occurrence resource is in one of the following states: FAILED_TEMP, PARTIALLY_COMPLETED. PARTIALLY_COMPLETED is a terminal state (see clause 6.6.2.2).

Error handling: The operation is rejected in case the "NS LCM operation occurrence" resource is in any other state than FAILED_TEMP, or in case Continue is not supported for the particular NS LCM operation for the particular NS.

6.3.10 Flow of failing an NS lifecycle management operation

This clause describes a sequence for declaring as "failed" an NS lifecycle management operation occurrence that is represented by an "Individual NS LCM operation occurrence" resource. If there is neither an assumption that the operation can eventually succeed after further retries, nor that the operation can be successfully rolled back, the operation can be declared as "failed". This will unblock the invocation of other LCM operations, such as HealNs, or non-graceful NS termination, on the affected NS instance.

A comprehensive description of the handling of NS lifecycle management errors is provided in clause 6.6.

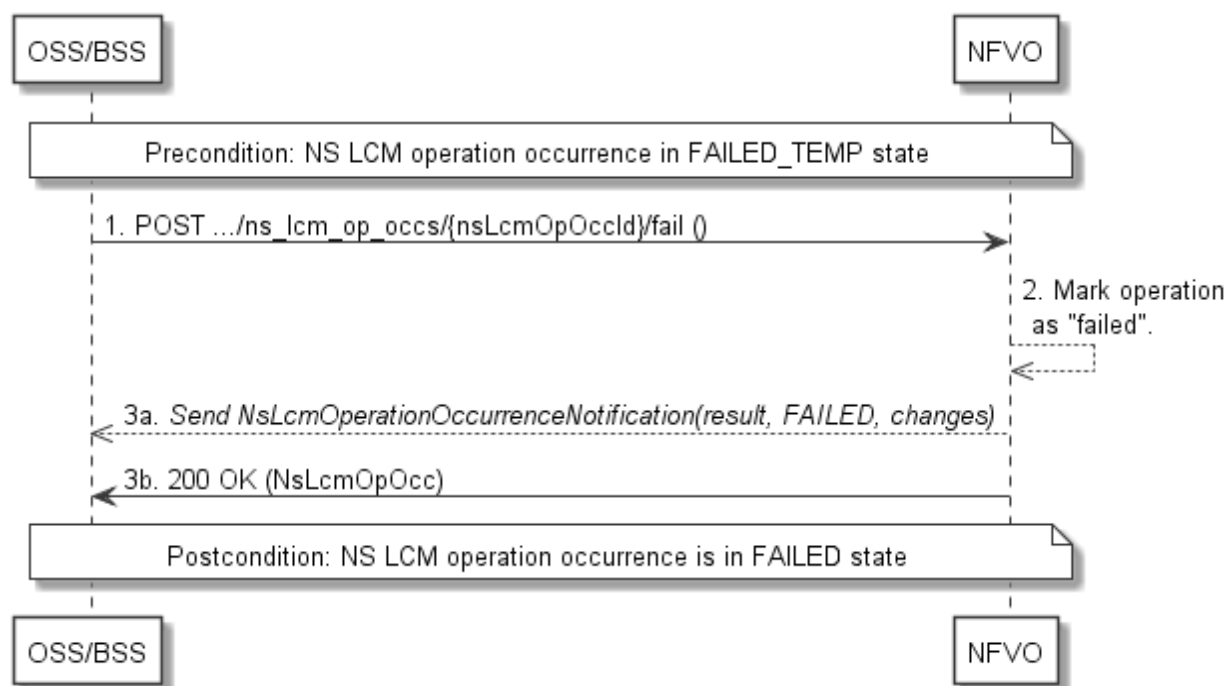


Figure 6.3.10-1: Flow of declaring an NS lifecycle management operation as failed

NOTE: Due to possible race conditions, the 200 response and the "FAILED" NsLcmOperationOccurrenceNotification can arrive in any order at the OSS/BSS.

Precondition: The NS lifecycle operation occurrence is in FAILED_TEMP state.

Declaring an NS lifecycle management operation as failed, as illustrated in Figure 6.3.10-1, consists of the following steps:

- 1) The OSS/BSS sends a POST request with an empty body to the "Fail operation task" resource of the NS LCM operation occurrence that is to be marked as failed.
- 2) The NFVO marks the operation as failed.

- 3) The NFVO sends an NS lifecycle management operation occurrence notification (see clause 6.3.6) to indicate the final failure of the operation, and inform the OSS/BSS about the changes on the NS components (e.g. VNFs, VLs). Furthermore, it returns a "200 OK" response, and includes in the body a NsLcmOpOcc structure. The order in which the response and the notification arrive at the OSS/BSS is not defined.

Postcondition: The NS lifecycle operation occurrence resource is FAILED state. This is a terminal state (see clause 6.6.2.2).

Error handling: The operation is rejected in case the NS lifecycle operation occurrence resource is in any other state than FAILED_TEMP.

6.3.11 Flow of cancelling an NS lifecycle management operation

This clause describes a sequence for cancelling an ongoing NS LCM operation occurrence, or a rollback of an NS LCM operation occurrence. The possibility and timing of cancellation is dependent on the implementation of the underlying lifecycle management operation.

A comprehensive description of the handling of NS lifecycle management errors is provided in clause 6.6.

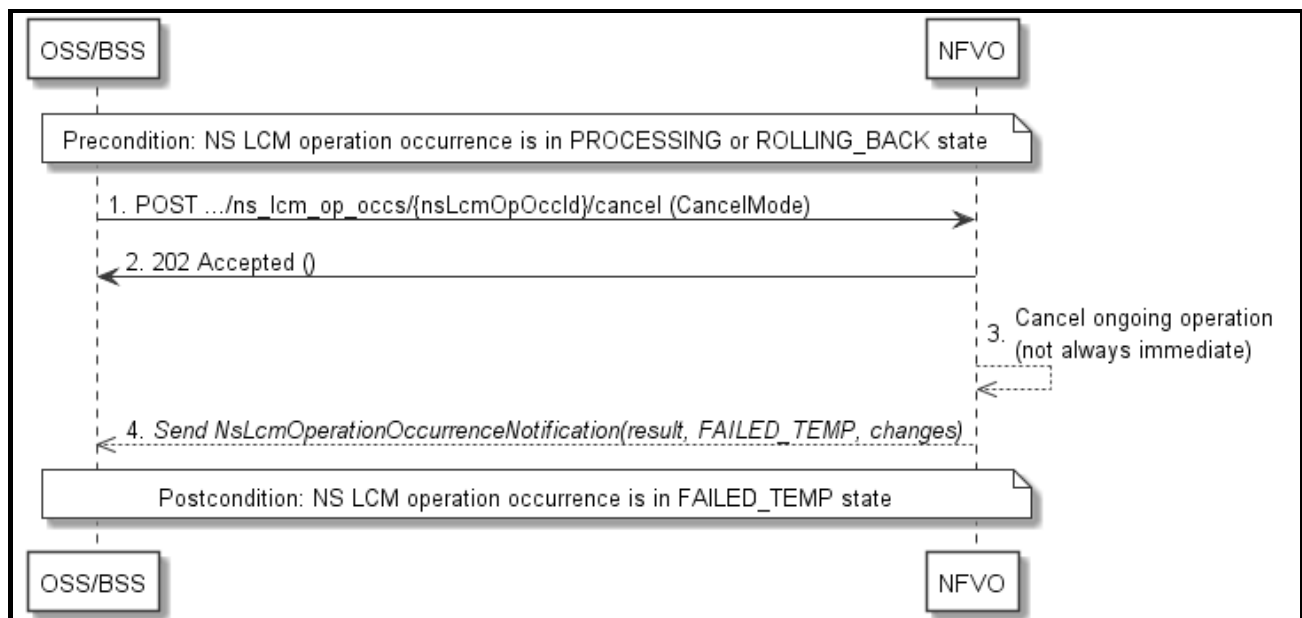


Figure 6.3.11-1: Flow of cancelling an NS lifecycle management operation in "PROCESSING" or "ROLLING_BACK" state

NOTE: Due to possible race conditions, the 202 response and the "FAILED_TEMP" NsLcmOperationOccurrenceNotification can arrive in any order at the OSS/BSS.

Precondition: The NS lifecycle operation occurrence is in PROCESSING or ROLLING_BACK state.

Cancelling an NS lifecycle operation when it is in "PROCESSING" or "ROLLING_BACK" state, as illustrated in Figure 6.3.11-1, consists of the following steps:

- 1) The OSS/BSS sends a POST request with a "CancelMode" structure in the body to the "Cancel operation task" resource of the NS LCM operation occurrence that is to be cancelled.
- 2) The NFVO returns a "202 Accepted" response.
- 3) The NFVO cancels the ongoing LCM operation. This can take some time.
- 4) The NFVO sends an NS lifecycle management operation occurrence notification (see clause 6.3.6) to indicate an intermediate error (cancelled) of the operation, and inform the OSS/BSS about the changes on the NS components (e.g. VNFs, VLs).

Postcondition: The NS lifecycle management operation occurrence resource is in FAILED_TEMP state.

Error handling: The operation is rejected in case the NS lifecycle operation occurrence is in any other state than PROCESSING or ROLLING_BACK, or in case Cancel is not supported for the particular NS LCM operation for the particular NS.

6.4 Resources

6.4.1 Introduction

6.4.1.1 Overview

This clause defines all the resources and methods provided by the NS lifecycle management interface.

6.4.1.2 Task resources that trigger NS LCM operations

A number of resources are defined as task resources to trigger NS LCM operations that are potentially long-running (e.g. Instantiate NS, Scale NS). To represent each occurrence of such an NS LCM operation, an "Individual NS LCM operation occurrence" resource is created as defined in clause 6.3.3.

When successfully executing the POST method on a task resource that triggers an NS LCM operation, asynchronous processing of the request is started, which shall include the following:

- 1) Before returning the "202 Accepted" response to the POST method, a new "Individual NS LCM operation occurrence" resource as defined in clause 6.4.10 shall be created, which represents the underlying NS LCM operation occurrence that is executed by the NFVO. The NFVO shall set the "operationState" in the representation of the "Individual NS LCM operation occurrence" resource to "PROCESSING".
- 2) Notifications of type "NsLcmOperationOccurrenceNotification" shall be triggered as part of executing the underlying NS LCM operation occurrence as defined in clauses 6.5.2.5 and 6.6.2.
- 3) If the NFVO has successfully completed the underlying NS LCM operation occurrence:
 - a) It shall update the representation of the "Individual NS instance" resource which has been changed by the LCM operation to reflect the result of the operation. For individual operations, specific additional conditions can be specified in the following clauses, if applicable.
 - b) It shall set the "operationState" attribute in the representation of the aforementioned "Individual NS LCM operation occurrence" resource to the value "COMPLETED", and shall reflect the changes performed during the LCM operation in the representation of that resource.
 - c) To indicate success, the NFVO shall send a notification of type "NsLcmOperationOccurrenceNotification" with the "operationState" attribute set to "COMPLETED" as defined in clause 6.6.2.
- 4) If executing the underlying NS LCM operation occurrence by the NFVO has failed in the "PROCESSING" phase, which NFVO knows that the NS instance can be brought into a consistent state by immediately rolling back the operation, the NFVO shall send a notification of type "NsLcmOperationOccurrenceNotification" with the "operationState" attribute set to "ROLLING_BACK" as defined in clause 6.6.2. It shall also set the "operationState" attribute in the representation of the aforementioned "Individual NS LCM operation occurrence" resource to the value "ROLLING_BACK".
- 5) If executing the underlying NS LCM operation occurrence by the NFVO has failed with no option to recover, the NFVO shall send a notification of type "NsLcmOperationOccurrenceNotification" with the "operationState" attribute set to "FAILED" as defined in clause 6.6.2. It shall also set the "operationState" attribute in the representation of the aforementioned "Individual NS LCM operation occurrence" resource to the value "FAILED", and shall reflect, at its best knowledge, the changes performed during the LCM operation.

- 6) If executing the underlying NS LCM operation occurrence by the NFVO has failed temporarily, the NFVO shall send a notification of type "NsLcmOperationOccurrenceNotification" with the "operationState" attribute set to "FAILED_TEMP" as defined in clause 6.6.2. It shall also set the "operationState" attribute in the representation of the aforementioned "Individual NS LCM operation occurrence" resource to the value "FAILED_TEMP", and shall reflect, at its best knowledge, the changes performed so far during the LCM operation in the representation of that resource.
- 7) If executing the underlying NS LCM operation occurrence by the NFVO has failed, which the NFVO knows that the failure is acceptable and continues the NS LCM till it finishes, the NFVO shall send a notification of type "NsLcmOperationOccurrenceNotification" with the "operationState" attribute set to "PARTIALLY_COMPLETED" as defined in clause 6.6.2. It shall also set the "operationState" attribute in the representation of the aforementioned "Individual NS LCM operation occurrence" resource to the value "PARTIALLY_COMPLETED", and shall reflect, at its best knowledge, the changes performed so far during the LCM operation in the representation of that resource has been partially completed with acceptable errors.

The preconditions and postconditions for a successful execution of each of the NS lifecycle management operations triggered by the corresponding task resources shall be as defined in Table 6.4.1.2-1.

Table 6.4.1.2-1: Preconditions, postconditions, and parameterization of the flow for different NS lifecycle management operations

Operation	Precondition	Task	RequestStructure	Postcondition
Instantiate NS	NS instance created and in NOT_INSTANTIATED state	instantiate	InstantiateNsRequest	NS instance in INSTANTIATED state
Scale NS	NS instance in INSTANTIATED state	scale	ScaleNsRequest	NS instance still in INSTANTIATED state and NS has been scaled
Update NS	NS instance in INSTANTIATED state	update	UpdateNsRequest	NS instance still in INSTANTIATED state and NS has been updated
Heal NS	NS instance in INSTANTIATED state	heal	HealNsRequest	NS instance still in INSTANTIATED state
Terminate NS	NS instance in INSTANTIATED state	terminate	TerminateNsRequest	NS instance in NOT_INSTANTIATED state

6.4.1a Resource: API versions

The "API versions" resources as defined in clause 9.3.3 of ETSI GS NFV-SOL 013 [16] are part of the NS lifecycle management interface.

6.4.2 Resource: NS Instances

6.4.2.1 Description

This resource represents NS instances. The API consumer can use this resource to create "Individual NS instance" resources, and to query NS instances.

6.4.2.2 Resource definition

The resource URI is:

{apiRoot}/nslcm/{apiMajorVersion}/ns_instances

This resource shall support the resource URI variables defined in Table 6.4.2.2-1.

Table 6.4.2.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 6.1a.

6.4.2.3 Resource methods

6.4.2.3.1 POST

The POST method creates a new NS instance resource.

This method shall follow the provisions specified in the Tables 6.4.2.3.1-1 and 6.4.2.3.1-2 for URI query parameters, request and response data structures, and response codes.

As the result of successfully executing this method, a new "Individual NS instance" resource as defined in clause 6.4.3 shall have been created, and the value of the "instantiationState" attribute in the representation of that resource shall be "NOT_INSTANTIATED". A notification of type NsIdentifierCreationNotification shall be triggered as part of successfully executing this method as defined in clause 6.5.2.6.

Table 6.4.2.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Remarks
n/a		

Table 6.4.2.3.1-2: Details of the POST request/response on this resource

Request body	Data type	Cardinality	Remarks	
	CreateNsRequest	1	The NS creation parameters, as defined in clause 6.5.2.9.	
Response body	Data type	Cardinality	Response Codes	Remarks
	NsInstance	1	201 Created	<p>Shall be returned when a new "Individual NS instance" resource and the associated NS instance identifier has been created successfully.</p> <p>The response body shall contain a representation of the created NS instance, as defined in clause 6.5.2.10.</p> <p>The HTTP response shall include a "Location" HTTP header that contains the resource URI of the created NS instance.</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

6.4.2.3.2 GET

The GET method queries information about multiple NS instances.

This method shall follow the provisions specified in the Tables 6.4.2.3.2-1 and 6.4.2.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 6.4.2.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
filter	0..1	<p>Attribute-based filtering expression according to clause 5.2 of ETSI GS NFV-SOL 013 [16].</p> <p>The NFVO shall support receiving this parameter as part of the URI query string. The OSS/BSS may supply this parameter.</p> <p>All attribute names that appear in the NsInstance and in data types referenced from it shall be supported by the NFVO in the filter expression.</p>
all_fields	0..1	<p>Include all complex attributes in the response. See clause 5.3 of ETSI GS NFV-SOL 013 [16] for details. The NFVO shall support this parameter.</p>
fields	0..1	<p>Complex attributes to be included into the response. See clause 5.3 of ETSI GS NFV-SOL 013 [16] for details. The NFVO should support this parameter.</p>
exclude_fields	0..1	<p>Complex attributes to be excluded from the response. See clause 5.3 of ETSI GS NFV-SOL 013 [16] for details. The NFVO should support this parameter.</p>
exclude-default	0..1	<p>Indicates to exclude the following complex attributes from the response. See clause 5.3 of ETSI GS NFV-SOL 013 [16] for details. The NFVO shall support this parameter.</p> <p>The following attributes shall be excluded from the NsInstance structure in the response body if this parameter is provided, or none of the parameters "all_fields", "fields", "exclude_fields", "exclude_default" are provided:</p> <ul style="list-style-type: none"> - vnInstances - pnfInfo - virtualLinkInfo - vnffgInfo - sapInfo - nsScaleStatus - additionalAffinityOrAntiAffinityRules
nextpage_opaque_marker	0..1	<p>Marker to obtain the next page of a paged response. Shall be supported by the NFVO if the NFVO supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource.</p>

Table 6.4.2.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Remarks	
	n/a			
Response body	Data type	Cardinality	Response Codes	Remarks
	NsInstance	0..N	200 OK	<p>Shall be returned when information about zero or more NS instances has been queried successfully.</p> <p>The response body shall contain in an array the representations of zero or more NS instances, as defined in clause 6.5.2.8.</p> <p>If the "filter" URI parameter or one of the "all_fields", "fields" (if supported), "exclude_fields" (if supported) or "exclude_default" URI parameters was supplied in the request, the data in the response body shall have been transformed according to the rules specified in clauses 5.2.2 and 5.3.2 of ETSI GS NFV-SOL 013 [16], respectively.</p> <p>If the NFVO supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource, inclusion of the Link HTTP header in this response shall follow the provisions in clause 5.4.2.3 of ETSI GS NFV-SOL 013 [16].</p>
	ProblemDetails	1	400 Bad Request	<p>Shall be returned upon the following error: Invalid attribute-based filtering expression.</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.</p>
	ProblemDetails	1	400 Bad Request	<p>Shall be returned upon the following error: Invalid attribute selector.</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.</p>
	ProblemDetails	1	400 Bad Request	<p>Shall be returned upon the following error: Response too big.</p> <p>If the NFVO supports alternative 1 (error) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource, this error response shall follow the provisions in clause 5.4.2.2 of ETSI GS NFV-SOL 013 [16].</p>
ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.	

6.4.2.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

6.4.2.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

6.4.2.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

6.4.3 Resource: Individual NS Instance

6.4.3.1 Description

This resource represents an individual NS instance. The API consumer can use this resource to modify and delete the underlying NS instance, and to read information about the NS instance.

6.4.3.2 Resource definition

The resource URI is:

{apiRoot}/nslcm/{apiMajorVersion}/ns_instances/{nsInstanceId}

The base resource URI variables for this resource are defined in Table 6.4.3.2-1.

Table 6.4.3.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 6.1a.
nsInstanceId	Identifier of the NS instance

6.4.3.3 Resource methods

6.4.3.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

6.4.3.3.2 GET

The GET method retrieves information about an NS instance by reading an "Individual NS instance" resource.

This method shall follow the provisions specified in the Tables 6.4.3.3.2-1 and 6.4.3.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 6.4.3.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Remarks
n/a		

Table 6.4.3.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Remarks	
	n/a			
Response body	Data type	Cardinality	Response codes	Remarks
	NsInstance	1	200 OK	Shall be returned when information about an individual NS instance has been read successfully. The response body shall contain a representation of the NS instance, as defined in clause 6.5.2.10.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

6.4.3.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

6.4.3.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

6.4.3.3.5 DELETE

This method deletes an "Individual NS instance" resource.

This method shall follow the provisions specified in the Tables 6.4.3.3.5-1 and 6.4.3.3.5-2 for URI query parameters, request and response data structures, and response codes.

As the result of successfully executing this method, the "Individual NS instance" resource shall not exist any longer. A notification of type "NsIdentifierDeletionNotification" shall be triggered as part of successfully executing this method as defined in clause 6.5.2.7.

Table 6.4.3.3.5-1: URI query parameters supported by the DELETE method on this resource

Name	Cardinality	Remarks
n/a		

Table 6.4.3.3.5-2: Details of the DELETE request/response on this resource

Request body	Data type	Cardinality	Remarks	
	n/a			
Response body	Data type	Cardinality	Response Codes	Remarks
	n/a		204 No Content	<p>Shall be returned when the "Individual NS instance" resource and the associated NS identifier have been deleted successfully.</p> <p>The response body shall be empty.</p>
	ProblemDetails	1	409 Conflict	<p>Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource.</p> <p>Typically, this is due to the fact that the NS instance resource is in INSTANTIATED state.</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

6.4.4 Resource: Instantiate NS task

6.4.4.1 Description

This task resource represents the "Instantiate NS" operation. The API consumer can use this resource to instantiate an NS instance.

6.4.4.2 Resource definition

The resource URI is:

{apiRoot}/nslcm/{apiMajorVersion}/ns_instances/{nsInstanceId}/instantiate

This resource shall support the resource URI variables defined in Table 6.4.4.2-1.

Table 6.4.4.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 6.1a.
nsInstanceId	Identifier of the NS instance to be instantiated.

6.4.4.3 Resource methods

6.4.4.3.1 POST

The POST method instantiates an NS instance.

This method shall follow the provisions specified in the Tables 6.4.4.3.1-1 and 6.4.4.3.1-2 for URI query parameters, request and response data structures, and response codes.

The steps and conditions that apply as the result of successfully executing this method are specified in clause 6.4.1.2.

In addition, once the NFVO has successfully completed the underlying NS LCM operation occurrence, it shall set the "nsState" attribute to the value "INSTANTIATED" in the representation of the "Individual NS instance" resource.

Table 6.4.4.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
None supported		

Table 6.4.4.3.1-2: Details of the POST request/response on this resource

Request body	Data type	Cardinality	Description	
	InstantiateNsRequest	1	Parameters for the instantiate NS operation, as defined in clause 6.5.2.11.	
Response body	Data type	Cardinality	Response Codes	Description
	n/a		202 Accepted	<p>Shall be returned when the request has been accepted for processing, but the processing has not been completed.</p> <p>The response body shall be empty.</p> <p>The HTTP response shall include a "Location" HTTP header that contains the URI of the newly-created "Individual NS LCM operation occurrence" resource corresponding to the operation.</p>
	ProblemDetails	1	409 Conflict	<p>Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource.</p> <p>Typically, this is due to the fact that the NS instance resource is in the INSTANTIATED state, or that another lifecycle management operation is ongoing.</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

6.4.4.3.2 GET

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

6.4.4.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

6.4.4.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

6.4.4.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

6.4.5 Resource: Scale NS task

6.4.5.1 Description

This task resource represents the "Scale NS" operation. The API consumer can use this resource to request scaling an NS instance.

Scaling an NS instance can be performed by explicitly adding/removing existing VNF instances to/from the NS instance, by leveraging on the abstraction mechanism provided by the NS scaling aspects and NS levels information elements declared in the NSD or by scaling individual VNF instances that are part of the NS itself. When adding VNFs and nested NSs - already existing or not - to the NS to be scaled, the NFVO shall follow the indications provided by the dependencies attribute, as specified in the corresponding NSD.

NOTE: In case the NS is a composite NS, it is also possible to scale directly its nested NS, as they are also NS and thus indirectly effectively scale the composite NS.

For the case of scaling individual VNF instances that are part of the NS instance, it depends on the VNF capabilities, and is declared in the corresponding VNFD, whether the scaling of the individual VNF instance is supported.

6.4.5.2 Resource definition

The resource URI is:

{apiRoot}/nslcm/{apiMajorVersion}/ns_instances/{nsInstanceId}/scale

This resource shall support the resource URI variables defined in Table 6.4.5.2-1.

Table 6.4.5.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 6.1a.
nsInstanceId	Identifier of the NS instance to be scaled.

6.4.5.3 Resource methods

6.4.5.3.1 POST

The POST method requests to scale an NS instance resource.

This method shall follow the provisions specified in the Tables 6.4.5.3.1-1 and 6.4.5.3.1-2 for URI query parameters, request and response data structures, and response codes.

The steps and conditions that apply as the result of successfully executing this method are specified in clause 6.4.1.2.

In addition, once the NFVO has successfully completed the underlying NS LCM operation occurrence, it shall reflect the result of scaling the NS instance by updating the "nsScaleStatus" attribute in the representation of the "Individual NS instance" resource.

Table 6.4.5.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
none supported		

Table 6.4.5.3.1-2: Details of the POST request/response on this resource

Request body	Data type	Cardinality	Description	
	ScaleNsRequest	1	Parameters for the scale NS operation, as defined in clause 6.5.2.14.	
Response body	Data type	Cardinality	Response Codes	Description
	n/a		202 Accepted	<p>Shall be returned when the request has been accepted for processing, but the processing has not been completed.</p> <p>The response body shall be empty.</p> <p>The HTTP response shall include a "Location" HTTP header that contains the URI of the newly-created "Individual NS lifecycle operation occurrence" resource corresponding to the operation.</p>
	ProblemDetails	1	409 Conflict	<p>Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource.</p> <p>Typically, this is due to the fact that the NS instance resource is in NOT_INSTANTIATED state, or that another lifecycle management operation is ongoing.</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</p>
ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.	

6.4.5.3.2 GET

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

6.4.5.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

6.4.5.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

6.4.5.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

6.4.6 Resource: Update NS task

6.4.6.1 Description

This task resource represents the "Update NS" operation. The API consumer can use this resource to update an NS instance.

For the cases of "change VNF deployment flavour" and "operate VNF" on individual VNF instances that are part of the NS instance, it depends on the VNF capabilities, and is declared in the VNFD, whether this operation is supported for a particular VNF. This operation may be service-disruptive.

6.4.6.2 Resource definition

The resource URI is:

{apiRoot}/nslcm/{apiMajorVersion}/ns_instances/{nsInstanceId}/update

This resource shall support the resource URI variables defined in Table 6.4.6.2-1.

Table 6.4.6.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 6.1a.
nsInstanceId	Identifier of the NS instance to be updated.

6.4.6.3 Resource methods

6.4.6.3.1 POST

The POST method updates an NS instance.

This method shall follow the provisions specified in the Tables 6.4.6.3.1-1 and 6.4.6.3.1-2 for URI query parameters, request and response data structures, and response codes.

The steps and conditions that apply as the result of successfully executing this method are specified in clause 6.4.1.2.

In addition, once the NFVO has successfully completed the underlying NS LCM operation occurrence, it shall reflect the result of the update of the NS instance by updating the representation of the "Individual NS instance" resource accordingly to the update.

Table 6.4.6.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
none supported		

Table 6.4.6.3.1-2: Details of the POST request/response on this resource

Request body	Data type	Cardinality	Description	
	UpdateNsRequest	1	Parameters for the update NS operation, as defined in clause 6.5.2.12.	
Response body	Data type	Cardinality	Response Codes	Description
	n/a		202 Accepted	<p>Shall be returned when the request has been accepted for processing, but the processing has not been completed.</p> <p>The response body shall be empty.</p> <p>The HTTP response shall include a "Location" HTTP header that contains the URI of the newly-created "Individual NS lifecycle operation occurrence" resource corresponding to the operation.</p>
	ProblemDetails	1	409 Conflict	<p>Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource.</p> <p>Typically, this is due to the fact that the NS instance resource is in NOT_INSTANTIATED state, or that another lifecycle management operation is ongoing.</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</p>
ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.	

6.4.6.3.2 GET

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

6.4.6.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

6.4.6.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

6.4.6.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

6.4.7 Resource: Heal NS task

6.4.7.1 Description

This task resource represents the "Heal NS" operation. The API consumer can use this resource to request healing an NS instance.

For the case of healing individual VNF instances that are part of the NS instance, it depends on VNF capabilities, and is declared in the VNFD, whether the healing of the individual VNF instance is supported.

6.4.7.2 Resource definition

The resource URI is:

{apiRoot}/nslcm/{apiMajorVersion}/ns_instances/{nsInstanceId}/heal

This resource shall support the resource URI variables defined in Table 6.4.7.2-1.

Table 6.4.7.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 6.1a.
nsInstanceId	Identifier of the NS instance to be healed.

6.4.7.3 Resource methods

6.4.7.3.1 POST

The POST method requests to heal an NS instance.

This method shall follow the provisions specified in the Tables 6.4.7.3.1-1 and 6.4.7.3.1-2 for URI query parameters, request and response data structures, and response codes.

The steps and conditions that apply as the result of successfully executing this method are specified in clause 6.4.1.2.

Table 6.4.7.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
none supported		

Table 6.4.7.3.1-2: Details of the POST request/response on this resource

Request body	Data type	Cardinality	Description	
	HealNsRequest	1	Parameters for the heal NS operation, as defined in clause 6.5.2.13.	
Response body	Data type	Cardinality	Response Codes	Description
	n/a		202 Accepted	<p>Shall be returned when the request has been accepted for processing, but the processing has not been completed.</p> <p>The response body shall be empty.</p> <p>The HTTP response shall include a "Location" HTTP header that contains the URI of the newly-created "Individual NS lifecycle operation occurrence" resource corresponding to the operation.</p>
	ProblemDetails	1	409 Conflict	<p>Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource.</p> <p>Typically, this is due to the fact that the NS instance resource is in NOT_INSTANTIATED state, or that another lifecycle management operation is ongoing.</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</p>
ProblemDetails	See clause 6.4 of [16]		4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

6.4.7.3.2 GET

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

6.4.7.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

6.4.7.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

6.4.7.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

6.4.8 Resource: Terminate NS task

6.4.8.1 Description

This task resource represents the "Terminate NS" operation. The API consumer can use this resource to terminate an NS instance.

6.4.8.2 Resource definition

The resource URI is:

{apiRoot}/nslcm/{apiMajorVersion}/ns_instances/{nsInstanceId}/terminate

This resource shall support the resource URI variables defined in Table 6.4.8.2-1.

Table 6.4.8.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 6.1a.
nsInstanceId	The identifier of the NS instance to be terminated.

6.4.8.3 Resource methods

6.4.8.3.1 POST

The POST method terminates an NS instance.

This method shall follow the provisions specified in the Tables 6.4.8.3.1-1 and 6.4.8.3.1-2 for URI query parameters, request and response data structures, and response codes.

The steps and conditions that apply as the result of successfully executing this method are specified in clause 6.4.1.2.

In addition, once the NFVO has successfully completed the underlying NS LCM operation occurrence, it shall set the "nsState" attribute in the representation of the "Individual NS instance" resource to the value "NOT_INSTANTIATED".

This method can only be used with an NS instance in the INSTANTIATED state. Terminating an NS instance does not delete the NS instance identifier, but rather transitions the NS into the NOT_INSTANTIATED state.

Table 6.4.8.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Remarks
n/a		

Table 6.4.8.3.1-2: Details of the POST request/response on this resource

Request body	Data type	Cardinality	Remarks	
	TerminateNsRequest	1	The terminate NS request parameters, as defined in clause 6.5.2.15.	
Response body	Data type	Cardinality	Response Codes	Remarks
	n/a	1	202 Accepted	<p>Shall be returned when the request has been accepted for processing.</p> <p>The response body shall be empty.</p> <p>The HTTP response shall include a "Location" HTTP header that contains the URI of the newly-created "Individual NS lifecycle operation occurrence" resource corresponding to the operation.</p>
	ProblemDetails	1	409 Conflict	<p>Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource.</p> <p>Typically, this is due to the fact that the NS instance resource is in NOT_INSTANTIATED state, or that another lifecycle management operation is ongoing.</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</p>
ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.	

6.4.8.3.2 GET

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

6.4.8.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

6.4.8.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

6.4.8.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

6.4.9 Resource: NS LCM operation occurrences

6.4.9.1 Description

This resource represents NS lifecycle management operation occurrences. The API consumer can use this resource to query status information about multiple NS lifecycle management operation occurrences.

6.4.9.2 Resource definition

The resource URI is:

{apiRoot}/nslcm/{apiMajorVersion}/ns_lcm_op_occs

The base resource URI variables for this resource are defined in Table 6.4.9.2-1.

Table 6.4.9.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 6.1a.

6.4.9.3 Resource methods

6.4.9.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

6.4.9.3.2 GET

The API consumer can use this method to query status information about multiple NS lifecycle management operation occurrences.

This method shall follow the provisions specified in the Tables 6.4.9.3.2-1 and 6.4.9.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 6.4.9.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
filter	0..1	Attribute-based filtering expression according to clause 5.2 of ETSI GS NFV-SOL 013 [16]. The NFVO shall support receiving this parameter as part of the URI query string. The OSS/BSS may supply this parameter. All attribute names that appear in the NsLcmOpOcc and in data types referenced from it shall be supported by the NFVO in the filter expression.
fields	0..1	Complex attributes to be included into the response. See clause 5.3 of ETSI GS NFV-SOL 013 [16] for details. The NFVO should support this parameter.
exclude_fields	0..1	Complex attributes to be excluded from the response. See clause 5.3 of ETSI GS NFV-SOL 013 [16] for details. The NFVO should support this parameter.
exclude_default	0..1	Indicates to exclude the following complex attributes from the response. See clause 5.3 of ETSI GS NFV-SOL 013 [16] for details. The NFVO shall support this parameter. The following attributes shall be excluded from the NsLcmOpOcc structure in the response body if this parameter is provided: <ul style="list-style-type: none"> - operationParams - changedVnfInfo - error - resourceChanges

Name	Cardinality	Description
nextpage_opaque_marker	0..1	Marker to obtain the next page of a paged response. Shall be supported by the NFVO if the NFVO supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource.

Table 6.4.9.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	NsLcmOpOcc	0..N	200 OK	<p>Shall be returned when status information for zero or more NS lifecycle management operation occurrences has been queried successfully.</p> <p>The response body shall contain in an array the status information about zero or more NS lifecycle operation occurrences, as defined in clause 6.5.2.3.</p> <p>If the "filter" URI parameter or one of the "all_fields", "fields", "exclude_fields" or "exclude_default" URI parameters was supplied in the request and is supported, the data in the response body shall have been transformed according to the rules specified in clauses 5.2.2 and 5.3.2 of ETSI GS NFV-SOL 013 [16], respectively.</p> <p>If the NFVO supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource, inclusion of the Link HTTP header in this response shall follow the provisions in clause 5.4.2.3 of ETSI GS NFV-SOL 013 [16].</p>
	ProblemDetails	1	400 Bad Request	<p>Shall be returned upon the following error: Invalid attribute-based filtering expression.</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.</p>
	ProblemDetails	1	400 Bad Request	<p>Shall be returned upon the following error: Invalid attribute selector.</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.</p>
	ProblemDetails	1	400 Bad Request	<p>Shall be returned upon the following error: Response too big.</p> <p>If the NFVO supports alternative 1 (error) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource, this error response shall follow the provisions in clause 5.4.2.2 of ETSI GS NFV-SOL 013 [16].</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

6.4.9.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

6.4.9.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

6.4.9.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

6.4.10 Resource: Individual NS LCM operation occurrence

6.4.10.1 Description

This resource represents an NS lifecycle management operation occurrence. The API consumer can use this resource to read status information about an individual NS lifecycle management operation occurrence. Further, the API consumer can use task resources which are children of this resource to request cancellation of an operation in progress, and to request the handling of operation errors via retrying the operation, rolling back the operation, or permanently failing the operation.

The NFVO may remove an "Individual NS LCM operation occurrence" resource some time after it has reached one of the terminal states (i.e. the "operationState" attribute of its representation is equal to one of the values "COMPLETED", "FAILED" or "ROLLED_BACK"). The minimum time how long the NFVO waits before deleting such a resource is defined by means outside the scope of the present document.

6.4.10.2 Resource definition

The resource URI is:

{apiRoot}/nslcm/{apiMajorVersion}/ns_lcm_op_occs/{nsLcmOpOccId}

The base resource URI variables for this resource are defined in Table 6.4.10.2-1.

Table 6.4.10.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 6.1a.
nsLcmOpOccId	Identifier of an NS lifecycle management operation occurrence.

6.4.10.3 Resource methods

6.4.10.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

6.4.10.3.2 GET

The API consumer can use this method to retrieve status information about an NS lifecycle management operation occurrence by reading an individual "NS LCM operation occurrence" resource.

This method shall follow the provisions specified in the Tables 6.4.10.3.2-1 and 6.4.10.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 6.4.10.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
none supported		

Table 6.4.10.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	NsLcmOpOcc	1	200 OK	Shall be returned when information about an NS LCM operation occurrence has been read successfully. The response body shall contain status information about an NS lifecycle management operation occurrence (see clause 6.5.2.3).
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

6.4.10.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

6.4.10.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

6.4.10.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

6.4.11 Resource: Retry operation task

6.4.11.1 Description

This task resource represents the "Retry operation" operation. The API consumer can use this resource to initiate retrying an NS lifecycle management operation that is in a transient failure state. See also clause 6.6.2.3.

6.4.11.2 Resource definition

The resource URI is:

{apiRoot}/nslcm/{apiMajorVersion}/ns_lcm_op_occs/{nsLcmOpOccId}/retry

This resource shall support the resource URI variables defined in Table 6.4.11.2-1.

Table 6.4.11.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 6.1a.
nsLcmOpOccId	Identifier of an NS lifecycle management operation occurrence to be retried. See note.
NOTE:	This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request triggering an NS LCM operation. It can also be retrieved from the "nsLcmOpOccId" attribute in the NsLcmOperationOccurrenceNotification.

6.4.11.3 Resource methods

6.4.11.3.1 POST

The POST method initiates retrying an NS lifecycle management operation if that operation has experienced a temporary failure, i.e. the related "NS LCM operation occurrence" is in "FAILED_TEMP" state.

This method shall follow the provisions specified in the Tables 6.4.11.3.1-1 and 6.4.11.3.1-2 for URI query parameters, request and response data structures, and response codes.

In case of success of processing the asynchronous request, the "operationState" attribute in the representation of the parent resource shall be changed to "PROCESSING" and the applicable "start" notification according to clause 6.6.2.2 shall be emitted to indicate that the underlying NS LCM operation occurrence proceeds.

Table 6.4.11.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
none supported		

Table 6.4.11.3.1-2: Details of the POST request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a		The POST request to this resource has an empty payload body.	
Response body	Data type	Cardinality	Response Codes	Description
	n/a		202 Accepted	<p>Shall be returned when the request has been accepted for processing, but processing has not been completed.</p> <p>The response shall have an empty payload body.</p>
	ProblemDetails	0..1	404 Not Found	<p>Shall be returned upon the following error: The API producer did not find a current representation for the target resource or is not willing to disclose that one exists.</p> <p>The general cause for this error and its handling is specified in clause 6.4 of ETSI GS NFV-SOL 013 [16], including rules for the presence of the response body.</p> <p>Specifically in case of this task resource, the response code 404 shall also be returned if the task is not supported for the NS LCM operation occurrence represented by the parent resource, which means that the task resource consequently does not exist.</p> <p>In this case, the response body shall be present, and shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</p>
	ProblemDetails	1	409 Conflict	<p>Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the NS LCM operation occurrence resource.</p> <p>Typically, this is due to the fact that the NS LCM operation occurrence is not in FAILED_TEMP state, or another error handling action is starting, such as rollback or fail.</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</p>
ProblemDetails	See clause 6.4 of [16]		4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

6.4.11.3.2 GET

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

6.4.11.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

6.4.11.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

6.4.11.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

6.4.12 Resource: Rollback operation task

6.4.12.1 Description

This task resource represents the "Rollback operation" operation. The API consumer can use this resource to initiate rolling back an NS lifecycle management operation. See also clause 6.6.2.3.

6.4.12.2 Resource definition

The resource URI is:

{apiRoot}/nslcm/{apiMajorVersion}/ns_lcm_op_occs/{nsLcmOpOccId}/rollback

This resource shall support the resource URI variables defined in Table 6.4.12.2-1.

Table 6.4.12.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 6.1a.
nsLcmOpOccId	Identifier of an NS lifecycle management operation occurrence to be rolled back. See note.
NOTE:	This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request triggering an NS LCM operation. It can also be retrieved from the "nsLcmOpOccId" attribute in the NsLcmOperationOccurrenceNotification.

6.4.12.3 Resource methods

6.4.12.3.1 POST

The POST method initiates rolling back an NS lifecycle operation if that operation has experienced a temporary failure, i.e. the related "NS LCM operation occurrence" is in "FAILED_TEMP" state.

This method shall follow the provisions specified in the Tables 6.4.12.3.1-1 and 6.4.12.3.1-2 for URI query parameters, request and response data structures, and response codes.

In case of success of processing the asynchronous request, the "operationState" attribute in the representation of the parent resource shall be changed to "ROLLING_BACK" and the applicable "start" notification according to clause 6.6.2.2 shall be emitted to indicate that rollback of the underlying NS LCM operation occurrence is attempted.

Table 6.4.12.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
none supported		

Table 6.4.12.3.1-2: Details of the POST request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a		The POST request to this resource has an empty payload body.	
Response body	Data type	Cardinality	Response Codes	Description
	n/a		202 Accepted	<p>Shall be returned when the request has been accepted for processing, but processing has not been completed.</p> <p>The response shall have an empty payload body.</p>
	ProblemDetails	0..1	404 Not Found	<p>Shall be returned upon the following error: The API producer did not find a current representation for the target resource or is not willing to disclose that one exists.</p> <p>The general cause for this error and its handling is specified in clause 6.4 of ETSI GS NFV-SOL 013 [16], including rules for the presence of the response body.</p> <p>Specifically, in case of this task resource, the response code 404 shall also be returned if the task is not supported for the NS LCM operation occurrence represented by the parent resource, which means that the task resource consequently does not exist.</p> <p>In this case, the response body shall be present, and shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</p>
	ProblemDetails	1	409 Conflict	<p>Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the NS LCM operation occurrence resource.</p> <p>Typically, this is due to the fact that the NS LCM operation occurrence is not in FAILED_TEMP state, or another error handling action is starting, such as retry or fail.</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</p>
ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.	

6.4.12.3.2 GET

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

6.4.12.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

6.4.12.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

6.4.12.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

6.4.13 Resource: Continue operation task

6.4.13.1 Description

This task resource represents the "Continue operation" operation. The API consumer can use this resource to initiate continuing an NS lifecycle management operation. See also clause 6.6.2.3.

6.4.13.2 Resource definition

The resource URI is:

{apiRoot}/nslcm/{apiMajorVersion}/ns_lcm_op_occs/{nsLcmOpOccId}/continue

This resource shall support the resource URI variables defined in Table 6.4.13.2-1.

Table 6.4.13.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 6.1a.
nsLcmOpOccId	Identifier of an NS lifecycle management operation occurrence to be continued. See note.
NOTE:	This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request triggering an NS LCM operation. It can also be retrieved from the "nsLcmOpOccId" attribute in the NsLcmOperationOccurrenceNotification.

6.4.13.3 Resource methods

6.4.13.3.1 POST

The POST method initiates continuing an NS lifecycle operation if that operation has experienced a temporary failure, i.e. the related "NS LCM operation occurrence" is in "FAILED_TEMP" state.

This method shall follow the provisions specified in the Tables 6.4.13.3.1-1 and 6.4.13.3.1-2 for URI query parameters, request and response data structures, and response codes.

In case of success of processing the asynchronous request, the "operationState" attribute in the representation of the parent resource shall be changed to "PROCESSING" and the applicable "start" notification according to clause 6.6.2.2 shall be emitted to indicate that the underlying NS LCM operation occurrence continues.

Table 6.4.13.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
none supported		

Table 6.4.13.3.1-2: Details of the POST request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			The POST request to this resource has an empty payload body.
Response body	Data type	Cardinality	Response Codes	Description
	n/a		202 Accepted	<p>Shall be returned when the request has been accepted for processing, but processing has not been completed.</p> <p>The response shall have an empty payload body.</p>
	ProblemDetails	0..1	404 Not Found	<p>Shall be returned upon the following error: The API producer did not find a current representation for the target resource or is not willing to disclose that one exists.</p> <p>The general cause for this error and its handling is specified in clause 6.4 of ETSI GS NFV-SOL 013 [16], including rules for the presence of the response body.</p> <p>Specifically, in case of this task resource, the response code 404 shall also be returned if the task is not supported for the NS LCM operation occurrence represented by the parent resource, which means that the task resource consequently does not exist.</p> <p>In this case, the response body shall be present, and shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</p>
	ProblemDetails	1	409 Conflict	<p>Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the NS LCM operation occurrence resource.</p> <p>Typically, this is due to the fact that the NS LCM operation occurrence is not in FAILED_TEMP state, or another error handling action is starting, such as retry or fail.</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

6.4.13.3.2 GET

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

6.4.13.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

6.4.13.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

6.4.13.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

6.4.14 Resource: Fail operation task

6.4.14.1 Description

This task resource represents the "Fail operation" operation. The API consumer can use this resource to mark an NS lifecycle management operation occurrence as "finally failed", i.e. change the state of the related NS LCM operation occurrence resource to "FAILED", if it is not assumed that a subsequent retry or rollback will succeed. Once the operation is marked as "finally failed", it cannot be retried or rolled back anymore. See also clause 6.6.2.3.

6.4.14.2 Resource definition

The resource URI is:

{apiRoot}/nslcm/{apiMajorVersion}/ns_lcm_op_occs/{nsLcmOpOccId}/fail

This resource shall support the resource URI variables defined in Table 6.4.14.2-1.

Table 6.4.14.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 6.1a.
nsLcmOpOccId	Identifier of an NS lifecycle management operation occurrence to be marked as "failed". See note.
NOTE:	This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request triggering an NS LCM operation. It can also be retrieved from the "nsLcmOpOccId" attribute in the NsLcmOperationOccurrenceNotification.

6.4.14.3 Resource methods

6.4.14.3.1 POST

The POST method marks an NS lifecycle management operation occurrence as "finally failed" if that operation occurrence is in "FAILED_TEMP" state.

This method shall follow the provisions specified in the Tables 6.4.14.3.1-1 and 6.4.14.3.1-2 for URI query parameters, request and response data structures, and response codes.

In case of success, the "operationState" attribute in the representation of the parent resource shall be changed to "FAILED" and the applicable "result" notification according to clause 6.6.2.2 shall be emitted to indicate that the execution of the underlying NS LCM operation occurrence has finally and unrecoverable failed.

Table 6.4.14.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
None supported		

Table 6.4.14.3.1-2: Details of the POST request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a		The POST request to this resource has an empty payload body.	
Response body	Data type	Cardinality	Response Codes	Description
	NsLcmOpOcc	1	200 OK	<p>Shall be returned when the state of the NS lifecycle management operation occurrence has been changed successfully.</p> <p>The response shall include a representation of the "Individual NS lifecycle management operation occurrence" resource.</p>
	ProblemDetails	0..1	404 Not Found	<p>Shall be returned upon the following error: The API producer did not find a current representation for the target resource or is not willing to disclose that one exists.</p> <p>The general cause for this error and its handling is specified in clause 6.4 of ETSI GS NFV-SOL 013 [16], including rules for the presence of the response body.</p> <p>Specifically in case of this task resource, the response code 404 shall also be returned if the task is not supported for the NS LCM operation occurrence represented by the parent resource, which means that the task resource consequently does not exist.</p> <p>In this case, the response body shall be present, and shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</p>
	ProblemDetails	1	409 Conflict	<p>Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the NS LCM operation occurrence resource.</p> <p>Typically, this is due to the fact that the NS LCM operation occurrence is not in FAILED_TEMP state, or another error handling action is starting, such as retry or rollback.</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

6.4.14.3.2 GET

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

6.4.14.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

6.4.14.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

6.4.14.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

6.4.15 Resource: Cancel operation task

6.4.15.1 Description

This task resource represents the "Cancel operation" operation. The API consumer can use this resource to cancel an ongoing NS lifecycle management operation. See also clause 6.6.2.3.

6.4.15.2 Resource definition

The resource URI is:

{apiRoot}/nslcm/{apiMajorVersion}/ns_lcm_op_occs/{nsLcmOpOccId}/cancel

This resource shall support the resource URI variables defined in Table 6.4.15.2-1.

Table 6.4.15.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 6.1a.
nsLcmOpOccId	Identifier of an NS lifecycle management operation occurrence to be cancelled. See note.
NOTE:	This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request triggering an NS LCM operation. It can also be retrieved from the "nsLcmOpOccId" attribute in the NsLcmOperationOccurrenceNotification.

6.4.15.3 Resource methods

6.4.15.3.1 POST

The POST method initiates cancelling an ongoing NS lifecycle management operation while it is being executed or rolled back, i.e. the related "NS LCM operation occurrence" is either in "PROCESSING" or "ROLLING_BACK" state.

This method shall follow the provisions specified in the Tables 6.4.15.3.1-1 and 6.4.15.3.1-2 for URI query parameters, request and response data structures, and response codes.

Before returning the "202 Accepted" response, the NFVO shall update the "isCancelPending" and "cancelMode" attributes in the representation of the parent resource according to the provisions in clause 6.5.2.3.

In case of success of processing the asynchronous request:

- 1) If the request has been processed in "PROCESSING" or "ROLLING_BACK" state, the "operationState" attribute in the representation of the parent resource shall be changed to "FAILED_TEMP".

In both cases, the NFVO shall update the "isCancelPending" and "cancelMode" attributes in the representation of the parent resource according to the provisions in clause 6.5.2.3 to reflect the new status, and the applicable "result" notification according to clause 6.6.2.2 shall be emitted to indicate that the execution of the underlying NS LCM operation occurrence has temporarily failed.

Due to race conditions, the processing of the actual operation that is to be cancelled may eventually still succeed, in which case the "operationState" attribute in the representation of the parent resource shall represent the result of that operation, rather than the result of the cancellation.

Table 6.4.15.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
None supported		

Table 6.4.15.3.1-2: Details of the POST request/response on this resource

Request body	Data type	Cardinality	Description	
	CancelMode	1	The POST request to this resource shall include a CancelMode structure in the payload body to choose between "graceful" and "forceful" cancellation.	
Response body	Data type	Cardinality	Response Codes	Description
	n/a		202 Accepted	<p>Shall be returned when the request has been accepted for processing, but processing has not been completed.</p> <p>The response shall have an empty entity body.</p>
	ProblemDetails	0..1	404 Not Found	<p>Shall be returned upon the following error: The API producer did not find a current representation for the target resource or is not willing to disclose that one exists.</p> <p>The general cause for this error and its handling is specified in clause 6.4 of ETSI GS NFV-SOL 013 [16], including rules for the presence of the response body.</p> <p>Specifically, in case of this task resource, the response code 404 shall also be returned if the task is not supported for the NS LCM operation occurrence represented by the parent resource, which means that the task resource consequently does not exist.</p> <p>In this case, the response body shall be present, and shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</p>
	ProblemDetails	1	409 Conflict	<p>Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the NS LCM operation occurrence resource.</p> <p>Typically, this is due to the fact that the operation occurrence is not in STARTING, PROCESSING or ROLLING_BACK state.</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</p>
ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.	

6.4.15.3.2 GET

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

6.4.15.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

6.4.15.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

6.4.15.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

6.4.16 Resource: Subscriptions

6.4.16.1 Description

This resource represents subscriptions. The API consumer can use this resource to subscribe to notifications related to NS lifecycle management, and to query its subscriptions.

6.4.16.2 Resource definition

The resource URI is:

{apiRoot}/nslcm/{apiMajorVersion}/subscriptions

This resource shall support the resource URI variables defined in Table 6.4.16.2-1.

Table 6.4.16.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 6.1a.

6.4.16.3 Resource methods

6.4.16.3.1 POST

The POST method creates a new subscription.

This method shall follow the provisions specified in the Tables 6.4.16.3.1-1 and 6.4.16.3.1-2 for URI query parameters, request and response data structures, and response codes.

As the result of successfully executing this method, a new "Individual subscription" resource as defined in clause 6.4.17 shall have been created. This method shall not trigger any notification.

Creation of two subscription resources with the same callback URI and the same filter can result in performance degradation and will provide duplicates of notifications to the OSS, and might make sense only in very rare use cases. Consequently, the NFVO may either allow creating a subscription resource if another subscription resource with the same filter and callback URI already exists (in which case it shall return the "201 Created" response code), or may decide to not create a duplicate subscription resource (in which case it shall return a "303 See Other" response code referencing the existing subscription resource with the same filter and callback URI).

Table 6.4.16.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Remarks
n/a		

Table 6.4.16.3.1-2: Details of the POST request/response on this resource

Request body	Data type	Cardinality	Remarks	
	LccnSubscriptionRequest	1	Details of the subscription to be created, as defined in clause 6.5.2.2.	
Response body	Data type	Cardinality	Response Codes	Remarks
	LccnSubscription	1	201 Created	<p>Shall be returned when the subscription has been created successfully.</p> <p>The response body shall contain a representation of the created "Individual subscription" resource.</p> <p>The HTTP response shall include a "Location:" HTTP header that points to the created "Individual subscription" resource.</p>
	n/a		303 See Other	<p>Shall be returned if a subscription with the same callback URI and the same filter already exists and the policy of the NFVO is to not create redundant subscriptions.</p> <p>The HTTP response shall include a "Location" HTTP header that contains the resource URI of the existing "Individual subscription" resource.</p> <p>The response body shall be empty.</p>
	ProblemDetails	1	422 Unprocessable Entity	<p>Shall be returned upon the following error: The content type of the payload body is supported and the payload body of a request contains syntactically correct data but the data cannot be processed.</p> <p>The general cause for this error and its handling is specified in clause 6.4 of ETSI GS NFV-SOL 013 [16], including rules for the presence of the response body.</p> <p>Specifically in case of this resource, the response code 422 shall also be returned if the NFVO has tested the Notification endpoint as described in clause 6.4.18.3.2 and the test has failed.</p> <p>In this case, the "detail" attribute in the "ProblemDetails" structure shall convey more information about the error.</p>
ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.	

6.4.16.3.2 GET

The GET method queries the list of active subscriptions of the functional block that invokes the method. It can be used e.g. for resynchronization after error situations.

This method shall follow the provisions specified in the Tables 6.4.16.3.2-1 and 6.4.16.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 6.4.16.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Remarks
filter	0..1	Attribute-based filtering expression according to clause 5.2 of ETSI GS NFV-SOL 013 [16]. The NFVO shall support receiving this parameter as part of the URI query string. The OSS/BSS may supply this parameter. All attribute names that appear in the LccnSubscription and in data types referenced from it shall be supported by the NFVO in the filter expression.
nextpage_opaque_marker	0..1	Marker to obtain the next page of a paged response. Shall be supported by the NFVO if the NFVO supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource.

Table 6.4.16.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Remarks	
	n/a			
Response body	Data type	Cardinality	Response Codes	Remarks
	LccnSubscription	0..N	200 OK	Shall be returned when the list of subscriptions has been queried successfully. The response body shall contain in an array the representations of all active subscriptions of the functional block that invokes the method, i.e. zero or more representations of lifecycle change notification subscriptions as defined in clause 6.5.2.4. If the "filter" URI parameter was supplied in the request, the data in the response body shall have been transformed according to the rules specified in clause 5.2.2 of ETSI GS NFV-SOL 013 [16]. If the NFVO supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource, inclusion of the Link HTTP header in this response shall follow the provisions in clause 5.4.2.3 of ETSI GS NFV-SOL 013 [16].
	ProblemDetails	1	400 Bad Request	Shall be returned upon the following error: Invalid attribute-based filtering expression. The response body shall contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.
	ProblemDetails	1	400 Bad Request	Shall be returned upon the following error: Response too big. If the NFVO supports alternative 1 (error) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource, this error response shall follow the provisions in clause 5.4.2.2 of ETSI GS NFV-SOL 013 [16].
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

6.4.16.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

6.4.16.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

6.4.16.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

6.4.17 Resource: Individual subscription

6.4.17.1 Description

This resource represents an individual subscription. The API consumer can use this resource to read and to terminate a subscription to notifications related to NS lifecycle management.

6.4.17.2 Resource definition

The resource URI is:

{apiRoot}/nslcm/{apiMajorVersion}/subscriptions/{subscriptionId}

This resource shall support the resource URI variables defined in Table 6.4.17.2-1.

Table 6.4.17.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 6.1a.
subscriptionId	Identifier of this subscription

6.4.17.3 Resource methods

6.4.17.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

6.4.17.3.2 GET

The GET method retrieves information about a subscription by reading an "Individual subscription" resource.

This method shall follow the provisions specified in the Tables 6.4.17.3.2-1 and 6.4.17.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 6.4.17.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Remarks
n/a		

Table 6.4.17.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Remarks	
n/a				
Response body	Data type	Cardinality	Response Codes	Remarks
	LccnSubscription	1	200 OK	Shall be returned when information about an individual subscription has been read successfully. The response body shall contain a representation of the "Individual subscription" resource.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

6.4.17.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

6.4.17.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

6.4.17.3.5 DELETE

The DELETE method terminates an individual subscription.

This method shall follow the provisions specified in the Tables 6.4.17.3.5-1 and 6.4.17.3.5-2 for URI query parameters, request and response data structures, and response codes.

As the result of successfully executing this method, the "Individual subscription" resource shall not exist any longer. This means that no notifications for that subscription shall be sent to the formerly-subscribed API consumer.

NOTE: Due to race conditions, some notifications might still be received by the formerly-subscribed API consumer for a certain time period after the deletion.

Table 6.4.17.3.5-1: URI query parameters supported by the DELETE method on this resource

Name	Cardinality	Remarks
n/a		

Table 6.4.17.3.5-2: Details of the DELETE request/response on this resource

Request body	Data type	Cardinality	Remarks	
n/a				
Response body	Data type	Cardinality	Response Codes	Remarks
	n/a		204 No Content	Shall be returned when the "Individual subscription" resource has been deleted successfully. The response body shall be empty.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

6.4.18 Resource: Notification endpoint

6.4.18.1 Description

This resource represents a notification endpoint. The API producer can use this resource to send notifications related to NS lifecycle changes to a subscribed API consumer, which has provided the URI of this resource during the subscription process.

6.4.18.2 Resource definition

The resource URI is provided by the API consumer when creating the subscription.

This resource shall support the resource URI variables defined in Table 6.4.18.2-1.

Table 6.4.18.2-1: Resource URI variables for this resource

Name	Definition
n/a	

6.4.18.3 Resource methods

6.4.18.3.1 POST

The POST method delivers a notification from the API producer to an API consumer. The API consumer shall have previously created an "Individual subscription" resource with a matching filter.

This method shall follow the provisions specified in the Tables 6.4.18.3.1-1 and 6.4.18.3.1-2 for URI query parameters, request and response data structures, and response codes.

Table 6.4.18.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Remarks
n/a		

Each notification request body shall include exactly one of the alternatives defined in Table 6.4.18.3.1-2.

Table 6.4.18.3.1-2: Details of the POST request/response on this resource

Request body	Data type	Cardinality	Remarks	
	NsLcmOperationOccurrenceNotification	1	A notification about lifecycle changes triggered by an NS LCM operation occurrence.	
	NsIdentifierCreationNotification	1	A notification about the creation of an NS identifier and the related NS instance resource.	
	NsIdentifierDeletionNotification	1	A notification about the deletion of an NS identifier and the related NS instance resource.	
Response body	Data type	Cardinality	Response Codes	Remarks
	n/a		204 No Content	Shall be returned when the notification has been delivered successfully.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

6.4.18.3.2 GET

The GET method allows the API producer to test the notification endpoint that is provided by the API consumer, e.g. during subscription.

This method shall follow the provisions specified in the Tables 6.4.18.3.2-1 and 6.4.18.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 6.4.18.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
none supported		

Table 6.4.18.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	n/a		204 No Content	Shall be returned to indicate that the notification endpoint has been tested successfully. The response body shall be empty.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

6.4.18.3.3 PUT

This method is not supported. When this method is requested on this resource, the OSS/BSS shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

6.4.18.3.4 PATCH

This method is not supported. When this method is requested on this resource, the OSS/BSS shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

6.4.18.3.5 DELETE

This method is not supported. When this method is requested on this resource, the OSS/BSS shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

6.5 Data model

6.5.1 Introduction

This clause defines the request and response data structures of the NS Lifecycle management interface.

6.5.2 Resource and notification data types

6.5.2.1 Introduction

This clause defines the data structures to be used in resource representations and notifications.

6.5.2.2 Type: LccnSubscriptionRequest

This type represents a subscription request related to notifications about NS lifecycle changes. It shall comply with the provisions defined in Table 6.5.2.2-1.

Table 6.5.2.2-1: Definition of the LccnSubscriptionRequest data type

Attribute name	Data type	Cardinality	Description
filter	LifecycleChangeNotificationsFilter	0..1	Filter settings for this subscription, to define the subset of all notifications this subscription relates to. A particular notification is sent to the subscriber if the filter matches, or if there is no filter.
callbackUri	Uri	1	The URI of the endpoint to send the notification to.
authentication	SubscriptionAuthentication	0..1	Authentication parameters to configure the use of Authorization when sending notifications corresponding to this subscription, as defined in clause 8.3.4 of ETSI GS NFV-SOL 013 [16]. This attribute shall only be present if the subscriber requires authorization of notifications.

6.5.2.3 Type: NsLcmOpOcc

This type represents an NS lifecycle management operation occurrence. It shall comply with the provisions defined in Table 6.5.2.3-1.

Table 6.5.2.3-1: Definition of the NsLcmOpOcc data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this NS lifecycle operation occurrence.
operationState	NsLcmOperationStateType	1	The state of the NS LCM operation.
statusEnteredTime	DateTime	1	Date-time when the current state has been entered.
nsInstancelid	Identifier	1	Identifier of the NS instance to which the operation applies.
lcmOperationType	NsLcmOpType	1	Type of the actual LCM operation represented by this lcm operation occurrence.
startTime	DateTime	1	Date-time of the start of the operation.
isAutomaticInvocation	Boolean	1	Set to true if this NS LCM operation occurrence has been automatically triggered by the NFVO. This occurs in the case of auto-scaling, auto-healing and when a nested NS is modified as a result of an operation on its composite NS. Set to false otherwise.
operationParams	Object	0..1	Input parameters of the LCM operation. This attribute shall be formatted according to the request data type of the related LCM operation. The following mapping between lcmOperationType and the data type of this attribute shall apply: <ul style="list-style-type: none"> • INSTANTIATE: InstantiateNsRequest • SCALE: ScaleNsRequest • UPDATE: UpdateNsRequest • HEAL: HealNsRequest • TERMINATE: TerminateNsRequest This attribute shall be present if this data type is returned in a response to reading an individual resource, and may be present according to the chosen attribute selector parameter if this data type is returned in a response to a query of a container resource.

Attribute name	Data type	Cardinality	Description
isCancelPending	Boolean	1	If the LCM operation occurrence is in "PROCESSING" or "ROLLING_BACK" state and the operation is being cancelled, this attribute shall be set to true. Otherwise, it shall be set to false.
cancelMode	CancelModeType	0..1	The mode of an ongoing cancellation. Shall be present when isCancelPending=true, and shall be absent otherwise.
error	ProblemDetails	0..1	If "operationState" is "FAILED_TEMP" or "FAILED" or "operationState" is "PROCESSING" or "ROLLING_BACK" and previous value of "operationState" was "FAILED_TEMP", this attribute shall be present and contain error information, unless it has been requested to be excluded via an attribute selector.
resourceChanges	Structure (inlined)	0..1	This attribute contains information about the cumulative changes to virtualised resources that were performed so far by the LCM operation since its start, if applicable.
>affectedVnfs	AffectedVnf	0..N	Information about the VNF instances that were affected during the lifecycle operation, if this notification represents the result of a lifecycle operation. See note.
>affectedPnfs	AffectedPnf	0..N	Information about the PNF instances that were affected during the lifecycle operation, if this notification represents the result of a lifecycle operation. See note.
>affectedVls	AffectedVirtualLink	0..N	Information about the VL instances that were affected during the lifecycle operation, if this notification represents the result of a lifecycle operation. See note.
>affectedVnffgs	AffectedVnffg	0..N	Information about the VNFFG instances that were affected during the lifecycle operation, if this notification represents the result of a lifecycle operation. See note.
>affectedNss	AffectedNs	0..N	Information about the nested NS instances that were affected during the lifecycle operation, if this notification represents the result of a lifecycle operation. See note.
>affectedSaps	AffectedSap	0..N	Information about the SAP instances that were affected during the lifecycle operation, if this notification represents the result of a lifecycle operation. See note.
_links	Structure (inlined)	1	Links to resources related to this resource.
>self	Link	1	URI of this resource.
>nsInstance	Link	1	Link to the NS instance that the operation applies to.
>cancel	Link	0..1	Link to the task resource that represents the "cancel" operation for this LCM operation occurrence, if cancelling is currently allowed.
>retry	Link	0..1	Link to the task resource that represents the "retry" operation for this LCM operation occurrence, if retrying is currently allowed.
>rollback	Link	0..1	Link to the task resource that represents the "rollback" operation for this LCM operation occurrence, if rolling back is currently allowed.
>continue	Link	0..1	Link to the task resource that represents the "continue" operation for this LCM operation occurrence, if continuing is currently allowed.
>fail	Link	0..1	Link to the task resource that represents the "fail" operation for this LCM operation occurrence, if declaring as failed is currently allowed.
NOTE:	This allows the OSS/BSS to obtain a copy of the latest "result" notification if it has not received it due to an error. If the notification represents the successful result of a lifecycle operation, at least an affectedVnf, or affectedPnf, or affectedVI, or affectedVnffg or affectedNs, or affectedSap shall be present.		

6.5.2.4 Type: LccnSubscription

This type represents a subscription related to notifications about NS lifecycle changes. It shall comply with the provisions defined in Table 6.5.2.4-1.

Table 6.5.2.4-1: Definition of the LccnSubscription data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this subscription resource.
filter	LifecycleChangeNotificationsFilter	0..1	Filter settings for this subscription, to define the subset of all notifications this subscription relates to. A particular notification is sent to the subscriber if the filter matches, or if there is no filter.
callbackUri	Uri	1	The URI of the endpoint to send the notification to.
_links	Structure (inlined)	1	Links to resources related to this resource.
>self	Link	1	URI of this resource.

6.5.2.5 Type: NsLcmOperationOccurrenceNotification

This type represents an NS lifecycle management operation occurrence notification, which informs the receiver of changes in the NS lifecycle caused by an NS LCM operation occurrence. The NS LCM operation occurrence may be triggered by the OSS/BSS or automatically triggered by the NFVO. The automatic trigger occurs in case of auto-scaling, auto-healing and when a nested NS is modified as a result of an operation on its composite NS.

It shall comply with the provisions defined in Table 6.5.2.5-1. The support of the notification is mandatory.

This notification shall be triggered by the NFVO when there is a change in the NS lifecycle caused by an NS LCM operation occurrence that changes the NS lifecycle, including:

- Instantiation of the NS (start and result)
- Scaling of the NS (start and result, including the auto-scaling)
- Update of the NS (start and result)
- Termination of the NS (start and result)
- Healing of the NS (start and result, including the auto-healing)
- When a nested NS is modified as a result of an operation on its composite NS

Clause 6.6.2 defines the states and state transition of an NS LCM operation occurrence, and also specifies details of the notifications to be emitted at each state transition.

If this is the initial notification about the start of an NS LCM operation occurrence, it is assumed that the notification is sent by the NFVO before any action is taken as part of the LCM operation. Due to possible race conditions, the "start" notification and the LCM operation acknowledgment (i.e. the "202 Accepted" response) can arrive in any order at the OSS/BSS, and the OSS/BSS shall be able to handle such a situation.

If this is a notification about a final or intermediate result state of an NS LCM operation occurrence, the notification shall be sent after all related actions of the LCM operation that led to this state have been executed.

The new state shall be set in the "Individual NS LCM operation occurrence" resource before the notification about the state change is sent.

See clause 6.6.2.2 for further provisions regarding sending this notification, including in cases of handling LCM operation errors.

Table 6.5.2.5-1: Definition of the NsLcmOperationOccurrenceNotification data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this notification. If a notification is sent multiple times due to multiple subscriptions, the "id" attribute of all these notifications shall have the same value.
nsInstancelid	Identifier	1	The identifier of the NS instance affected.
nsLcmOpOcclid	Identifier	1	The identifier of the NS lifecycle operation occurrence associated to the notification.
operation	LcmOpType	1	The lifecycle operation.
notificationType	String	1	Discriminator for the different notification types. Shall be set to "NsLcmOperationOccurrenceNotification" for this notification type.
subscriptionId	Identifier	1	Identifier of the subscription that this notification relates to.
timestamp	DateTime	1	Date-time of the generation of the notification.
notificationStatus	Enum (inlined)	1	Indicates whether this notification reports about the start of an NS lifecycle operation or the result of an NS lifecycle operation. Permitted values: - START: Informs about the start of the NS LCM operation occurrence. - RESULT: Informs about the final or intermediate result of the NS LCM operation occurrence.
operationState	NsLcmOperationStateType	1	The state of the NS lifecycle operation occurrence.
isAutomaticInvocation	Boolean	1	Set to true if this NS LCM operation occurrence has been automatically triggered by the NFVO. This occurs in case of auto-scaling, auto-healing and when a nested NS is modified as a result of an operation on its composite NS. Set to false otherwise.
affectedVnf	AffectedVnf	0..N	Information about the VNF instances that were affected during the lifecycle operation. See note.
affectedPnf	AffectedPnf	0..N	Information about the PNF instances that were affected during the lifecycle operation. See note.
affectedVI	AffectedVirtualLink	0..N	Information about the VL instances that were affected during the lifecycle operation. See note.
affectedVnffg	AffectedVnffg	0..N	Information about the VNFFG instances that were affected during the lifecycle operation. See note.
affectedNs	AffectedNs	0..N	Information about the NS instances that were affected during the lifecycle operation. See note.
affectedSap	AffectedSap	0..N	Information about the SAP instances that were affected during the lifecycle operation. See note.
error	ProblemDetails	0..1	Details of the latest error, if one has occurred during executing the LCM operation (see clause 6.3 of ETSI GS NFV-SOL 013 [16]). Shall be present if operationState is "FAILED_TEMP" or "FAILED", and shall be absent otherwise.
links	LccnLinks	1	Links to resources related to this notification.
NOTE: Shall be present if the "notificationStatus" is set to "RESULT" and the operation has performed any resource modification. Shall be absent otherwise.			

6.5.2.6 Type: NsIdentifierCreationNotification

This type represents an NS identifier creation notification, which informs the receiver of the creation of a new "Individual NS instance" resource and the associated NS instance identifier. It shall comply with the provisions defined in Table 6.5.2.6-1. The support of the notification is mandatory.

This notification shall be triggered by the NFVO when it has created an "Individual NS instance" resource and the associated NS instance identifier.

Table 6.5.2.6-1: Definition of the NsIdentifierCreationNotification data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this notification. If a notification is sent multiple times due to multiple subscriptions, the "id" attribute of all these notifications shall have the same value.
notificationType	String	1	Discriminator for the different notification types. Shall be set to "NsIdentifierCreationNotification" for this notification type.
subscriptionId	Identifier	1	Identifier of the subscription that this notification relates to.
timestamp	DateTime	1	Date-time of the generation of the notification.
nsInstanceid	Identifier	1	The created NS instance identifier. Shall be set to the same "id" attribute value of the associated "NsInstance" representation of the "Individual NS instance" resource.
_links	LccnLinks	1	Links to resources related to this notification.

6.5.2.7 Type: NsIdentifierDeletionNotification

This type represents an NS identifier deletion notification, which informs the receiver of the deletion of a new "Individual NS instance" resource and the associated NS instance identifier. It shall comply with the provisions defined in Table 6.5.2.7-1. The support of the notification is mandatory.

This notification shall be triggered by the NFVO when it has deleted an "Individual NS instance" resource and the associated NS instance identifier.

Table 6.5.2.7-1: Definition of the NsInstanceDeletionNotification data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this notification. If a notification is sent multiple times due to multiple subscriptions, the "id" attribute of all these notifications shall have the same value.
notificationType	String	1	Discriminator for the different notification types. Shall be set to "NsIdentifierDeletionNotification" for this notification type.
subscriptionId	Identifier	1	Identifier of the subscription that this notification relates to.
timeStamp	DateTime	1	Date-time of the generation of the notification.
nsInstanceid	Identifier	1	The deleted NS instance identifier. Shall be set to the same "id" attribute value of the associated "NsInstance" representation of the "Individual NS instance" resource.
_links	LccnLinks	1	Links to resources related to this notification.

6.5.2.8 Type: NsChangeNotification

This type represents an NS change notification, which informs the receiver of changes on the NS instance caused by the LCM operation occurrence, which directly or indirectly impacts its NS component and is triggered without any context of this NS instance.

This notification is different from the NsLcmOperationOccurrenceNotification (see clause 6.5.2.5), which is triggered by the NS LCM operation occurrence on the NS instance itself. It shall comply with the provisions defined in Table 6.5.2.8-1. The support of the notification is mandatory.

The trigger conditions include:

- 1) NS LCM operation occurrence which directly or indirectly impacts the NS component (start and result).

If this is a notification about the start of an LCM operation occurrence impacting the NS component, the notification shall be sent as soon as the impact on the NS component is identified.

If this is a notification about a final result state of an NS LCM operation occurrence impacting the NS component, the notification shall be sent after the impact on the NS component has been executed.

Table 6.5.2.8-1: Definition of the NsChangeNotification data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this notification. If a notification is sent multiple times due to multiple subscriptions, the "id" attribute of all these notifications shall have the same value.
nsInstanceId	Identifier	1	The identifier of the NS instance affected.
nsComponentType	NsComponentType	1	Indicates the impacted NS component type.
nsComponentId	Identifier	1	The identifier of the impacted NS component.
lcmOpOccIdImpactingNsComponent	Identifier	1	The identifier of the lifecycle operation occurrence which is associated to the notification and impacts the NS component directly or indirectly.
lcmOpNameImpactingNsComponent	LcmOpNameForChangeNotificationType	1	Indicates the name of the lifecycle operation occurrence which is associated to the notification and impacts the NS component directly or indirectly.
lcmOpOccStatusImpactingNsComponent	LcmOpOccStatusForChangeNotificationType	1	Indicates this status of the lifecycle operation occurrence which is associated to the notification and impacts the NS component directly or indirectly.
notificationType	String	1	Discriminator for the different notification types. Shall be set to "NsChangeNotification" for this notification type.
subscriptionId	Identifier	1	Identifier of the subscription that this notification relates to.
timeStamp	DateTime	1	Date-time of the generation of the notification.
error	ProblemDetails	0..1	Details of the latest error, if one has occurred during executing the LCM operation (see clause 6.3 of ETSI GS NFV-SOL 013 [16]). Shall be present if lcmOpOccStatusImpactingNsComponent is "PARTIALLY_COMPLETED" or "FAILED", and shall be absent otherwise.
_links	LccnLinks	1	Links to resources related to this notification.

6.5.2.9 Type: CreateNsRequest

This type represents request parameters for the "Create NS identifier" operation. It shall comply with the provisions defined in Table 6.5.2.9-1.

Table 6.5.2.9-1: Definition of the CreateNsRequest data type

Attribute name	Data type	Cardinality	Description
nsId	Identifier	1	Identifier of the NSD that defines the NS instance to be created.
nsName	String	1	Human-readable name of the NS instance to be created.
nsDescription	String	1	Human-readable description of the NS instance to be created.

6.5.2.10 Type: NsInstance

This type represents an NS instance. It shall comply with the provisions defined in Table 6.5.2.10-1.

Table 6.5.2.10-1: Definition of the NsInstance data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of the NS instance.
nsInstanceName	String	1	Human readable name of the NS instance.
nsInstanceDescription	String	1	Human readable description of the NS instance.
nsdId	Identifier	1	Identifier of the NSD on which the NS instance is based. The value is copied from the NSD.
nsdInfold	Identifier	1	Identifier of the NSD information object on which the NS instance is based. This identifier has been allocated by the NFVO.
flavourId	IdentifierInNsd	0..1	Identifier of the NS deployment flavour applied to the NS instance. This attribute shall be present if the nsState attribute value is INSTANTIATED.
vnfInstance	VnfInstance	0..N	Information on constituent VNF(s) of the NS instance. See note.
pnfInfo	PnfInfo	0..N	Information on the PNF(s) that are part of the NS instance.
virtualLinkInfo	NsVirtualLinkInfo	0..N	Information on the VL(s) of the NS instance. This attribute shall be present if the nsState attribute value is INSTANTIATED and if the NS instance has specified connectivity.
vnffgInfo	VnffgInfo	0..N	Information on the VNFFG(s) of the NS instance.
sapInfo	SapInfo	0..N	Information on the SAP(s) of the NS instance.
nestedNsInstanceCid	Identifier	0..N	Identifier of the nested NS(s) of the NS instance. See note.
nsState	Enum (inlined)	1	The state of the NS instance. Permitted values: NOT_INSTANTIATED: The NS instance is terminated or not instantiated. INSTANTIATED: The NS instance is instantiated.
monitoringParameter	NsMonitoringParameter	0..N	Performance metrics tracked by the NFVO (e.g. for auto-scaling purposes) as identified by the NS designer in the NSD.
nsScaleStatus	NsScaleInfo	0..N	Status of each NS scaling aspect declared in the applicable DF, how "big" the NS instance has been scaled w.r.t. that aspect. This attribute shall be present if the nsState attribute value is INSTANTIATED.
additionalAffinityOrAntiAffinityRule	AffinityOrAntiAffinityRule	0..N	Information on the additional affinity or anti-affinity rule from NS instantiation operation. Shall not conflict with rules already specified in the NSD.
_links	Structure (inlined)	1	Links to resources related to this resource.
>self	Link	1	URI of this resource.
>nestedNsInstances	Link	0..N	Links to the nested NS instances of the present NS instance.
>instantiate	Link	0..1	Link to the "instantiate" task resource, if the related operation is possible based on the current status of this NS instance resource (i.e. NS instance in NOT_INSTANTIATED state).

Attribute name	Data type	Cardinality	Description
>terminate	Link	0..1	Link to the "terminate" task resource, if the related operation is possible based on the current status of this NS instance resource (i.e. NS instance is in INSTANTIATED state).
>update	Link	0..1	Link to the "update" task resource, if the related operation is possible based on the current status of this NS instance resource (i.e. NS instance is in INSTANTIATED state).
>scale	Link	0..1	Link to the "scale" task resource, if the related operation is supported for this NS instance, and is possible based on the current status of this NS instance resource (i.e. NS instance is in INSTANTIATED state).
>heal	Link	0..1	Link to the "heal" task resource, if the related operation is supported for this NS instance, and is possible based on the current status of this NS instance resource (i.e. NS instance is in INSTANTIATED state).
NOTE: If the "nsState" attribute is INSTANTIATED, at least either one "vnfInstance" attribute or one "nestedNsInstanceId" attribute shall be present.			

6.5.2.11 Type: InstantiateNsRequest

This type represents request parameters for the "Instantiate NS" operation. It shall comply with the provisions defined in Table 6.5.2.11-1.

Table 6.5.2.11-1: Definition of the InstantiateNsRequest data type

Attribute name	Data type	Cardinality	Description
nsFlavourId	IdentifierInNsd	1	Identifier of the NS deployment flavour to be instantiated.
sapData	SapData	0..N	Create data concerning the SAPs of this NS.
addpnfData	AddPnfData	0..N	Information on the PNF(s) that are part of this NS.
vnfInstanceData	VnfInstanceData	0..N	Specify an existing VNF instance to be used in the NS. If needed, the VNF Profile to be used for this VNF instance is also provided. See note 1.
nestedNsInstanceData	NestedNsInstanceData	0..N	Specify an existing NS instance to be used as a nested NS within the NS. If needed, the NS Profile to be used for this nested NS instance is also provided. See note 2 and note 3.
locationConstraints	VnfLocationConstraint	0..N	Defines the location constraints for the VNF to be instantiated as part of the NS instantiation. An example can be a constraint for the VNF to be in a specific geographic location.
additionalParamsForNs	KeyValuePairs	0..1	Allows the OSS/BSS to provide additional parameter(s) at the composite NS level (as opposed to the VNF level, which is covered in additionalParamsForVnf), and as opposed to the nested NS level, which is covered in additionalParamForNestedNs.
additionalParamForNestedNs	ParamsForNestedNs	0..N	Allows the OSS/BSS to provide additional parameter(s) per nested NS instance (as opposed to the composite NS level, which is covered in additionalParamForNs, and as opposed to the VNF level, which is covered in additionalParamForVnf). This is for nested NS instances that are to be created by the NFVO as part of the NS instantiation and not for existing nested NS instances that are referenced for reuse.

Attribute name	Data type	Cardinality	Description
additionalParamsForVnf	ParamsForVnf	0..N	Allows the OSS/BSS to provide additional parameter(s) per VNF instance (as opposed to the composite NS level, which is covered in additionalParamsForNs), and as opposed to the nested NS level, which is covered in additionalParamForNestedNs). This is for VNFs that are to be created by the NFVO as part of the NS instantiation and not for existing VNF that are referenced for reuse.
startTime	DateTime	0..1	Timestamp indicating the earliest time to instantiate the NS. Cardinality "0" indicates the NS instantiation takes place immediately.
nsInstantiationLevelId	IdentifierInNsd	0..1	Identifies one of the NS instantiation levels declared in the DF applicable to this NS instance. If not present, the default NS instantiation level as declared in the NSD shall be used.
additionalAffinityOrAntiAffinityRule	AffinityOrAntiAffinityRule	0..N	Specifies additional affinity or anti-affinity constraint for the VNF instances to be instantiated as part of the NS instantiation. Shall not conflict with rules already specified in the NSD.
NOTE 1: The DF of the VNF instance shall match the VNF DF present in the associated VNF Profile.			
NOTE 2: The NS DF of each nested NS shall be one of the allowed flavours in the associated NSD (as referenced in the nestedNsd attribute of the NSD of the NS to be instantiated).			
NOTE 3: The NSD of each referenced NSs (i.e. each nestedInstanceId) shall match the one of the nested NSD in the composite NSD.			

6.5.2.12 Type: UpdateNsRequest

This type represents request parameters for the "Update NS" operation. It shall comply with the provisions defined in Table 6.5.2.12-1.

Table 6.5.2.12-1: Definition of the UpdateNsRequest data type

Attribute name	Data type	Cardinality	Description
updateType	Enum (inlined)	1	The type of update. It determines also which one of the following parameters is present in the operation. Possible values include: <ul style="list-style-type: none"> - ADD_VNF: Adding existing VNF instance(s) - REMOVE_VNF: Removing VNF instance(s) - INSTANTIATE_VNF: Instantiating new VNF(s) - CHANGE_VNF_DF: Changing VNF DF - OPERATE_VNF: Changing VNF state - MODIFY_VNF_INFORMATION: Modifying VNF information and/or the configurable properties of VNF instance(s) - CHANGE_EXTERNAL_VNF_CONNECTIVITY: Changing the external connectivity of VNF instance(s) - ADD_SAP: Adding SAP(s) - REMOVE_SAP: Removing SAP(s) - ADD_NESTED_NS: Adding existing NS instance(s) as nested NS(s) - REMOVE_NESTED_NS: Removing existing nested NS instance(s) - ASSOC_NEW_NSD_VERSION: Associating a new NSD version to the NS instance - MOVE_VNF: Moving VNF instance(s) from one origin NS instance to another target NS instance - ADD_VNFFG: Adding VNFFG(s) - REMOVE_VNFFG: Removing VNFFG(s) - UPDATE_VNFFG: Updating VNFFG(s) - CHANGE_NS_DF: Changing NS DF - ADD_PNF: Adding PNF - MODIFY_PNF: Modifying PNF - REMOVE_PNF: Removing PNF

Attribute name	Data type	Cardinality	Description
addVnfInstance	VnfInstanceData	0..N	Identifies an existing VNF instance to be added to the NS instance. It shall be present only if updateType = "ADD_VNF".
removeVnfInstanceId	Identifier	0..N	Identifies an existing VNF instance to be removed from the NS instance. It contains the identifier(s) of the VNF instances to be removed. It shall be present only if updateType = "REMOVE_VNF". See note.
instantiateVnfData	InstantiateVnfData	0..N	Identifies the new VNF to be instantiated. It can be used e.g. for the bottom-up NS creation. It shall be present only if updateType = "INSTANTIATE_VNF".
changeVnfFlavourData	ChangeVnfFlavourData	0..N	Identifies the new DF of the VNF instance to be changed to. It shall be present only if updateType = "CHANGE_VNF_DF".
operateVnfData	OperateVnfData	0..N	Identifies the state of the VNF instance to be changed. It shall be present only if updateType = "OPERATE_VNF".
modifyVnfInfoData	ModifyVnfInfoData	0..N	Identifies the VNF information parameters and/or the configurable properties of VNF instance to be modified. It shall be present only if updateType = "MODIFY_VNF_INFORMATION".
changeExtVnfConnectivityData	ChangeExtVnfConnectivityData	0..N	Specifies the new external connectivity data of the VNF instance to be changed. It shall be present only if updateType = "CHANGE_EXTERNAL_VNF_CONNECTIVITY".
addSap	SapData	0..N	Identifies a new SAP to be added to the NS instance. It shall be present only if updateType = "ADD_SAP".
removeSapId	Identifier	0..N	The identifier an existing SAP to be removed from the NS instance. It shall be present only if updateType = "REMOVE_SAP".
addNestedNsData	NestedNsInstanceData	0..N	The identifier of an existing nested NS instance to be added to (nested within) the NS instance. It shall be present only if updateType = "ADD_NESTED_NS".
removeNestedNsId	IdentifierInNs	0..N	The identifier of an existing nested NS instance to be removed from the NS instance. It shall be present only if updateType = "REMOVE_NESTED_NS".
assocNewNsdVersionData	AssocNewNsdVersionData	0..1	Specify the new NSD to be used for the NS instance. It shall be present only if updateType = "ASSOC_NEW_NSD_VERSION".
moveVnfInstanceData	MoveVnfInstanceData	0..N	Specify existing VNF instance to be moved from one NS instance to another NS instance. It shall be present only if updateType = "MOVE_VNF".
addVnffg	AddVnffgData	0..N	Specify the new VNFFG to be created to the NS Instance. It shall be present only if updateType = "ADD_VNFFG".
removeVnffgId	Identifier	0..N	Identifier of an existing VNFFG to be removed from the NS Instance. It shall be present only if updateType = "REMOVE_VNFFG".
updateVnffg	UpdateVnffgData	0..N	Specify the new VNFFG Information data to be updated for a VNFFG of the NS Instance. It shall be present only if updateType = "UPDATE_VNFFG".
changeNsFlavourData	ChangeNsFlavourData	0..1	Specifies the new DF to be applied to the NS instance. It shall be present only if updateType = "CHANGE_NS_DF".
addPnfData	AddPnfData	0..N	Specifies the PNF to be added into the NS instance. It shall be present only if updateType = "ADD_PNF".
modifyPnfData	ModifyPnfData	0..N	Specifies the PNF to be modified in the NS instance. It shall be present only if updateType = "MODIFY_PNF".
removePnfId	Identifier	0..N	Identifier of the PNF to be deleted from the NS instance. It shall be present only if updateType = "REMOVE_PNF".
updateTime	DateTime	0..1	Timestamp indicating the update time of the NS, i.e. the NS will be updated at this timestamp. Cardinality "0" indicates the NS update takes place immediately.
NOTE: If a VNF instance is removed from an NS and this NS was the last one for which this VNF instance was a part, the VNF instance is terminated by the NFVO.			

6.5.2.13 Type: HealNsRequest

This type represents request parameters for the "Heal NS" operation. This operation supports the healing of an NS instance, either by healing the complete NS instance or by healing one of more of the VNF instances that are part of this NS. It shall comply with the provisions defined in Table 6.5.2.13-1.

Table 6.5.2.13-1: Definition of the HealNsRequest data type

Attribute name	Data type	Cardinality	Description
healNsData	HealNsData	0..1	Provides the information needed to heal an NS. See note.
healVnfData	HealVnfData	0..N	Provides the information needed to heal a VNF. See note.
NOTE: Either the parameter healNsData or the parameter healVnfData, but not both shall be provided.			

6.5.2.14 Type: ScaleNsRequest

This type represents request parameters for the "Scale NS" operation. It shall comply with the provisions defined in Table 6.5.2.14-1.

Table 6.5.2.14-1: Definition of the ScaleNsRequest data type

Attribute name	Data type	Cardinality	Description
scaleType	Enum (inlined)	1	Indicates the type of scaling to be performed. Possible values: - SCALE_NS - SCALE_VNF
scaleNsData	ScaleNsData	0..1	The necessary information to scale the referenced NS instance. It shall be present when scaleType = SCALE_NS. See note.
scaleVnfData	ScaleVnfData	0..N	The necessary information to scale the referenced NS instance. It shall be present when scaleType = SCALE_VNF. See note.
scaleTime	DateTime	0..1	Timestamp indicating the scale time of the NS, i.e. the NS will be scaled at this timestamp. Cardinality "0" indicates the NS scaling takes place immediately.
NOTE: Either the parameter scaleNsData or the parameter scaleVnfData, but not both shall be provided.			

6.5.2.15 Type: TerminateNsRequest

This type represents request parameters for the "Terminate NS" operation. It shall comply with the provisions defined in Table 6.5.2.15-1.

Table 6.5.2.15-1: Definition of the TerminateNsRequest data type

Attribute name	Data type	Cardinality	Description
terminationTime	DateTime	0..1	Timestamp indicating the end time of the NS, i.e. the NS will be terminated automatically at this timestamp. Cardinality "0" indicates the NS termination takes place immediately.

6.5.2.16 Type: CancelMode

This type represents a parameter to select the mode of cancelling an ongoing NS LCM operation occurrence. It shall comply with the provisions defined in Table 6.5.2.16-1.

Table 6.5.2.16-1: Definition of the CancelMode data type

Attribute name	Data type	Cardinality	Description
cancelMode	CancelModeType	1	Cancellation mode to apply.

6.5.3 Referenced structured data types

6.5.3.1 Introduction

This clause defines data structures that can be referenced from data structures defined in the previous clauses, but can neither be resource representations nor bound to any subscribe/notify mechanism.

6.5.3.2 Type: AffectedVnf

This type provides information about added, deleted and modified VNFs. It shall comply with the provisions in Table 6.5.3.2-1.

Table 6.5.3.2-1: Definition of the AffectedVnf data type

Attribute name	Data type	Cardinality	Description
vnfInstanceld	Identifier	1	Identifier of the VNF instance.
vnfdId	Identifier	1	Identifier of the VNFD of the VNF Instance.
vnfProfileId	IdentifierInNsd	1	Identifier of the VNF profile of the NSD.
vnfName	String	1	Name of the VNF Instance.
changeType	Enum (inlined)	1	Signals the type of change Permitted values: <ul style="list-style-type: none"> - ADD - REMOVE - INSTANTIATE - TERMINATE - SCALE - CHANGE_FLAVOUR - HEAL - OPERATE - MODIFY_INFORMATION - CHANGE_EXTERNAL_VNF_CONNECTIVITY
changeResult	Enum (inlined)	1	Signals the result of change identified by the "changeType" attribute. Permitted values: <ul style="list-style-type: none"> - COMPLETED - ROLLED_BACK - FAILED
changedInfo	Structure (inlined)	0..1	Information about the changed VNF instance information, including VNF configurable properties, if applicable.
>changedVnfInfo	ModifyVnfInfoData	0..1	Information about the changed VNF instance information, including configurable properties, if applicable. See note.
>changedExtConnectivity	ExtVirtualLinkInfo	0..N	Information about changed external connectivity, if applicable. Only information about external VL instances that have been added or modified shall be provided. See note.
NOTE: When the "changedInfo" attribute is present, either the "changedVnfInfo" attribute or the "changedExtConnectivity" attribute or both shall be present.			

6.5.3.3 Type: AffectedPnf

This type provides information about added, deleted and modified PNFs. It shall comply with the provisions in Table 6.5.3.3-1.

Table 6.5.3.3-1: Definition of the AffectedPnf data type

Attribute name	Data type	Cardinality	Description
pnfId	Identifier	1	Identifier of the affected PNF. This identifier is allocated by the OSS/BSS.
pnfDId	IdentifierInNsd	1	Identifier of the PNFD on which the PNF is based.
pnfProfileId	IdentifierInNsd	1	Identifier of the PNF profile of the NSD.
pnfName	String	1	Name of the PNF.
cpInstanceId	IdentifierInPnf	1..N	Identifier of the CP in the scope of the PNF.
changeType	Enum (inlined)	1	Signals the type of change. Permitted values: - ADD - REMOVE - MODIFY
changeResult	Enum (inlined)	1	Signals the result of change identified by the "changeType" attribute. Permitted values: - COMPLETED - ROLLED_BACK - FAILED

6.5.3.4 Type: AffectedVirtualLink

This type provides information about added, deleted and modified VLs. It shall comply with the provisions in Table 6.5.3.4-1.

Table 6.5.3.4-1: Definition of the AffectedVirtualLink data type

Attribute name	Data type	Cardinality	Description
nsVirtualLinkInstancId	IdentifierInNs	1	Identifier of the VL Instance.
nsVirtualLinkDescId	IdentifierInNsd	1	Identifier of the VLD in the NSD for this VL.
viProfileId	IdentifierInNsd	1	Name of the VL profile.
changeType	Enum (inlined)	1	Signals the type of change. Permitted values: - ADD - DELETE - MODIFY - ADD_LINK_PORT - REMOVE_LINK_PORT
linkPortIds	IdentifierInNs	0..N	Identifiers of the link ports of the affected VL related to the change. Each identifier references an "NsLinkPortInfo" structure. Shall be set when changeType is equal to "ADD_LINK_PORT" or "REMOVE_LINK_PORT", and the related "NsLinkPortInfo" structures are present (case "add") or have been present (case "remove") in the "NsVirtualLinkInfo" structure that is represented by the "virtualLinkInfo" attribute in the "NsInstance" structure. See note.

Attribute name	Data type	Cardinality	Description
changeResult	Enum (inlined)	1	Signals the result of change identified by the "changeType" attribute. Permitted values: - COMPLETED - ROLLED_BACK - FAILED
NOTE: The resource handles of the affected NS link ports can be found by dereferencing the identifiers in the "linkPortIds" attribute.			

6.5.3.5 Type: AffectedVnffg

This type provides information about added, deleted and modified VNFFG instances. It shall comply with the provisions in Table 6.5.3.5-1.

Table 6.5.3.5-1: Definition of the AffectedVnffg data type

Attribute name	Data type	Cardinality	Description
vnffgInstanceld	IdentifierInNs	1	Identifier of the VNFFG instance.
vnffgdId	IdentifierInNsd	1	Identifier of the VNFFGD of the VNFFG instance.
changeType	Enum (inlined)	1	Signals the type of lifecycle change. Permitted values: - ADD - REMOVE - MODIFY
changeResult	Enum (inlined)	1	Signals the result of change identified by the "changeType" attribute. Permitted values: - COMPLETED - ROLLED_BACK - FAILED

6.5.3.6 Type: AffectedNs

This type provides information about added, deleted and modified nested NSs. It shall comply with the provisions in Table 6.5.3.6-1.

Table 6.5.3.6-1: Definition of the AffectedNs data type

Attribute name	Data type	Cardinality	Description
nsInstanceld	Identifier	1	Identifier of the nested NS instance.
nsdId	Identifier	1	Identifier of the NSD of the nested NS instance.
changeType	Enum (inlined)	1	Signals the type of lifecycle change. Permitted values: - ADD - REMOVE - INSTANTIATE - INSTANTIATE - SCALE - UPDATE - HEAL - TERMINATE

Attribute name	Data type	Cardinality	Description
changeResult	Enum (inlined)	1	Signals the result of change identified by the "changeType" attribute. Permitted values: - COMPLETED - ROLLED_BACK - FAILED - PARTIALLY_COMPLETED

6.5.3.7 Type: AffectedSap

This type provides information about added, deleted and modified SAP of an NS. It shall comply with the provisions in Table 6.5.3.7-1.

Table 6.5.3.7-1: Definition of the AffectedSap data type

Attribute name	Data type	Cardinality	Description
sapInstanceld	IdentifierInNs	1	Identifier of the SAP instance.
sapId	IdentifierInNsd	1	Identifier of the SAPD for this SAP.
sapName	String	1	Human readable name for the SAP.
changeType	Enum (inlined)	1	Signals the type of lifecycle change. Permitted values: - ADD - REMOVE - MODIFY
changeResult	Enum (inlined)	1	Signals the result of change identified by the "changeType" attribute. Permitted values: - COMPLETED - ROLLED_BACK - FAILED

6.5.3.8 Type: LifecycleChangeNotificationsFilter

This type represents a subscription filter related to notifications about NS lifecycle changes. It shall comply with the provisions defined in Table 6.5.3.8-1.

At a particular nesting level in the filter structure, the following applies: All attributes shall match in order for the filter to match (logical "and" between different filter attributes). If an attribute is an array, the attribute shall match if at least one of the values in the array matches (logical "or" between the values of one filter attribute).

Table 6.5.3.8-1: Definition of the LifecycleChangeNotificationsFilter data type

Attribute name	Data type	Cardinality	Description
nsInstanceSubscriptionFilter	NsInstanceSubscriptionFilter	0..1	Filter criteria to select NS instances about which to notify.
notificationTypes	Enum (inlined)	0..N	Match particular notification types. Permitted values: - NsLcmOperationOccurrenceNotification - NsIdentifierCreationNotification - NsIdentifierDeletionNotification - NsChangeNotification See note.

Attribute name	Data type	Cardinality	Description
operationTypes	NsLcmOpType	0..N	Match particular NS lifecycle operation types for the notification of type NsLcmOperationOccurrenceNotification. May be present if the "notificationTypes" attribute contains the value "NsLcmOperationOccurrenceNotification", and shall be absent otherwise.
operationStates	LcmOperationStateType	0..N	Match particular LCM operation state values as reported in notifications of type NsLcmOperationOccurrenceNotification. May be present if the "notificationTypes" attribute contains the value "NsLcmOperationOccurrenceNotification", and shall be absent otherwise.
nsComponentTypes	NsComponentType	0..N	Match particular NS component types for the notification of type NsChangeNotification. May be present if the "notificationTypes" attribute contains the value "NsChangeNotification", and shall be absent otherwise.
lcmOpNameImpactingNsComponent	LcmOpNameForChangeNotificationType	0..N	Match particular LCM operation names for the notification of type NsChangeNotification. May be present if the "notificationTypes" attribute contains the value "NsChangeNotification", and shall be absent otherwise.
lcmOpOccStatusImpactingNsComponent	LcmOpOccStatusForChangeNotificationType	0..N	Match particular LCM operation status values as reported in notifications of type NsChangeNotification. May be present if the "notificationTypes" attribute contains the value "NsChangeNotification", and shall be absent otherwise.
NOTE: The permitted values of the "notificationTypes" attribute are spelled exactly as the names of the notification types to facilitate automated code generation systems.			

6.5.3.9 Type: LccnLinks

This type represents the links to resources that a notification can contain. It shall comply with the provisions defined in Table 6.5.3.9-1.

Table 6.5.3.9-1: Definition of the LccnLinks data type

Attribute name	Data type	Cardinality	Description
nsInstance	NotificationLink	1	Link to the resource representing the NS instance to which the notified change applies.
subscription	NotificationLink	1	Link to the subscription that triggered this notification.
nsLcmOpOcc	NotificationLink	0..1	Link to the lifecycle operation occurrence that this notification is related to. Shall be present if there is a related lifecycle operation occurrence.

6.5.3.10 Type: SapData

This type represents the information related to a SAP of an NS. It shall comply with the provisions defined in Table 6.5.3.10-1.

Table 6.5.3.10-1: Definition of the SapData data type

Attribute name	Data type	Cardinality	Description
sapId	IdentifierInNsd	1	Reference to the SAPD for this SAP.
sapName	String	1	Human readable name for the SAP.
description	String	1	Human readable description for the SAP.
sapProtocolData	CpProtocolData	0..N	Parameters for configuring the network protocols on the SAP.

6.5.3.11 Type: CpProtocolData

This type represents network protocol data. It shall comply with the provisions defined in Table 6.5.3.11-1.

Table 6.5.3.11-1: Definition of the CpProtocolData data type

Attribute name	Data type	Cardinality	Description
layerProtocol	Enum (inlined)	1	Identifier of layer(s) and protocol(s). Permitted values: IP_OVER_ETHERNET See note.
ipOverEthernet	IpOverEthernetAddressData	0..1	Network address data for IP over Ethernet to assign to the extCP instance. Shall be present if layerProtocol is equal to "IP_OVER_ETHERNET", and shall be absent otherwise.
NOTE:	This attribute allows to signal the addition of further types of layer and protocol in future versions of the present document in a backwards-compatible way. In the current version of the present document, only IP over Ethernet is supported.		

6.5.3.12 Type: IpOverEthernetAddressData

This type represents network address data for IP over Ethernet. It shall comply with the provisions defined in Table 6.5.3.12-1.

Table 6.5.3.12-1: Definition of the IpOverEthernetAddressData data type

Attribute name	Data type	Cardinality	Description
macAddress	MacAddress	0..1	MAC address. If this attribute is not present, it shall be chosen by the NFV MANO. See note 1.
ipAddresses	Structure (inlined)	0..N	List of IP addresses to assign to the extCP instance. Each entry represents IP address data for fixed or dynamic IP address assignment per subnet. If this attribute is not present, no IP address shall be assigned. See note 1.
>type	Enum (inlined)	1	The type of the IP addresses. Permitted values: IPV4, IPV6.
>fixedAddresses	IpAddress	0..N	Fixed addresses to assign (from the subnet defined by "subnetId" if provided). See note 2.
>numDynamicAddresses	Integer	0..1	Number of dynamic addresses to assign (from the subnet defined by "subnetId" if provided). See note 2.
>addressRange	Structure (inlined)	0..1	An IP address range to be used, e.g. in case of egress connections. In case this attribute is present, IP addresses from the range will be used. See note 2.
>>minAddress	IpAddress	1	Lowest IP address belonging to the range.
>>maxAddress	IpAddress	1	Highest IP address belonging to the range.
>subnetId	IdentifierInVim	0..1	Subnet defined by the identifier of the subnet resource in the VIM. In case this attribute is present, IP addresses from that subnet will be assigned; otherwise, IP addresses not bound to a subnet will be assigned.
NOTE 1: At least one of "macAddress" or "ipAddresses" shall be present.			
NOTE 2: Exactly one of "fixedAddresses", "numDynamicAddresses" or "ipAddressRange" shall be present.			

6.5.3.13 Type: PnfInfo

This type represents the information about a PNF that is part of an NS instance. It shall comply with the provisions defined in Table 6.5.3.13-1.

Table 6.5.3.13-1: Definition of the PnfInfo data type

Attribute name	Data type	Cardinality	Description
pnfId	Identifier	1	Identifier of the PNF. This identifier is allocated by the OSS/BSS.
pnfName	String	1	Name of the PNF.
pnfId	Identifier	1	Identifier of the PNFD on which the PNF is based.
pnfIdInfold	Identifier	1	Identifier of the PNFD information object related to this PNF. This identifier is allocated by the NFVO.
pnfProfileId	IdentifierInNsd	1	Identifier of the related PnfProfile in the NSD on which the PNF is based.
cpInfo	PnfExtCpInfo	1..N	Information on the external CP of the PNF.

6.5.3.14 Type: AddPnfData

This type specifies an PNF to be added to the NS instance and the PNF Profile to use for this PNF. It shall comply with the provisions defined in Table 6.5.3.14-1.

Table 6.5.3.14-1: Definition of the AddPnfData data type

Attribute name	Data type	Cardinality	Description
pnfId	Identifier	1	Identifier of the PNF. This identifier is allocated by the OSS/BSS.
pnfName	String	1	Name of the PNF.
pnfdId	Identifier	1	Identifier of the PNFD on which the PNF is based.
pnfProfileId	IdentifierInNsd	1	Identifier of related PnfProfile in the NSD on which the PNF is based.
cpData	PnfExtCpData	0..N	Address assigned for the PNF external CP(s).

6.5.3.15 Type: ModifyPnfData

This type specifies an PNF to be modified in the NS instance. It shall comply with the provisions defined in Table 6.5.3.15-1.

Table 6.5.3.15-1: Definition of the ModifyPnfData data type

Attribute name	Data type	Cardinality	Description
pnfId	Identifier	1	Identifier of the PNF. This identifier is allocated by the OSS/BSS.
pnfName	String	0..1	Name of the PNF. See note.
cpData	PnfExtCpData	0..N	Address assigned for the PNF external CP(s). See note.

NOTE: At least one attribute shall be present.

6.5.3.16 Type: PnfExtCpData

This type represents the configuration data on the external CP of the PNF. It shall comply with the provisions defined in Table 6.5.3.16-1.

Table 6.5.3.16-1: Definition of the PnfExtCpData data type

Attribute name	Data type	Cardinality	Description
cpInstanceId	IdentifierInPnf	0..1	Identifier of the CP. Shall be present for existing CP.
cpdId	IdentifierInNsd	0..1	Identifier of the Connection Point Descriptor (CPD) for this CP. Shall be present for new CP.
cpProtocolData	CpProtocolData	1..N	Address assigned for this CP.

6.5.3.17 Type: PnfExtCpInfo

This type represents the information about the external CP of the PNF. It shall comply with the provisions defined in Table 6.5.3.17-1.

Table 6.5.3.17-1: Definition of the PnfExtCpInfo data type

Attribute name	Data type	Cardinality	Description
cpInstanceId	IdentifierInPnf	1	Identifier of the CP in the scope of the PNF.
cpdId	IdentifierInNsd	1	Identifier of (reference to) the Connection Point Descriptor (CPD) for this CP.
cpProtocolData	cpProtocolData	1..N	Parameters for configuring the network protocols on the CP.

6.5.3.18 Type: IpOverEthernetAddressInfo

This type represents information about a network address that has been assigned. It shall comply with the provisions defined in Table 6.5.3.18-1.

Table 6.5.3.18-1: Definition of the IpOverEthernetAddressInfo data type

Attribute name	Data type	Cardinality	Description
macAddress	MacAddress	0..1	Assigned MAC address. See note 1.
ipAddresses	Structure (inlined)	0..N	Addresses assigned to the CP or SAP instance. Each entry represents IP addresses assigned by fixed or dynamic IP address assignment per subnet. See note 1.
>type	Enum (inlined)	1	The type of the IP addresses. Permitted values: IPV4, IPV6.
>addresses	IpAddress	0..N	Fixed addresses assigned (from the subnet defined by "subnetId" if provided). See note 2.
>isDynamic	Boolean	0..1	Indicates whether this set of addresses was assigned dynamically (true) or based on address information provided as input from the API consumer (false). Shall be present if "addresses" is present and shall be absent otherwise.
>addressRange	Structure (inlined)	0..1	An IP address range used, e.g. in case of egress connections. See note 2.
>>minAddress	IpAddress	1	Lowest IP address belonging to the range
>>maxAddress	IpAddress	1	Highest IP address belonging to the range
>subnetId	IdentifierInVim	0..1	Subnet defined by the identifier of the subnet resource in the VIM. In case this attribute is present, IP addresses are bound to that subnet.

NOTE 1: At least one of "macAddress" or "ipAddresses" shall be present.
NOTE 2: Exactly one of "addresses" or "addressRange" shall be present.

6.5.3.19 Type: VnfInstanceData

This type specifies an existing VNF instance to be used in the NS instance and if needed, the VNF Profile to use for this VNF instance. It shall comply with the provisions defined in Table 6.5.3.19-1.

Table 6.5.3.19-1: Definition of the VnfInstanceData data type

Attribute name	Data type	Cardinality	Description
vnfInstanceid	Identifier	1	Identifier of the existing VNF instance to be used in the NS.
vnfProfileid	IdentifierInNsd	0..1	Identifier of (Reference to) a vnfProfile defined in the NSD which the existing VNF instance shall be matched with. If not present, the NFVO will select the VnfProfile matching the information in the VNF instance.

6.5.3.19a Type: NestedNsInstanceData

This type specifies an existing nested NS instance to be used in the NS instance and if needed, the NsProfile to use for this nested NS instance. It shall comply with the provisions defined in Table 6.5.3.19a-1.

Table 6.5.3.19a-1: Definition of the NestedNsInstanceData data type

Attribute name	Data type	Cardinality	Description
nestedNsInstanceid	Identifier	1	Identifier of the existing nested NS instance to be used in the NS.
nsProfileid	IdentifierInNsd	0..1	Identifier of an NsProfile defined in the NSD which the existing nested NS instance shall be matched with. If not present, the NFVO will select the NsProfile matching the information in the nested NS instance.

6.5.3.20 Type: VnfLocationConstraint

This type represents the association of location constraints to a VNF instance to be created according to a specific VNF profile. It shall comply with the provisions defined in Table 6.5.3.20-1.

Table 6.5.3.20-1: Definition of the VnfLocationConstraint data type

Attribute name	Data type	Cardinality	Description
vnfProfileId	IdentifierInNsd	1	Identifier (reference to) of a VnfProfile in the NSD used to manage the lifecycle of the VNF instance.
locationConstraints	LocationConstraints	1	Defines the location constraints for the VNF instance to be created based on the VNF profile. See note.
NOTE: These constraints are typically determined by the OSS from service requirements (e.g. latency requirements, regulatory requirements). The NFVO can map such location constraints to eligible NFVI-PoPs/resource zones where the VNF instance is to be created.			

6.5.3.21 Type: LocationConstraints

This type represents location constraints for a VNF to be instantiated. The location constraints shall be presented as a country code, optionally followed by a civic address based on the format defined by IETF RFC 4776 [8]. The LocationConstraints data type shall comply with the provisions defined in Table 6.5.3.21-1.

Table 6.5.3.21-1: Definition of the LocationConstraints data type

Attribute name	Data type	Cardinality	Description
countryCode	String	1	The two-letter ISO 3166 [13] country code in capital letters.
civicAddressElement	Structure (inlined)	0..N	Zero or more elements comprising the civic address.
>caType	Integer	1	Describe the content type of caValue. The value of caType shall comply with section 3.4 of IETF RFC 4776 [8].
>caValue	String	1	Content of civic address element corresponding to the caType. The format caValue shall comply with section 3.4 of IETF RFC 4776 [8].

6.5.3.21a Type: ParamsForNestedNs

This type specifies additional parameters on a per-nested NS instance basis. It shall comply with the provisions defined in Table 6.5.3.21a-1.

Table 6.5.3.21a-1: Definition of the ParamsForNestedNs data type

Attribute name	Data type	Cardinality	Description
nsProfileId	IdentifierInNsd	1	Identifier of an NSProfile to which the additional parameters apply.
additionalParam	KeyValuePairs	0..N	Additional parameters that are to be applied on a per nested NS instance.

6.5.3.22 Type: ParamsForVnf

This type defines the additional parameters for the VNF instance to be created associated with an NS instance. It shall comply with the provisions defined in Table 6.5.3.22-1.

Table 6.5.3.22-1: Definition of the ParamsForVnf data type

Attribute name	Data type	Cardinality	Description
vnfProfileId	IdentifierInNsd	1	Identifier of the vnfProfile applicable to the VNF instance to which the additional parameters apply.
vnfInstanceName	String	0..1	Human-readable name of the VNF instance to be created.
vnfInstanceDescription	String	0..1	Human-readable description of the VNF instance to be created.
vnfConfigurableProperties	KeyValuePairs	0..1	Values for the "vnfConfigurableProperties" input parameter of the Instantiate VNF operation defined in ETSI GS NFV-SOL 003 [4].
metadata	KeyValuePairs	0..1	Values for the "metadata" input parameter of the Create VNF Identifier operation defined in ETSI GS NFV-SOL 003 [4].
extensions	KeyValuePairs	0..1	Values for the "extensions" attribute of the Instantiate VNF operation defined in ETSI GS NFV-SOL 003 [4].
additionalParams	KeyValuePairs	0..1	Additional input parameters for the instantiation process, specific to the VNF being instantiated, as declared in the VNFD as part of "InstantiateVnfOpConfig".

6.5.3.23 Type: AffinityOrAntiAffinityRule

This type describes the additional affinity or anti-affinity rule applicable between the VNF instances to be instantiated in the NS instantiation operation request or between the VNF instances to be instantiated in the NS instantiation operation request and the existing VNF instances. It shall comply with the provisions defined in Table 6.5.3.23-1.

Table 6.5.3.23-1: Definition of the AffinityOrAntiAffinityRule data type

Attribute name	Data type	Cardinality	Description
vnfdId	Identifier	0..N	Reference to a VNFD. When the VNFD which is not used to instantiate VNF, it presents all VNF instances of this type as the subjects of the affinity or anti-affinity rule. The VNF instance which the VNFD presents is not necessary as a part of the NS to be instantiated.
vnfProfileId	IdentifierInNsd	1..N	Reference to a vnfProfile defined in the NSD. At least one VnfProfile which is used to instantiate VNF for the NS to be instantiated as the subject of the affinity or anti-affinity rule shall be present. When the VnfProfile which is not used to instantiate VNF, it presents all VNF instances of this type as the subjects of the affinity or anti-affinity rule. The VNF instance which the VnfProfile presents is not necessary as a part of the NS to be instantiated.
vnfInstanceId	Identifier	0..N	Reference to the existing VNF instance as the subject of the affinity or anti-affinity rule. The existing VNF instance is not necessary as a part of the NS to be instantiated.
affinityOrAntiAffinity	Enum (inlined)	1	The type of the constraint. Permitted values: <ul style="list-style-type: none"> • AFFINITY • ANTI_AFFINITY
scope	Enum (inlined)	1	Specifies the scope of the rule where the placement constraint applies. Permitted values: <ul style="list-style-type: none"> • NFVI_POP • ZONE • ZONE_GROUP • NFVI_NODE

6.5.3.24 Type: InstantiateVnfData

This type represents the information that are needed for VNF instantiation. This information element is used for the bottom-up NS creation when the OSS/BSS explicitly requests VNF instantiation for a given NS. When the NFVO invokes the Instantiate VNF update operation, a set of these parameters are then passed by the NFVO to the VNFM. It shall comply with the provisions defined in Table 6.5.3.24-1.

Table 6.5.3.24-1: Definition of the InstantiateVnfData data type

Attribute name	Data type	Cardinality	Description
vnfdId	Identifier	1	Information sufficient to identify the VNFD which defines the VNF to be instantiated.
vnfFlavourId	IdentifierInVnfd	1	Identifier of the VNF deployment flavour to be instantiated.
vnfInstantiationLevelId	IdentifierInVnfd	0..1	Identifier of the instantiation level of the deployment flavour to be instantiated. If not present, the default instantiation level as declared in the VNFD is instantiated.
vnfInstanceName	String	0..1	Human-readable name of the VNF instance to be created.
vnfInstanceDescription	String	0..1	Human-readable description of the VNF instance to be created.
extVirtualLinks	ExtVirtualLinkData	0..N	Information about external VLS to connect the VNF to.
extManagedVirtualLinks	ExtManagedVirtualLinkData	0..N	Information about internal VLS that are managed by other entities than the VNFM.
localizationLanguage	String	0..1	Localization language of the VNF to be instantiated. The value shall comply with the format defined in IETF RFC 5646 [9].
vnfConfigurableProperties	KeyValuePairs	0..1	Values for the "vnfConfigurableProperties" input parameter of the Instantiate VNF operation defined in ETSI GS NFV-SOL 003 [4].
additionalParams	KeyValuePairs	0..1	Additional input parameters for the instantiation process, specific to the VNF being instantiated, as declared in the VNFD as part of "InstantiateVnfOpConfig".
metadata	KeyValuePairs	0..1	This attribute provides values for the "metadata" input parameter of the Create VNF Identifier operation defined in ETSI GS NFV-SOL 003 [4].
extensions	KeyValuePairs	0..1	This attribute provides values for the "extensions" input parameter of the Instantiate VNF operation defined in ETSI GS NFV-SOL 003 [4].
locationConstraints	VnfLocationConstraint	0..1	Defines the location constraints for the VNF to be instantiated as part of the NS Update. An example can be a constraint for the VNF to be in a specific geographic location.

6.5.3.25 Type: ChangeVnfFlavourData

The type represents the information that is requested to be changed deployment flavour for an existing VNF instance. It shall comply with the provisions defined in Table 6.5.3.25-1.

Table 6.5.3.25-1: Definition of the ChangeVnfFlavourData data type

Attribute name	Data type	Cardinality	Description
vnfInstanceId	Identifier	1	Identifier of the VNF instance to be modified.
newFlavourId	IdentifierInVnfd	1	Identifier of the VNF deployment flavour to be instantiated.
instantiationLevelId	IdentifierInVnfd	0..1	Identifier of the instantiation level of the deployment flavour to be instantiated. If not present, the default instantiation level as declared in the VNFD is instantiated.
extVirtualLinks	ExtVirtualLinkData	0..N	Information about external VLs to connect the VNF to. Entries in the list of external VLs that are unchanged need not be supplied.
extManagedVirtualLinks	ExtManagedVirtualLinkData	0..N	Information about internal VLs that are managed by NFVO. See note.
additionalParams	KeyValuePairs	0..1	Additional input parameters for the flavour change process, specific to the VNF being modified, as declared in the VNFD as part of "ChangeVnfFlavourOpConfig".
extensions	KeyValuePairs	0..1	This attribute provides values for the "extensions" input parameter of the Change VNF Flavour operation defined in ETSI GS NFV-SOL 003 [4].
vnfConfigurableProperties	KeyValuePairs	0..1	This attribute provides values for the "vnfConfigurableProperties" input parameter of the Change VNF Flavour operation defined in ETSI GS NFV-SOL 003 [4].
NOTE: 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.			

6.5.3.26 Type: ExtVirtualLinkData

This type represents an external VL. It shall comply with the provisions defined in Table 6.5.3.26-1.

Table 6.5.3.26-1: Definition of the ExtVirtualLinkData data type

Attribute name	Data type	Cardinality	Description
extVirtualLinkId	Identifier	0..1	The identifier of the external VL instance, if provided.
vimId	Identifier	0..1	Identifier of the VIM that manages this resource. This attribute shall only be supported and present if VNF-related resource management in direct mode is applicable.
resourceProviderId	Identifier	0..1	Identifies the entity responsible for the management of this resource. This attribute shall only be supported and present if VNF-related resource management in indirect mode is applicable. The identification scheme is outside the scope of the present document.
resourceId	IdentifierInVim	1	The identifier of the resource in the scope of the VIM or the resource provider.
extCps	VnfExtCpData	1..N	External CPs of the VNF to be connected to this external VL.
extLinkPorts	ExtLinkPortData	0..N	Externally provided link ports to be used to connect external connection points to this external VL.

6.5.3.27 Type: ExtManagedVirtualLinkData

This type represents an externally-managed internal VL. It shall comply with the provisions defined in Table 6.5.3.27-1.

Table 6.5.3.27-1: Definition of the ExtManagedVirtualLinkData data type

Attribute name	Data type	Cardinality	Description
extManagedVirtualLinkId	Identifier	0..1	The identifier of the externally-managed internal VL instance, if provided.
vnfVirtualLinkDescId	IdentifierInVnfd	1	The identifier of the VLD in the VNFD for this VL.
vimId	Identifier	0..1	Identifier of the VIM that manage this resource. This attribute shall only be supported and present if VNF-related resource management in direct mode is applicable.
resourceProviderId	Identifier	0..1	Identifies the entity responsible for the management of this resource. This attribute shall only be supported and present if VNF-related resource management in indirect mode is applicable. The identification scheme is outside the scope of the present document.
resourceId	IdentifierInVim	1	The identifier of the resource in the scope of the VIM or the resource provider.

6.5.3.28 Type: ExtLinkPortData

This type represents an externally provided link port to be used to connect a VNF external connection point to an external VL. It shall comply with the provisions defined in Table 6.5.3.28-1.

Table 6.5.3.28-1: Definition of the ExtLinkPortData data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this link port as provided by the entity that has created the link port.
resourceHandle	ResourceHandle	1	Reference to the virtualised resource realizing this link port.

6.5.3.29 Type: VnfExtCpData

This type represents configuration information for external CPs created from a CPD. It shall comply with the provisions defined in Table 6.5.3.29-1.

Table 6.5.3.29-1: Definition of the VnfExtCpData data type

Attribute name	Data type	Cardinality	Description
cpId	IdentifierInVnfd	1	The identifier of the CPD in the VNFD.
cpConfig	VnfExtCpConfig	1..N	List of instance data that need to be configured on the CP instances created from the respective CPD.

6.5.3.30 Type: VnfExtCpConfig

This type represents an externally provided link port or network address information per instance of a VNF external connection point. In case a link port is provided, the NFVO shall use that link port when connecting the VNF external CP to the external VL. In case a link port is not provided, the NFVO or VNFM shall create a link port on the external VL, and use that link port to connect the VNF external CP to the external VL.

This type shall comply with the provisions defined in Table 6.5.3.30-1.

Table 6.5.3.30-1: Definition of the VnfExtCpConfig data type

Attribute name	Data type	Cardinality	Description
cpInstanceld	IdentifierInVnf	0..1	Identifier of the external CP instance to which this set of configuration parameters is requested to be applied. Shall be present if this instance has already been created.
linkPortId	Identifier	0..1	Identifier of a pre-configured link port to which the external CP will be associated. See note.
cpProtocolData	CpProtocolData	0..N	Parameters for configuring the network protocols on the link port that connects the CP to a VL. See note.
<p>NOTE: The following conditions apply to the attributes "linkPortId" and "cpProtocolData":</p> <ul style="list-style-type: none"> – The "linkPortId" and "cpProtocolData" attributes shall both be absent for the deletion of an existing external CP instance addressed by cpInstanceld. – At least one of these attributes shall be present for a to-be-created external CP instance or an existing external CP instance. – If the "cpProtocolData" attribute is absent, the "linkPortId" attribute shall be provided referencing a pre-created link port with pre-configured address information. – If both "cpProtocolData" and "linkportId" are provided, the API consumer shall ensure that the cpProtocolData can be used with the pre-created link port referenced by "linkPortId". 			

6.5.3.31 Type: OperateVnfData

This type represents a VNF instance for which the operational state needs to be changed and the requested new state. It shall comply with the provisions defined in Table 6.5.3.31-1.

Table 6.5.3.31-1: Definition of the OperateVnfData data type

Attribute name	Data type	Cardinality	Description
vnfInstanceld	Identifier	1	Identifier of the VNF instance.
changeStateTo	OperationalStates	1	The desired operational state (i.e. started or stopped) to change the VNF to.
stopType	StopType	0..1	It signals whether forceful or graceful stop is requested. See note.
gracefulStopTimeout	Integer	0..1	The time interval (in seconds) to wait for the VNF to be taken out of service during graceful stop, before stopping the VNF. See note.
additionalParam	KeyValuePairs	0..1	Additional parameters passed by the OSS/BSS as input to the Operate VNF operation, specific to the VNF being operated.
<p>NOTE: The "stopType" and "gracefulStopTimeout" attributes shall be absent, when the "changeStateTo" attribute is equal to "STARTED". The "gracefulStopTimeout" attribute shall be present, when the "changeStateTo" attribute is equal to "STOPPED" and the "stopType" attribute is equal to "GRACEFUL". The "gracefulStopTimeout" attribute shall be absent, when the "changeStateTo" attribute is equal to "STOPPED" and the "stopType" attribute is equal to "FORCEFUL". The request shall be treated as if the "stopType" attribute was set to "FORCEFUL", when the "changeStateTo" attribute is equal to "STOPPED" and the "stopType" attribute is absent.</p>			

6.5.3.32 Type: ModifyVnfInfoData

This type represents the information that is requested to be modified for a VNF instance. The information to be modified shall comply with the associated VNFD. This type shall comply with the provisions defined in Table 6.5.3.32-1.

Table 6.5.3.32-1: Definition of the ModifyVnfInfoData data type

Attribute name	Data type	Cardinality	Description
vnfInstanceId	Identifier	1	Identifier of the VNF instance.
vnfInstanceName	String	0..1	New value of the "vnfInstanceName" attribute in "VnfInstance", or "null" to remove the attribute.
vnfInstanceDescription	String	0..1	New value of the "vnfInstanceDescription" attribute in "VnfInstance", or "null" to remove the attribute.
vnfdId	Identifier	0..1	New value of the "vnfdId" attribute in "VnfInstance". The value "null" is not permitted.
vnfConfigurableProperties	KeyValuePairs	0..1	Modifications to entries in the "vnfConfigurableProperties" attribute in "VnfInstance", as defined in clause 6.5.3.57.
metadata	KeyValuePairs	0..1	Modifications to entries in the "metadata" attribute in "VnfInstance", as defined in clause 6.5.3.57.
extensions	KeyValuePairs	0..1	Modifications to entries in the "extensions" attribute in "VnfInstance", as defined in clause 6.5.3.57.

6.5.3.33 Type: ChangeExtVnfConnectivityData

This type describes the information invoked by the NFVO to change the external VNF connectivity information maintained by the VNFM. The types of changes that this operation supports are:

- 1) Disconnect the external CPs that are connected to a particular external VL, and connect them to a different external VL.
- 2) Change the connectivity parameters of the existing external CPs, including changing addresses.

NOTE: Depending on the capabilities of the underlying VIM resources, certain changes (e.g. modifying the IP address assignment) might not be supported without deleting the resource and creating another one with the modified configuration.

This type shall comply with the provisions defined in Table 6.5.3.33-1.

Table 6.5.3.33-1: Definition of the ChangeExtVnfConnectivityData data type

Attribute name	Data type	Cardinality	Description
vnfInstanceId	Identifier	1	Identifier of the VNF instance.
extVirtualLinks	ExtVirtualLinkData	1..N	Information about external VLs to change (e.g. connect the VNF to). Entries in the list of external VLs that are unchanged need not be supplied.
additionalParams	KeyValuePairs	0..1	Additional parameters passed by the OSS as input to the external connectivity change process, specific to the VNF instance being changed.

6.5.3.34 Type: AssocNewNsdVersionData

This type specifies a new NSD version that is associated to the NS instance. After issuing the Update NS operation with updateType = "AssocNewNsdVersion", the NFVO shall use the referred NSD as a basis for the given NS instance. Different versions of the same NSD have same nsdInvariantId, but different nsdId attributes, therefore if the nsdInvariantId of the NSD version that is to be associated to this NS instance is different from the one used before, the NFVO shall reject the request. Only new versions of the same NSD can be associated to an existing NS instance. This data type shall comply with the provisions defined in Table 6.5.3.34-1.

Table 6.5.3.34-1: Definition of the AssocNewNsdVersionData data type

Attribute name	Data type	Cardinality	Description
newNsdId	Identifier	1	Identifier of the new NSD version that is to be associated to the NS instance.
sync	Boolean	0..1	Specify whether the NS instance shall be automatically synchronized to the new NSD by the NFVO (in case of true value) or the NFVO shall not do any action (in case of a false value) and wait for further guidance from OSS/BSS (i.e. waiting for OSS/BSS to issue NS lifecycle management operation to explicitly add/remove VNFs and modify information of VNF instances according to the new NSD). The synchronization to the new NSD means e.g. instantiating/adding those VNFs whose VNFD is referenced by the new NSD version but not referenced by the old one, terminating/removing those VNFs whose VNFD is referenced by the old NSD version but not referenced by the new NSD version, modifying information of VNF instances to the new applicable VNFD provided in the new NSD version. A cardinality of 0 indicates that synchronization shall not be done.

6.5.3.35 Type: MoveVnfInstanceData

This type specifies existing VNF instances to be moved from one NS instance (source) to another NS instance (destination). The NS instance defined in the Update NS operation indicates the source NS instance and the destination NS instance is specified in this data type (referred to targetNsInstanceId). It shall comply with the provisions defined in Table 6.5.3.35-1.

Table 6.5.3.35-1: Definition of the MoveVnfInstanceData data type

Attribute name	Data type	Cardinality	Description
targetNsInstanceId	Identifier	1	Specify the target NS instance where the VNF instances are moved to.
vnfInstanceId	Identifier	1..N	Specify the VNF instance that is moved.

6.5.3.36 Type: AddVnffgData

This type specifies the parameters used for the creation of a new VNFFG instance. It shall comply with the provisions defined in Table 6.5.3.36-1.

Table 6.5.3.36-1: Definition of the AddVnffgData data type

Attribute name	Data type	Cardinality	Description
vnffgId	IdentifierInNsd	1	Identifier of the VNFFGD used to create this VNFFG instance.
vnffgName	String	1	Human readable name for the VNFFG.
description	String	1	Human readable description for the VNFFG.

6.5.3.37 Type: UpdateVnffgData

This type specifies the parameters used for the update of an existing VNFFG instance. It shall comply with the provisions defined in Table 6.5.3.37-1.

Table 6.5.3.37-1: Definition of the UpdateVnffgData data type

Attribute name	Data type	Cardinality	Description
vnffgInfold	IdentifierInNs	1	Identifier of an existing VNFFG to be updated for the NS Instance.
nfp	NfpData	0..N	Indicate the desired new NFP(s) for a given VNFFG after the operations of addition/removal of NS components (e.g. VNFs, VLs, etc.) have been completed, or indicate the updated or newly created NFP classification and selection rule which applied to an existing NFP.
nfpInfold	IdentifierInNs	0..N	Identifier(s) of the NFP to be deleted from a given VNFFG.

6.5.3.38 Type: NfpData

This type contains information used to create or modify NFP instance parameters for the update of an existing VNFFG instance. It shall comply with the provisions defined in Table 6.5.3.38-1.

Table 6.5.3.38-1: Definition of the NfpData data type

Attribute name	Data type	Cardinality	Description
nfpInfold	IdentifierInNs	0..1	Identifier of the NFP to be modified. It shall be present for modified NFPs and shall be absent for the new NFP. See note 1.
nfpName	String	0..1	Human readable name for the NFP. It shall be present for the new NFP, and it may be present otherwise. See note 2.
description	String	0..1	Human readable description for the NFP. It shall be present for the new NFP, and it may be present otherwise. See note 2.
cpGroup	CpGroupInfo	0..N	Group(s) of CPs and/or SAPs which the NFP passes by. Cardinality can be 0 if only updated or newly created NFP classification and selection rule which applied to an existing NFP is provided. See note 3 and note 4.
nfpRule	NfpRule	0..1	NFP classification and selection rule. See note 1.
NOTE 1: It shall be present for modified NFPs and shall be absent for the new NFP.			
NOTE 2: It shall be present for the new NFP, and it may be present otherwise.			
NOTE 3: At least a CP or an nfpRule shall be present.			
NOTE 4: When multiple identifiers are included, the position of the identifier in the cpGroup value specifies the position of the group in the path.			

6.5.3.39 Type: ChangeNsFlavourData

This type specifies an existing NS instance for which the DF needs to be changed. This specifies the new DF, the instantiationLevel of the new DF that may be used. It shall comply with the provisions defined in Table 6.5.3.39-1.

Table 6.5.3.39-1: Definition of the ChangeNsFlavourData data type

Attribute name	Data type	Cardinality	Description
newNsFlavourId	IdentifierInNsd	1	Identifier of the new NS DF to apply to this NS instance.
instantiationLevelId	IdentifierInNsd	0..1	Identifier of the instantiation level of the deployment flavour to be instantiated. If not present, the default instantiation level as declared in the NSD is instantiated.

6.5.3.40 Type: NfpRule

The NfpRule data type is an expression of the conditions that shall be met in order for the NFP to be applicable to the packet. The condition acts as a flow classifier and it is met only if all the values expressed in the condition are matched by those in the packet. It shall comply with the provisions defined in Table 6.5.3.40-1.

Table 6.5.3.40-1: Definition of the NfpRule data type

Attribute name	Data type	Cardinality	Description
etherDestinationAddress	MacAddress	0..1	Indicates a destination Mac address See note.
etherSourceAddress	MacAddress	0..1	Indicates a source Mac address See note.
etherType	Enum (inlined)	0..1	Indicates the protocol carried over the Ethernet layer. Permitted values: IPV4 IPV6 See note.
vlanTag	String	0..N	Indicates a VLAN identifier in an IEEE 802.1Q-2018 tag [6] Multiple tags can be included for QinQ stacking. See note.
protocol	Enum (inlined)	0..1	Indicates the L4 protocol, For IPv4 [7] this corresponds to the field called "Protocol" to identify the next level protocol. For IPv6 [12] this corresponds to the field is called the "Next Header" field. Permitted values: Any keyword defined in the IANA protocol registry [1], e.g.: - TCP - UDP - ICMP See note.
dscp	String	0..1	For IPv4 [7] a string of "0" and "1" digits that corresponds to the 6-bit Differentiated Services Code Point (DSCP) field of the IP header. For IPv6 [12] a string of "0" and "1" digits that corresponds to the 6 differentiated services bits of the traffic class header field. See note.
sourcePortRange	PortRange	0..1	Indicates a range of source ports. See note.
destinationPortRange	PortRange	0..1	Indicates a range of destination ports. See note.
sourceIpAddressPrefix	IpAddressPrefix	0..1	Indicates the source IP address range in CIDR format. See note.
destinationIpAddressPrefix	IpAddressPrefix	0..1	Indicates the destination IP address range in CIDR format. See note.
extendedCriteria	Mask	0..N	Indicates values of specific bits in a frame. See note.
NOTE: At least one attribute shall be present. If multiple attributes are present, a logical "AND" operation shall be applied to those attributes when matching packets against the rule.			

6.5.3.41 Type: Mask

The Mask data type identifies the value to be matched for a sequence of bits at a particular location in a frame. It shall comply with the provisions defined in Table 6.5.3.41-1.

Table 6.5.3.41-1: Definition of the Mask data type

Attribute name	Data type	Cardinality	Description
startingPoint	Integer	1	Indicates the offset between the last bit of the source mac address and the first bit of the sequence of bits to be matched.
length	Integer	1	Indicates the number of bits to be matched.
value	String	1	Provide the sequence of bit values to be matched.

6.5.3.42 Type: PortRange

The PortRange data type provides the lower and upper bounds of a range of Internet ports. It shall comply with the provisions defined in Table 6.5.3.42-1.

Table 6.5.3.42-1: Definition of the PortRange data type

Attribute name	Data type	Cardinality	Description
lowerPort	Integer	1	Identifies the lower bound of the port range.
upperPort	Integer	1	Identifies the upper bound of the port range.

6.5.3.43 Type: HealNsData

This type represents the information used to heal an NS. It shall comply with the provisions defined in Table 6.5.3.43-1.

Table 6.5.3.43-1: Definition of the HealNsData data type

Attribute name	Data type	Cardinality	Description
degreeHealing	Enum (inlined)	1	Indicates the degree of healing. Possible values include: <ul style="list-style-type: none"> - HEAL_RESTORE: Complete the healing of the NS restoring the state of the NS before the failure occurred - HEAL_QOS: Complete the healing of the NS based on the newest QoS values - HEAL_RESET: Complete the healing of the NS resetting to the original instantiation state of the NS - PARTIAL_HEALING
actionsHealing	String	0..N	Used to specify dedicated healing actions in a particular order (e.g. as a script). The actionsHealing attribute can be used to provide a specific script whose content and actions might only be possible to be derived during runtime. See note.
healScript	IdentifierInNsd	0..1	Reference to a script from the NSD that shall be used to execute dedicated healing actions in a particular order. The healScript, since it refers to a script in the NSD, can be used to execute healing actions which are defined during NS design time. See note.
additionalParamsforNs	KeyValuePairs	0..1	Allows the OSS/BSS to provide additional parameter(s) to the healing process at the NS level.
NOTE: Either the actionsHealing or healScript attribute shall be present, not both attributes.			

6.5.3.44 Type: HealVnfData

This type represents the information to heal a VNF that is part of an NS. The NFVO shall then invoke the Heal VNF operation towards the appropriate VNFM. It shall comply with the provisions defined in Table 6.5.3.44-1.

Table 6.5.3.44-1: Definition of the HealVnfData data type

Attribute name	Data type	Cardinality	Description
vnfInstanceid	Identifier	1	Identifies the VNF instance, part of the NS, requiring a healing action.
cause	String	0..1	Indicates the reason why a healing procedure is required.
additionalParams	KeyValuePairs	0..1	Additional parameters passed by the NFVO as input to the healing process, specific to the VNF being healed. EXAMPLE: Input parameters to VNF-specific healing procedures.

6.5.3.45 Type: ScaleNsData

This type represents the information to scale an NS. It shall comply with the provisions defined in Table 6.5.3.45-1.

Table 6.5.3.45-1: Definition of the ScaleNsData data type

Attribute name	Data type	Cardinality	Description
vnfInstanceToBeAdded	VnfInstanceData	0..N	An existing VNF instance to be added to the NS instance as part of the scaling operation. If needed, the VNF Profile to be used for this VNF instance may also be provided. See note 1, note 2 and note 3.
vnfInstanceToBeRemoved	Identifier	0..N	The VNF instance to be removed from the NS instance as part of the scaling operation. See note 1 and note 4.
scaleNsByStepsData	ScaleNsByStepsData	0..1	The information used to scale an NS instance by one or more scaling steps. See note 1.
scaleNsToLevelData	ScaleNsToLevelData	0..1	The information used to scale an NS instance to a target size. See note 1.
additionalParamsForNs	KeyValuePairs	0..1	Allows the OSS/BSS to provide additional parameter(s) at the NS level necessary for the NS scaling (as opposed to the VNF level, which is covered in additionalParamForVnf).
additionalParamsForVnf	ParamsForVnf	0..N	Allows the OSS/BSS to provide additional parameter(s) per VNF instance (as opposed to the NS level, which is covered in additionalParamforNs). This is for VNFs that are to be created by the NFVO as part of the NS scaling and not for existing VNF that are covered by the scaleVnfData.
locationConstraints	VnfLocationConstraint	0..N	The location constraints for the VNF to be instantiated as part of the NS scaling. An example can be a constraint for the VNF to be in a specific geographic location.
NOTE 1: No more than two attributes between vnfInstanceToBeAdded, vnfInstanceToBeRemoved, scaleNsByStepsData and scaleNsToLevelData shall be present. In case of two, the attributes shall be vnfInstanceToBeAdded and vnfInstanceToBeRemoved.			
NOTE 2: The DF of the VNF instance shall match the VNF DF present in the associated VNF Profile of the new NS flavour.			
NOTE 3: This functionality is the same as the one provided by the Update NS operation when the AddVnf update type is selected (see clause 7.3.5).			
NOTE 4: This functionality is the same as the one provided by the Update NS operation when the RemoveVnf update type is selected (see clause 7.3.5).			

6.5.3.46 Type: ScaleNsByStepsData

This type represents the information used to scale an NS instance by one or more scaling steps, with respect to a particular NS scaling aspect. Performing a scaling step means increasing/decreasing the capacity of an NS instance in a discrete manner, i.e. moving from one NS scale level to another. The NS scaling aspects and their corresponding NS scale levels applicable to the NS instance are declared in the NSD. It shall comply with the provisions defined in Table 6.5.3.46-1.

Table 6.5.3.46-1: Definition of the ScaleNsByStepsData data type

Attribute name	Data type	Cardinality	Description
scalingDirection	Enum (inlined)	1	The scaling direction. Possible values are: - SCALE_IN - SCALE_OUT
aspectId	IdentifierInNsd	1	The aspect of the NS that is requested to be scaled, as declared in the NSD.
numberOfSteps	Integer	0..1	The number of scaling steps to be performed. Defaults to 1.

6.5.3.47 Type: ScaleNsToLevelData

This type represents the information used to scale an NS instance to a target size. The target size is either expressed as an NS instantiation level or as a list of NS scale levels, one per NS scaling aspect, of the current DF. The NS instantiation levels, the NS scaling aspects and their corresponding NS scale levels applicable to the NS instance are declared in the NSD. It shall comply with the provisions defined in Table 6.5.3.47-1.

Table 6.5.3.47-1: Definition of the ScaleNsToLevelData data type

Attribute name	Data type	Cardinality	Description
nsInstantiationLevel	IdentifierInNsd	0..1	Identifier of the target NS instantiation level of the current DF to which the NS instance is requested to be scaled. See note.
nsScaleInfo	NsScaleInfo	0..N	For each NS scaling aspect of the current DF, defines the target NS scale level to which the NS instance is to be scaled. See note.
NOTE: Either nsInstantiationLevel or nsScaleInfo, but not both, shall be present.			

6.5.3.48 Type: NsScaleInfo

This type represents the target NS Scale level for each NS scaling aspect of the current deployment flavour. It shall comply with the provisions defined in Table 6.5.3.48-1.

Table 6.5.3.48-1: Definition of the NsScaleInfo data type

Attribute name	Data type	Cardinality	Description
nsScalingAspectId	IdentifierInNsd	1	Identifier of the NS scaling aspect.
nsScaleLevelId	IdentifierInNsd	1	Identifier of the NS scale level.

6.5.3.49 Type: ScaleVnfData

This type represents defines the information to scale a VNF instance to a given level, or to scale a VNF instance by steps. It shall comply with the provisions defined in Table 6.5.3.49-1.

Table 6.5.3.49-1: Definition of the ScaleVnfData data type

Attribute name	Data type	Cardinality	Description
vnfInstancelId	Identifier	1	Identifier of the VNF instance being scaled.
scaleVnfType	Enum (inlined)	1	Type of the scale VNF operation requested. Allowed values are: <ul style="list-style-type: none"> - SCALE_OUT - SCALE_IN - SCALE_TO_INSTANTIATION_LEVEL - SCALE_TO_SCALE_LEVEL(S) The set of types actually supported depends on the capabilities of the VNF being managed. See note 1.
scaleToLevelData	ScaleToLevelData	0..1	The information used for scaling to a given level. See note 2.
scaleByStepData	ScaleByStepData	0..1	The information used for scaling by steps. See note 2.
NOTE 1: ETSI GS NFV-IFA 010 [2] specifies that the lifecycle management operations that expand or contract a VNF instance include scale in, scale out, scale up and scale down. Vertical scaling (scale up, scale down) is not supported in the present document.			
NOTE 2: Either scaleToLevelData or scaleByStepData but not both shall be present. The scaleByStepData is used for scale out/in type of scaling, and the scaleToLevelData is used for scale to instantiation/scale level type of scaling.			

6.5.3.50 Type: ScaleToLevelData

This type describes the information used to scale a VNF instance to a target size. The target size is either expressed as an instantiation level of that DF as defined in the VNFD, or given as a list of scale levels, one per scaling aspect of that DF. Instantiation levels and scaling aspects are declared in the VNFD. The NFVO shall then invoke the ScaleVnfToLevel operation towards the appropriate VNFM. It shall comply with the provisions defined in Table 6.5.3.50-1.

Table 6.5.3.50-1: Definition of the ScaleToLevelData data type

Attribute name	Data type	Cardinality	Description
vnfInstantiationLevelId	IdentifierInVnfd	0..1	Identifier of the target instantiation level of the current deployment flavour to which the VNF is requested to be scaled. See note.
vnfScaleInfo	VnfScaleInfo	0..N	For each scaling aspect of the current deployment flavour, indicates the target scale level to which the VNF is to be scaled. See note.
additionalParams	KeyValuePairs	0..1	Additional parameters passed by the NFVO as input to the scaling process, specific to the VNF being scaled.
NOTE: Either the instantiationLevelId attribute or the scaleInfo attribute shall be included.			

6.5.3.51 Type: VnfScaleInfo

This type describes the provides information about the scale level of a VNF instance with respect to one scaling aspect. It shall comply with the provisions defined in Table 6.5.3.51-1.

Table 6.5.3.51-1: Definition of the VnfScaleInfo data type

Attribute name	Data type	Cardinality	Description
aspectId	IdentifierInVnfd	1	The scaling aspect.
scaleLevel	Integer	1	The scale level for that aspect. Minimum value 0, maximum value maxScaleLevel as declared in the VNFD.

6.5.3.52 Type: ScaleByStepData

This type describes the information to scale a VNF instance by steps. The NFVO shall then invoke the Scale VNF operation towards the appropriate VNFM. It shall comply with the provisions defined in Table 6.5.3.52-1.

Table 6.5.3.52-1: Definition of the ScaleByStepData data type

Attribute name	Data type	Cardinality	Description
aspectId	IdentifierInVnfd	1	Identifier of (reference to) the aspect of the VNF that is requested to be scaled, as declared in the VNFD.
numberOfSteps	Integer	0..1	Number of scaling steps. It shall be a positive number. Defaults to 1. The VNF provider defines in the VNFD whether or not a particular VNF supports performing more than one step at a time. Such a property in the VNFD applies for all instances of a particular VNF. See note.
additionalParams	KeyValuePairs	0..1	Additional parameters passed by the NFVO as input to the scaling process, specific to the VNF instance being scaled.
NOTE: A scaling step is the smallest unit by which a VNF instance can be scaled w.r.t a particular scaling aspect.			

6.5.3.53 Type: NsVirtualLinkInfo

This type specifies the information about an NS VL instance. It shall comply with the provisions defined in Table 6.5.3.53-1.

Table 6.5.3.53-1: Definition of the NsVirtualLinkInfo data type

Attribute name	Data type	Cardinality	Description
id	IdentifierInNs	1	Identifier of the VL instance.
nsVirtualLinkDescId	IdentifierInNsd	1	Identifier of the VLD in the NSD.
nsVirtualLinkProfileId	IdentifierInNsd	1	Identifier of the VL profile in the NSD.
resourceHandle	ResourceHandle	1..N	Identifier(s) of the virtualised network resource(s) realizing the VL instance. See note.
linkPort	NsLinkPortInfo	0..N	Link ports of the VL instance. Cardinality of zero indicates that no port has yet been created for the VL instance.

NOTE: As an NS can include NFs deployed in NFVI PoPs under the control of several different VIMs, deploying an NS VL can involve several VIMs each allocating different virtualised network resources. When this NsVirtualLink is provided as an ExtVirtualLink as input of a VNF LCM operation, the id of the ExtVirtualLink shall be the same as the corresponding NsVirtualLink. The connectivity between virtualised network resources allocated in different VIMs and part of the same VL is not addressed in the present document.

6.5.3.54 Type: VirtualStorageResourceInfo

This type represents the information that allows addressing a virtualised resource that is used by a VNF instance. It shall comply with the provisions defined in table 6.5.3.54-1.

Table 6.5.3.54-1: Definition of the VirtualStorageResourceInfo data type

Attribute name	Data type	Cardinality	Description
id	IdentifierInVnf	1	Identifier of this VirtualStorageResourceInfo instance.
virtualStorageDescId	IdentifierInVnfd	1	Identifier of the VirtualStorageDesc in the VNFD.
storageResource	ResourceHandle	1	Reference to the VirtualStorage resource.
reservationId	Identifier	0..1	The reservation identifier applicable to the resource. It shall be present when an applicable reservation exists.
metadata	KeyValuePairs	0..1	Metadata about this resource.

6.5.3.55 Type: NsLinkPortInfo

This type represents information about a link port of a VL instance. It shall comply with the provisions defined in Table 6.5.3.55-1.

Table 6.5.3.55-1: Definition of the NsLinkPortInfo data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this link port as provided by the entity that has created the link port.
resourceHandle	ResourceHandle	1	Identifier of the virtualised resource realizing this link port.
nsCpHandle	NsCpHandle	0..1	Identifier of the CP/SAP instance to be connected to this link port. The value refers to a vnfExtCpInfo item in the VnfInstance, or a pnfExtCpInfo item in the PnfInfo, or a sapInfo item in the NS instance. There shall be at most one link port associated with any connection point instance.

NOTE: When the NsVirtualLink, from which the present NsLinkPort is part of, is provided as an ExtVirtualLink as input of a VNF LCM operation, the id of the ExtLinkPort shall be the same as the corresponding NsLinkPort.

6.5.3.56 Type: NsCpHandle

This type represents an identifier of the CP or SAP instance. It shall comply with the provisions defined in Table 6.5.3.56-1.

Table 6.5.3.56-1: Definition of the NsCpHandle data type

Attribute name	Data type	Cardinality	Description
vnfInstancelid	Identifier	0..1	Identifier of the VNF instance associated to the CP instance. This attribute shall be present if the CP instance is VNF external CP. See note 1 and note 4.
vnfExtCpInstancelid	IdentifierInVnf	0..1	Identifier of the VNF external CP instance in the scope of the VNF instance. This attribute shall be present if the CP instance is VNF external CP. See note 1 and note 4.
pnfInfold	Identifier	0..1	Identifier of the PNF instance associated to the CP instance. This attribute shall be present if the CP instance is PNF external CP. See note 2 and note 4.
pnfExtCpInstancelid	IdentifierInPnf	0..1	Identifier of the PNF external CP instance in the scope of the PNF. This attribute shall be present if the CP instance is PNF external CP. See note 2 and note 4.
nsInstancelid	Identifier	0..1	Identifier of the NS instance associated to the SAP instance. This attribute shall be present if the CP instance is NS SAP. See note 3 and note 4.
nsSapInstancelid	IdentifierInNs	0..1	Identifier of the SAP instance in the scope of the NS instance. This attribute shall be present if the CP instance is NS SAP. See note 3 and note 4.
NOTE 1: For the VNF external CP instance, both vnfInstancelid and vnfExtCpInstancelid shall be present as a pair. NOTE 2: For the PNF external CP instance, both pnfInfold and PnfExtCpInstancelid shall be present as a pair. NOTE 3: For the SAP instance, both nsInstancelid and nsSapInstancelid shall be present as a pair. NOTE 4: One pair of identifiers (VNF external CP, PNF external CP or SAP) shall be present.			

6.5.3.57 Type: VnfInstance

This type represents a VNF instance. It shall comply with the provisions defined in Table 6.5.3.57-1.

NOTE: Clause B.3.2 of ETSI GS NFV-SOL 003 [4] provides examples illustrating the relationship among the different run-time information elements (CP, VL and link ports) used to represent the connectivity of a VNF.

Table 6.5.3.57-1: Definition of the VnfInstance data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of the VNF instance.
vnfInstanceName	String	0..1	Name of the VNF instance. Modifications to this attribute can be requested using the "ModifyVnfInfoData" structure.
vnfInstanceDescription	String	0..1	Human-readable description of the VNF instance. Modifications to this attribute can be requested using the "ModifyVnfInfoData" structure.
vnfdId	Identifier	1	Identifier of the VNFD on which the VNF instance is based. Modifications to this attribute can be requested using the "ModifyVnfInfoData" structure. See note 1.
vnfProvider	String	1	Provider of the VNF and the VNFD. The value is copied from the VNFD.
vnfProductName	String	1	Name to identify the VNF Product. The value is copied from the VNFD.
vnfSoftwareVersion	Version	1	Software version of the VNF. The value is copied from the VNFD.

Attribute name	Data type	Cardinality	Description
vnfdVersion	Version	1	Identifies the version of the VNFD. The value is copied from the VNFD.
vnfPkgId	Identifier	1	Identifier of information held by the NFVO about the specific VNF package on which the VNF is based. This identifier has been allocated by the NFVO.
vnfConfigurableProperties	KeyValuePairs	0..1	<p>Additional VNF-specific attributes that provide the current values of the configurable properties of the VNF instance.</p> <p>These attributes represent values that are stored persistently in the VnfInstance structure and that correspond to configuration parameters of the VNF instance.</p> <p>Modifying these attributes affects the configuration of the VNF instance either directly (if the VNF instance is in INSTANTIATED state at the time of the modification) or as part of the subsequent VNF instantiation operation (if the VNF instance is in NOT_INSTANTIATED state at the time of the modification).</p> <p>Configurable properties referred in these attributes are declared in the VNFD (see note 2 and note 3).</p> <p>These configurable properties include the following standard attributes, which are declared in the VNFD if auto-scaling and/or auto-healing are supported by the VNF:</p> <ul style="list-style-type: none"> - isAutoscaleEnabled: If present, the VNF supports auto-scaling. If set to true, auto-scaling is currently enabled. If set to false, auto-scaling is currently disabled. - isAutohealEnabled: If present, the VNF supports auto-healing. If set to true, auto-healing is currently enabled. If set to false, auto-healing is currently disabled. <p>Modifications to these attributes can be requested using the "ModifyVnfInfoData" structure.</p>
vimId	Identifier	0..N	Identifier of a VIM that manages resources for the VNF instance.
instantiationState	Enum (inlined)	1	<p>The instantiation state of the VNF.</p> <p>Permitted values: NOT_INSTANTIATED: The VNF instance is terminated or not instantiated. INSTANTIATED: The VNF instance is instantiated.</p>
instantiatedVnfInfo	Structure (inlined)	0..1	<p>Information specific to an instantiated VNF instance.</p> <p>This attribute shall be present if the instantiateState attribute value is INSTANTIATED.</p>
>flavourId	IdentifierInVnfd	1	Identifier of the VNF deployment flavour applied to this VNF instance.
>vnfState	VnfOperationalStateType	1	State of the VNF instance.
>scaleStatus	VnfScaleInfo	0..N	Scale status of the VNF, one entry per aspect. Represents for every scaling aspect how "big" the VNF has been scaled w.r.t. that aspect.

Attribute name	Data type	Cardinality	Description
			<p>This attribute shall be present if the VNF supports scaling.</p> <p>See clause B.2 of ETSI GS NFV-SOL 003 [4] for an explanation of VNF scaling.</p>
>maxScaleLevels	VnfScaleInfo	0..N	<p>Maximum allowed scale levels of the VNF, one entry per aspect.</p> <p>This attribute shall be present if the VNF supports scaling.</p>
>extCplInfo	VnfExtCplInfo	1..N	Information about the external CPs exposed by the VNF instance.
>extVirtualLinkInfo	ExtVirtualLinkInfo	0..N	Information about the external VLs the VNF instance is connected to.
>extManagedVirtualLinkInfo	ExtManagedVirtualLinkInfo	0..N	Information about the externally-managed internal VLs of the VNF instance.
>monitoringParameters	VnfMonitoringParameter	0..N	Performance metrics tracked by the VNFM (e.g. for auto-scaling purposes) as identified by the VNF provider in the VNFD.
>localizationLanguage	String	0..1	<p>Information about localization language of the VNF (includes e.g. strings in the VNFD).</p> <p>The localization languages supported by a VNF can be declared in the VNFD, and localization language selection can take place at instantiation time.</p> <p>The value shall comply with the format defined in IETF RFC 5646 [9].</p>
>vnfcResourceInfo	VnfcResourceInfo	0..N	Information about the virtualised compute and storage resources used by the VNFCs of the VNF instance.
>vnfVirtualLinkResourceInfo	VnfVirtualLinkResourceInfo	0..N	Information about the virtualised network resources used by the VLs of the VNF instance.
>virtualStorageResourceInfo	VirtualStorageResourceInfo	0..N	Information about the virtualised storage resources used as storage for the VNF instance.
metadata	KeyValuePairs	0..1	<p>Additional VNF-specific attributes that provide metadata describing the VNF instance.</p> <p>These attributes represent values that are stored persistently in the VnfInstance structure for consumption by functional blocks that invoke the VNF lifecycle management interface. They are not consumed by the VNFM, or the lifecycle management scripts.</p> <p>Modifying the values of these attributes has no effect on the VNF instance, it only affects the information represented in the VnfInstance structure.</p> <p>Metadata that the VNF provider foresees are expected to be declared in the VNFD (see note 2).</p> <p>Modifications to these attributes can be requested using the "ModifyVnfInfoData" structure.</p>

Attribute name	Data type	Cardinality	Description
extensions	KeyValuePairs	0..1	<p>Additional VNF-specific attributes that affect the lifecycle management of this VNF instance. These attributes represent values that are stored persistently in the VnfInstance structure for consumption by the VNFM or by the lifecycle management scripts during the execution of VNF lifecycle management operations.</p> <p>Modifying the values of these attributes has no direct effect on the VNF instance; however, the modified attribute values can be considered during subsequent VNF lifecycle management operations, which means that the modified values can indirectly affect the configuration of the VNF instance.</p> <p>All extensions that are allowed for the VNF are declared in the VNFD.</p> <p>Modifications to these attributes can be requested using the "ModifyVnfInfoData" structure.</p>
<p>NOTE 1: Modifying the value of this attribute shall not be performed when conflicts exist between the previous and the newly referred VNF package, i.e. when the new VNFD is changed with respect to the previous VNFD in other aspects than merely referencing to other VNF software images. In order to avoid misalignment of the VnfInstance with the current VNF's on-boarded VNF Package, the values of attributes in the VnfInstance that have corresponding attributes in the VNFD shall be kept in sync with the values in the VNFD.</p> <p>NOTE 2: ETSI GS NFV-SOL 001 [i.1] specifies the structure and format of the VNFD based on TOSCA specifications.</p> <p>NOTE 3: VNF configurable properties are sometimes also referred to as configuration parameters applicable to a VNF. Some of these are set prior to instantiation and cannot be modified if the VNF is instantiated, some are set prior to instantiation (are part of initial configuration) and can be modified later, and others can be set only after instantiation. The applicability of certain configuration may depend on the VNF and the required operation of the VNF at a certain point in time.</p>			

6.5.3.58 Type: CpProtocollInfo

This type describes the protocol layer(s) that a CP or SAP uses together with protocol-related information, like addresses. It shall comply with the provisions defined in Table 6.5.3.58-1.

Table 6.5.3.58-1: Definition of the CpProtocollInfo data type

Attribute name	Data type	Cardinality	Description
layerProtocol	Enum (inlined)	1	<p>The identifier of layer(s) and protocol(s) associated to the network address information.</p> <p>Permitted values: IP_OVER_ETHERNET</p> <p>See note.</p>
ipOverEthernet	IpOverEthernetAddressInfo	0..1	<p>IP addresses over Ethernet to assign to the CP or SAP instance. Shall be present if layerProtocol is equal to "IP_OVER_ETHERNET", and shall be absent otherwise.</p>
<p>NOTE: This attribute allows to signal the addition of further types of layer and protocol in future versions of the present document in a backwards-compatible way. In the current version of the present document, only IP over Ethernet is supported.</p>			

6.5.3.59 Type: ExtManagedVirtualLinkInfo

This type provides information about an externally-managed virtual link for VNFs. It shall comply with the provisions defined in Table 6.5.3.59-1.

Table 6.5.3.59-1: Definition of the ExtManagedVirtualLinkInfo data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of the externally-managed internal VL and the related externally-managed VL information instance.
vnfVirtualLinkDescId	IdentifierInVnfd	1	Identifier of the VNF Virtual Link Descriptor (VLD) in the VNFD.
networkResource	ResourceHandle	1	Reference to the VirtualNetwork resource.
vnfLinkPorts	VnfLinkPortInfo	0..N	Link ports of this VL.

6.5.3.60 Type: VnfcResourceInfo

This type represents the information on virtualised compute and storage resources used by a VNFC in a VNF instance. It shall comply with the provisions defined in Table 6.5.3.60-1.

Table 6.5.3.60-1: Definition of the VnfcResourceInfo data type

Attribute name	Data type	Cardinality	Description
id	IdentifierInVnf	1	Identifier of this VnfcResourceInfo instance.
vduld	IdentifierInVnfd	1	Reference to the applicable VDU in the VNFD. See note 1.
computeResource	ResourceHandle	1	Reference to the VirtualCompute resource.
storageResourceIds	IdentifierInVnf	0..N	References to the VirtualStorage resources. The value refers to a VirtualStorageResourceInfo item in the VnfInstance.
reservationId	Identifier	0..1	The reservation identifier applicable to the resource. It shall be present when an applicable reservation exists.
vnfcCpInfo	Structure (inlined)	0..N	CPs of the VNFC instance. Shall be present when that particular CP of the VNFC instance is exposed as an external CP of the VNF instance or is connected to an external CP of the VNF instance. See note 2. May be present otherwise.
>id	IdentifierInVnf	1	Identifier of this VNFC CP instance and the associated array entry.
>cpId	IdentifierInVnfd	1	Identifier of the VDU CPD, cpId, in the VNFD. See note 1.
>vnfExtCpId	IdentifierInVnf	0..1	Identifier of the related external CP. Shall be present when the VNFC CP is exposed as an external CP of the VNF instance or connected to an external CP of the VNF instance (see note 2) and shall be absent otherwise.
>cpProtocolInfo	CpProtocolInfo	0..N	Network protocol information for this CP. May be omitted if the VNFC CP is exposed as an external CP. See note 3.
>vnfLinkPortId	IdentifierInVnf	0..1	Identifier of the "vnfLinkPortInfo" structure in the "VnfVirtualLinkResourceInfo" structure. Shall be present if the CP is associated to a link port on an internal VL of the VNF instance and shall be absent otherwise.
>metadata	KeyValuePairs	0..1	Metadata about this CP.
metadata	KeyValuePairs	0..1	Metadata about this resource.
NOTE 1: ETSI GS NFV-SOL 001 [i.1] specifies the structure and format of the VNFD based on TOSCA specifications.			
NOTE 2: A VNFC CP is "connected to" an external CP if the VNFC CP is connected to an internal VL that exposes an external CP. A VNFC CP is "exposed as" an external CP if it is connected directly to an external VL.			
NOTE 3: The information can be omitted because it is already available as part of the external CP information.			

6.5.3.61 Type: VnfVirtualLinkResourceInfo

This type represents the information that allows addressing a virtualised resource that is used by an internal VL instance in a VNF instance. It shall comply with the provisions defined in Table 6.5.3.61-1.

Table 6.5.3.61-1: Definition of the VnfVirtualLinkResourceInfo data type

Attribute name	Data type	Cardinality	Description
id	IdentifierInVnf	1	Identifier of this VnfVirtualLinkResourceInfo instance.
vnfVirtualLinkDescId	IdentifierInVnfd	1	Identifier of the VNF Virtual Link Descriptor (VLD) in the VNFD.
networkResource	ResourceHandle	1	Reference to the VirtualNetwork resource.
reservationId	Identifier	0..1	The reservation identifier applicable to the resource. It shall be present when an applicable reservation exists.
vnfLinkPorts	VnfLinkPortInfo	0..N	Links ports of this VL. Shall be present when the linkPort is used for external connectivity by the VNF (refer to VnfLinkPortInfo). May be present otherwise.
metadata	KeyValuePairs	0..1	Metadata about this resource.

6.5.3.62 Type: ExtVirtualLinkInfo

This type represents information about an VNF external VLs. It shall comply with the provisions defined in Table 6.5.3.62-1.

Table 6.5.3.62-1: Definition of the ExtVirtualLinkInfo data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of the external VL and the related external VL information instance.
resourceHandle	ResourceHandle	1	Reference to the resource realizing this VL.
extLinkPorts	ExtLinkPortInfo	0..N	Link ports of this VL.

6.5.3.63 Type: ExtLinkPortInfo

This type represents information about a link port of an external VL, i.e. a port providing connectivity for the VNF to an NS VL. It shall comply with the provisions defined in Table 6.5.3.63-1.

Table 6.5.3.63-1: Definition of the ExtLinkPortInfo data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this link port as provided by the entity that has created the link port.
resourceHandle	ResourceHandle	1	Reference to the virtualised resource realizing this link port.
cpInstanceId	IdentifierInVnf	0..1	Identifier of the external CP of the VNF connected to this link port. There shall be at most one link port associated with any external connection point instance. The value refers to an "extCpInfo" item in the VnfInstance.

6.5.3.64 Type: VnfLinkPortInfo

This type represents a link port of an internal VL of a VNF. It shall comply with the provisions defined in Table 6.5.3.64-1.

Table 6.5.3.64-1: Definition of the VnfLinkPortInfo data type

Attribute name	Data type	Cardinality	Description
id	IdentifierInVnf	1	Identifier of this link port as provided by the entity that has created the link port.
resourceHandle	ResourceHandle	1	Reference to the virtualised resource realizing this link port.
cplInstanceIcd	IdentifierInVnf	0..1	<p>When the link port is used for external connectivity by the VNF, this attribute represents the identifier of the external CP associated with this link port.</p> <p>When the link port is used for internal connectivity in the VNF, this attribute represents the identifier of the VNFC CP to be connected to this link port.</p> <p>Shall be present when the link port is used for external connectivity by the VNF.</p> <p>May be present if used to reference a VNFC CP instance.</p> <p>There shall be at most one link port associated with any external connection point instance or internal connection point (i.e. VNFC CP) instance.</p> <p>The value refers to an "extCpInfo" item in the VnfInstance or a "vnfcCpInfo" item of a "vnfcResouceInfo" item in the VnfInstance.</p>
cplInstanceType	Enum (inlined)	0..1	<p>Type of the CP instance that is identified by cplInstanceIcd.</p> <p>Shall be present if "cplInstanceIcd" is present, and shall be absent otherwise.</p> <p>Permitted values:</p> <ul style="list-style-type: none"> - VNFC_CP: The link port is connected to a VNFC CP. - EXT_CP: The link port is associated to an external CP.

6.5.3.65 Type: VnffgInfo

This type specifies the information about a VNFFG instance. It shall comply with the provisions defined in Table 6.5.3.65-1.

Table 6.5.3.65-1: Definition of the VnffgInfo data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this VNFFG instance.
vnffgdId	IdentifierInNsd	1	Identifier of the VNFFGD in the NSD.
vnfInstanceIcd	Identifier	1..N	Identifier(s) of the constituent VNF instance(s) of this VNFFG instance.
pnfInfoIcd	Identifier	0..N	Identifier(s) of the constituent PNF instance(s) of this VNFFG instance.
nsVirtualLinkInfoIcd	IdentifierInNs	1..N	Identifier(s) of the constituent VL instance(s) of this VNFFG instance.
nsCpHandle	NsCpHandle	1..N	Identifiers of the CP instances attached to the constituent VNFs and PNFs or the SAP instances of the VNFFG. See note.
nfpInfo	NfpInfo	1..N	Information on the NFP instances.
NOTE: It indicates an exhaustive list of all the CP instances and SAP instances of the VNFFG.			

6.5.3.66 Type: NfpInfo

This type represents an NFP instance. It shall comply with the provisions defined in Table 6.5.3.66-1.

Table 6.5.3.66-1: Definition of the NfpInfo data type

Attribute name	Data type	Cardinality	Description
id	IdentifierInNs	1	Identifier of this NFP instance.
nfpDId	IdentifierInNsd	0..1	Identifier of the NFPD used to instantiate this NFP instance. It shall be present if the NFP instance is instantiated from the NFPD.
nfpName	String	0..1	Human readable name for the NFP instance.
description	String	0..1	Human readable description for the NFP instance.
cpGroup	CpGroupInfo	1..N	Group(s) of CPs and/or SAPs which the NFP passes through. See note.
totalCp	Integer	0..1	Total number of CP and SAP instances in this NFP instance.
nfpRule	NfpRule	1	NFP classification and selection rule.
nfpState	Enum (inlined)	1	The state of the NFP instance. Permitted values: ENABLED: The NFP instance is enabled. DISABLED: The NFP instance is disabled.
NOTE: When multiple identifiers are included, the position of the identifier in the CpGroup data type specifies the position of the group in the path.			

6.5.3.67 Type: SapInfo

This type represents an SAP instance. It shall comply with the provisions defined in Table 6.5.3.67-1.

Table 6.5.3.67-1: Definition of the SapInfo data type

Attribute name	Data type	Cardinality	Description
id	IdentifierInNs	1	Identifier of the SAP instance.
sapDId	IdentifierInNsd	1	Identifier of the SAPD in the NSD.
sapName	String	1	Human readable name for the SAP instance.
description	String	1	Human readable description for the SAP instance.
sapProtocolInfo	CpProtocolInfo	1..N	Network protocol information for this SAP.

6.5.3.68 Type: NsMonitoringParameter

This type represents a monitoring parameter that is tracked by the NFVO, for example, for auto-scaling purposes. It shall comply with the provisions defined in Table 6.5.3.68-1.

Table 6.5.3.68-1: Definition of the NsMonitoringParameter data type

Attribute name	Data type	Cardinality	Description
id	IdentifierInNsd	1	Identifier of the monitoring parameter defined in the NSD.
name	String	0..1	Human readable name of the monitoring parameter, as defined in the NSD.
performanceMetric	String	1	Performance metric that is monitored. This attribute shall contain the related "Measurement Name" value as defined in clause 7.2 of ETSI GS NFV-IFA 027 [15].

6.5.3.69 Type: VnfMonitoringParameter

This type represents a monitoring parameter that is tracked by the VNFM, for example, for auto-scaling purposes. It shall comply with the provisions defined in Table 6.5.3.69-1.

Table 6.5.3.69-1: Definition of the VnfMonitoringParameter data type

Attribute name	Data type	Cardinality	Description
id	IdentifierInVnfd	1	Identifier of the monitoring parameter defined in the VNFD.
name	String	0..1	Human readable name of the monitoring parameter, as defined in the VNFD.
performanceMetric	String	1	Performance metric that is monitored. This attribute shall contain the related "Measurement Name" value as defined in clause 7.2 of ETSI GS NFV-IFA 027 [15].

6.5.3.70 Type: VnfExtCplInfo

This type represents information about an external CP of a VNF. It shall comply with the provisions defined in Table 6.5.3.70-1.

Table 6.5.3.70-1: Definition of the VnfExtCplInfo data type

Attribute name	Data type	Cardinality	Description
id	IdentifierInVnf	1	Identifier of the external CP instance and the related information instance.
cpdId	IdentifierInVnfd	1	Identifier of the external CPD, VnfExtCpd, in the VNFD.
cpProtocolInfo	CpProtocolInfo	1..N	Network protocol information for this CP.
extLinkPortId	Identifier	0..1	Identifier of the "extLinkPortInfo" structure inside the "extVirtualLinkInfo" structure. Shall be present if the CP is associated to a link port.
metadata	KeyValuePairs	0..1	Metadata about this external CP.
associatedVnfcCpld	IdentifierInVnf	0..1	Identifier of the "vnfcCplInfo" structure in "VnfcResourceInfo" structure that represents the VNFC CP which is exposed by this external CP instance. Shall be present in case this CP instance maps to a VNFC CP. See note.
associatedVnfVirtualLinkId	IdentifierInVnf	0..1	Identifier of the "VnfVirtualLinkResourceInfo" structure that represents the internal VL, which is exposed by this external CP instance. Shall be present in case this CP instance maps to an internal VL. See note.
NOTE: The attributes "associatedVnfcCpld" and "associatedVnfVirtualLinkId" are mutually exclusive. One and only one shall be present.			

6.5.3.71 Type: CpGroupInfo

This type represents describes a group of CPs and/or SAPs pairs associated to the same position in an NFP. It shall comply with the provisions defined in Table 6.5.3.71-1.

Table 6.5.3.71-1: Definition of the CpGroupInfo data type

Attribute name	Data type	Cardinality	Description
cpPairInfo	CpPairInfo	1..N	One or more pair(s) of ingress and egress CPs or SAPs which the NFP passes by. See note.
forwardingBehaviour	Enum (inlined)	0..1	Identifies a rule to apply to forward traffic to the ingress CPs or SAPs of the group. Permitted values: ALL = Traffic flows shall be forwarded simultaneously to all CPs or SAPs of the group. LB = Traffic flows shall be forwarded to one CP or SAP of the group selected based on a load-balancing algorithm.
forwardingBehaviourInputParameters	ForwardingBehaviourInputParameters	0..1	Provides input parameters to configure the forwarding behaviour (e.g. identifies a load balancing algorithm and criteria).
NOTE: All CP or SAP pairs in a group shall be instantiated from connection point descriptors or service access point descriptors referenced in the corresponding NfpPositionDesc.			

6.5.3.72 Type: CpPairInfo

This type represents describes a pair of ingress and egress CPs or SAPs which the NFP passes by. It shall comply with the provisions defined in Table 6.5.3.72-1.

Table 6.5.3.72-1: Definition of the CpPairInfo data type

Attribute name	Data type	Cardinality	Description
vnfExtCplds	IdentifierInVnf	0..2	Identifier(s) of the VNF CP(s) which form the pair. See note 1 and note 2.
pnfExtCplds	IdentifierInPnf	0..2	Identifier(s) of the PNF CP(s) which form the pair. See note 1 and note 2.
saplds	IdentifierInNs	0..2	Identifier(s) of the SAP(s) which form the pair. See note 1 and note 2.
NOTE 1: The presence of a single vnfExpCpld, pnfExtCpld, or sapld occurrence indicates that the CP or SAP is used both as an ingress and egress port at a particular NFP position.			
NOTE 2: Only one of these three attributes shall be present.			

6.5.3.73 Type: ForwardingBehaviourInputParameters

This type represents provides input parameters to configure the forwarding behaviour. It shall comply with the provisions defined in Table 6.5.3.73-1.

Table 6.5.3.73-1: Definition of the ForwardingBehaviourInputParameters data type

Attribute name	Data type	Cardinality	Description
algorithmName	Enum (Inlined)	0..1	May be included if forwarding behaviour is equal to LB. Shall not be included otherwise. Permitted values: <ul style="list-style-type: none"> - ROUND_ROBIN - LEAST_CONNECTION - LEAST_TRAFFIC - LEAST_RESPONSE_TIME - CHAINED_FAILOVER - SOURCE_IP_HASH - SOURCE_MAC_HASH
algorithmWeights	Integer	0..N	Percentage of messages sent to a CP instance. May be included if applicable to the algorithm. See note 1 and note 2.
NOTE 1: If applicable to the algorithm but not provided, default values determined by the VIM or NFVI are expected to be used.			
NOTE 2: Weight applies to the CP instances in the order they have been created.			

6.5.4 Referenced simple data types and enumerations

6.5.4.1 Introduction

This clause defines simple data types that can be referenced from data structures defined in the previous clauses.

6.5.4.2 Simple data types

No particular simple data types are defined for this interface, in addition to those defined in clause 4.4.2.

6.5.4.3 Enumeration: NsLcmOpType

The enumeration NsLcmOpType represents those lifecycle operations that trigger an NS lifecycle management operation occurrence notification. It shall comply with the provisions defined in Table 6.5.4.3-1.

Table 6.5.4.3-1: Enumeration NsLcmOpType

Enumeration value	Description
INSTANTIATE	Represents the "Instantiate NS" LCM operation.
SCALE	Represents the "Scale NS" LCM operation.
UPDATE	Represents the "Update NS" LCM operation.
TERMINATE	Represents the "Terminate NS" LCM operation.
HEAL	Represents the "Heal NS" LCM operation.

6.5.4.4 Enumeration: NsLcmOperationStateType

The enumeration NsLcmOperationStateType shall comply with the provisions defined in Table 6.5.4.4-1. More information of the meaning of the states can be found in clause 6.6.2.2.

Table 6.5.4.4-1: Enumeration NsLcmOperationStateType

Enumeration value	Description
PROCESSING	The LCM operation is currently in execution.
COMPLETED	The LCM operation has been completed successfully.
PARTIALLY_COMPLETED	The LCM operation has been partially completed with acceptable errors.
FAILED_TEMP	The LCM operation has failed and execution has stopped, but the execution of the operation is not considered to be closed.
FAILED	The LCM operation has failed and it cannot be retried or rolled back, as it is determined that such action will not succeed.
ROLLING_BACK	The LCM operation is currently being rolled back.
ROLLED_BACK	The LCM operation has been successfully rolled back, i.e. The state of the NS prior to the original operation invocation has been restored as closely as possible.

6.5.4.5 Enumeration: NsComponentType

The enumeration NsComponentType represents the NS component type. It shall comply with the provisions defined in Table 6.5.4.5-1.

Table 6.5.4.5-1: Enumeration NsComponentType

Enumeration value	Description
VNF	Represents the impacted NS component is a VNF.
PNF	Represents the impacted NS component is a PNF.
NS	Represents the impacted NS component is a nested NS.

6.5.4.6 Enumeration: LcmOpNameForChangeNotificationType

The enumeration LcmOpNameForChangeNotificationType represents the name of the lifecycle operation that impacts the NS component and trigger an NS change notification. It shall comply with the provisions defined in Table 6.5.4.6-1.

Table 6.5.4.6-1: Enumeration LcmOpNameForChangeNotificationType

Enumeration value	Description
VNF_INSTANTIATE	Represents the "Instantiate VNF" LCM operation.
VNF_SCALE	Represents the "Scale VNF" LCM operation.
VNF_SCALE_TO_LEVEL	Represents the "Scale VNF to Level" LCM operation.
VNF_CHANGE_FLAVOUR	Represents the "Change VNF Flavour" LCM operation.
VNF_TERMINATE	Represents the "Terminate VNF" LCM operation.
VNF_HEAL	Represents the "Heal VNF" LCM operation.
VNF_OPERATE	Represents the "Operate VNF" LCM operation.
VNF_CHANGE_EXT_CONN	Represents the "Change external VNF connectivity" LCM operation.
VNF_MODIFY_INFO	Represents the "Modify VNF Information" LCM operation.
NS_INSTANTIATE	Represents the "Instantiate NS" LCM operation.
NS_SCALE	Represents the "Scale NS" LCM operation.
NS_UPDATE	Represents the "Update NS" LCM operation.
NS_TERMINATE	Represents the "Terminate NS" LCM operation.
NS_HEAL	Represents the "Heal NS" LCM operation.

6.5.4.7 Enumeration: LcmOpOccStatusForChangeNotificationType

The enumeration LcmOpOccStatusForChangeNotificationType represents the status of the lifecycle management operation occurrence that impacts the NS component and triggers an NS change notification. It shall comply with the provisions defined in Table 6.5.4.7-1.

Table 6.5.4.7-1: Enumeration LcmOpOccStatusForChangeNotificationType

Enumeration value	Description
START	The impact on the NS component is identified.
COMPLETED	The impact on the NS component stops and related lifecycle operation completes successfully.
PARTIALLY_COMPLETED	The impact on the NS component stops and related lifecycle operation partially completes. Inconsistency state may exist on the NS component.
FAILED	The impact on the NS component stops and related lifecycle operation fails. Inconsistency state may exist for the NS component.
ROLLED_BACK	The impact on the NS component stops and related lifecycle operation is rolled back.

6.5.4.8 Enumeration: OperationalStates

The enumeration OperationalStates shall comply with the provisions defined in Table 6.5.4.8-1.

Table 6.5.4.8-1: Enumeration OperationalStates

Enumeration value	Description
STARTED	The VNF instance is up and running.
STOPPED	The VNF instance has been shut down.

6.5.4.9 Enumeration: StopType

The enumeration StopType shall comply with the provisions defined in Table 6.5.4.9-1.

Table 6.5.4.9-1: Enumeration StopType

Enumeration value	Description
FORCEFUL	The VNFM will stop the VNF immediately after accepting the request.
GRACEFUL	The VNFM will first arrange to take the VNF out of service after accepting the request. Once that operation is successful or once the timer value specified in the "gracefulStopTimeout" attribute expires, the VNFM will stop the VNF.

6.5.4.10 Enumeration: CancelModeType

The enumeration CancelModeType defines the valid modes of cancelling an NS LCM operation occurrence. It shall comply with the provisions defined in Table 6.5.4.10-1.

Table 6.5.4.10-1: Enumeration CancelModeType

Enumeration value	Description
GRACEFUL	The NFVO shall not start any new VNF lifecycle management and resource management operation, and shall wait for the ongoing VNF lifecycle management and resource management operations in the underlying system, typically the VNFM and VIM, to finish execution or to time out. After that, the NFVO shall put the operation occurrence into the FAILED_TEMP state.
FORCEFUL	The NFVO shall not start any new VNF lifecycle management and resource management operation, shall cancel the ongoing VNF lifecycle management and resource management operations in the underlying system, typically the VNFM and VIM, and shall wait for the cancellation to finish or to time out. After that, the NFVO shall put the operation occurrence into the FAILED_TEMP state.

6.6 Handling of errors during NS lifecycle management operations

6.6.1 Basic concepts (informative)

6.6.1.1 Motivation

NS lifecycle management operation occurrences can fail. Failure can be caused by multiple reasons, which generally fall into the following categories:

- Transient errors which do not require intervention from a human operator or a higher-layer management entity for resolution, e.g. momentary network outage.
- "Permanent" errors which require such intervention.

It is unreasonable to expect that all errors can be resolved automatically, therefore the possibility of intervention will usually be incorporated in the system design as acknowledged means of error resolution.

6.6.1.2 Failure resolution strategies: Retry, Rollback and Continue

Most transient errors are handled best with a retry mechanism. Retry might happen automatically at the point of failure within the same NS LCM workflow (where it makes sense to limit the number of automatic retries). It is important to strive for designing retry operations that have no unintended side effects from the original invocation of the operation. This is called *idempotent retry*. Idempotent retry can also be used as an on-demand error resolution mechanism (see below) if the original operation failed because of a condition that has been resolved manually by the human operator or by a higher-level management entity, so idempotent retry is suitable for general error resolution in most cases.

However, even if a system is designed with idempotent retry capabilities, eventual success of the operation cannot be guaranteed. In this case, the system needs to decide the error handling strategy, either by a backward action or a forward action. By a backward action, it means the concerned error is not acceptable and permanent. Therefore, the system attempts to resolve the inconsistent state by requesting to roll back the changes made by the operation. By a forward action, it means the concerned error is acceptable and can be fixed later (typically after current operation). Therefore, the system decides to skip the concerned error and continues the operation, e.g. based on policy configuration. Given that, rollback and continue as error handling strategies are also desired to be allowed in the system design.

In many cases, idempotent retry can resolve transient errors and lead to success eventually. Depending on the situation, rollback followed by a repetition of the operation could take longer than a successful retry, as rollback first removes allocated resources and then the repetition of the operation allocates them again, which costs time. Therefore, it often makes sense to perform first idempotent retry, which is followed by either rollback or continue if the retry has failed.

Idempotent retry is meaningful and useful for all operation types. For some operations, rollback is better suited and has a better chance of success. In general, rollback is well-suited for additive operations such as InstantiateNs or scale out, while ill-suited for subtractive ones such as scale in or TerminateNs, or for HealNs. For some operations, continue is better suited if the concerned error is acceptable.

Both rollback and idempotent retry can fail. In that case, the system can be left in an inconsistent state after a failed operation, which requires resolution by a higher-level entity such as the OSS/BSS or human operator.

6.6.1.3 Error handling at NFVO and OSS/BSS

If the NFVO executes an NS LCM workflow and encounters a problem, the following options are possible:

- Stop on first error:
 - Once the NFVO encounters an error, the normal execution of the NS LCM workflow is interrupted, and an error handling procedure is triggered (i.e. automatic retry, automatic rollback, automatic fail, escalate). See the clauses below for description of error handling procedures.
 - It is assumed that all NSs and all NFVOs support "stop on first error".

EXAMPLE 1: OSS/BSS is attempting to instantiate an NS with 10 VNFs. The first 8 VNFs are instantiated successfully, however, an error occurs when attempting to instantiate VNF #9. The NFVO stops execution and chooses which of the error handling options it invokes (note that it even could try multiple options after each other).

- Best Effort:
 - Each time the NFVO encounters an error, it is decided whether the execution of a part or all of the remaining steps of the NS LCM workflow is performed, or whether the execution is interrupted and an error handling procedure is triggered (i.e. automatic retry, automatic rollback, automatic fail, escalate). See the clauses below for description of error handling procedures.
 - Support of "best effort" requires a suitable workflow design.
 - It is therefore assumed that not all NSs and not all NFVOs support "best effort".

EXAMPLE 2: Same example as above. After the error occurs attempting to instantiate VNF #8, the NFVO continues by creating #9 and #10, and then chooses which error handling options it invokes.

The NFVO has the following error handling procedures to react to errors (see clause 6.6.1.2 for general elaboration regarding retry, rollback, and continue):

- Automatic Retry: The NFVO retries (once or more) to continue the execution of the workflow without involving an external entity. Automatic retry of failed parts of the workflow might even be built into the workflow itself. Retry can eventually succeed or fail. Successful retry leads to the NS LCM operation to be reported as successful. Failed retry is typically escalated.
- Automatic Rollback: The NFVO rolls back the NS to the state prior to starting the NS LCM operation without involving an external entity. Rollback can eventually succeed or can fail, preventing the NS from reaching that previous state. Successful rollback leads to the NS LCM operation to be reported as rolled back. Failed rollback is typically escalated.
- Automatic Continue: The NFVO skips the error and continue the NS LCM operation without involving an external entity. Continue can eventually succeed or fail. Successful continue leads to the NS LCM operation to be reported as partially completed. Failed continue is typically escalated or trying other error handling procedures like automatic rollback.
- Escalate: After failed automatic retry/retries, automatic rollback or automatic continue is typically not the first option in most situations, but the error is preferably reported to the OSS/BSS for further resolution. The same applies if no automatic error resolution was attempted by the NFVO, or if automatic rollback has failed or if automatic continue is not appropriate (e.g. based on policy configuration). This is done by sending an NS LCM operation occurrence notification.
- Unresolvable Error: The NFVO determines that the operation has failed and definitely cannot be recovered (e.g. if no retry, no continue, and no rollback is possible), and that escalating the error to the OSS/BSS will have no chance to lead to a resolution either. In this case, the NFVO would report that the operation has terminally failed. After that, other means of resolution can be attempted, such as the invocation of Heal NS, or manual procedures using the GUI of the NFVO or VIM to release stranded resources.

The OSS/BSS has the following error handling procedures to react to error reports from the NFVO:

- On-demand retry: After the NFVO has reported the error to the OSS/BSS, the OSS/BSS or the human operator takes steps to resolve the situation that has led to the occurrence of the error. Subsequently, the retry of the operation is triggered towards the NFVO by the OSS/BSS via the NS LCM interface.
- On-demand rollback: After the NFVO has reported the error to the OSS/BSS, and after the OSS/BSS or the human operator has decided to roll back the operation, the rollback of the operation is triggered towards the NFVO by the OSS/BSS via the NS LCM interface.
- On-demand continue: After the NFVO has reported the error to the OSS/BSS, and after the OSS/BSS or the human operator has decided to continue the operation, the continue of the operation is triggered towards the NFVO by the OSS/BSS via the NS LCM interface.

- Fail: After the NFVO has reported the error to the OSS/BSS, and after the OSS/BSS or the human operator has determined that neither on-demand retry nor on-demand rollback will fix the error, or on-demand continue is not appropriate, the NS LCM operation can be declared as terminally failed towards the NFVO by the OSS/BSS via the NS LCM interface. After that, other means of resolution can be attempted, such as the invocation of HealNs, or manual procedures using the GUI of the NFVO or VIM to release stranded resources.

6.6.2 States and state transitions of an NS lifecycle management operation occurrence

6.6.2.1 General

An NS lifecycle management operation occurrence supports a number of states and error handling operations. The states and state transitions that shall be supported by the NFVO are shown in Figure 6.6.2.1-1. Transitions labelled with underlined text represent error handling operations; other transitions represent conditions.

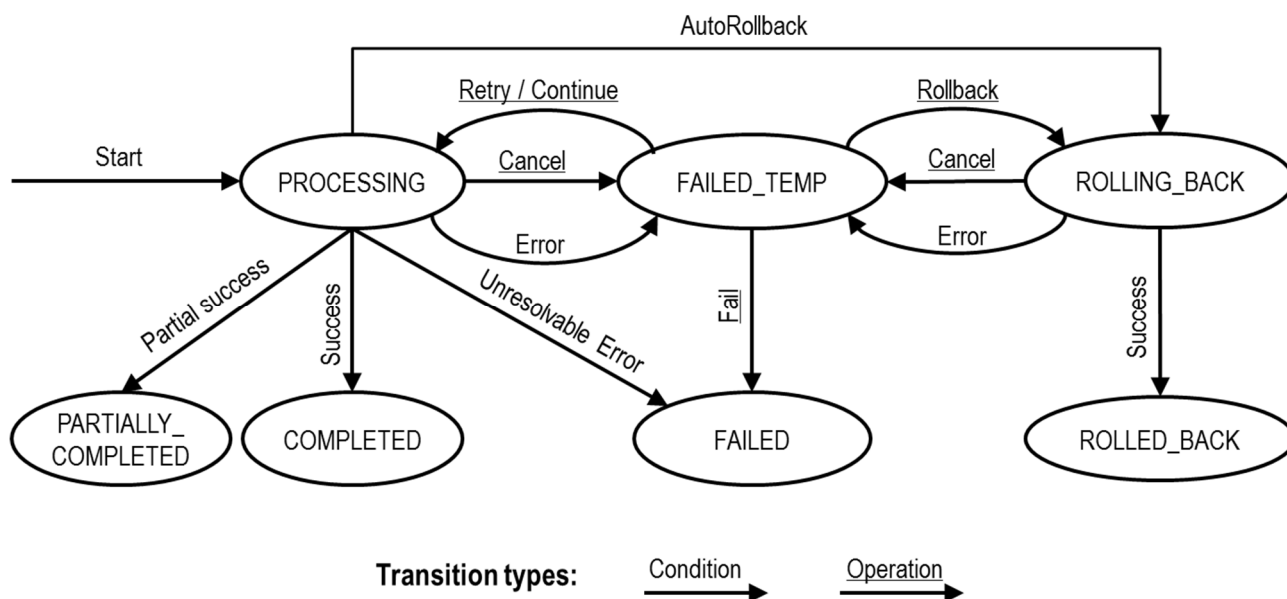


Figure 6.6.2.1-1: States of an NS lifecycle management operation occurrence

6.6.2.2 States of an NS lifecycle management operation occurrence

At each time, an NS lifecycle management operation occurrence is in one of the following states. There are transitional states (states from which a different state can be reached) and terminal states (states from which no other state can be reached; i.e. the state of an NS lifecycle management operation occurrence in a terminal state cannot change anymore).

PROCESSING: The NS LCM operation is currently in execution. This state has the following characteristics:

- This is the initial state for any NS operation.
- This is a transient state.
- This state may block other NS LCM operations from being executed on the same NS instance (up to NS and NFVO implementation).
- The operations "Retry", "Continue", "Fail", and "Rollback" shall not be permitted to be invoked for an operation that is in this state.

- All failures of procedures executed by the NFVO as part of the NS LCM operation while in "PROCESSING" state should result by default in transiting to FAILED_TEMP, with the following three alternative options:
 - If a failure occurs in the "PROCESSING" state from which the NFVO knows that the NS instance can be brought into a consistent state by immediately rolling back the operation, the NS lifecycle management operation occurrence may transit directly into the "ROLLING_BACK" state ("AutoRollback").
 - If a failure occurs in the "PROCESSING" state from which the NFVO knows that it can neither be fixed by retrying nor be rolled back nor be skipped by continuing, the NS lifecycle management operation occurrence may transit directly into the "FAILED" state ("Unresolvable Error").
 - If a failure occurs in the "PROCESSING" state from which the NFVO knows that the failure is acceptable and continues the NS LCM operation till it finishes, the NS lifecycle management operation occurrence may transit directly into the "PARTIALLY_COMPLETED" state ("Partial success").
- If a "cancel" request was issued during the operation is in "PROCESSING" state, processing will be cancelled but this might not be immediate. This is represented by a flag in the data model that indicates there is a pending "cancel" request for this state. Upon successful cancellation, the NS lifecycle management operation occurrence shall transit into the "FAILED_TEMP" state.

COMPLETED: The operation has completed successfully. This is a terminal state.

PARTIALLY COMPLETED: The operation has completed partially, i.e. with acceptable errors. This state has the following characteristics:

- This is a terminal state.
- Such an operation state is typically the result of an automatic continue operation inside the NFVO or an on-demand continue operation from a higher layer management entity (i.e. OSS/BSS) for a given error.
- The result of the NS LCM operation (the actual resource changes) can show an inconsistent state of the NS. Nevertheless, these changes shall be synchronized between the NFVO and OSS/BSS (by reporting them in the LCCN, and by allowing the OSS/BSS to obtain them on request) in order for other NS LCM operations (e.g. Heal, Terminate, Update) to be guaranteed to work on resources that are known to the OSS/BSS.

The fact that a LCM operation is in "PARTIALLY_COMPLETED" state shall not block other operations from execution on the NS instance by the NFVO. However, the NS instance may itself be in a state that disallows certain operations.

FAILED_TEMP: The operation has failed and execution has stopped, but the execution of the operation is not considered to be closed. This state has the following characteristics:

- This is a transient state.
- This state may block other NS LCM operations from being executed on the same NS instance (enforced by the NFVO, and up to NS and NFVO capabilities).
- Retry and/or rollback and/or continue and/or fail may be invoked for the operation.
- If the NS LCM operation is retried or continued, the NS lifecycle management operation occurrence shall transit into the "PROCESSING" state.
- If the NS LCM operation is rolled back, the NS lifecycle management operation occurrence shall transit into the "ROLLING_BACK" state.
- If the NS LCM operation is marked as "failed", the NS lifecycle management operation occurrence shall transit into the "FAILED" state.
- Operation cancellation and failure to roll back should result in FAILED_TEMP.

FAILED: The operation has failed and it cannot be retried, rolled back, or continued, as it is determined that such action will not succeed. This state has the following characteristics:

- This is a terminal state.

- Such an operation state is typically the result of a decision of a higher layer management entity (i.e. OSS/BSS) or its human operator that an operation in "FAILED_TEMP" state cannot be retried or rolled back or continued ("Fail").
- Such an operation state can also be reached immediately in case of failure of an operation in "PROCESSING" state that can neither be retried, rolled back, nor continued ("Unresolvable Error").
- The result of the NS LCM operation (the actual resource changes) can show an inconsistent state of the NS. Nevertheless, these changes shall be synchronized between the NFVO and OSS/BSS (by reporting them in the LCCN, and by allowing the OSS/BSS to obtain them on request) in order for other NS LCM operations (e.g. Heal, Terminate) to be guaranteed to work on resources that are known to the OSS/BSS.
- The fact that a LCM operation is in "FAILED" state shall not block other operations from execution on the NS instance by the NFVO. However, the NS instance may itself be in a state that disallows certain operations.

ROLLED_BACK: The state of the NS prior to the original operation invocation has been restored as closely as possible. This state has the following characteristics:

- This is a terminal state.
- This may involve recreating some resources that have been deleted by the operation, the recreated resources should be as similar as possible to the deleted ones. Differences between original resources and re-created ones may include a different resource identity, but also different dynamic attributes such as an IP address.

ROLLING_BACK: The NS LCM operation is currently being rolled back. This state has the following characteristics:

- This is a transient state.
- This state may block other NS LCM operations from being executed on the same NS instance (up to NS and NFVO implementation).
- The operations "Retry", "Continue", and "Rollback" shall not be permitted to be invoked for an operation that is in this state.
- If a "Cancel" request was issued during the operation is in "ROLLING_BACK" state, rolling back will be cancelled but this might not be immediate. This is represented by a flag in the data model that indicates there is a pending "Cancel" request for this state. Upon successful cancellation, the NS lifecycle management operation occurrence shall transit into the "FAILED_TEMP" state.
- If a failure occurs during rolling back, the operation should transition to the "FAILED_TEMP" state.
- Upon successful rollback, the NS lifecycle management operation occurrence shall transit into the "ROLLED_BACK" state.

In addition, the following provisions apply to NS lifecycle management operation occurrence notifications:

- The "start" notification (i.e. notificationStatus="START") shall be sent when the operation enters one of states "PROCESSING" and "ROLLING_BACK" from another state, indicating the state entered.
- The "result" notification (i.e. notificationStatus="RESULT") shall be sent when the NS LCM operation occurrence enters one of the error states "FAILED_TEMP", "FAILED", "ROLLED_BACK", indicating the state entered, the error cause and the changes to the NS's resources since the operation was initially started.
- The "result" notification (i.e. notificationStatus="RESULT") shall be sent when the operation enters the success state "COMPLETED" or partial success state "PARTIALLY_COMPLETED", indicating the state entered and the changes to the NS's resources.

Such a notification scheme allows the OSS/BSS to keep in sync with changes to the NS's resources by an ongoing NS LCM operation. If the notification relates to a transient state, further changes can be expected. If the notification relates to a terminal state, no further changes to the NS's resources will be performed by the related NS lifecycle management operation occurrence, and the OSS/BSS can use the information in the notification to synchronize its internal state with the result of the LCM operation. In case of loss of notifications, a query of the resource that represents the NS lifecycle operation occurrence can be used by the OSS/BSS to obtain the same information.

6.6.2.3 Error handling operations that change the state of an NS lifecycle operation

Retry: This operation retries an NS lifecycle operation. It has the following characteristics:

- Execution of "Retry" for an actual NS LCM operation on a particular NS may be supported, depending on characteristics of the NS and the NS LCM operation.
- The operation may be invoked via an interface, or the NFVO may invoke the operation per its own decision.

Rollback: This operation rolls back an NS lifecycle operation. It has the following characteristics:

- Execution of "Rollback" for an actual NS LCM operation on a particular NS may be supported, depending on characteristics of the NS and the NS LCM operation.
- The operation may be invoked via an interface, or the NFVO may invoke the operation per its own decision.

Continue: This operation continues an NS lifecycle operation. It has the following characteristics:

- Execution of "Continue" for an actual NS LCM operation on a particular NS may be supported, depending on characteristics of the NS and the NS LCM operation.
- The operation may be invoked via an interface, or the NFVO may invoke the operation per its own decision.

Fail: This operation transits the NS lifecycle operation occurrence into the terminal "FAILED" state. It has the following characteristics:

- Execution of "Fail" shall be supported for a LCM operation on a particular NS if at least one of following - Retry, Rollback, Continue, or Cancel - is supported for this operation.
- The operation may be invoked via an interface, or the NFVO may invoke the operation per its own decision.

Cancel: This operation cancels an ongoing NS lifecycle management operation, its Retry, Rollback, or Continue. It has the following characteristics:

- Execution of "Cancel" for an actual NS LCM operation on a particular NS may be supported, depending on characteristics of the NS and the NS LCM operation.
- The "Cancel" operation need not have immediate effect, depending on the capabilities of the underlying systems, and the currently executed resource management operation.
- Two modes of cancellation are supported: graceful and forceful:
 - When executing the *graceful* "Cancel" operation, the NFVO will not initiate any new operation towards the underlying systems, will wait until the currently executed operations finish or time out, and will then put the NS lifecycle management operation occurrence into the "FAILED_TEMP" state.
 - When executing the *forceful* "Cancel" operation, the NFVO will cancel all ongoing operations in the underlying systems for which cancellation is supported, will not initiate any new operation towards the underlying systems, will wait for the requested cancellations to finish or time out, and will the put the NS lifecycle management operation occurrence into the "FAILED_TEMP" state.
- Executing "Cancel" can lead to inconsistencies between the information that the NFVO has about the state of the resources of the NS, and their actual state. The probability of such inconsistencies is bigger when using the *forceful* cancellation mode.

6.6.3 Detailed flows

6.6.3.1 Immediate failure

If the NS LCM operation fails immediately, i.e. it returns an HTTP error, then the operation has not started, and no "NS LCM operation occurrence resource" has been created. Also, a "start" lifecycle management operation occurrence notification has not been sent. The operation cannot be retried, but the same operation may be invoked again from the API. The NS instance is not changed by a synchronous failure, so no special error handling is required.

Figure 6.6.3.1-1 illustrates the flow.

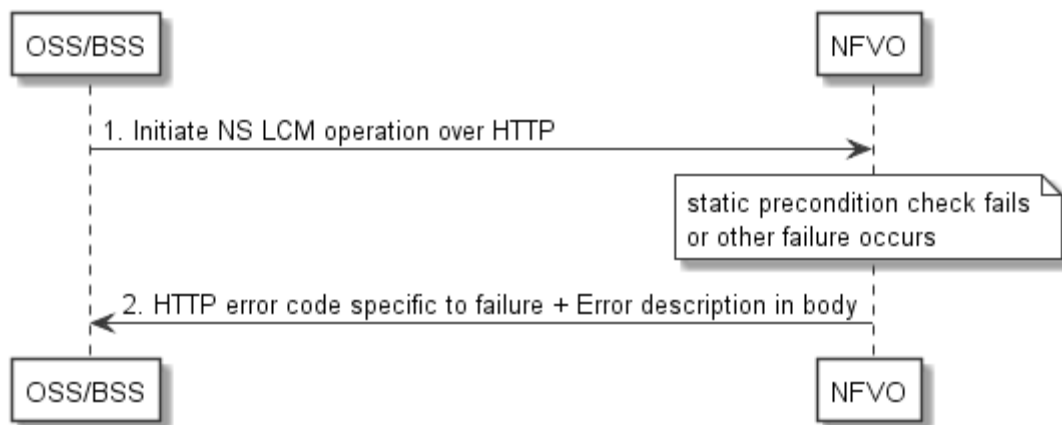


Figure 6.6.3.1-1: Immediate failure of an NS LCM operation

6.6.3.2 Failure during actual NS LCM operation execution

After a failed resource management operation, automatic retry can be invoked by the NFVO itself. These invocations are not visible outside of the NFVO, as the NS LCM operation occurrence stays in "PROCESSING" state during these automatic retries. If these do not resolve the issue, intervention (typically by a human operator) is necessary. For that purpose, the NS LCM operation is set into a temporary failure state, and the OSS/BSS is notified. The human operator performs a root cause analysis and eventually resolves the obstacle. Subsequently, and if supported, the operation can be retried, rolled-back or determined as permanently failed. Figure 6.6.3.2-1 illustrates the possible options.

NOTE 1: Excluding automated rollback which is seen as a rare option.

NOTE 2: Excluding "start" notifications (i.e. notificationStatus="START") for simplification purposes.

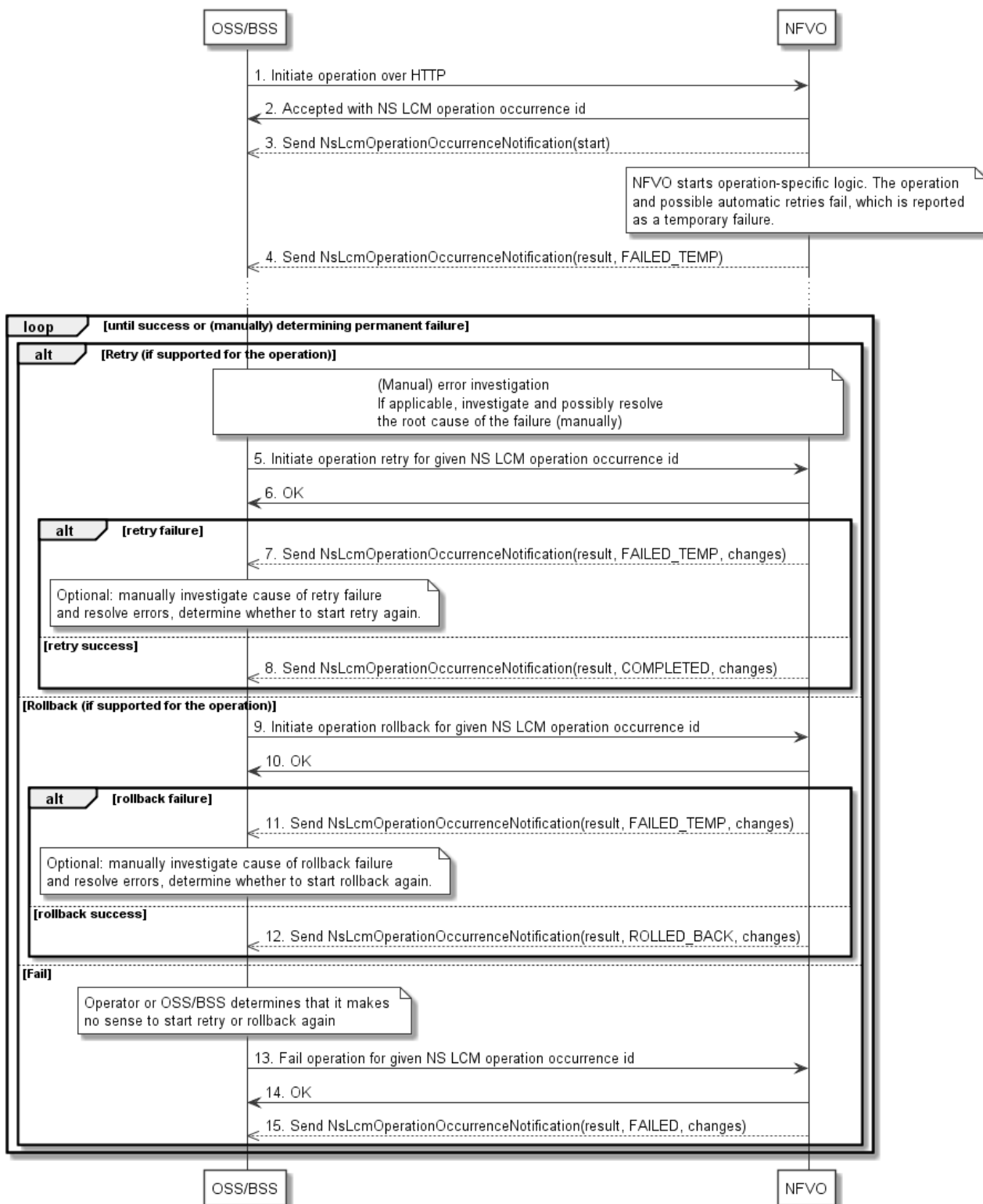


Figure 6.6.3.2-1: Handling failures during the actual execution of an NS LCM operation

6.6.3.3 LCM operation cancellation

The cancellation of an NS LCM operation that is in PROCESSING or ROLLING_BACK state is handled like any other error that leads to stopping the execution of the NS LCM workflow before it can be successfully completed. The NS LCM operation transits into the FAILED_TEMP state which allows root cause analysis, possible fixing of the root cause, followed by retrying, rolling back, or finally failing of the operation.

7 NS Performance Management interface

7.1 Description

This interface allows providing performance management (measurement results collection and notifications) related to NSs. Performance information on a given NS instance is sent by the NFVO to the OSS/BSS. Collection and reporting of performance information is controlled by a PM job that groups details of performance collection and reporting information. Further, this interface allows API version information retrieval.

When new performance information is available, the API consumer is notified using the notification `PerformanceInformationAvailableNotification`.

The operations provided through this interface are:

- Create PM Job
- Query PM Job
- Delete PM Job
- Create Threshold
- Query Threshold
- Delete Threshold
- Notify

7.1a API version

For the NS performance management interface version as specified in the present document, the MAJOR version field shall be 2, the MINOR version field shall be 1, and the PATCH version field shall be 0 (see clause 9.1 of ETSI GS NFV-SOL 013 [16] for a definition of the version fields). Consequently, the `{apiMajorVersion}` URI variable shall be set to "v2".

7.2 Resource structure and methods

All resource URIs of the API shall use the base URI specification defined in clause 4.1 of ETSI GS NFV-SOL 013 [16].

The string "nspm" shall be used to represent `{apiName}`. All resource URIs in the clauses below are defined relative to the above base URI.

Figure 7.2-1 shows the overall resource URI structure defined for the performance management API.

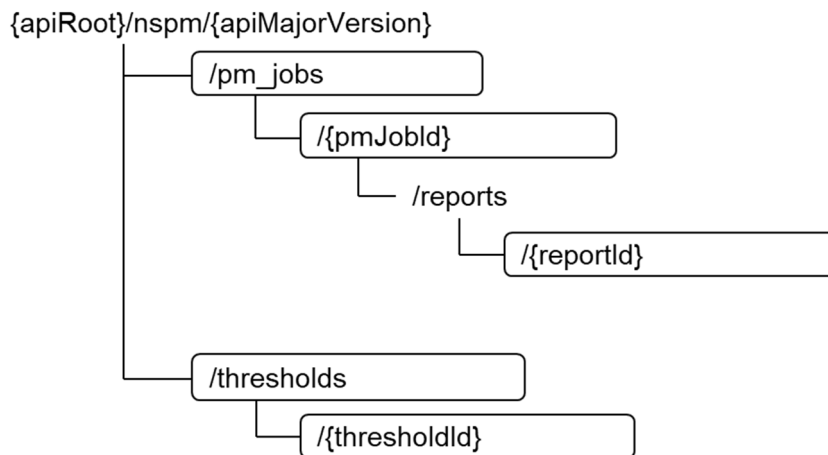


Figure 7.2-1: Resource URI structure of the NS Performance Management interface

Table 7.2-1 lists the individual resources defined, and the applicable HTTP methods.

The NFVO shall support responding to requests for all HTTP methods on the resources in Table 7.2-1 that are marked as "M" (mandatory) in the "Cat" column. The NFVO shall also support the "API versions" resources as specified in clause 9.3.2 of ETSI GS NFV-SOL 013 [16].

Table 7.2-1: Resources and methods overview of the NS Performance Management interface

Resource name	Resource URI	HTTP Method	Cat	Meaning
PM jobs	/pm_jobs	POST	M	Create a PM job.
		GET	M	Query PM jobs.
Individual PM job	/pm_jobs/{pmJobId}	GET	M	Read a single PM job.
		PATCH	M	Update PM job callback.
		DELETE	M	Delete a PM job.
Individual performance report	/pm_jobs/{pmJobId}/reports/{reportId}	GET	M	Read an individual performance report.
Thresholds	/thresholds	POST	M	Create a threshold.
		GET	M	Query thresholds.
Individual threshold	/thresholds/{thresholdId}	GET	M	Read a single threshold.
		PATCH	M	Update threshold callback.
		DELETE	M	Delete a threshold.
Notification endpoint	(provided by API consumer)	POST	See note	Notify about PM related events. See note.
		GET	See note	Test the notification endpoint. See note.
NOTE:	The NFVO shall support invoking the HTTP methods defined for the "Notification endpoint" resource exposed by the OSS/BSS. If the OSS/BSS supports invoking the POST method on the "PM jobs" or "Thresholds" resource towards the NFVO, it shall also support responding to the HTTP requests defined for the "Notification endpoint" resource.			

7.3 Sequence diagrams (informative)

7.3.1 Flow of creating a PM job

This clause describes a sequence for creating a performance management jobs.

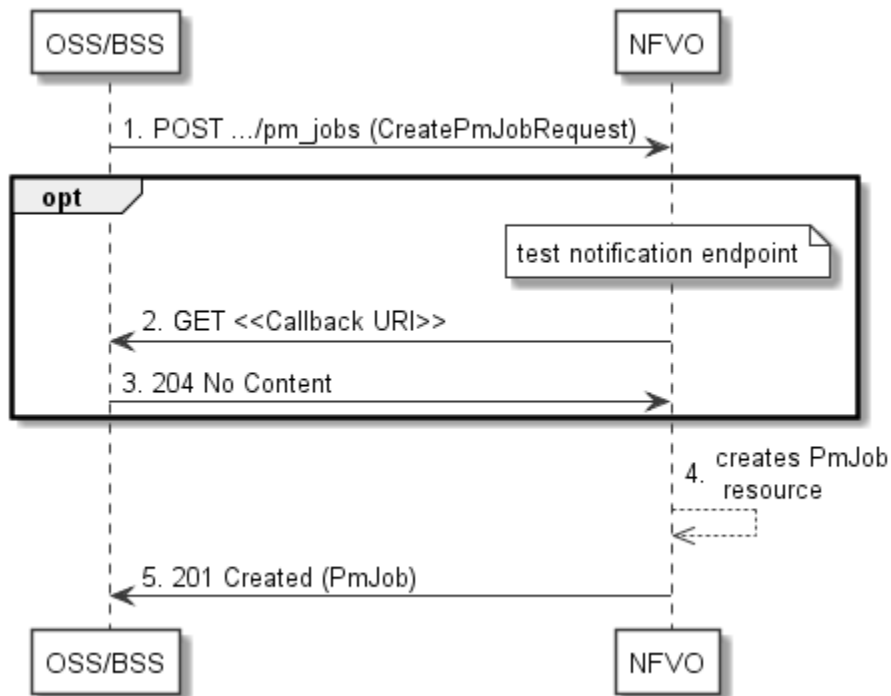


Figure 7.3.1-1: Flow of PM job creation

PM job creation, as illustrated in Figure 7.3.1-1, consists of the following steps:

- 1) If the OSS/BSS intends to create a PM job, it sends a POST request to the "PM jobs" resource, including one data structure of type "CreatePmJobRequest" in the payload body.
- 2) Optionally, to test the notification endpoint that was registered by the OSS/BSS during PM job creation, the NFVO sends a GET request to the notification endpoint URI.
- 3) In that case, the OSS/BSS returns a "204 No Content" response to indicate success.
- 4) The NFVO creates a PM job instance.
- 5) The NFVO returns a "201 Created" response to the OSS/BSS and includes in the payload body a representation of the PM job just created.

Error handling: In case of failure, including an invalid notification endpoint, appropriate error information is provided in the response.

7.3.1a Flow of updating the callback URI of a PM job

This clause describes a sequence for updating the callback URI in a PM job.

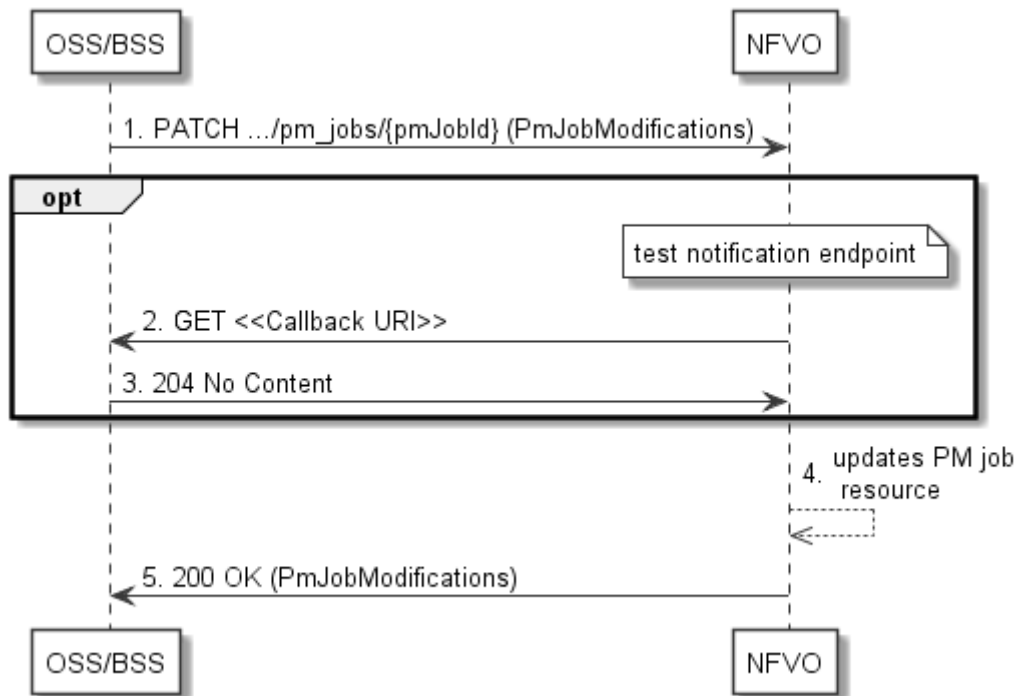


Figure 7.3.1a-1: Flow of PM job callback URI update

PM job callback URI update, as illustrated in Figure 7.3.1a-1, consists of the following steps:

- 1) If the OSS/BSS intends to update the callback URI in a PM job, it sends a PATCH request to the "Individual PM job" resource, including a data structure of type "PmJobModifications" in the payload body.
- 2) Optionally, to test the notification endpoint that is addressed by the new callback URI, the NFVO sends a GET request to the notification endpoint URI.
- 3) In that case, the OSS/BSS returns a "204 No Content" response to indicate success.
- 4) The NFVO returns a "200 OK" response to the OSS/BSS and includes in the payload body a data structure of type "PmJobModifications" to indicate the performed modifications.

Error handling: In case of failure, including an invalid notification endpoint, appropriate error information is provided in the response.

7.3.2 Flow of querying/reading PM jobs

This clause describes a sequence for querying/reading performance management jobs.

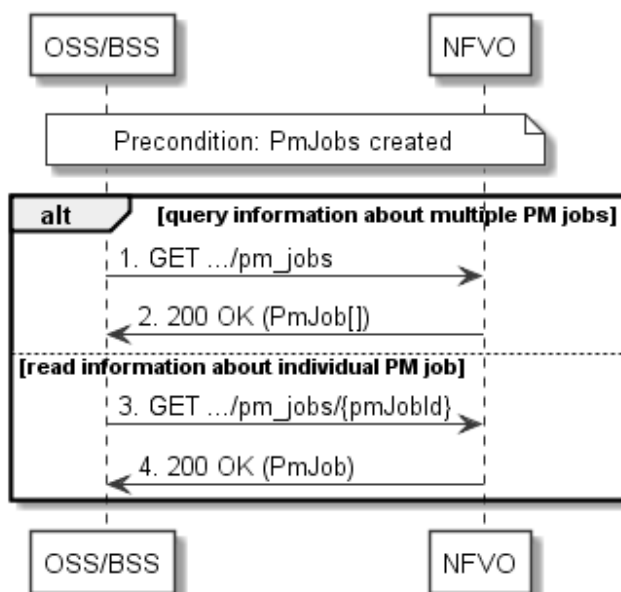


Figure 7.3.2-1: Flow of PM jobs query/read

PM jobs query/read, as illustrated in Figure 7.3.2-1, consists of the following steps:

- 1) If the OSS/BSS intends to query all PM jobs, it sends a GET request to the "PM jobs" resource.
- 2) The NFVO returns a "200 OK" response to the OSS/BSS, and includes zero or more data structures of type "PmJob" in the payload body.
- 3) If the OSS/BSS intends to read information about a particular PM job, it sends a GET request to the "Individual PM job" resource, addressed by the appropriate PM job identifier in its resource URI.
- 4) The NFVO returns a "200 OK" response to the OSS/BSS, and includes one data structure of type "PmJob" in the payload body.

Error handling: In case of failure, appropriate error information is provided in the response.

7.3.3 Flow of deleting a PM job

This clause describes a sequence for deleting a performance management jobs.

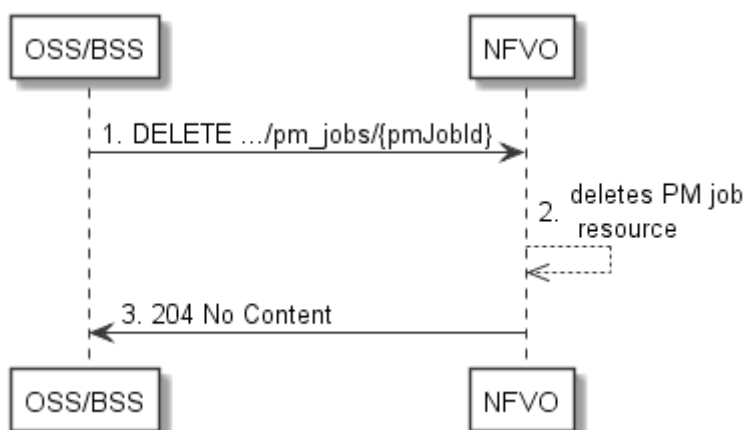


Figure 7.3.3-1: Flow of PM job deletion

PM job deletion, as illustrated in Figure 7.3.3-1, consists of the following steps:

- 1) If the OSS/BSS intends to delete a PM job, it sends a DELETE request to the "Individual PM job" resource, addressed by the appropriate PM job identifier in its resource URI.

- 2) The NFVO deletes the "Individual PM job" resource.
- 3) The NFVO returns a response with a "204 No Content" response code and an empty payload body to the OSS/BSS.

Error handling: In case of failure, appropriate error information is provided in the response.

7.3.4 Flow of obtaining performance reports

This clause describes a sequence for obtaining performance reports.

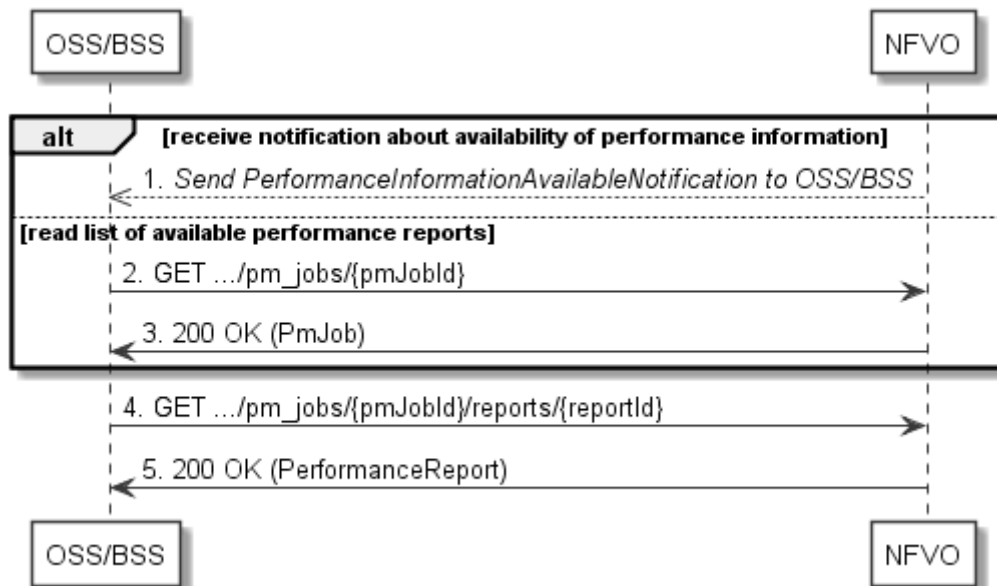


Figure 7.3.4-1: Flow of obtaining performance reports

Obtaining a performance report, as illustrated in Figure 7.3.4-1, consists of the following steps:

- 1) The NFVO sends to the OSS/BSS a PerformanceInformationAvailableNotification (see clause 7.3.9) that indicates the availability of a new performance report, including a link from which the report can be obtained.
- 2) Alternatively, the OSS/BSS sends a GET request to the "Individual PM job" resource, to obtain a representation of the PM job resource including information about performance reports that are available for this PM job, including their URIs.
- 3) In that case, the NFVO returns a "200 OK" response to the OSS/BSS, and includes a data structure of type "PmJob" in the payload body.
- 4) The OSS/BSS sends to the NFVO a GET request to the URI obtained either in step (1) or step (3), in order to read an "Individual performance report" resource.
- 5) The NFVO returns a "200 OK" response to the OSS/BSS, and includes a data structure of type "PerformanceReport" in the payload body.

7.3.5 Flow of creating a threshold

This clause describes a sequence for creating a performance management threshold.

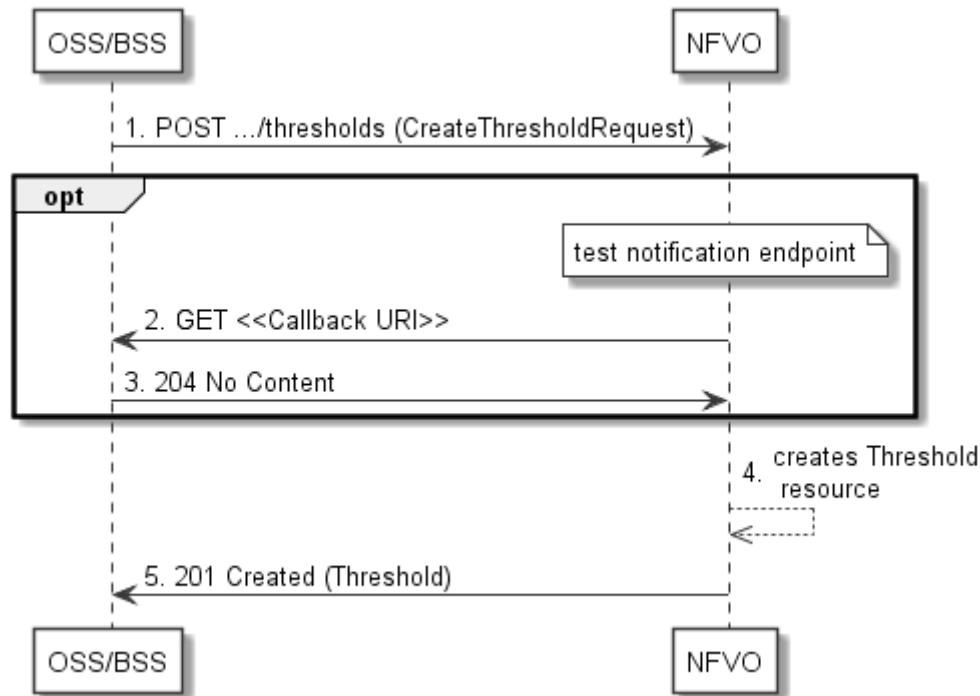


Figure 7.3.5-1: Flow of threshold creation

Threshold creation, as illustrated in Figure 7.3.5-1, consists of the following steps:

- 1) If the OSS/BSS intends to create a threshold, it sends a POST request to the "Thresholds" resource, including a data structure of type "CreateThresholdRequest" in the payload body.
- 2) Optionally, to test the notification endpoint that was registered by the OSS/BSS during threshold creation, the NFVO sends a GET request to the notification endpoint URI.
- 3) In that case, the OSS/BSS returns a "204 No Content" response to indicate success.
- 4) The NFVO creates a threshold instance.
- 5) The NFVO returns a "201 Created" response to the OSS/BSS and includes in the payload body a representation of the threshold just created.

Error handling: In case of failure, including an invalid notification endpoint, appropriate error information is provided in the response.

7.3.5a Flow of updating the callback URI of a threshold

This clause describes a sequence for updating the callback URI in a performance management threshold.

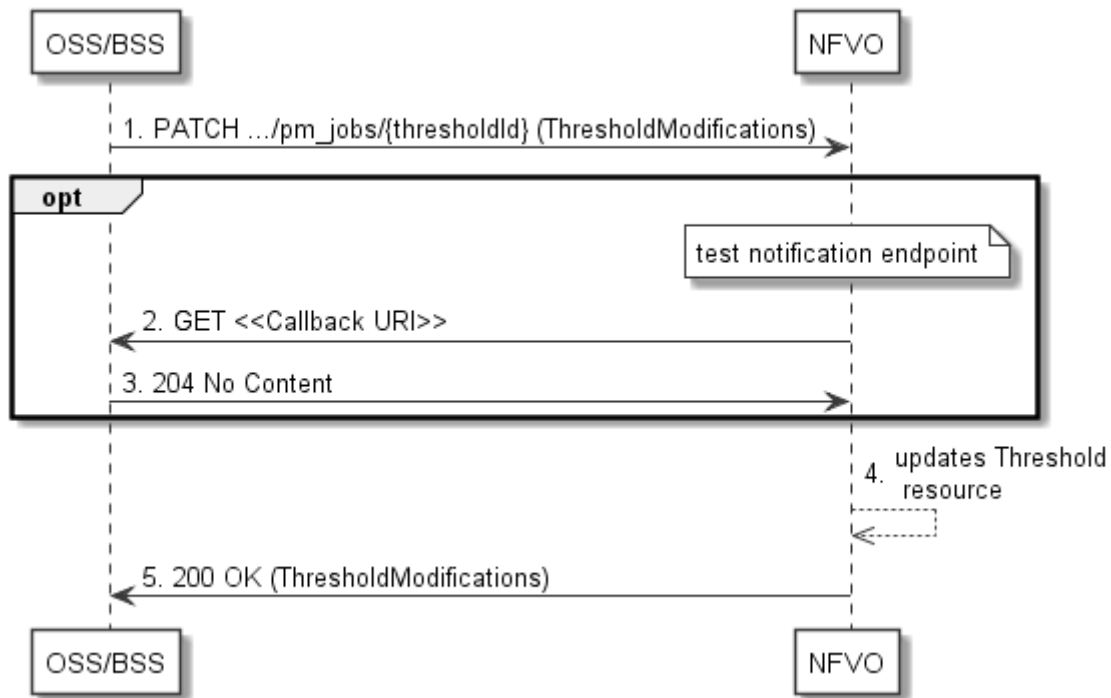


Figure 7.3.5a-1: Flow of threshold callback URI update

Threshold callback URI update, as illustrated in Figure 7.3.5a-1, consists of the following steps:

- 1) If the OSS/BSS intends to update the callback URI in a threshold, it sends a PATCH request to the "Individual threshold" resource, including a data structure of type "ThresholdModifications" in the payload body.
- 2) Optionally, to test the notification endpoint that is addressed by the new callback URI, the NFVO sends a GET request to the notification endpoint URI.
- 3) In that case, the OSS/BSS returns a "204 No Content" response to indicate success.
- 4) The NFVO returns a "200 OK" response to the EM and includes in the payload body a data structure of type "ThresholdModifications" to indicate the performed modifications.

Error handling: In case of failure, including an invalid notification endpoint, appropriate error information is provided in the response.

7.3.6 Flow of querying/reading thresholds

This clause describes a sequence for querying/reading performance management thresholds.

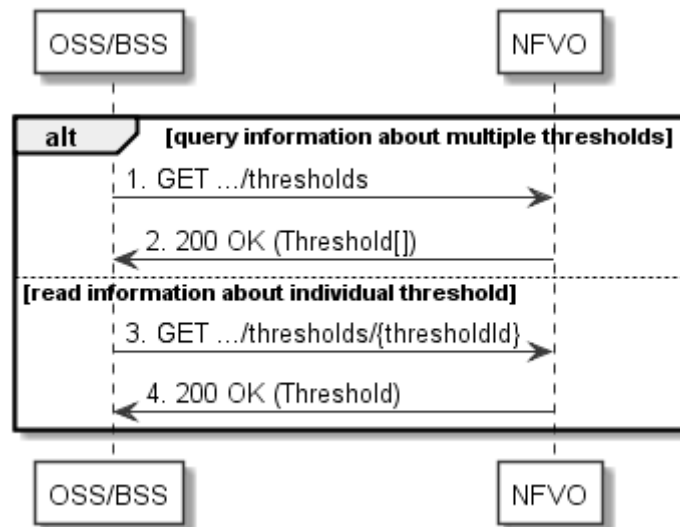


Figure 7.3.6-1: Flow of thresholds query/read

Threshold query/read, as illustrated in Figure 7.3.6-1, consists of the following steps:

- 1) If the OSS/BSS intends to query all thresholds, it sends a GET request to the "Thresholds" resource.
- 2) The NFVO returns a "200 OK" response to the OSS/BSS, and includes zero or more data structures of type "Threshold" in the payload body.
- 3) If the OSS/BSS intends to read information about a particular threshold, it sends a GET request to the "Individual threshold" resource with the appropriate threshold identifier in its resource URI.
- 4) The NFVO returns a "200 OK" response to the OSS/BSS, and includes a data structure of type "Threshold" in the payload body.

Error handling: In case of failure, appropriate error information is provided in the response.

7.3.7 Flow of deleting thresholds

This clause describes a sequence for deleting performance management thresholds.

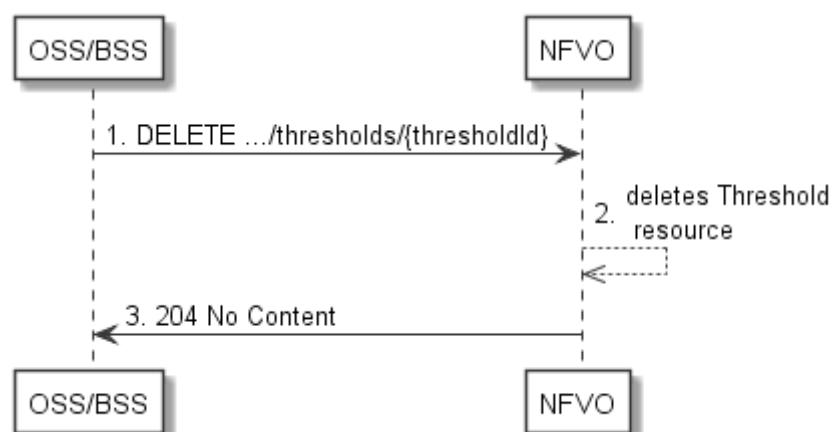


Figure 7.3.7-1: Flow of threshold deletion

Threshold deletion, as illustrated in Figure 7.3.7-1, consists of the following steps:

- 1) If the OSS/BSS intends to delete a particular threshold, it sends a DELETE request to the "Individual threshold" resource, addressed by the appropriate threshold identifier in its resource URI.

- 2) The NFVO deletes the "Individual threshold" resource.
- 3) The NFVO returns a "204 No Content" response code to the NFVO. The response body shall be empty.

Error handling: In case of failure, appropriate error information is provided in the response.

7.3.8 Void

7.3.9 Flow of sending notifications

This clause describes the procedure for sending notifications related to NS performance management.

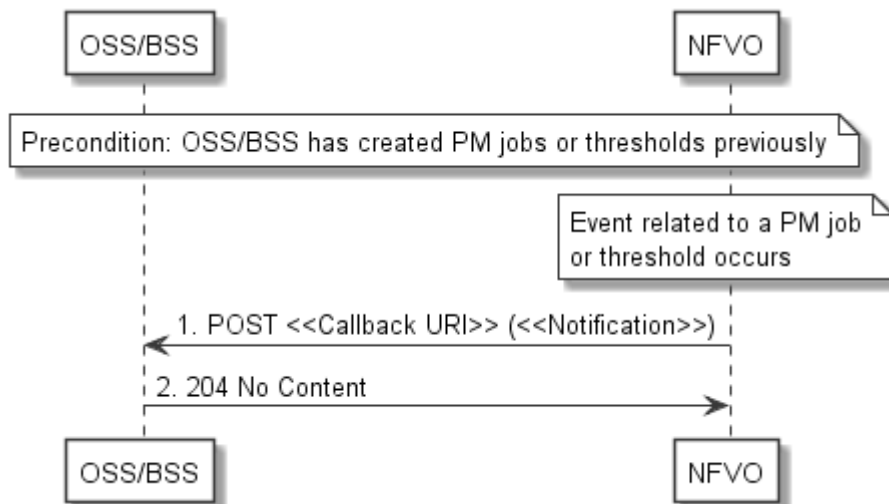


Figure 7.3.9-1: Flow of sending notifications

Precondition: The OSS/BSS has previously created thresholds and or PM jobs which trigger notifications related to NS performance management.

The procedure consists of the following steps as illustrated in Figure 7.3.9-1:

- 1) If an event occurs that indicates a threshold crossing or availability of performance information in a PM job, the NFVO generates a notification that includes information about the event, and sends it in the body of a POST request to the URI which the OSS/BSS has registered as part of creating the threshold or PM job. The variable <<Notification>> in the flow is a placeholder for the different types of notifications that can be sent by this API.
- 2) The OSS/BSS acknowledges the successful delivery of the notification by returning a "204 No Content" response.

Error handling: If the NFVO does not receive the "204 No Content" response from the NFVO, it can retry sending the notification.

7.4 Resources

7.4.1 Introduction

This clause defines all the resources and methods provided by the performance management interface.

7.4.1a Resource: API versions

The "API versions" resources as defined in clause 9.3.3 of ETSI GS NFV-SOL 013 [16] are part of the NS performance management interface.

7.4.2 Resource: PM jobs

7.4.2.1 Description

This resource represents PM jobs. The API consumer can use this resource to create and query PM jobs.

7.4.2.2 Resource definition

The resource URI is:

{apiRoot}/nspm/{apiMajorVersion}/pm_jobs

This resource shall support the resource URI variables defined in Table 7.4.2.2-1.

Table 7.4.2.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 7.1a.

7.4.2.3 Resource methods

7.4.2.3.1 POST

The POST method creates a PM job.

This method shall follow the provisions specified in the Tables 7.4.2.3.1-1 and 7.4.2.3.1-2 for URI query parameters, request and response data structures, and response codes.

As the result of successfully executing this method, a new "Individual PM job" resource as defined in clause 7.4.3 shall have been created.

Table 7.4.2.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
none supported		

Table 7.4.2.3.1-2: Details of the POST request/response on this resource

Request body	Data type	Cardinality	Description	
	CreatePmJobRequest	1	PM job creation request	
Response body	Data type	Cardinality	Response Codes	Description
	PmJob	1	201 Created	<p>Shall be returned when the PM job has been created successfully.</p> <p>The response body shall contain a representation of the created "Individual PM job" resource, as defined in clause 7.5.2.7.</p> <p>The HTTP response shall include a "Location" HTTP header that points to the created "Individual PM job" resource.</p>
	ProblemDetails	1	422 Unprocessable Entity	<p>Shall be returned upon the following error: The content type of the payload body is supported and the payload body of a request contains syntactically correct data but the data cannot be processed.</p> <p>The general cause for this error and its handling is specified in clause 6.4 of ETSI GS NFV-SOL 013 [16], including rules for the presence of the response body.</p> <p>Specifically in case of this resource, the response code 422 shall also be returned if the NFVO has tested the Notification endpoint as described in clause 7.4.9.3.2 and the test has failed.</p> <p>In this case, the "detail" attribute in the "ProblemDetails" structure shall convey more information about the error.</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

7.4.2.3.2 GET

The API consumer can use this method to retrieve information about PM jobs.

This method shall follow the provisions specified in the Tables 7.4.2.3.2-1 and 7.4.2.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 7.4.2.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
filter	0..1	<p>Attribute-based filtering expression according to clause 5.2 of ETSI GS NFV-SOL 013 [16].</p> <p>The NFVO shall support receiving this parameter as part of the URI query string. The OSS/BSS may supply this parameter.</p> <p>All attribute names that appear in the PmJob and in data types referenced from it shall be supported by the NFVO in the filter expression.</p>
all_fields	0..1	Include all complex attributes in the response. See clause 5.3 of ETSI GS NFV-SOL 013 [16] for details. The NFVO shall support this parameter.
fields	0..1	Complex attributes to be included into the response. See clause 5.3 of ETSI GS NFV-SOL 013 [16] for details. The NFVO should support this parameter.
exclude_fields	0..1	Complex attributes to be excluded from the response. See clause 5.3 of ETSI GS NFV-SOL 013 [16] for details. The NFVO should support this parameter.

Name	Cardinality	Description
exclude-default	0..1	Indicates to exclude the following complex attributes from the response. See clause 5.3 of ETSI GS NFV-SOL 013 [16] for details. The NFVO shall support this parameter. The following attributes shall be excluded from the PmJob structure in the response body if this parameter is provided, or none of the parameters "all_fields", "fields", "exclude_fields", "exclude_default" are provided: - Reports
nextpage_opaque_marker	0..1	Marker to obtain the next page of a paged response. Shall be supported by the NFVO if the NFVO supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource.

Table 7.4.2.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	PmJob	0..N	200 OK	Shall be returned when information about zero or more PM jobs has been queried successfully. The response body shall contain in an array the representations of zero or more PM jobs, as defined in clause 7.5.2.7. If the "filter" URI parameter or one of the "all_fields", "fields", "include_fields", "exclude_fields" or "exclude_default" URI parameters was supplied in the request and is supported, the data in the response body shall have been transformed according to the rules specified in clauses 5.2.2 and 5.3.2 of ETSI GS NFV-SOL 013 [16], respectively. If the NFVO supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource, inclusion of the Link HTTP header in this response shall follow the provisions in clause 5.4.2.3 of ETSI GS NFV-SOL 013 [16].
	ProblemDetails	1	400 Bad Request	Shall be returned upon the following error: Invalid attribute-based filtering expression. The response body shall contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.
	ProblemDetails	1	400 Bad Request	Shall be returned upon the following error: Invalid attribute selector. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	400 Bad Request	Shall be returned upon the following error: Response too big. If the NFVO supports alternative 1 (error) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource, this error response shall follow the provisions in clause 5.4.2.2 of ETSI GS NFV-SOL 013 [16].
ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.	

7.4.2.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

7.4.2.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

7.4.2.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

7.4.3 Resource: Individual PM job

7.4.3.1 Description

This resource represents an individual PM job. The API consumer can use this resource to delete and read the underlying PM job.

7.4.3.2 Resource definition

The resource URI is:

{apiRoot}/nspm/{apiMajorVersion}/pm_jobs/{pmJobId}

This resource shall support the resource URI variables defined in Table 7.4.3.2-1.

Table 7.4.3.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 7.1a.
pmJobId	Identifier of the PM job. See note.
NOTE:	This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request creating a new "Individual PM job" resource. It can also be retrieved from the "id" attribute in the payload body of that response.

7.4.3.3 Resource methods

7.4.3.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

7.4.3.3.2 GET

The API consumer can use this method for reading an individual PM job.

This method shall follow the provisions specified in the Tables 7.4.3.3.2-1 and 7.4.3.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 7.4.3.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
none supported		

Table 7.4.3.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	PmJob	1	200 OK	<p>Shall be returned when information about an individual PM job has been read successfully.</p> <p>The response body shall contain a representation of the "Individual PM job" resource, as defined in clause 7.5.2.7.</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

7.4.3.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

7.4.3.3.4 PATCH

This method allows to modify an "individual PM job" resource.

This method shall follow the provisions specified in the Tables 7.4.3.3.4-1 and 7.4.3.3.4-2 for URI query parameters, request and response data structures, and response codes.

Table 7.4.3.3.4-1: URI query parameters supported by the PATCH method on this resource

Name	Cardinality	Description
none supported		

Table 7.4.3.3.4-2: Details of the PATCH request/response on this resource

Request body	Data type	Cardinality	Description	
	PmJobModifications	1	<p>Parameters for the PM job modification.</p> <p>The Content-Type header shall be set to "application/merge-patch+json" according to IETF RFC 7396 [11].</p>	
Response body	Data type	Cardinality	Response Codes	Description
	PmJobModifications	1	200 OK	<p>Shall be returned when the request has been processed successfully.</p> <p>The response body shall contain a data structure of type "PmJobModifications".</p>
	ProblemDetails	0..1	412 Precondition failed	<p>Shall be returned upon the following error: A precondition given in an HTTP request header is not fulfilled.</p> <p>Typically, this is due to an ETag mismatch, indicating that the resource was modified by another entity.</p> <p>The response body should contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.</p>

	ProblemDetails	1	422 Unprocessable Entity	<p>Shall be returned upon the following error: The content type of the payload body is supported and the payload body of a request contains syntactically correct data but the data cannot be processed.</p> <p>The general cause for this error and its handling is specified in clause 6.4 of ETSI GS NFV-SOL 013 [16], including rules for the presence of the response body.</p> <p>Specifically in case of this resource, the response code 422 shall also be returned if the NFVO has tested the Notification endpoint as described in clause 7.4.9.3.2 and the test has failed.</p> <p>In this case, the "detail" attribute in the "ProblemDetails" structure shall convey more information about the error.</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

7.4.3.3.5 DELETE

This method terminates an individual PM job.

This method shall follow the provisions specified in the Tables 7.4.3.3.5-1 and 7.4.3.3.5-2 for URI query parameters, request and response data structures, and response codes.

As the result of successfully executing this method, the "Individual PM job" resource shall not exist any longer.

Table 7.4.3.3.5-1: URI query parameters supported by the DELETE method on this resource

Name	Cardinality	Description
none supported		

Table 7.4.3.3.5-2: Details of the DELETE request/response on this resource

Request body	Data type	Cardinality	Description	
n/a				
Response body	Data type	Cardinality	Response Codes	Description
	n/a		204 No Content	<p>Shall be returned when the PM job has been deleted successfully.</p> <p>The response body shall be empty.</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

7.4.4 Resource: Individual performance report

7.4.4.1 Description

This resource represents an individual performance report that has been collected by a PM job. The API consumer can use this resource to read the performance report. The URI of this report can be obtained from a PerformanceInformationAvailableNotification (see clause 7.5.2.5) or from the representation of the "Individual PM job" resource.

It is determined by means outside the scope of the present document, such as configuration or policy, how long an individual performance report is available.

7.4.4.2 Resource definition

The resource URI is:

{apiRoot}/nspm/{apiMajorVersion}/pm_jobs/{pmJobId}/reports/{reportId}

This resource shall support the resource URI variables defined in Table 7.4.4.2-1.

Table 7.4.4.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 7.1a.
pmJobId	Identifier of the PM job.
reportId	Identifier of the performance report.

7.4.4.3 Resource methods

7.4.4.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

7.4.4.3.2 GET

The API consumer can use this method for reading an individual performance report.

This method shall follow the provisions specified in the Tables 7.4.4.3.2-1 and 7.4.4.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 7.4.4.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
none supported		

Table 7.4.4.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	PerformanceReport	1	200 OK	Shall be returned when information of an individual performance report has been read successfully. The response body shall contain a representation of the "Individual performance report" resource, as defined in clause 7.5.2.10.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

7.4.4.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

7.4.4.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

7.4.4.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

7.4.5 Resource: Thresholds

7.4.5.1 Description

This resource represents thresholds. The API consumer can use this resource to create and query thresholds.

7.4.5.2 Resource definition

The resource URI is:

{apiRoot}/nspm/{apiMajorVersion}/thresholds

This resource shall support the resource URI variables defined in Table 7.4.5.2-1.

Table 7.4.5.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 7.1a.

7.4.5.3 Resource methods

7.4.5.3.1 POST

The POST method can be used by the API consumer to create a threshold.

This method shall follow the provisions specified in the Tables 7.4.5.3.1-1 and 7.4.5.3.1-2 for URI query parameters, request and response data structures, and response codes.

As the result of successfully executing this method, a new "Individual threshold" resource as defined in clause 7.4.6 shall have been created.

Table 7.4.5.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
None supported		

Table 7.4.5.3.1-2: Details of the POST request/response on this resource

Request body	Data type	Cardinality	Description	
	CreateThresholdRequest	1	Request parameters to create a new "Individual threshold" resource.	
Response body	Data type	Cardinality	Response Codes	Description
	Threshold	1	201 Created	<p>Shall be returned when a threshold has been created successfully.</p> <p>The response body shall contain a representation of the created "Individual threshold" resource, as defined in clause 7.5.2.9.</p> <p>The HTTP response shall include a "Location" HTTP header that contains the resource URI of the created threshold resource.</p>
	ProblemDetails	1	422 Unprocessable Entity	<p>Shall be returned upon the following error: The content type of the payload body is supported and the payload body of a request contains syntactically correct data but the data cannot be processed.</p> <p>The general cause for this error and its handling is specified in clause 6.4 of ETSI GS NFV-SOL 013 [16], including rules for the presence of the response body.</p> <p>Specifically in case of this resource, the response code 422 shall also be returned if the NFVO has tested the Notification endpoint as described in clause 7.4.9.3.2 and the test has failed.</p> <p>In this case, the "detail" attribute in the "ProblemDetails" structure shall convey more information about the error.</p>
ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.	

7.4.5.3.2 GET

The API consumer can use this method to query information about thresholds.

This method shall follow the provisions specified in the Tables 7.4.5.3.2-1 and 7.4.5.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 7.4.5.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
filter	0..1	<p>Attribute-based filtering expression according to clause 5.2 of ETSI GS NFV-SOL 013 [16].</p> <p>The NFVO shall support receiving this parameter as part of the URI query string. The OSS/BSS may supply this parameter.</p> <p>All attribute names that appear in the Thresholds data type and in data types referenced from it shall be supported by the NFVO in the filter expression.</p>
nextpage_opaque_marker	0..1	Marker to obtain the next page of a paged response. Shall be supported by the NFVO if the NFVO supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource.

NOTE: There are no attribute selectors defined for this resource as the threshold attributes with cardinality 0..1 or 0..N are not structurally complex in nature.

Table 7.4.5.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	Threshold	0..N	200 OK	<p>Shall be returned when information about zero or more thresholds has been queried successfully.</p> <p>If the "filter" URI parameter was supplied in the request, the data in the response body shall have been transformed according to the rules specified in clause 5.2.2 of ETSI GS NFV-SOL 013 [16].</p> <p>The response body shall contain in an array the representations of zero or more thresholds, as defined in clause 7.5.2.9.</p> <p>If the NFVO supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource, inclusion of the Link HTTP header in this response shall follow the provisions in clause 5.4.2.3 of ETSI GS NFV-SOL 013 [16].</p>
	ProblemDetails	1	400 Bad Request	<p>Shall be returned upon the following error: Invalid attribute-based filtering expression.</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.</p>
	ProblemDetails	1	400 Bad Request	<p>Shall be returned upon the following error: Response too big.</p> <p>If the NFVO supports alternative 1 (error) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource, this error response shall follow the provisions in clause 5.4.2.2 of ETSI GS NFV-SOL 013 [16].</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

7.4.5.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

7.4.5.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

7.4.5.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

7.4.6 Resource: Individual threshold

7.4.6.1 Description

This resource represents an individual threshold.

7.4.6.2 Resource definition

The resource URI is:

{apiRoot}/nspm/{apiMajorVersion}/thresholds/{thresholdId}

This resource shall support the resource URI variables defined in Table 7.4.6.2-1.

Table 7.4.6.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 7.1a.
thresholdId	Identifier of the threshold. See note.
NOTE:	This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request creating a new "Individual threshold" resource. It can also be retrieved from the "id" attribute in the payload body of that response.

7.4.6.3 Resource methods

7.4.6.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

7.4.6.3.2 GET

The API consumer can use this method for reading an individual threshold.

This method shall follow the provisions specified in the Tables 7.4.6.3.2-1 and 7.4.6.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 7.4.6.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
none supported		

Table 7.4.6.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	Threshold	1	200 OK	Shall be returned when information about an individual threshold has been read successfully. The response body shall contain a representation of the threshold, as defined in clause 7.5.2.9.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

7.4.6.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

7.4.6.3.4 PATCH

This method allows to modify an "Individual threshold" resource.

This method shall follow the provisions specified in the Tables 7.4.6.3.4-1 and 7.4.6.3.4-2 for URI query parameters, request and response data structures, and response codes.

Table 7.4.6.3.4-1: URI query parameters supported by the PATCH method on this resource

Name	Cardinality	Description
none supported		

Table 7.4.6.3.4-2: Details of the PATCH request/response on this resource

Request body	Data type	Cardinality	Description	
	ThresholdModifications	1	Parameters for the threshold modification. The Content-Type header shall be set to "application/merge-patch+json" according to IETF RFC 7396 [11].	
Response body	Data type	Cardinality	Response Codes	Description
	ThresholdModifications	1	200 OK	Shall be returned when the request has been processed successfully. The response body shall contain a data structure of type "ThresholdModifications".
	ProblemDetails	0..1	412 Precondition failed	Shall be returned upon the following error: A precondition given in an HTTP request header is not fulfilled. Typically, this is due to an ETag mismatch, indicating that the resource was modified by another entity. The response body should contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.
	ProblemDetails	1	422 Unprocessable Entity	Shall be returned upon the following error: The content type of the payload body is supported and the payload body of a request contains syntactically correct data but the data cannot be processed. The general cause for this error and its handling is specified in clause 6.4 of ETSI GS NFV-SOL 013 [16], including rules for the presence of the response body. Specifically in case of this resource, the response code 422 shall also be returned if the NFVO has tested the Notification endpoint as described in clause 7.4.9.3.2 and the test has failed. In this case, the "detail" attribute in the "ProblemDetails" structure shall convey more information about the error.
ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.	

7.4.6.3.5 DELETE

This method allows to delete a threshold.

This method shall follow the provisions specified in the Tables 7.4.6.3.5-1 and 7.4.6.3.5-2 for URI query parameters, request and response data structures, and response codes.

As the result of successfully executing this method, the "Individual threshold" resource shall not exist any longer.

Table 7.4.6.3.5-1: URI query parameters supported by the DELETE method on this resource

Name	Cardinality	Description

Table 7.4.6.3.5-2: Details of the DELETE request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	n/a		204 No Content	<p>Shall be returned when the threshold has been deleted successfully.</p> <p>The response body shall be empty.</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

7.4.7 Void

7.4.8 Void

7.4.9 Resource: Notification endpoint

7.4.9.1 Description

This resource represents a notification endpoint for NS performance management.

The API producer can use this resource to send notifications related to performance management events to an API consumer which has provided the URI of this resource during the PM job or threshold creation process.

7.4.9.2 Resource definition

The resource URI is provided by the API consumer when creating the PM job or threshold.

This resource shall support the resource URI variables defined in Table 7.4.9.2-1.

Table 7.4.9.2-1: Resource URI variables for this resource

Name	Definition
n/a	

7.4.9.3 Resource methods

7.4.9.3.1 POST

The POST method delivers a notification regarding a performance management event from the API producer to an API consumer. The API consumer shall have previously created an "Individual PM job resource" or "Individual threshold resource".

This method shall follow the provisions specified in the Tables 7.4.9.3.1-1 and 7.4.9.3.1-2 for URI query parameters, request and response data structures, and response codes.

Table 7.4.9.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
none supported		

Table 7.4.9.3.1-2: Details of the POST request/response on this resource

Request body	Data type	Cardinality	Description	
	PerformanceInformationAvailableNotification	1	Notification about performance information availability	
	ThresholdCrossedNotification	1	Notification about threshold crossing	
Response body	Data type	Cardinality	Response Codes	Description
	n/a		204 No Content	Shall be returned when the notification has been delivered successfully.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

7.4.9.3.2 GET

The GET method allows the API producer to test the notification endpoint that is provided by the API consumer, e.g. during creation of the PM job or threshold resource.

This method shall follow the provisions specified in the Tables 7.4.9.3.2-1 and 7.4.9.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 7.4.9.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
none supported		

Table 7.4.9.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	n/a		204 No Content	Shall be returned to indicate that the notification endpoint has been tested successfully. The response body shall be empty.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

7.4.9.3.3 PUT

This method is not supported. When this method is requested on this resource, the OSS/BSS shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

7.4.9.3.4 PATCH

This method is not supported. When this method is requested on this resource, the OSS/BSS shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

7.4.9.3.5 DELETE

This method is not supported. When this method is requested on this resource, the OSS/BSS shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

7.5 Data Model

7.5.1 Introduction

This clause defines the request and response data structures of the NS Performance Management interface. If a request or response contains attributes not defined in the present document, a receiving functional block that does not understand these attributes shall not treat their presence as an error, and may choose to ignore them.

7.5.2 Resource and notification data types

7.5.2.1 Introduction

This clause defines the data structures to be used in resource representations and notifications.

7.5.2.2 Void

7.5.2.3 Void

7.5.2.4 Type: ThresholdCrossedNotification

This type represents a notification that is sent when a threshold has been crossed. It shall comply with the provisions defined in Table 7.5.2.4-1.

NOTE: The timing of sending this notification is determined by the capability of the producing entity to evaluate the threshold crossing condition.

The notification shall be triggered by the NFVO when a threshold has been crossed.

Table 7.5.2.4-1: Definition of the ThresholdCrossedNotification data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this notification. If a notification is sent multiple times due to multiple subscriptions, the "id" attribute of all these notifications shall have the same value.
notificationType	String	1	Discriminator for the different notification types. Shall be set to "ThresholdCrossedNotification" for this notification type.
timeStamp	DateTime	1	Date and time of the generation of the notification.
thresholdId	Identifier	1	Identifier of the threshold which has been crossed.
crossingDirection	CrossingDirectionType	1	An indication of whether the threshold was crossed in upward or downward direction.

Attribute name	Data type	Cardinality	Description
objectType	String	1	Type of the measured object. The applicable measured object type for a measurement is defined in clause 7.3 of ETSI GS NFV-IFA 027 [15].
objectInstanceid	Identifier	1	Identifier of the measured object instance as per clause 6.2 of ETSI GS NFV-IFA 027 [15].
subObjectInstanceid	IdentifierInNs	0..1	Identifier of the sub-object of the measured object to which the measurement applies. Shall be present if this is required in clause 6.2 of ETSI GS NFV-IFA 027 [15] for the related measured object type. See note.
performanceMetric	String	1	Performance metric associated with the threshold. This attribute shall contain the related "Measurement Name" value as defined in clause 7.3 of ETSI GS NFV-IFA 027 [15].
performanceValue	(any type)	1	Value of the metric that resulted in threshold crossing. The type of this attribute shall correspond to the related "Measurement Unit" as defined in clause 7.3 of ETSI GS NFV-IFA 027 [15].
context	KeyValuePairs	0..1	Measurement context information related to the measured value. The set of applicable keys is defined per measurement in the related "Measurement Context" in clause 7.3 of ETSI GS NFV-IFA 027 [15].
_links	Structure (inlined)	1	Links to resources related to this notification.
>objectInstance	NotificationLink	0..1	Link to the resource representing the measured object instance to which the notified change applies. Shall be present if the measured object instance information is accessible as a resource.
>threshold	NotificationLink	1	Link to the resource that represents the threshold that was crossed.
NOTE: The sub-object allows to structure the measured object but is not to be confused with sub-counters which allow to structure the measurement.			

7.5.2.5 Type: PerformanceInformationAvailableNotification

This notification informs the receiver that performance information is available. It shall comply with the provisions defined in Table 7.5.2.5-1.

The notification shall be triggered by the NFVO when new performance information collected by a PM job is available.

The periodicity of triggering this notification is influenced by the "reportingPeriod" attribute in the "PmJobCriteria" data structure as defined in clause 7.5.3.3.

Table 7.5.2.5-1: Definition of the PerformanceInformationAvailableNotification data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this notification. If a notification is sent multiple times due to multiple subscriptions, the "id" attribute of all these notifications shall have the same value.
notificationType	String	1	Discriminator for the different notification types. Shall be set to "PerformanceInformationAvailableNotification" for this notification type.
timeStamp	DateTime	1	Date and time of the generation of the notification.
pmJobId	Identifier	1	Identifier of the PM job for which performance information is available.
objectType	String	1	Type of the measured object. The applicable measured object type for a measurement is defined in clause 7.3 of ETSI GS NFV-IFA 027 [15].
objectInstanceid	Identifier	1	Identifier of the measured object instance as per clause 6.2 of ETSI GS NFV-IFA 027 [15].
subObjectInstanceIds	IdentifierInNs	0..N	Identifiers of the sub-object instances of the measured object instance for which the measurements have been taken. Shall be present if the related PM job has been set up to measure only a subset of all sub-object instances of the measured object instance and a sub-object is defined in clause 6.2 of ETSI GS NFV-IFA 027 [15] for the related measured object type. Shall be absent otherwise.
_links	Structure (inlined)	1	Links to resources related to this notification.
>objectInstance	NotificationLink	0..1	Link to the resource representing the measured object instance to which the notification applies. Shall be present if the measured object instance information is accessible as a resource.
>pmJob	NotificationLink	1	Link to the resource that represents the PM job for which performance information is available.
>performanceReport	NotificationLink	1	Link from which the available performance information of data type "PerformanceReport" (see clause 7.5.2.10) can be obtained. This link should point to an "Individual performance report" resource as defined in clause 7.4.4.

7.5.2.6 Type: CreatePmJobRequest

This type represents a request to create a PM job. It shall comply with the provisions defined in Table 7.5.2.6-1.

Table 7.5.2.6-1: Definition of the CreatePmJobRequest data type

Attribute name	Data type	Cardinality	Description
objectType	String	1	Type of the measured object. The applicable measured object type for a measurement is defined in clause 7.3 of ETSI GS NFV-IFA 027 [15].
objectInstanceIds	Identifier	1..N	Identifiers of the measured object instances for which performance information is requested to be collected.
subObjectInstanceIds	IdentifierInNs	0..N	Identifiers of the sub-object instances of the measured object instance for which performance information is requested to be collected. May be present if a sub-object is defined in clause 6.2 of ETSI GS NFV-IFA 027 [15] for the related measured object type. If this attribute is present, the cardinality of the "objectInstanceIds" attribute shall be 1. If this attribute is absent and a sub-object is defined in clause 6.2 of ETSI GS NFV-IFA 027 [15] for the related measured object type, measurements will be taken for all sub-object instances of the measured object instance.
criteria	PmJobCriteria	1	Criteria of the collection of performance information.
callbackUri	Uri	1	The URI of the endpoint to send the notification to.
authentication	SubscriptionAuthentication	0..1	Authentication parameters to configure the use of Authorization when sending notifications corresponding to this PM job, as defined in clause 8.3.4 of ETSI GS NFV-SOL 013 [16]. This attribute shall only be present if the API consumer requires authorization of notifications.

7.5.2.7 Type: PmJob

This type represents a PM job. It shall comply with the provisions defined in Table 7.5.2.7-1.

Table 7.5.2.7-1: Definition of the PmJob data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this PM job.
objectType	String	1	Type of the measured object. The applicable measured object type for a measurement is defined in clause 7.3 of ETSI GS NFV-IFA 027 [15].
objectInstancelds	Identifier	1..N	Identifiers of the measured object instances for which performance information is collected.
subObjectInstancelds	IdentifierInNs	0..N	Identifiers of the sub-object instances of the measured object instance for which performance information is requested to be collected. May be present if a sub-object is defined in clause 6.2 of ETSI GS NFV-IFA 027 [15] for the related measured object type. If this attribute is present, the cardinality of the "objectInstancelds" attribute shall be 1. If this attribute is absent and a sub-object is defined in clause 6.2 of ETSI GS NFV-IFA 027 [15] for the related measured object type, measurements will be taken for all sub-object instances of the measured object instance.
criteria	PmJobCriteria	1	Criteria of the collection of performance information.
callbackUri	Uri	1	The URI of the endpoint to send the notification to.
reports	Structure (inlined)	0..N	Information about available reports collected by this PM job.
>href	Uri	1	The URI where the report can be obtained.
>readyTime	DateTime	1	The time when the report was made available.
>expiryTime	DateTime	0..1	The time when the report will expire.
>fileSize	UnsigendInt	0..1	The size of the report file in bytes, if known.
_links	Structure (inlined)	1	Links for this resource.
>self	Link	1	URI of this resource.
>objects	Link	0..N	Links to resources representing the measured object instances for which performance information is collected. Shall be present if the measured object instance information is accessible as a resource.

7.5.2.8 Type: CreateThresholdRequest

This type represents a request to create a threshold. It shall comply with the provisions defined in Table 7.5.2.8-1.

Table 7.5.2.8-1: Definition of the CreateThresholdRequest data type

Attribute name	Data type	Cardinality	Description
objectType	String	1	Type of the measured object. The applicable measured object type for a measurement is defined in clause 7.3 of ETSI GS NFV-IFA 027 [15].
objectInstancelid	Identifier	1	Identifier of the measured object instance associated with this threshold.
subObjectInstancelids	IdentifierInNs	0..N	Identifiers of the sub-object instances of the measured object instance associated with this threshold. May be present if a sub-object is defined in clause 6.2 of ETSI GS NFV-IFA 027 [15] for the related measured object type. If this attribute is absent and a sub-object is defined in clause 6.2 of ETSI GS NFV-IFA 027 [15] for the measured object type, measurements will be taken for all sub-object instances of the measured object instance.
criteria	ThresholdCriteria	1	Criteria that define this threshold.
callbackUri	Uri	1	The URI of the endpoint to send the notification to.
authentication	SubscriptionAuthentication	0..1	Authentication parameters to configure the use of Authorization when sending notifications corresponding to this threshold, as defined in clause 8.3.4 of ETSI GS NFV-SOL 013 [16]. This attribute shall only be present if the API consumer requires authorization of notifications.

7.5.2.9 Type: Threshold

This type represents a threshold. It shall comply with the provisions defined in Table 7.5.2.9-1.

Table 7.5.2.9-1: Definition of the Threshold data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this threshold resource.
objectType	String	1	Type of the measured object. The applicable measured object type for a measurement is defined in clause 7.3 of ETSI GS NFV-IFA 027 [15].
objectInstancelid	Identifier	1	Identifier of the measured object instance associated with the threshold.
subObjectInstancelids	IdentifierInNs	0..N	Identifiers of the sub-object instances of the measured object instance associated with the threshold. May be present if a sub-object is defined in clause 6.2 of ETSI GS NFV-IFA 027 [15] for the related measurement type. If this attribute is absent and a sub-object is defined in clause 6.2 of ETSI GS NFV-IFA 027 [15] for the related measured object type, measurements will be taken for all sub-object instances of the measured object instance.
criteria	ThresholdCriteria	1	Criteria that define this threshold.
callbackUri	Uri	1	The URI of the endpoint to send the notification to.
_links	Structure (inlined)	1	Links for this resource.
>self	Link	1	URI of this resource.
>object	Link	0..1	Link to a resource representing the measured object instance for which performance information is collected. Shall be present if the measured object instance information is accessible as a resource.

7.5.2.10 Type: PerformanceReport

This type defines the format of a performance report provided by the NFVO to the OSS/BSS as a result of collecting performance information as part of a PM job. The type shall comply with the provisions defined in Table 7.5.2.10-1.

Table 7.5.2.10-1: Definition of the PerformanceReport data type

Attribute name	Data type	Cardinality	Description
entries	Structure (inlined)	1..N	List of performance information entries. Each performance report entry is for a given metric of a given object (i.e. NS instance), but can include multiple collected values.
>objectType	String	1	Type of the measured object. The applicable measured object type for a measurement is defined in clause 7.3 of ETSI GS NFV-IFA 027 [15].
>objectInstanceld	Identifier	1	Identifier of the measured object instance for which the performance metric is reported.
>subObjectInstanceld	IdentifierInNs	0..1	Identifier of the sub-object instance of the measured object instance for which the performance metric is reported. Shall be present if this is required in clause 6.2 of ETSI GS NFV-IFA 027 [15] for the related measured object type. See note.
>performanceMetric	String	1	Name of the metric collected. This attribute shall contain the related "Measurement Name" value as defined in clause 7.3 of ETSI GS NFV-IFA 027 [15].
>performanceValues	Structure (inlined)	1..N	List of performance values with associated timestamp.
>>timeStamp	DateTime	1	Time stamp indicating when the data has been collected.
>>value	(any type)	1	Value of the metric collected. The type of this attribute shall correspond to the related "Measurement Unit" as defined in clause 7.3 of ETSI GS NFV-IFA 027 [15].
>>context	KeyValuePairs	0..1	Measurement context information related to the measured value. The set of applicable keys is defined per measurement in the related "Measurement Context" in clause 7.3 of ETSI GS NFV-IFA 027 [15].
NOTE:	The sub-object allows to structure the measured object but is not to be confused with sub-counters which allow to structure the measurement value. EXAMPLE: Measured object: VnfcInstanceXYZ Sub-object: VnfcInstance1 Measurement: vCPU_utilization Sub-counters: vCPU utilization of each of the vCPUs of VnfcInstance1 (vCPU_utilization.vCPU1, vCPU_utilization.vCPU2, etc.).		

7.5.2.11 Type: ThresholdModifications

This type represents modifications to a threshold. It shall comply with the provisions defined in Table 7.5.2.11-1.

Table 7.5.2.11-1: Definition of the ThresholdModifications data type

Attribute name	Data type	Cardinality	Description
callbackUri	Uri	0..1	New value of the "callbackUri" attribute. The value "null" is not permitted. See note.
authentication	SubscriptionAuthentication	0..1	New value of the "authentication" attribute, or "null" to remove the attribute. If present in a request body, these modifications shall be applied according to the rules of JSON Merge Patch (see IETF RFC 7396 [11]). This attribute shall not be present in response bodies. See note.
NOTE:	At least one of the attributes defined in this type shall be present in request bodies.		

7.5.2.12 Type: PmJobModifications

This type represents modifications to a PM job. It shall comply with the provisions defined in Table 7.5.2.12-1.

Table 7.5.2.12-1: Definition of the PmJobModifications data type

Attribute name	Data type	Cardinality	Description
callbackUri	Uri	0..1	New value of the "callbackUri" attribute. The value "null" is not permitted. See note.
authentication	SubscriptionAuthentication	0..1	New value of the "authentication" attribute, or "null" to remove the attribute. If present in a request body, these modifications shall be applied according to the rules of JSON Merge Patch (see IETF RFC 7396 [11]). This attribute shall not be present in response bodies. See note.
NOTE: At least one of the attributes defined in this type shall be present in request bodies.			

7.5.3 Referenced structured data types

7.5.3.1 Introduction

This clause defines data structures that can be referenced from data structures defined in the previous clauses, but can neither be resource representations nor bound to any subscribe/notify mechanism.

7.5.3.2 Void

7.5.3.3 Type: PmJobCriteria

This type represents collection criteria for PM jobs. It shall comply with the provisions defined in Table 7.5.3.3-1.

Table 7.5.3.3-1: Definition of the PmJobCriteria data type

Attribute name	Data type	Cardinality	Description
performanceMetric	String	0..N	This defines the types of performance metrics for the specified object instances. Valid values are specified as "Measurement Name" values in clause 7.3 of ETSI GS NFV-IFA 027 [15]. At least one of the two attributes (performance metric or group) shall be present.
performanceMetricGroup	String	0..N	Group of performance metrics. A metric group is a pre-defined list of metrics, known to the API producer that it can decompose to individual metrics. Valid values are specified as "Measurement Group" values in clause 7.3 of ETSI GS NFV-IFA 027 [15]. At least one of the two attributes (performance metric or group) shall be present.
collectionPeriod	UnsignedInt	1	Specifies the periodicity at which the API producer will collect performance information. The unit shall be seconds. See note 1 and note 2.
reportingPeriod	UnsignedInt	1	Specifies the periodicity at which the API producer will report to the API consumer about performance information. The unit shall be seconds. See note 1 and note 2.
reportingBoundary	DateTime	0..1	Identifies a time boundary after which the reporting will stop. The boundary shall allow a single reporting as well as periodic reporting up to the boundary.

Attribute name	Data type	Cardinality	Description
NOTE 1: At the end of each reportingPeriod, the API producer will inform the API consumer about availability of the performance data collected for each completed collection period during this reportingPeriod. The reportingPeriod should be equal to or a multiple of the collectionPeriod. In the latter case, the performance data for the collection periods within one reporting period are reported together.			
NOTE 2: In particular when choosing short collection and reporting periods, the number of PM jobs that can be supported depends on the capability of the producing entity.			

7.5.3.4 Type: ThresholdCriteria

This type represents criteria that define a threshold. It shall comply with the provisions defined in Table 7.5.3.4-1.

Table 7.5.3.4-1: Definition of the ThresholdCriteria data type

Attribute name	Data type	Cardinality	Description
performanceMetric	String	1	Defines the performance metric associated with the threshold. Valid values are specified as "Measurement Name" values in clause 7.2 of ETSI GS NFV-IFA 027 [15].
thresholdType	Enum (inlined)	1	Type of threshold. This attribute determines which other attributes are present in the data structure. Permitted values: - SIMPLE: Single-valued static threshold See note 1.
simpleThresholdDetails	Structure (inlined)	0..1	Details of a simple threshold. Shall be present if thresholdType="SIMPLE".
>thresholdValue	Number	1	The threshold value. Shall be represented as a floating point number.
>hysteresis	Number	1	The hysteresis of the threshold. Shall be represented as a non-negative floating point number. A notification with crossing direction "UP" will be generated if the measured value reaches or exceeds "thresholdValue" + "hysteresis". A notification with crossing direction "DOWN" will be generated if the measured value reaches or undercuts "thresholdValue" - "hysteresis". See note 2.
NOTE 1: In the present document, simple thresholds are defined. The definition of additional threshold types is left for future specification.			
NOTE 2: The hysteresis is defined to prevent storms of threshold crossing notifications. When processing a request to create a threshold, implementations should enforce a suitable minimum value for this attribute (e.g. override the value or reject the request).			

7.5.4 Referenced simple data types and enumerations

7.5.4.1 Introduction

This clause defines simple data types and enumerations that can be referenced from data structures defined in the previous clauses.

7.5.4.2 Simple data types

No particular simple data types are defined for this interface, in addition to those defined in clause 4.4.2.

7.5.4.3 Enumeration: CrossingDirectionType

The enumeration CrossingDirectionType shall comply with the provisions defined in Table 7.5.4.3-1.

Table 7.5.4.3-1: Enumeration CrossingDirectionType

Enumeration value	Description
UP	The threshold was crossed in upward direction.
DOWN	The threshold was crossed in downward direction.

8 NS Fault Management interface

8.1 Description

This interface allows the OSS/BSS to subscribe to notifications regarding NS alarms provided by the NFVO, and API version information retrieval. An alarm on a given NS results from either a collected virtualised resource fault impacting the connectivity of the NS instance or a VNF alarm, resulting from a virtualised resource alarm, issued by the VNFM for a VNF that is part of this NS instance.

The operations provided through this interface are:

- Get Alarm List
- Acknowledge Alarm
- Subscribe
- Query Subscription Information
- Terminate Subscription
- Notify

8.1a API version

For the NS fault management interface version as specified in the present document, the MAJOR version field shall be 1, the MINOR version field shall be 2, and the PATCH version field shall be 0 (see clause 9.1 of ETSI GS NFV-SOL 013 [16] for a definition of the version fields). Consequently, the {apiMajorVersion} URI variable shall be set to "v1".

NOTE: In the present document, there were no changes to the clauses defining the NS fault management interface that are visible at interface level compared to the previous version of the present document; hence, the MAJOR/MINOR/PATCH version fields are kept the same.

8.2 Resource structure and methods

All resource URIs of the API shall use the base URI specification defined in clause 4.1 of ETSI GS NFV-SOL 013 [16]. The string "nsfm" shall be used to represent {apiName}. All resource URIs in the clauses below are defined relative to the above base URI.

Figure 8.2-1 shows the overall resource URI structure defined for the NS fault management interface.

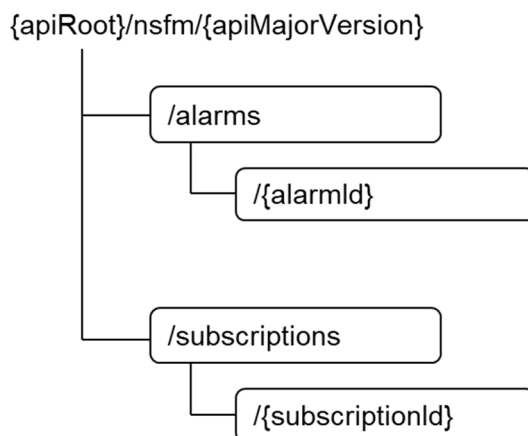


Figure 8.2-1: Resource URI structure of the NS Fault Management interface

Table 8.2-1 lists the individual resources defined, and the applicable HTTP methods.

The NFVO shall support responding to requests for all HTTP methods on the resources in Table 8.2-1 that are marked as "M" (mandatory) in the "Cat" column. The NFVO shall also support the "API versions" resources as specified in clause 9.3.2 of ETSI GS NFV-SOL 013 [16].

Table 8.2-1: Resources and methods overview of the NS Fault Management interface

Resource name	Resource URI	HTTP Method	Cat	Meaning
Alarms	/alarms	GET	M	Query alarms related to NS instances.
Individual alarm	/alarms/{alarmId}	GET	M	Read individual alarm.
		PATCH	M	Acknowledge individual alarm.
Subscriptions	/subscriptions	POST	M	Subscribe to alarms related to NSs.
		GET	M	Query multiple subscriptions.
Individual subscription	/subscriptions/{subscriptionId}	GET	M	Read an individual subscription.
		DELETE	M	Terminate a subscription.
Notification endpoint	(provided by API consumer)	POST	See note	Notify about NS alarms. See note.
		GET	See note	Test the notification endpoint. See note.
NOTE:	The NFVO shall support invoking the HTTP methods defined for the "Notification endpoint" resource exposed by the OSS/BSS. If the OSS/BSS supports invoking the POST method on the "Subscriptions" resource towards the NFVO, it shall also support responding to the HTTP requests defined for the "Notification endpoint" resource.			

8.3 Sequence diagrams (informative)

8.3.1 Flow of the Get Alarm List operation

This clause describes a sequence flow for querying one or multiple alarms.

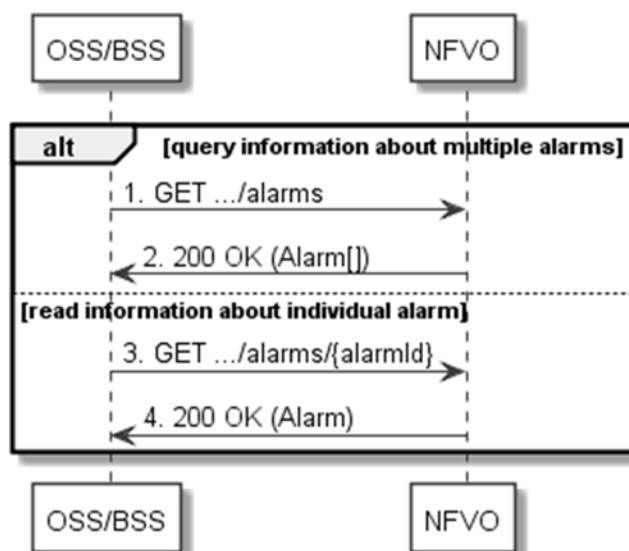


Figure 8.3.1-1: Flow of alarm query/read

Alarm query, as illustrated in Figure 8.3.1-1, consists of the following steps:

- 1) If the OSS/BSS intends to query all alarms, it sends a GET request to the "Alarms" resource.
- 2) The NFVO returns a "200 OK" response to the OSS/BSS, and includes zero or more data structures of type "Alarm" in the payload body.
- 3) If the OSS/BSS intends to read a particular alarm, it sends a GET request to the "Individual alarm" resource, addressed by the appropriate alarm identifier in its resource URI.
- 4) The NFVO returns a "200 OK" response to the OSS/BSS, and includes a data structure of type "Alarm" in the payload body.

Error handling: In case of failure, appropriate error information is provided in the response.

8.3.2 Flow of acknowledging alarm

This clause describes the procedure to acknowledge an individual alarm.

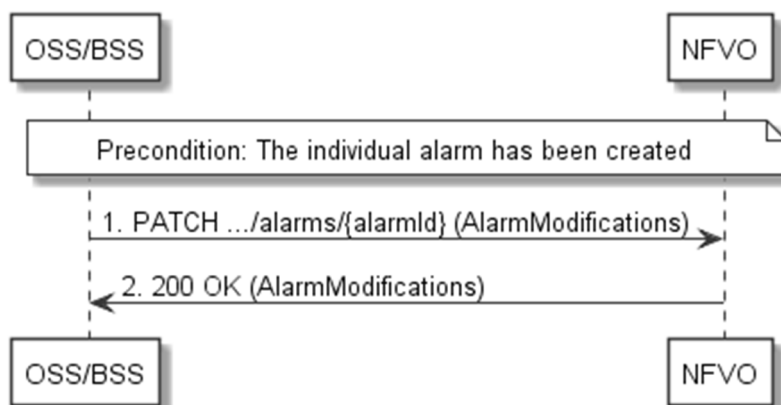


Figure 8.3.2-1: Flow of acknowledging alarm

Precondition: The resource representing the individual alarm has been created.

Acknowledge alarm, as illustrated in Figure 8.3.2-1, consists of the following steps:

- 1) The OSS/BSS sends a PATCH request to the individual alarm.

- 2) The NFVO returns a "200 OK" response to the OSS/BSS, and includes a data structure of type "AlarmModifications" in the payload body.

Error handling: In case of failure, appropriate error information is provided in the response.

8.3.3 Flow of managing subscriptions

This clause describes the procedure for creating, reading and terminating subscriptions to notifications related to NS fault management.

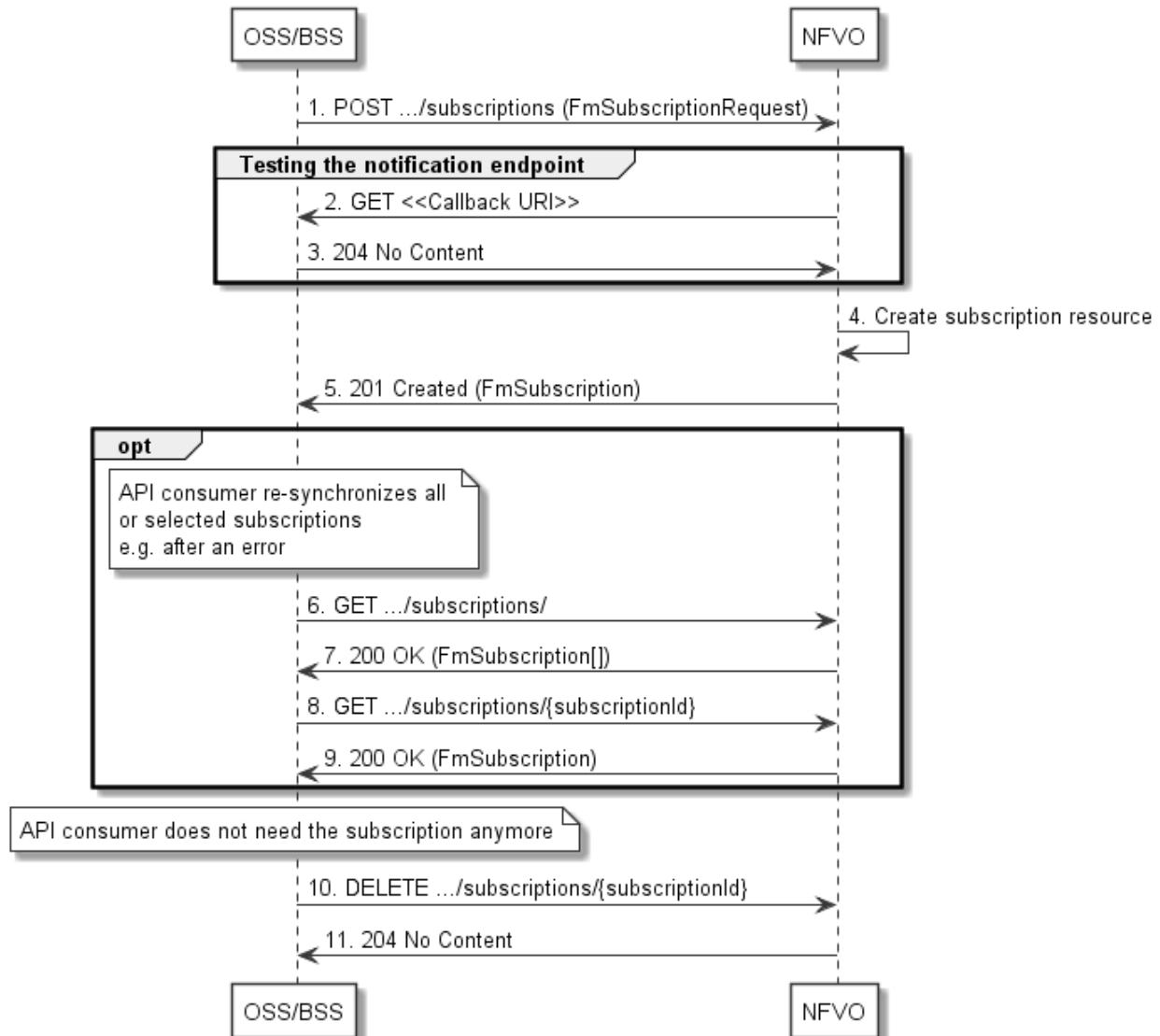


Figure 8.3.3-1: Flow of managing subscriptions

The procedure consists of the following steps as illustrated in Figure 8.3.3-1:

- 1) The OSS/BSS sends a POST request to the "Subscriptions" resource including in the payload body a data structure of type "FmSubscriptionRequest". This data structure contains filtering criteria and a callback URI to which the NFVO will subsequently send notifications about events that match the filter.
- 2) To test the notification endpoint that has been registered by the OSS/BSS as part of the subscription, the NFVO sends a GET request to the notification endpoint URI.
- 3) The OSS/BSS returns a "204 No Content" response to indicate success.

- 4) The NFVO creates a new subscription for notifications related to NS fault management, and a resource that represents this subscription.
- 5) The NFVO returns a "201 Created" response containing a data structure of type "FmSubscription", representing the "Individual subscription" resource just created by the NFVO, and provides the URI of the newly-created resource in the "Location" HTTP header.
- 6) If desired, e.g. to recover from an error situation, the OSS/BSS can query information about its subscriptions by sending a GET request to the "Subscriptions" resource.
- 7) In that case, the NFVO returns a "200 OK" response that contains the list of representations of all existing subscriptions that were created by the NFVO.
- 8) If desired, e.g. to recover from an error situation, the OSS/BSS can read information about a particular subscription by sending a GET request to the resource representing that individual subscription.
- 9) In that case, the NFVO returns a "200 OK" response that contains a representation of that individual subscription.
- 10) When the OSS/BSS does not need the subscription anymore, it terminates the subscription by sending a DELETE request to the resource that represents the individual subscription.
- 11) The NFVO acknowledges the successful termination of the subscription by returning a "204 No Content" response.

Error handling: The NFVO rejects a subscription if the subscription information is not valid: endpoint cannot be reached, subscription information is malformed, etc.

8.3.4 Flow of sending notifications

This clause describes the procedure for sending notifications related to NS fault management.

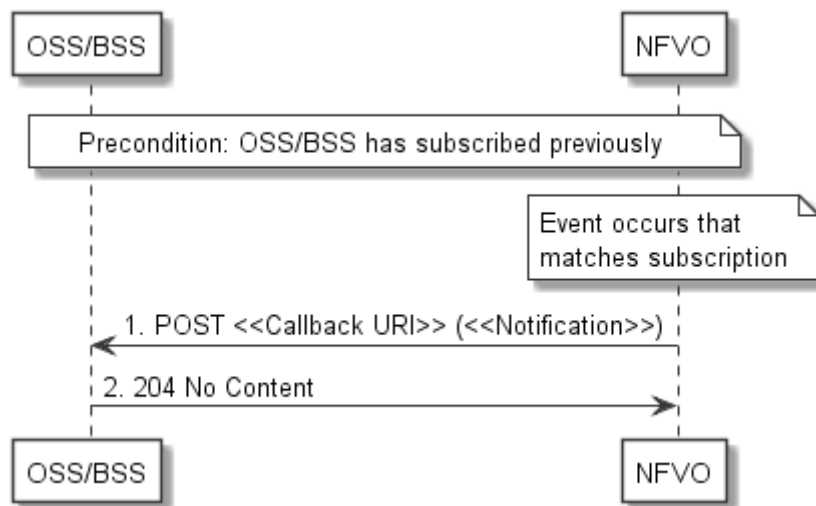


Figure 8.3.4-1: Flow of sending notifications

Precondition: The OSS/BSS has subscribed previously for notifications related to NS fault management.

The procedure consists of the following steps as illustrated in Figure 8.3.4-1:

- 1) If an event occurs that matches the filtering criteria which are part of the subscription, the NFVO generates a notification that includes information about the event, and sends it in the body of a POST request to the URI which the NFVO has registered as part of the subscription request. The variable <<Notification>> in the flow is a placeholder for the different types of notifications that can be sent by this API (see clauses 8.5.2.5, 8.5.2.6 and 8.5.2.7).

- 2) The OSS/BSS acknowledges the successful delivery of the notification by returning a "204 No Content" response.

Error handling: If the NFVO does not receive the "204 No Content" response from the OSS/BSS, it can retry sending the notification.

8.4 Resources

8.4.1 Introduction

This clause defines all the resources and methods provided by the NS fault management interface.

8.4.1a Resource: API versions

The "API versions" resources as defined in clause 9.3.3 of ETSI GS NFV-SOL 013 [16] are part of the NS fault management interface.

8.4.2 Resource: Alarms

8.4.2.1 Description

This resource represents a list of alarms related to NS instances.

8.4.2.2 Resource definition

The resource URI is:

{apiRoot}/nsfm/{apiMajorVersion}/alarms

This resource shall support the resource URI variables defined in Table 8.4.2.2-1.

Table 8.4.2.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 8.1a.

8.4.2.3 Resource methods

8.4.2.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

8.4.2.3.2 GET

The API consumer can use this method to retrieve information about the alarm list.

This method shall follow the provisions specified in the Tables 8.4.2.3.2-1 and 8.4.2.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 8.4.2.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
filter	0..1	<p>Attribute-based filtering expression according to clause 5.2 of ETSI GS NFV-SOL 013 [16].</p> <p>The NFVO shall support receiving this parameter as part of the URI query string. The OSS/BSS may supply this parameter.</p> <p>The following attribute names shall be supported by the NFVO in the filter expression:</p> <ul style="list-style-type: none"> - id - nsInstancelid - rootCauseFaultyComponent.faultyNestedNsInstancelid - rootCauseFaultyComponent.faultyNsVirtualLinkInstancelid - rootCauseFaultyComponent.faultyVnfInstancelid - rootCauseFaultyResource.faultyResourceType - eventType - perceivedSeverity - probableCause
nextpage_opaque_marker	0..1	Marker to obtain the next page of a paged response. Shall be supported by the NFVO if the NFVO supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource.

NOTE: There are no attribute selectors defined for this resource as the Alarm attributes with cardinality 0..1 or 0..N are not structurally complex in nature.

Table 8.4.2.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	Alarm	0..N	200 OK	<p>Shall be returned when information about zero or more alarms has been queried successfully.</p> <p>The response body shall contain in an array the representations of zero or more alarms as defined in clause 8.5.2.4.</p> <p>If the "filter" URI parameter was supplied in the request, the data in the response body shall have been transformed according to the rules specified in clause 5.2.2 of ETSI GS NFV-SOL 013 [16].</p> <p>If the NFVO supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource, inclusion of the Link HTTP header in this response shall follow the provisions in clause 5.4.2.3 of ETSI GS NFV-SOL 013 [16].</p>
	ProblemDetails	1	400 Bad Request	<p>Shall be returned upon the following error: Invalid attribute-based filtering expression.</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.</p>
	ProblemDetails	1	400 Bad Request	<p>Shall be returned upon the following error: Response too big.</p> <p>If the NFVO supports alternative 1 (error) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource, this error response shall follow the provisions in clause 5.4.2.2 of ETSI GS NFV-SOL 013 [16].</p>

	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.
--	----------------	------------------------	---------	---

8.4.2.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

8.4.2.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

8.4.2.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

8.4.3 Resource: Individual alarm

8.4.3.1 Description

This resource represents an individual alarm.

8.4.3.2 Resource definition

The resource URI is:

{apiRoot}/nsfm/{apiMajorVersion}/alarms/{alarmId}

This resource shall support the resource URI variables defined in Table 8.4.3.2-1.

Table 8.4.3.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 8.1a.
alarmId	Identifier of the alarm. See note.
NOTE:	This identifier can be retrieved from the "id" attribute of the "alarm" attribute in the AlarmNotification or AlarmClearedNotification. It can also be retrieved from the "id" attribute of the applicable array element in the payload body of the response to a GET request to the "Alarms" resource.

8.4.3.3 Resource methods

8.4.3.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

8.4.3.3.2 GET

The API consumer can use this method to read an individual alarm.

This method shall follow the provisions specified in the Tables 8.4.3.3.2-1 and 8.4.3.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 8.4.3.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
none supported		

Table 8.4.3.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	Alarm	1	200 OK	Shall be returned when information about an individual alarm has been read successfully. The response body shall contain a representation of the individual alarm.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

8.4.3.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

8.4.3.3.4 PATCH

This method modifies an individual alarm resource.

This method shall follow the provisions specified in the Tables 8.4.3.3.4-1 and 8.4.3.3.4-2 for URI query parameters, request and response data structures, and response codes.

Table 8.4.3.3.4-1: URI query parameters supported by the PATCH method on this resource

Name	Cardinality	Description
none supported		

Table 8.4.3.3.4-2: Details of the PATCH request/response on this resource

Request body	Data type	Cardinality	Description	
	AlarmModifications	1	The parameter for the alarm modification, as defined in clause 8.5.2.8. The Content-Type header shall be set to "application/merge-patch+json" according to IETF RFC 7396 [11].	
Response body	Data type	Cardinality	Response Codes	Description
	AlarmModifications	1	200 OK	Shall be returned when the request has been accepted and completed. The response body shall contain attribute modifications for an 'Individual alarm' resource (see clause 8.5.2.8).
	ProblemDetails	1	409 Conflict	Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the "Individual alarm" resource. Typically, this is due to the fact that the alarm is already in the state that is requested to be set (such as trying to acknowledge an already-acknowledged alarm). The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.
	ProblemDetails	0..1	412 Precondition failed	Shall be returned upon the following error: A precondition given in an HTTP request header is not fulfilled. Typically, this is due to an ETag mismatch, indicating that the resource was modified by another entity. The response body should contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.
ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.	

8.4.3.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

8.4.4 Resource: Subscriptions

8.4.4.1 Description

This resource represents subscriptions. The API consumer can use this resource to subscribe to notifications related to alarms related to an NS and to query its subscriptions.

8.4.4.2 Resource definition

The resource URI is:

{apiRoot}/nsfm/{apiMajorVersion}/subscriptions

This resource shall support the resource URI variables defined in Table 8.4.4.2-1.

Table 8.4.4.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 8.1a.

8.4.4.3 Resource methods

8.4.4.3.1 POST

The POST method creates a new subscription.

This method shall follow the provisions specified in the Tables 8.4.4.3.1-1 and 8.4.4.3.1-2 for URI query parameters, request and response data structures, and response codes.

As the result of successfully executing this method, a new "Individual subscription" resource as defined in clause 8.4.5 shall have been created. This method shall not trigger any notification.

Creation of two "Individual subscription" resources with the same callback URI and the same filter can result in performance degradation and will provide duplicates of notifications to the OSS, and might make sense only in very rare use cases. Consequently, the NFVO may either allow creating a new "Individual subscription" resource if another "Individual subscription" resource with the same filter and callback URI already exists (in which case it shall return the "201 Created" response code), or may decide to not create a duplicate "Individual subscription" resource (in which case it shall return a "303 See Other" response code referencing the existing "Individual subscription" resource with the same filter and callback URI).

Table 8.4.4.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Remarks
none supported		

Table 8.4.4.3.1-2: Details of the POST request/response on this resource

Request body	Data type	Cardinality	Description	
	FmSubscriptionRequest	1	Details of the subscription to be created, as defined in clause 8.5.2.2.	
Response body	Data type	Cardinality	Response Codes	Description
	FmSubscription	1	201 Created	<p>Shall be returned when the subscription has been created successfully.</p> <p>The response body shall contain a representation of the created "Individual subscription" resource.</p> <p>The HTTP response shall include a "Location:" HTTP header that points to the created "Individual subscription" resource.</p>
	n/a		303 See Other	<p>Shall be returned when a subscription with the same callback URI and the same filter already exists and the policy of the NFVO is to not create redundant subscriptions.</p> <p>The HTTP response shall include a "Location" HTTP header that contains the resource URI of the existing "Individual subscription" resource.</p> <p>The response body shall be empty.</p>

	ProblemDetails	1	422 Unprocessable Entity	<p>Shall be returned upon the following error: The content type of the payload body is supported and the payload body of a request contains syntactically correct data but the data cannot be processed.</p> <p>The general cause for this error and its handling is specified in clause 6.4 of ETSI GS NFV-SOL 013 [16], including rules for the presence of the response body.</p> <p>Specifically in case of this resource, the response code 422 shall also be returned if the NFVO has tested the Notification endpoint as described in clause 8.4.6.3.2 and the test has failed.</p> <p>In this case, the "detail" attribute in the "ProblemDetails" structure shall convey more information about the error.</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

8.4.4.3.2 GET

The API consumer can use this method to retrieve the list of active subscriptions for alarms related to an NS subscribed by the API consumer. It can be used e.g. for resynchronization after error situations.

This method shall follow the provisions specified in the Tables 8.4.4.3.2-1 and 8.4.4.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 8.4.4.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Remarks
filter	0..1	<p>Attribute-based filtering expression according to clause 5.2 of ETSI GS NFV-SOL 013 [16].</p> <p>The NFVO shall support receiving this parameter as part of the URI query string. The OSS/BSS may supply this parameter.</p> <p>All attribute names that appear in the FmSubscription and in data types referenced from it shall be supported by the NFVO in the filter expression.</p>
nextpage_opaque_marker	0..1	Marker to obtain the next page of a paged response. Shall be supported by the NFVO if the NFVO supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource.

Table 8.4.4.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	FmSubscription	0..N	200 OK	<p>Shall be returned when the list of subscriptions has been queried successfully.</p> <p>The response body shall contain in an array the representations of all active subscriptions of the functional block that invokes the method, i.e. zero or more representations of FM subscriptions, as defined in clause 8.5.2.3.</p> <p>If the "filter" URI parameter was supplied in the request, the data in the response body shall have been transformed according to the rules specified in clause 5.2.2 of ETSI GS NFV-SOL 013 [16].</p> <p>If the NFVO supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource, inclusion of the Link HTTP header in this response shall follow the provisions in clause 5.4.2.3 of ETSI GS NFV-SOL 013 [16].</p>
	ProblemDetails	1	400 Bad Request	<p>Shall be returned upon the following error: Invalid attribute-based filtering expression.</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.</p>
	ProblemDetails	1	400 Bad Request	<p>Shall be returned upon the following error: Response too big.</p> <p>If the NFVO supports alternative 1 (error) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource, this error response shall follow the provisions in clause 5.4.2.2 of ETSI GS NFV-SOL 013 [16].</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

8.4.4.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

8.4.4.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

8.4.4.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

8.4.5 Resource: Individual subscription

8.4.5.1 Description

This resource represents an individual subscription for alarms related to NSs. The API consumer can use this resource to read and to terminate a subscription to notifications related to NS fault management.

8.4.5.2 Resource definition

The resource URI is:

{apiRoot}/nsfm/{apiMajorVersion}/subscriptions/{subscriptionId}

This resource shall support the resource URI variables defined in Table 8.4.5.2-1.

Table 8.4.5.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 8.1a.
subscriptionId	Identifier of this subscription. See note.
NOTE:	This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request creating a new "Individual subscription" resource. It can also be retrieved from the "id" attribute in the payload body of that response.

8.4.5.3 Resource methods

8.4.5.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

8.4.5.3.2 GET

The API consumer can use this method for reading an individual subscription for alarms related to NSs subscribed by the API consumer.

This method shall follow the provisions specified in the Tables 8.4.5.3.2-1 and 8.4.5.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 8.4.5.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
none supported		

Table 8.4.5.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	FmSubscription	1	200 OK	Shall be returned when information about an individual subscription has been read successfully. The response body shall contain a representation of the "Individual subscription" resource.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

8.4.5.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

8.4.5.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

8.4.5.3.5 DELETE

This method terminates an individual subscription.

This method shall follow the provisions specified in the Tables 8.4.5.3.5-1 and 8.4.5.3.5-2 for URI query parameters, request and response data structures, and response codes.

As the result of successfully executing this method, the "Individual subscription" resource shall not exist any longer. This means that no notifications for that subscription shall be sent to the formerly-subscribed API consumer.

NOTE: Due to race conditions, some notifications might still be received by the formerly-subscribed API consumer for a certain time period after the deletion.

Table 8.4.5.3.5-1: URI query parameters supported by the DELETE method on this resource

Name	Cardinality	Description
none supported		

Table 8.4.5.3.5-2: Details of the DELETE request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	n/a		204 No Content	Shall be returned when the "Individual subscription" resource has been deleted successfully. The response body shall be empty.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

8.4.6 Resource: Notification endpoint

8.4.6.1 Description

This resource represents a notification endpoint for alarms related to NSs.

The API producer can use this resource to send notifications related to alarms related to NSs or about a rebuilt alarm list to a subscribed API consumer, which has provided the URI of this resource during the subscription process.

8.4.6.2 Resource definition

The resource URI is provided by the API consumer when creating the subscription.

This resource shall support the resource URI variables defined in Table 8.4.6.2-1.

Table 8.4.6.2-1: Resource URI variables for this resource

Name	Definition
none supported	

8.4.6.3 Resource methods

8.4.6.3.1 POST

The POST method notifies an alarm related to an NS or that the alarm list has been rebuilt. The API consumer shall have previously created an "individual subscription resource" with a matching filter.

This method shall follow the provisions specified in the Tables 8.4.6.3.1-1 and 8.4.6.3.1-2 for URI query parameters, request and response data structures, and response codes.

Table 8.4.6.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
none supported		

Each notification request body shall include exactly one of the alternatives defined in Table 8.4.6.3.1-2.

Table 8.4.6.3.1-2: Details of the POST request/response on this resource

Request body	Data type	Cardinality	Description	
	AlarmNotification	1	Information of an NS alarm.	
AlarmClearedNotification	1	Information of the clearance of an NS alarm.		
AlarmListRebuiltNotification	1	Information that the alarm list has been rebuilt by the NFVO.		
Response body	Data type	Cardinality	Response Codes	Description
	n/a		204 No Content	<p>Shall be returned when the notification has been delivered successfully.</p> <p>The response body shall be empty.</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

8.4.6.3.2 GET

The GET method allows the API producer to test the notification endpoint that is provided by the API consumer, e.g. during subscription.

This method shall follow the provisions specified in the Tables 8.4.6.3.2-1 and 8.4.6.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 8.4.6.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
none supported		

Table 8.4.6.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	n/a		204 No Content	Shall be returned to indicate that the notification endpoint has been tested successfully. The response body shall be empty.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

8.4.6.3.3 PUT

This method is not supported. When this method is requested on this resource, the OSS/BSS shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

8.4.6.3.4 PATCH

This method is not supported. When this method is requested on this resource, the OSS/BSS shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

8.4.6.3.5 DELETE

This method is not supported. When this method is requested on this resource, the OSS/BSS shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

8.5 Data Model

8.5.1 Introduction

This clause defines the request and response data structures of the NS fault management interface. If a request or response contains attributes not defined in the present document, a receiving functional block that does not understand these attributes shall not treat their presence as an error, and may choose to ignore them.

8.5.2 Resource and notification data types

8.5.2.1 Introduction

This clause defines the data structures to be used in the resource representations and notifications for the NS fault management interface.

8.5.2.2 Type: FmSubscriptionRequest

This type represents a subscription request related to notifications about NS faults. It shall comply with the provisions defined in Table 8.5.2.2-1.

Table 8.5.2.2-1: Definition of the FmSubscriptionRequest data type

Attribute name	Data type	Cardinality	Description
filter	FmNotificationsFilter	0..1	Filter settings for this subscription, to define the subset of all notifications this subscription relates to. A particular notification is sent to the subscriber if the filter matches, or if there is no filter.
callbackUri	Uri	1	The URI of the endpoint to send the notification to.
authentication	SubscriptionAuthentication	0..1	Authentication parameters to configure the use of Authorization when sending notifications corresponding to this subscription, as defined in clause 8.3.4 of ETSI GS NFV-SOL 013 [16]. This attribute shall only be present if the subscriber requires authorization of notifications.

8.5.2.3 Type: FmSubscription

This type represents a subscription related to notifications about NS faults. It shall comply with the provisions defined in Table 8.5.2.3-1.

Table 8.5.2.3-1: Definition of the FmSubscription data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this "Individual subscription" resource.
filter	FmNotificationsFilter	0..1	Filter settings for this subscription, to define the subset of all notifications this subscription relates to. A particular notification is sent to the subscriber if the filter matches, or if there is no filter.
callbackUri	Uri	1	The URI of the endpoint to send the notification to.
_links	Structure (inlined)	1	Links for this resource.
>self	Link	1	URI of this resource.

8.5.2.4 Type: Alarm

The alarm data type encapsulates information about an alarm. It shall comply with the provisions defined in Table 8.5.2.4-1.

Table 8.5.2.4-1: Definition of the Alarm data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this Alarm information element.
managedObjectId	Identifier	1	Identifier of the affected NS instance.
rootCauseFaultyComponent	FaultyComponentInfo	0..1	The NS components that are causing the NS fault. It shall be present if the alarm affects at least one NS component.
rootCauseFaultyResource	FaultyResourceInfo	0..1	The virtualised resources that are causing the NS fault. It shall be present when the faulty component is "NS Virtual Link" (see clause 8.5.3.4).
alarmRaisedTime	DateTime	1	Time stamp indicating when the alarm is raised by the managed object.
alarmChangedTime	DateTime	0..1	Time stamp indicating when the alarm was last changed. It shall be present if the alarm has been updated.
alarmClearedTime	DateTime	0..1	Time stamp indicating when the alarm was cleared. It shall be present if the alarm has been cleared.
alarmAcknowledgedTime	DateTime	0..1	Time stamp indicating when the alarm was acknowledged. It shall be present if the alarm has been acknowledged.

Attribute name	Data type	Cardinality	Description
ackState	Enum (inlined)	1	Acknowledgement state of the alarm. Permitted values: - UNACKNOWLEDGED - ACKNOWLEDGED
perceivedSeverity	PerceivedSeverityType	1	Perceived severity of the managed object failure.
eventTime	DateTime	1	Time stamp indicating when the fault was observed.
eventType	EventType	1	Type of event.
faultType	String	0..1	Additional information to clarify the type of the fault.
probableCause	String	1	Information about the probable cause of the fault.
isRootCause	Boolean	1	Attribute indicating if this fault is the root for other correlated alarms. If TRUE, then the alarms listed in the attribute "correlatedAlarmIds" are caused by this fault.
correlatedAlarmIds	Identifier	0..N	List of identifiers of other alarms correlated to this fault.
faultDetails	String	0..N	Provides additional information about the fault.
_links	Structure (inlined)	1	Links for this resource.
>self	Link	1	URI of this resource.
>objectInstance	Link	0..1	Link to the resource representing the NS instance to which the notified alarm is correlated. Shall be present if the NS instance information is accessible as a resource.

8.5.2.5 Type: AlarmNotification

This type represents an alarm notification about NS faults. It shall comply with the provisions defined in Table 8.5.2.5-1.

This notification shall be triggered by the NFVO when:

- An alarm has been created.
- An alarm has been updated, e.g. the severity of the alarm has changed.

Table 8.5.2.5-1: Definition of the AlarmNotification data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this notification. If a notification is sent multiple times due to multiple subscriptions, the "id" attribute of all these notifications shall have the same value.
notificationType	String	1	Discriminator for the different notification types. Shall be set to "AlarmNotification" for this notification type.
subscriptionId	Identifier	1	Identifier of the subscription that this notification relates to.
timeStamp	DateTime	1	Date-time of the generation of the notification.
alarm	Alarm	1	Information about an alarm including AlarmId, affected NS identifier, and FaultDetails.
_links	Structure (inlined)	1	Links to resources related to this notification.
>subscription	NotificationLink	1	Link to the related subscription.

8.5.2.6 Type: AlarmClearedNotification

This type represents an alarm cleared notification about NS faults. It shall comply with the provisions defined in Table 8.5.2.6-1.

The notification shall be triggered by the NFVO when an alarm has been cleared.

Table 8.5.2.6-1: Definition of the AlarmClearedNotification data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this notification. If a notification is sent multiple times due to multiple subscriptions, the "id" attribute of all these notifications shall have the same value.
notificationType	String	1	Discriminator for the different notification types. Shall be set to "AlarmClearedNotification" for this notification type.
subscriptionId	Identifier	1	Identifier of the subscription that this notification relates to.
timeStamp	DateTime	1	Date-time of the generation of the notification.
alarmId	Identifier	1	Alarm identifier.
alarmClearedTime	DateTime	1	The time stamp indicating when the alarm was cleared.
_links	Structure (inlined)	1	Links to resources related to this notification.
>subscription	NotificationLink	1	Link to the related subscription.
>alarm	NotificationLink	1	Link to the resource that represents the related alarm.

8.5.2.7 Type: AlarmListRebuiltNotification

This type represents a notification that the alarm list has been rebuilt, e.g. if the NFVO detects its storage holding the alarm list is corrupted. It shall comply with the provisions defined in Table 8.5.2.7-1.

The notification shall be triggered by the NFVO when the alarm list has been rebuilt, e.g. because the NFVO has detected that its storage holding the alarm list was corrupted.

Table 8.5.2.7-1: Definition of the AlarmListRebuiltNotification data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this notification. If a notification is sent multiple times due to multiple subscriptions, the "id" attribute of all these notifications shall have the same value.
notificationType	String	1	Discriminator for the different notification types. Shall be set to "AlarmListRebuiltNotification" for this notification type.
subscriptionId	Identifier	1	Identifier of the subscription that this notification relates to.
timeStamp	DateTime	1	Date-time of the generation of the notification.
_links	Structure (inlined)	1	Links to resources related to this notification.
>subscription	NotificationLink	1	Link to the related subscription.
>alarms	NotificationLink	1	Link to the alarm list, i.e. the "Alarms" resource.

8.5.2.8 Type: AlarmModifications

This type represents attribute modifications for an "Individual alarm" resource, i.e. modifications to a resource representation based on the "Alarm" data type. The attributes of "Alarm" that can be modified according to the provisions in clause 8.5.2.4 are included in the "AlarmModifications" data type.

The "AlarmModifications" data type shall comply with the provisions defined in Table 8.5.2.8-1.

Table 8.5.2.8-1: Definition of the AlarmModifications data type

Attribute name	Data type	Cardinality	Description
ackState	Enum (inlined)	1	New value of the "ackState" attribute in "Alarm". Permitted values: - ACKNOWLEDGED - UNACKNOWLEDGED

8.5.3 Referenced structured data types

8.5.3.1 Introduction

This clause defines data structures that can be referenced from data structures defined in the previous clauses, but can neither be resource representations nor bound to any subscribe/notify mechanism.

8.5.3.2 Type: FmNotificationsFilter

This type represents a subscription filter related to notifications about NS faults. It shall comply with the provisions defined in Table 8.5.3.2-1.

At a particular nesting level in the filter structure, the following applies: All attributes shall match in order for the filter to match (logical "and" between different filter attributes). If an attribute is an array, the attribute shall match if at least one of the values in the array matches (logical "or" between the values of one filter attribute).

Table 8.5.3.2-1: Definition of the FmNotificationsFilter data type

Attribute name	Data type	Cardinality	Description
nsInstanceSubscriptionFilter	NsInstanceSubscriptionFilter	0..1	Filter criteria to select NS instances about which to notify.
notificationTypes	Enum (inlined)	0..N	Match particular notification types. Permitted values: - AlarmNotification - AlarmClearedNotification - AlarmListRebuiltNotification See note.
faultyResourceTypes	FaultyResourceType	0..N	Match alarms related to NSs with a faulty resource type listed in this attribute.
perceivedSeverities	PerceivedSeverityType	0..N	Match alarms related to NSs with a perceived severity listed in this attribute.
eventTypes	EventType	0..N	Match alarms related to NSs with an event type listed in this attribute.
probableCauses	String	0..N	Match alarms related to NSs with a probable cause listed in this attribute.
NOTE: The permitted values of the "notificationTypes" attribute are spelled exactly as the names of the notification types to facilitate automated code generation systems.			

8.5.3.3 Type: FaultyResourceInfo

This type represents the faulty virtual resources that have a negative impact on an NS. It shall comply with the provisions defined in Table 8.5.3.3-1.

Table 8.5.3.3-1: Definition of the FaultyResourceInfo data type

Attribute name	Data type	Cardinality	Description
faultyResource	ResourceHandle	1	Information that identifies the faulty resource instance and its managing entity.
faultyResourceType	FaultyResourceType	1	Type of the faulty resource.

8.5.3.4 Type: FaultyComponentInfo

This type represents the faulty component that has a negative impact on an NS. It shall comply with the provisions defined in Table 8.5.3.4-1.

Table 8.5.3.4-1: Definition of the FaultyComponentInfo data type

Attribute name	Data type	Cardinality	Description
faultyNestedNsInstancelid	Identifier	0..1	Identifier of the faulty nested NS instance. See note.
faultyNsVirtualLinkInstancelid	Identifier	0..1	Identifier of the faulty NS virtual link instance. See note.
faultyVnfInstancelid	Identifier	0..1	Identifier of the faulty VNF instance. See note.
NOTE: At least one of the attributes shall be present.			

8.5.4 Referenced simple data types and enumerations

8.5.4.1 Introduction

This clause defines simple data types and enumerations that can be referenced from data structures defined in the previous clauses.

8.5.4.2 Simple data types

No particular simple data types are defined for this interface, in addition to those defined in clause 4.4.2.

8.5.4.3 Enumeration: PerceivedSeverityType

The enumeration PerceivedSeverityType shall comply with the provisions defined in Table 8.5.4.3-1. It indicates the relative level of urgency for operator attention.

Table 8.5.4.3-1: Enumeration PerceivedSeverityType

Enumeration value	Description
CRITICAL	The Critical severity level indicates that a service affecting condition has occurred and an immediate corrective action is required. Such a severity can be reported, for example, when a managed object becomes totally out of service and its capability needs to be restored (Recommendation ITU-T X.733 [14]).
MAJOR	The Major severity level indicates that a service affecting condition has developed and an urgent corrective action is required. Such a severity can be reported, for example, when there is a severe degradation in the capability of the managed object and its full capability needs to be restored (Recommendation ITU-T X.733 [14]).
MINOR	The Minor severity level indicates the existence of a non-service affecting fault condition and that corrective action should be taken in order to prevent a more serious (for example, service affecting) fault. Such a severity can be reported, for example, when the detected alarm condition is not currently degrading the capacity of the managed object (Recommendation ITU-T X.733 [14]).
WARNING	The Warning severity level indicates the detection of a potential or impending service affecting fault, before any significant effects have been felt. Action should be taken to further diagnose (if necessary) and correct the problem in order to prevent it from becoming a more serious service affecting fault (Recommendation ITU-T X.733 [14]).
INDETERMINATE	The Indeterminate severity level indicates that the severity level cannot be determined (Recommendation ITU-T X.733 [14]).
CLEARED	The Cleared severity level indicates the clearing of one or more previously reported alarms. This alarm clears all alarms for this managed object that have the same Alarm type, Probable cause and Specific problems (if given) (Recommendation ITU-T X.733 [14]).

8.5.4.4 Enumeration: EventType

The enumeration EventType represents those types of events that trigger an alarm. It shall comply with the provisions defined in Table 8.5.4.4-1.

Table 8.5.4.4-1: Enumeration EventType

Enumeration value	Description
COMMUNICATIONS_ALARM	An alarm of this type is associated with the procedure and/or process required conveying information from one point to another (Recommendation ITU-T X.733 [14]).
PROCESSING_ERROR_ALARM	An alarm of this type is associated with a software or processing fault (Recommendation ITU-T X.733 [14]).
ENVIRONMENTAL_ALARM	An alarm of this type is associated with a condition related to an enclosure in which the equipment resides (Recommendation ITU-T X.733 [14]).
QOS_ALARM	An alarm of this type is associated with degradation in the quality of a service (Recommendation ITU-T X.733 [14]).
EQUIPMENT_ALARM	An alarm of this type is associated with an equipment fault (Recommendation ITU-T X.733 [14]).

8.5.4.5 Enumeration: FaultyResourceType

The enumeration FaultyResourceType represents those types of faulty resource. It shall comply with the provisions defined in Table 8.5.4.5-1.

Table 8.5.4.5-1: Enumeration FaultyResourceType

Enumeration value	Description
COMPUTE	Virtual compute resource
STORAGE	Virtual storage resource
NETWORK	Virtual network resource

9 VNF Package Management interface

9.1 Description

This interface allows the OSS/BSS to invoke VNF package management operations towards the NFVO, and to subscribe to notifications regarding VNF package on-boarding or changes provided by the NFVO. Further, this interface allows API version information retrieval.

The operations provided through this interface are as follows:

- Create VNF Package Info
- Upload VNF Package
- Update VNF Package Info
- Delete VNF Package
- Query VNF Package, including obtaining the VNFD
- Fetch VNF Package
- Fetch VNF Package Artifacts
- Subscribe
- Query Subscription Info
- Notify
- Terminate Subscription

State changes of a VNF package are illustrated in clause B.2.

9.1a API version

For the VNF package management interface version as specified in the present document, the MAJOR version field shall be 2, the MINOR version field shall be 1, and the PATCH version field shall be 0 (see clause 9.1 of ETSI GS NFV-SOL 013 [16] for a definition of the version fields). Consequently, the {apiMajorVersion} URI variable shall be set to "v2".

9.2 Resource structure and methods

All resource URIs of the API shall use the base URI specification defined in clause 4.1 of ETSI GS NFV-SOL 013 [16]. The string "vnfpkgm" shall be used to represent {apiName}. All resource URIs in the clauses below are defined relative to the above base URI.

Figure 9.2-1 shows the overall resource URI structure defined for the VNF package management interface.

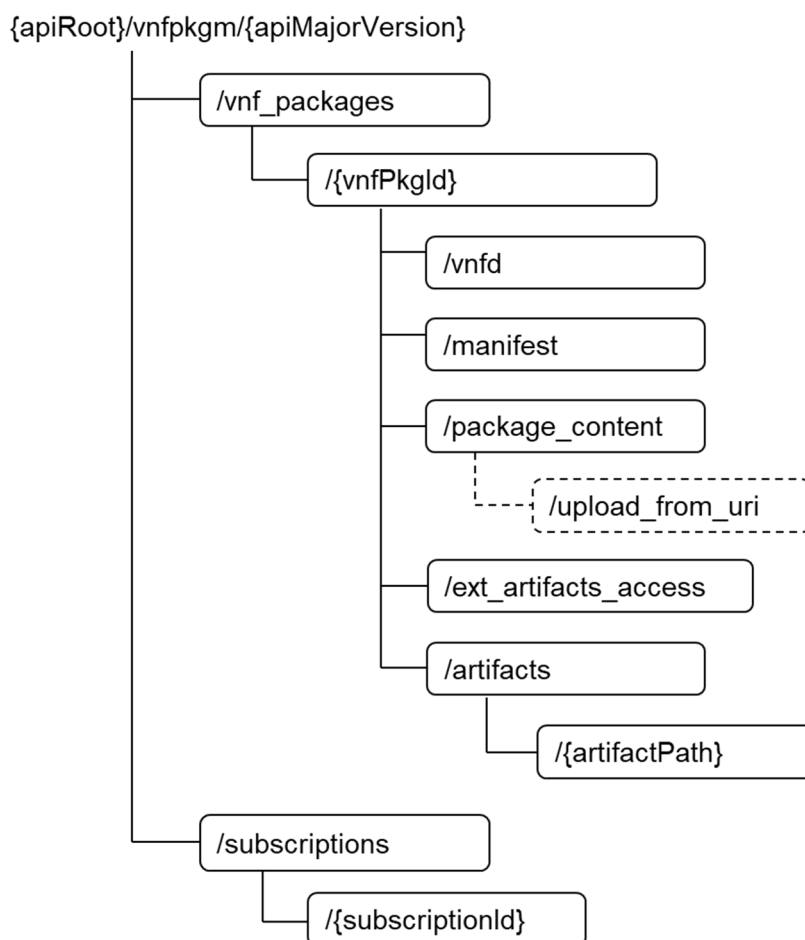


Figure 9.2-1: Resource URI structure of the VNF Package Management Interface

Table 9.2-1 lists the individual resources defined, and the applicable HTTP methods. The NFVO shall support responding to requests for all HTTP methods on the resources in Table 9.2-1 that are marked as "M" (mandatory) in the "Cat" column. The NFVO shall also support the "API versions" resources as specified in clause 9.3.2 of ETSI GS NFV-SOL 013 [16].

Table 9.2-1: Resources and methods overview of the VNF Package Management interface

Resource name	Resource URI	HTTP Method	Cat	Meaning
VNF packages	/vnf_packages	GET	M	Query VNF packages information.
		POST	M	Create a new individual VNF package resource.
Individual VNF package	/vnf_packages/{vnfPkgId}	GET	M	Read information about an individual VNF package.
		PATCH	M	Update information about an individual VNF package.
		DELETE	M	Delete an individual VNF package.
VNFD of an individual VNF package	/vnf_packages/{vnfPkgId}/vnfd	GET	M	Read VNFD of an on-boarded VNF package.
Manifest of an individual VNF package	/vnf_packages/{vnfPkgId}/manifest	GET	M	Read the manifest of an on-boarded VNF package
VNF package content	/vnf_packages/{vnfPkgId}/package_content	GET	M	Fetch an on-boarded VNF package.
		PUT	M	Upload a VNF package by providing the content of the VNF package.
VNF package artifacts	/vnf_packages/{vnfPkgId}/artifacts/	GET	M	Bulk-fetch artifacts that are not images.
Upload VNF package from URI task	/vnf_packages/{vnfPkgId}/package_content/upload_from_uri	POST	M	Upload a VNF package by providing the address information of the VNF package.
Access configuration for external artifact download	/vnf_packages/{vnfPkgId}/ext_artifacts_access	PUT	M	Provide access configuration information for the download of external artifacts.
		GET	M	Read access configuration information for the download of external artifacts.
Individual VNF package artifact	/vnf_packages/{vnfPkgId}/artifacts/{artifactPath}	GET	M	Fetch individual VNF package artifact.
Subscriptions	/subscriptions	POST	M	Subscribe to notifications related to on-boarding and/or changes of VNF packages.
		GET	M	Query multiple subscriptions.
Individual subscription	/subscriptions/{subscriptionId}	GET	M	Read information about an individual subscription.
		DELETE	M	Terminate a subscription.
Notification endpoint	(provided by API consumer)	POST	See note	Notify about VNF package on-boarding or change. See note.
		GET	See note	Test the notification endpoint. See note.

Resource name	Resource URI	HTTP Method	Cat	Meaning
NOTE:	The NFVO shall support invoking the HTTP methods defined for the "Notification endpoint" resource exposed by the OSS/BSS. If the OSS/BSS supports invoking the POST method on the "Subscriptions" resource towards the NFVO, it shall also support responding to the HTTP requests defined for the "Notification endpoint" resource.			

9.3 Sequence diagrams (informative)

9.3.1 Flow of the creation of an individual VNF package resource

This clause describes the procedure for creating an individual VNF package resource.

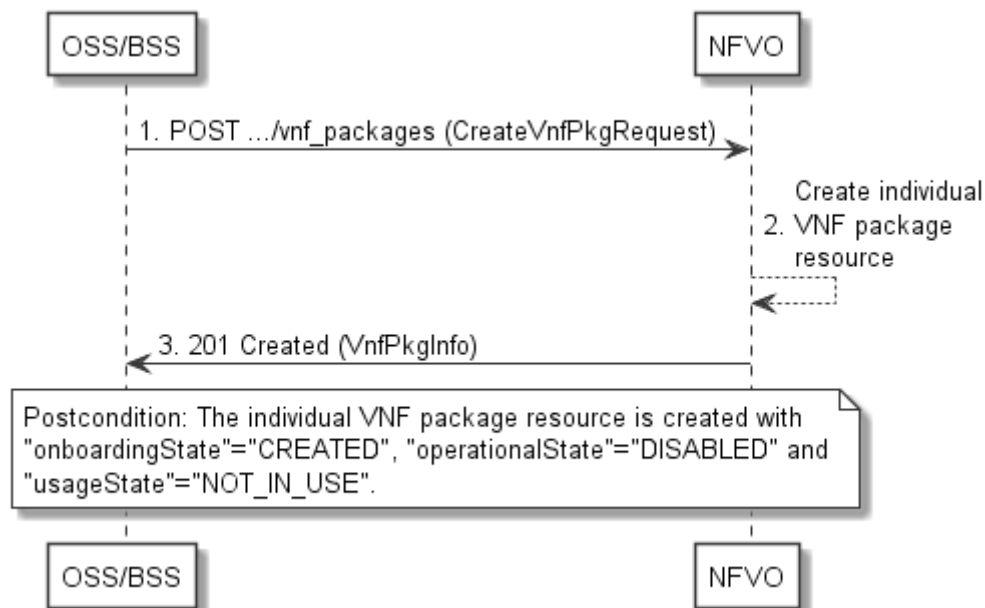


Figure 9.3.1-1: Flow of the creation of an individual VNF package resource

Creation of an individual VNF package resource, as illustrated in Figure 9.3.1-1, consists of the following steps:

- 1) The OSS/BSS sends a POST request to the "VNF packages" resource including in the payload body a data structure of type "CreateVnfPkgInfoRequest".
- 2) The NFVO creates a new individual VNF package resource.
- 3) The NFVO returns a "201 Created" response containing a representation of the individual VNF package resource and a "Location" HTTP header that points to the new "individual VNF package" resource.

Postcondition: Upon successful completion, the individual VNF package resource is created with the value of the "onboardingState" attribute equals to "CREATED", the value of the "operationalState" attribute equals to "DISABLED" and the value of "usageState" attribute equals to "NOT_IN_USE".

Error handling: In case of failure, appropriate error information is provided in the response.

9.3.2 Flow of the uploading of VNF package content

This clause describes the procedure of uploading the content of a VNF package.

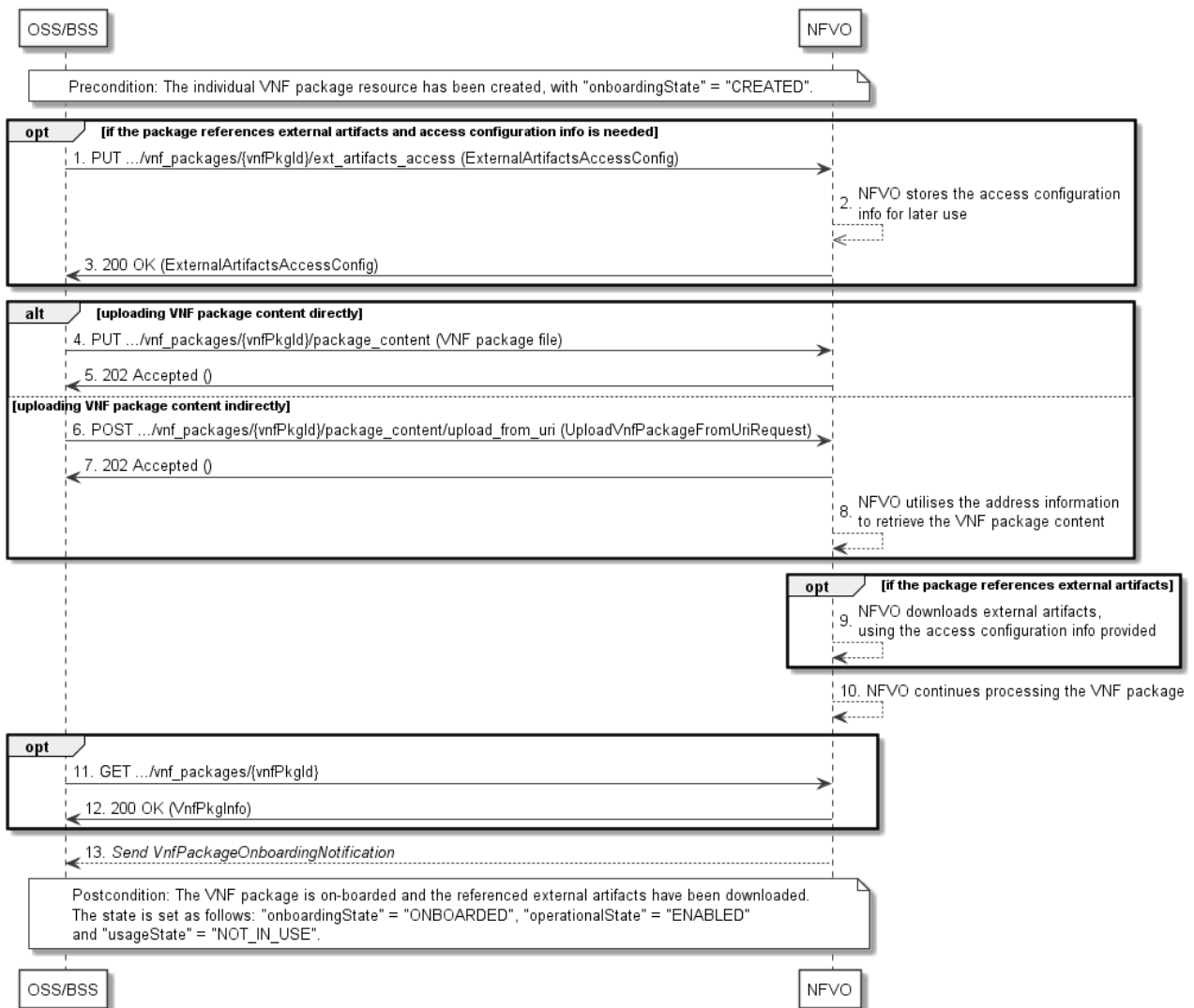


Figure 9.3.2-1: Flow of the uploading of VNF package content

Precondition: The individual VNF package resource has been created with the value of "onboardingState" attribute equals to "CREATED".

Uploading the content of a VNF package, as illustrated in Figure 9.3.2-1, consists of the following steps:

- 1) If the package to be uploaded references external artifacts and access configuration information is not available to the NFVO, the OSS/BSS provides access configuration information for the subsequent download of the external VNF package artifacts to the NFVO by sending a PUT request to the "Access configuration for external artifacts" resource including in the payload body a data structure of type "ExternalArtifactsAccessConfig".
- 2) In case step 1 was performed, the NFVO stores the information.
- 3) In case step 1 was performed, the NFVO returns a "200 OK" response with a copy of the stored information included in the payload body to indicate that the access configuration information was successfully stored.
- 4) If the OSS/BSS uploads the VNF package content directly to the NFVO, it sends a PUT request to the "VNF package content" resource including in the payload body a copy of the VNF package content. When the NFVO receives the PUT request, it sets the "onboardingState" to "UPLOADING". The NFVO then continues to receive the package data which might be a lengthy operation, depending in the size of the package.
- 5) The NFVO returns a "202 Accepted" response with an empty payload body.

- 6) If the OSS/BSS uploads the VNF package content indirectly to the NFVO, it sends a POST request to the "Upload VNF package from URI task" resource including in the payload body a data structure of type "UploadVnfPackageFromUriRequest".
- 7) The NFVO sets the "onboardingState" to "UPLOADING" and returns a "202 Accepted" response with an empty payload body to indicate the address information is successfully received.
- 8) The NFVO utilizes the address information to retrieve the VNF package content.
- 9) If the package references external artifacts, the NFVO utilizes the access configuration information and the information in the VNF package to download the external artifacts.
- 10) The NFVO sets the "onboardingState" to "PROCESSING". It then continues processing the VNF package (e.g. validation) after it has received the package content.
- 11) Optionally, the OSS/BSS can send a GET request to the "individual VNF package" resource to check the onboarding state of the VNF package resource.
- 12) The NFVO returns a "200 OK" response containing the information of the VNF package resource.
- 13) After successful processing, the NFVO sends a VnfPackageOnboardingNotification to the OSS/BSS to indicate the successful on-boarding of the VNF package content.

Postcondition: Upon successful completion, the content of the VNF package is on-boarded and the referenced external artifacts have been downloaded. The state of the VNF package is changed as follows: the value of the "onboardingState" attribute equals to "ONBOARDED", the value of the "operationalState" attribute equals to "ENABLED" and the value of the "usageState" attribute equals to "NOT_IN_USE".

Error handling: In case of failure, appropriate error information is provided in the response.

9.3.2a Flow of handling errors during upload of VNF package content

This clause describes how to handle errors during the uploading and processing of the content of a VNF package or external artifacts.

In case a long-running operation has failed, such as uploading of the VNF package, obtaining of external artifacts or processing/validation of the package, the "onboardingState" of the VNF package is set to "ERROR". Also, appropriate error information is provided in the "onboardingFailureDetails" attribute.

To handle the error, the OSS/BSS has the following options:

- 1) Delete the VNF Package resource and create a new one.
- 2) Upload the VNF package again, or upload a modified VNF package, by restarting the flow in clause 9.3.2 at step 4 or 6.
- 3) Provide new or updated access configuration information for the external artifacts as illustrated in Figure 9.3.2a-1.

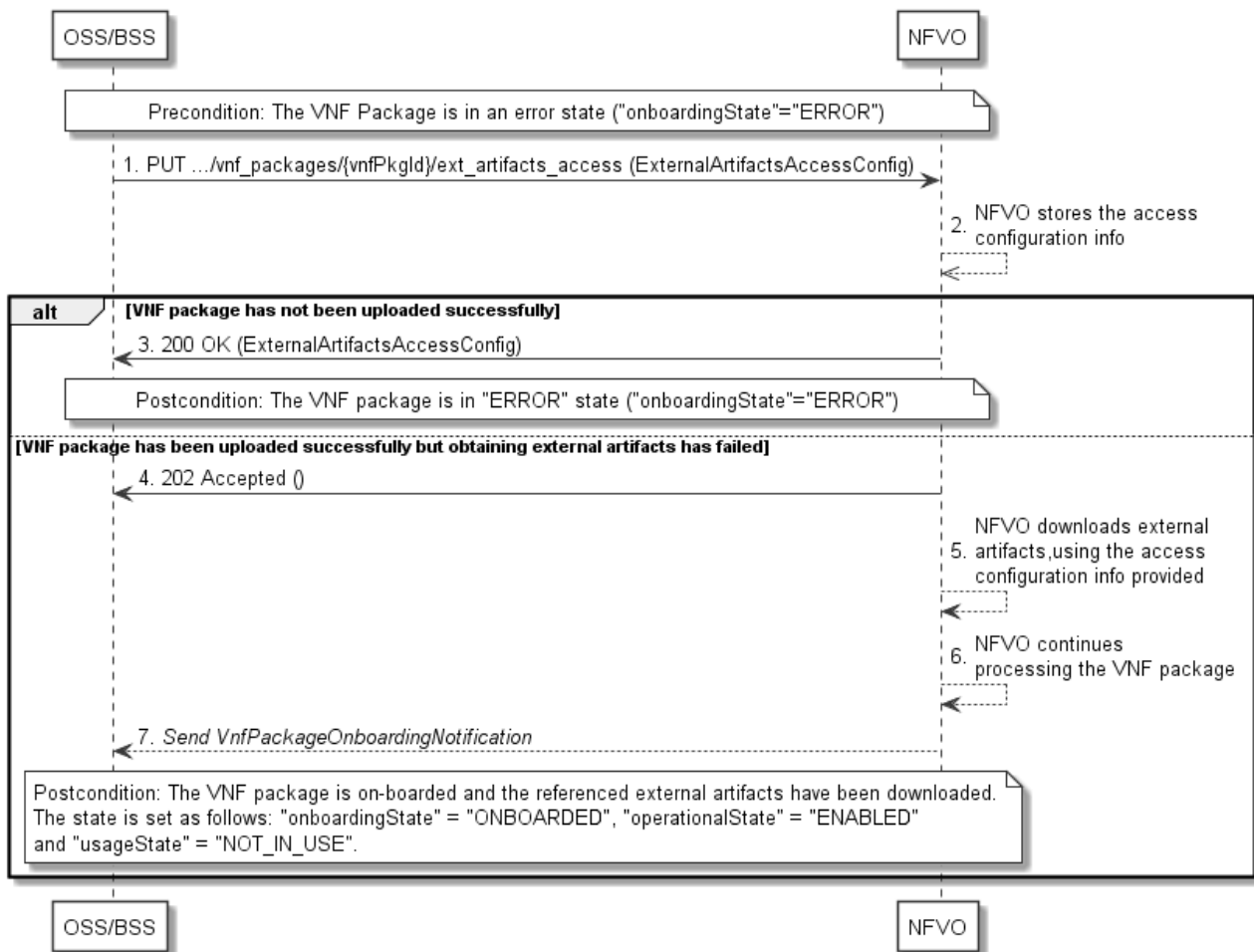


Figure 9.3.2a-1: Flow of re-downloading external VNF package artifacts in error case

Precondition: The VNF package is in an error state.

Providing new or updated access configuration information for the external artifacts, is handled in error state as illustrated in Figure 9.3.2a-1 and consists of the following steps:

- 1) The OSS/BSS provides access configuration information for the subsequent download of the external VNF package artifacts to the NFVO by sending a PUT request to the "Access configuration for external artifacts" resource including in the payload body a data structure of type "ExternalArtifactsAccessConfig".
- 2) The NFVO stores the information.

In case the VNF package has not been uploaded successfully previously:

- 3) The NFVO returns a "200 OK" response with a copy of the stored information included in the payload body to indicate that the access configuration information was successfully stored.

In case the VNF package has been uploaded successfully previously:

- 4) The NFVO sets the "onboardingState" to "UPLOADING" and returns a "202 Accepted" response with an empty payload body.
- 5) The NFVO utilizes the access configuration information and the information in the VNF package to retry the download of those external artifacts that were not successfully downloaded previously.
- 6) The NFVO sets the "onboardingState" to "PROCESSING". It then continues processing the VNF package (e.g. validation).
- 7) After successful processing, the NFVO sends a VnfPackageOnboardingNotification to the OSS/BSS to indicate the successful on-boarding of the VNF package content.

Postcondition:

- In case the VNF package was not successfully downloaded previously, the VNF package stays in "ERROR" state.
- In case the VNF package was successfully downloaded previously and this procedure was completed successfully, the content of the VNF package is on-boarded and the referenced external artifacts have been downloaded. The state of the VNF package is changed as follows: the value of the "onboardingState" attribute equals to "ONBOARDED", the value of the "operationalState" attribute equals to "ENABLED" and the value of the "usageState" attribute equals to "NOT_IN_USE".

Error handling: In case of failure, appropriate error information is provided in the response.

9.3.2b Flow of fetching the VNF package manifest

This clause describes a sequence for fetching the VNF package manifest.

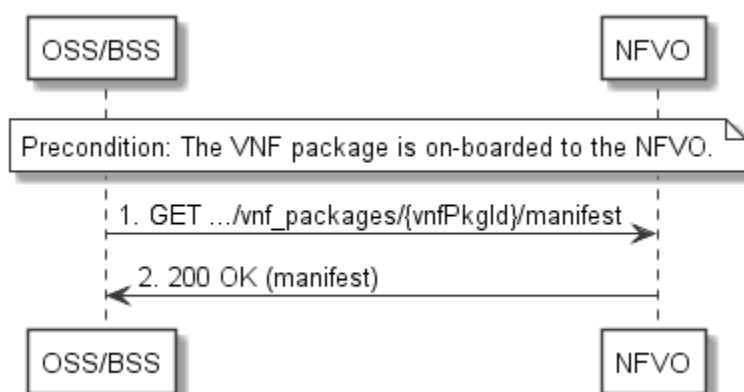


Figure 9.3.2b-1: Flow of fetching the VNF package manifest

Precondition: The VNF package is on-boarded to the NFVO.

Reading the manifest of an on-boarded VNF package, as illustrated in Figure 9.3.2b-1, consists of the following steps:

- 1) The OSS/BSS sends a GET request to the "Manifest of an individual VNF package" resource.
- 2) The NFVO returns a "200 OK" response with a payload body that contains a copy of the manifest file in the VNF package.
- 3) Error handling: In case of failure, appropriate error information is provided in the response.

9.3.3 Flow of querying/reading VNF package information

This clause describes a sequence for querying information about one or multiple VNF packages.

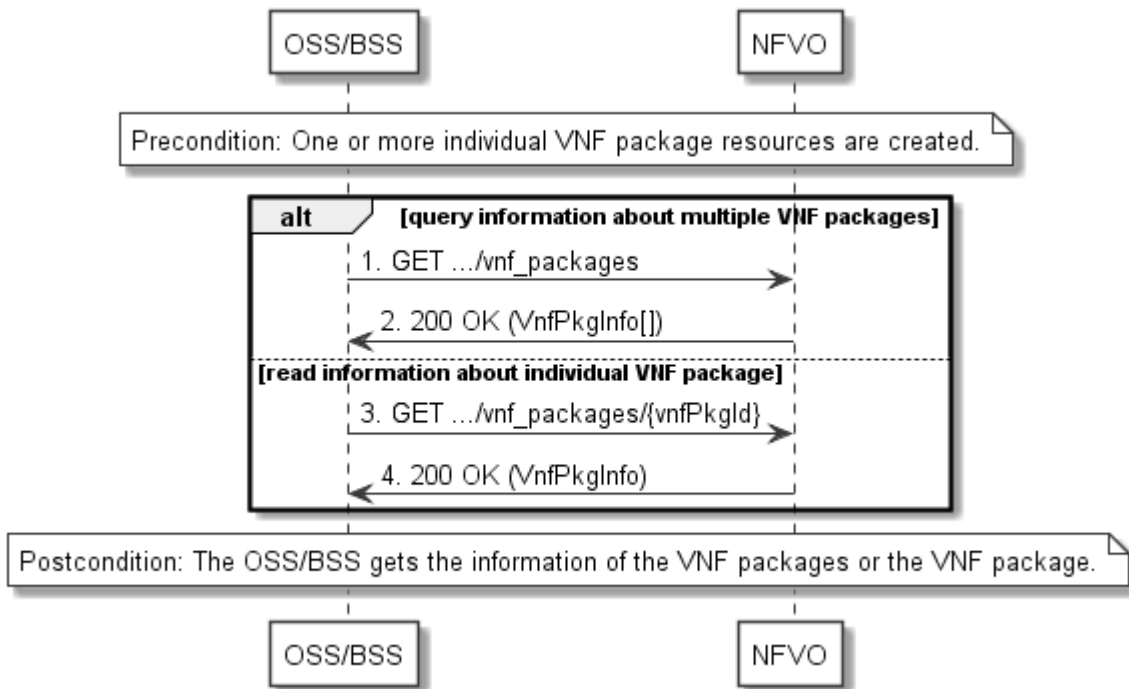


Figure 9.3.3-1: Flow of querying/reading VNF package information

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

VNF package information query, as illustrated in Figure 9.3.3-1, consists of the following steps:

- 1) If the OSS/BSS intends to query information about multiple VNF packages, it sends a GET request to the "VNF packages" resource.
- 2) The NFVO returns a "200 OK" response, and includes in the payload body zero or more data structures of type "VnfPkgInfo".
- 3) If the OSS/BSS intends to read information about a particular VNF package, the OSS/BSS sends a GET request to the "Individual VNF package" resource, addressed by the appropriate VNF package identifier in its resource URI.
- 4) The NFVO returns a "200 OK" response, and includes in the payload body a data structure of type "VnfPkgInfo".

Postcondition: Upon successful completion, the OSS/BSS gets the information of the VNF packages or the VNF package.

Error handling: In case of failure, appropriate error information is provided in the response.

9.3.4 Flow of reading the VNFD of an on-boarded VNF package

This clause describes the procedure for reading the VNFD of an on-boarded VNF package.

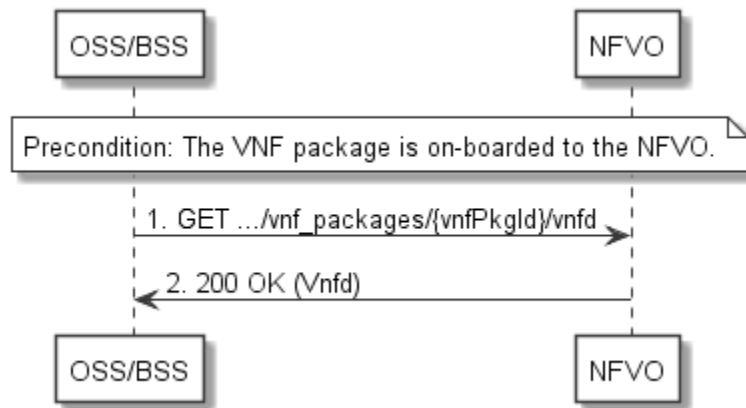


Figure 9.3.4-1: Flow of reading VNF D

Precondition: The VNF package is on-boarded to the NFVO.

The procedure consists of the following steps as illustrated in Figure 9.3.4-1:

- 1) The OSS/BSS sends a GET request to the "VNFD in an individual VNF package" resource.
- 2) The NFVO returns a "200 OK" response, and includes a copy of the VNFD from the VNF package in the payload body.

9.3.4a Flow of bulk-fetching VNF package artifacts that are not images

This clause describes a sequence for bulk-fetching artifacts that are not images from an on-boarded VNF package.

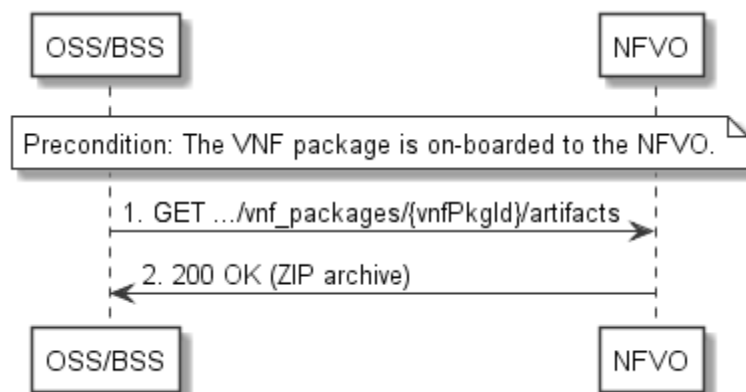


Figure 9.3.4a-1: Flow of bulk-fetching VNF package artifacts

Precondition: The VNF package is on-boarded to the NFVO.

Bulk-fetching of artifacts from an on-boarded VNF package allows the OSS/BSS to request all artifacts that are not images, and further may allow to specify additional filtering criteria for the artifacts to be included in that set.

NOTE: It is assumed that, due to their size, images are excluded from the bulk fetch operation. These can be fetched as individual artifacts.

The procedure consists of the following steps, as illustrated in Figure 9.3.4a-1:

- 1) The OSS/BSS sends a GET request to the "VNF package artifacts" resource and specifies, if supported, the appropriate URI query parameters to define the requested set of artifacts.
- 2) The NFVO returns a "200 OK" response with a payload body that contains a ZIP archive which contains the applicable artifacts, embedded in the appropriate directory structure in the VNF package.

Error handling: In case of failure, appropriate error information is provided in the response.

9.3.5 Flow of updating information of a VNF package

This clause describes the procedure for enabling/disabling/abort deletion of a VNF package.

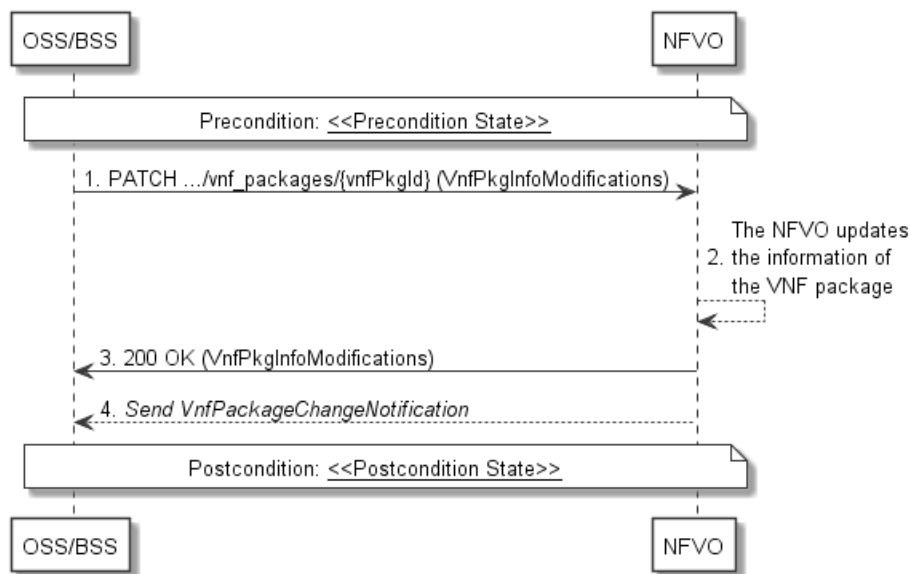


Figure 9.3.5-1: Flow of updating information of a VNF package

NOTE: Due to possible race conditions, the 200 response and the VnfPackageChangeNotification can arrive in any order at the OSS/BSS.

Precondition: The VNF package is in <<Precondition State>>, the value of <<Precondition State>> depends on the actual requested operation, and is described in Table 9.3.5-1.

The procedure consists of the following steps as illustrated in Figure 9.3.5-1:

- 1) The OSS/BSS sends a PATCH request to the "individual VNF package" resource including in the payload body a data structure of type "VnfPkgInfoModifications".
- 2) The NFVO updates the information of the VNF package.
- 3) The NFVO returns a "200 OK" response with a payload body containing a data structure of type "VnfPkgInfoModifications".
- 4) If the operational state of the VNF package is modified, the NFVO sends to OSS/BSS a VnfPackageChangeNotification to indicate the state change of the VNF package.

Postcondition: The VNF package is in << Postcondition State>>.

Table 9.3.5-1 describes how the <<Precondition State>> and << Postcondition State>> are parameterized in the above flow.

Table 9.3.5-1: Parameterization of the flow for updating information of a VNF package

Operation	<<Precondition State>>	<<PostconditionState>>
Enable a VNF package	The on-boarding state of the VNF package is ONBOARDED and the operational state of the VNF package is DISABLED	The operational state of the VNF package is ENABLED
Disable a VNF package	The on-boarding state of the VNF package is ONBOARDED and the operational state of the VNF package is ENABLED	The operational state of the VNF package is DISABLED
Update user defined data	The individual VNF package resource is created	The user defined data is updated

9.3.6 Flow of deleting a VNF package resource

This clause describes a sequence for deleting a VNF package resource.

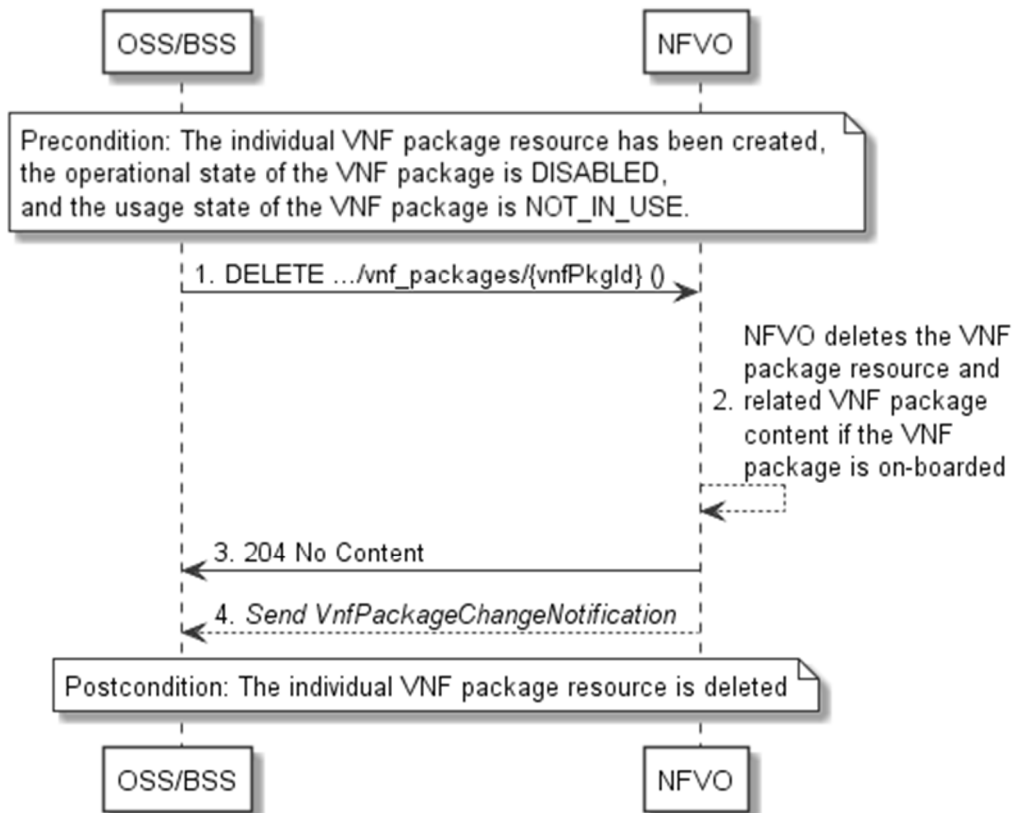


Figure 9.3.6-1: Flow of deleting a VNF package resource

NOTE: Due to possible race conditions, the 204 response and the `VnfPackageChangeNotification` can arrive in any order at the OSS/BSS.

Precondition: The individual VNF package resource has been created, the operational state of the VNF package is DISABLED, and the usage state of the VNF package is NOT_IN_USE.

Deleting a VNF package resource, as illustrated in Figure 9.3.6-1, consists of the following steps:

- 1) The OSS/BSS sends a DELETE request to the "individual VNF package" resource.
- 2) The NFVO deletes the "individual VNF package" resource and related VNF package content if the VNF package is on-boarded.
- 3) The NFVO returns a "204 No Content" response with an empty payload body.
- 4) The NFVO sends to the OSS/BSS a `VnfPackageChangeNotification` to indicate the deletion of the VNF package resource.

Postcondition: Upon successful completion, the individual VNF package resource is deleted.

Error handling: In case of failure, appropriate error information is provided in the response.

9.3.7 Flow of fetching an on-boarded VNF package

This clause describes a sequence for fetching the content of an on-boarded VNF package.

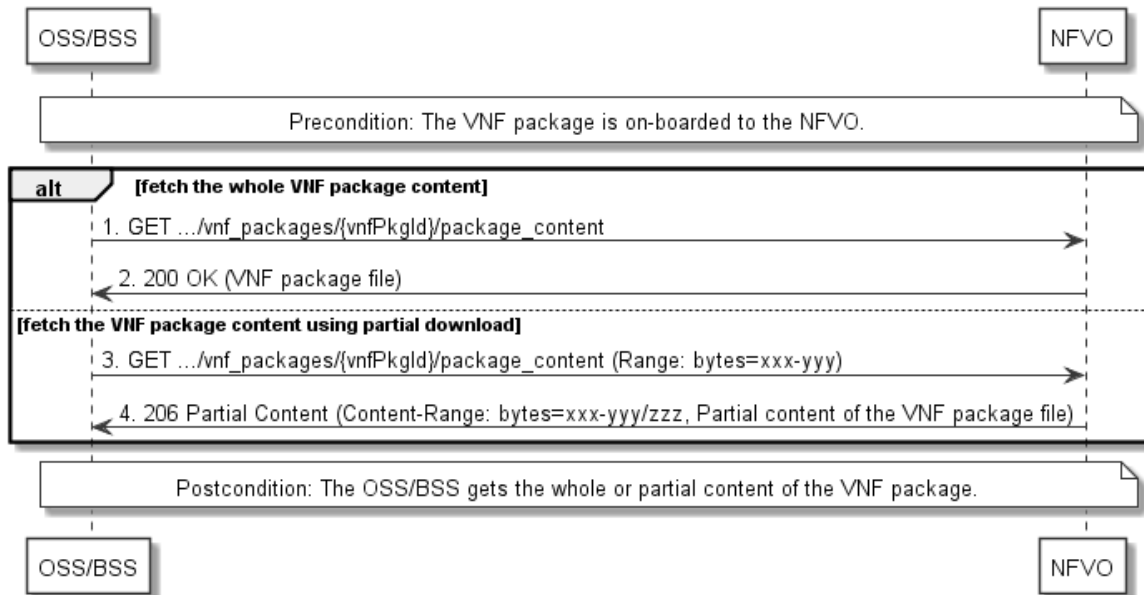


Figure 9.3.7-1: Flow of fetching an on-boarded VNF package

Precondition: The VNF package is on-boarded to the NFVO.

Fetching an on-boarded VNF package, as illustrated in Figure 9.3.7-1, consists of the following steps:

- 1) If fetching the whole VNF package content, the OSS/BSS sends a GET request to the "VNF package content" resource.
- 2) The NFVO returns a "200 OK" response, and includes a copy of the VNF package file in the payload body.
- 3) If fetching the VNF package content using partial download, the OSS/BSS sends a GET request to the "VNF package content" resource, and includes a "Range" HTTP header indicating the partition of the VNF package content needs to be transferred.
- 4) The NFVO returns a "206 Partial Content" response with a payload body containing the partial content of the VNF package, and a "Content-Range" HTTP header indicating the byte range enclosed in the payload and the complete length of the VNF package content.

Postcondition: Upon successful completion, the OSS/BSS gets the whole or partial content of the VNF package.

Error handling: In case of failure, appropriate error information is provided in the response.

9.3.8 Flow of fetching a VNF package artifact

This clause describes a sequence for fetching an individual artifact contained in an on-boarded VNF package.

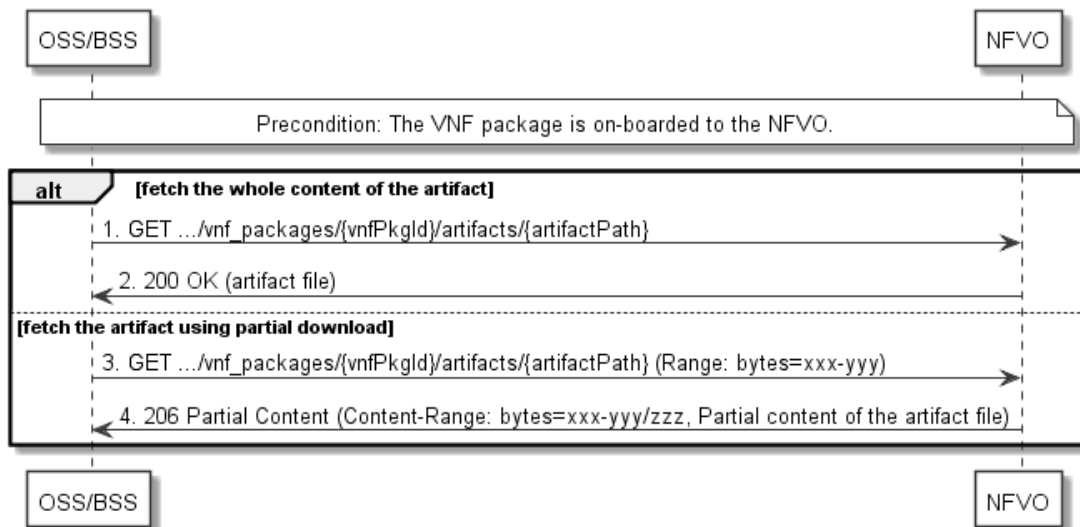


Figure 9.3.8-1: Flow of fetching a VNF package artifact

Precondition: The VNF package is on-boarded to the NFVO.

Fetching an individual artifact contained in an on-boarded VNF package, as illustrated in Figure 9.3.8-1, consists of the following steps:

- 1) If fetching the whole content of the artifact, the OSS/BSS sends a GET request to the "Individual VNF package artifact" resource.
- 2) The NFVO returns a "200 OK" response, and includes a copy of the applicable artifact file from the VNF package in the payload body.
- 3) If fetching the artifact using partial download, the OSS/BSS sends a GET request to the "Individual VNF package artifact" resource, and includes a "Range" HTTP header indicating the partition of the artifact needs to be transferred.
- 4) The NFVO returns a "206 Partial Content" response with a payload body containing the partial content of the artifact file, and a "Content-Range" HTTP header indicating the byte range enclosed in the payload and the complete length of the artifact file.

Error handling: In case of failure, appropriate error information is provided in the response.

9.3.9 Flow of managing subscriptions

This clause describes the procedure for creating, reading and terminating subscriptions to notifications related to VNF package management.

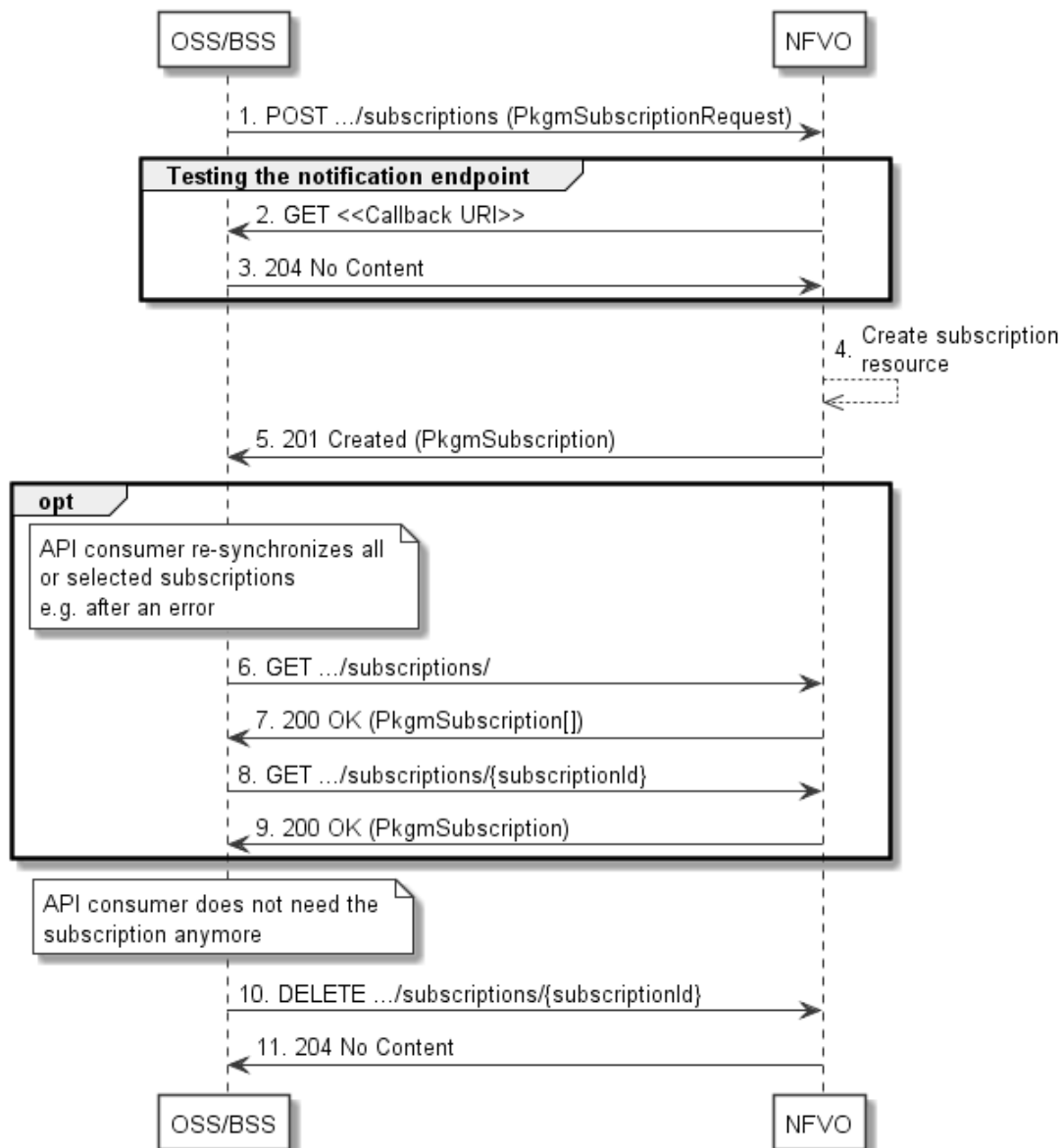


Figure 9.3.9-1: Flow of managing subscriptions

The procedure consists of the following steps as illustrated in Figure 9.3.9-1:

- 1) The OSS/BSS sends a POST request to the "Subscriptions" resource including in the payload body a data structure of type "PkgmSubscriptionRequest". That data structure contains filtering criteria and a callback URI to which the NFVO will subsequently send notifications about events that match the filter.
- 2) To test the notification endpoint that has been registered by the OSS/BSS as part of the subscription, the NFVO sends a GET request to the notification endpoint URI.
- 3) The OSS/BSS returns a "204 No Content" response to indicate success.
- 4) The NFVO creates a new subscription to notifications related to VNF package on-boarding or changes, and a resource that represents this subscription.
- 5) The NFVO returns a "201 Created" response containing a data structure of type "PkgmSubscription" representing the "Individual subscription" resource just created by the NFVO, and provides the URI of the newly-created resource in the "Location" HTTP header.
- 6) If desired, e.g. to recover from an error situation, the OSS/BSS can obtain information about its subscriptions by sending a GET request to the resource representing the subscriptions.

- 7) In that case, the NFVO returns a "200 OK" response that contains the list of representations of all existing subscriptions that were created by the OSS/BSS.
- 8) If desired, e.g. to recover from an error situation, the OSS/BSS can obtain information about a particular subscription by sending a GET request to the resource representing that individual subscription.
- 9) In that case, the NFVO returns a "200 OK" response that contains a representation of that individual subscription.
- 10) If the OSS/BSS does not need the subscription anymore, it terminates the subscription by sending a DELETE request to the resource that represents the individual subscription to remove.
- 11) The NFVO acknowledges the successful termination of the subscription by returning a "204 No Content" response.

Error handling: The NFVO rejects a subscription if the subscription information is not valid: endpoint cannot be reached, subscription information is malformed, etc.

9.3.10 Flow of sending notifications

This clause describes the procedure for sending notifications related to VNF package management.

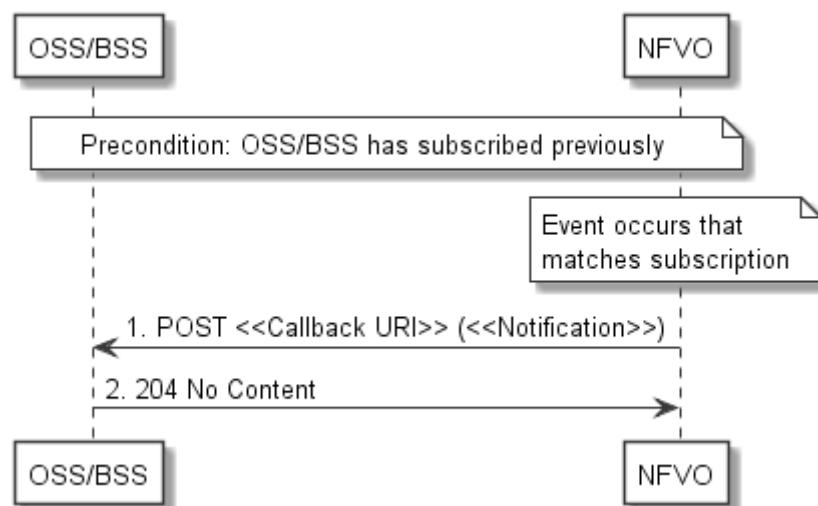


Figure 9.3.10-1: Flow of sending notifications

Precondition: The OSS/BSS has subscribed previously for notifications related to VNF package management.

The procedure consists of the following steps as illustrated in Figure 9.3.10-1:

- 1) If an event occurs that matches the filtering criteria which are part of the subscription, the NFVO generates a notification that includes information about the event, and sends it in the body of a POST request to the URI which the OSS/BSS has registered as part of the subscription request. The variable <<Notification>> in the flow is a placeholder for the different types of notifications that can be sent by this API (see clauses 9.5.2.8 and 9.5.2.9).
- 2) The OSS/BSS acknowledges the successful delivery of the notification by returning a "204 No Content" response.

Error handling: If the NFVO does not receive the "204 No Content" response from the OSS/BSS, it can retry sending the notification.

9.4 Resources

9.4.1 Introduction

This clause defines all the resources and methods provided by the VNF package management interface.

9.4.1a Resource: API versions

The "API versions" resources as defined in clause 9.3.3 of ETSI GS NFV-SOL 013 [16] are part of the VNF package management interface.

9.4.2 Resource: VNF packages

9.4.2.1 Description

This resource represents VNF packages. The API consumer can use this resource to create individual VNF package resources, and to query information of the VNF packages.

9.4.2.2 Resource definition

The resource URI is:

{apiRoot}/vnfpkgm/{apiMajorVersion}/vnf_packages

This resource shall support the resource URI variables defined in Table 9.4.2.2-1.

Table 9.4.2.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 9.1a.

9.4.2.3 Resource methods

9.4.2.3.1 POST

The POST method creates a new individual VNF package resource.

Upon successful creation of the "Individual VNF package" resource, the NFVO shall set the "onboardingState" attribute in the "VnPkgInfo" structure to "CREATED", the "operationalState" attribute to "DISABLED", and the "usageState" attribute to "NOT_IN_USE".

This method shall follow the provisions specified in the Tables 9.4.2.3.1-1 and 9.4.2.3.1-2 for URI query parameters, request and response data structures, and response codes.

Table 9.4.2.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
none supported		

Table 9.4.2.3.1-2: Details of the POST request/response on this resource

Request body	Data type	Cardinality	Description	
	CreateVnfPkgInfoRequest		1	Individual VNF package resource creation parameters, as defined in clause 9.5.2.2.
Response body	Data type	Cardinality	Response Codes	Description
	VnfPkgInfo	1	201 Created	<p>Shall be returned when an "Individual VNF package" resource has been created successfully.</p> <p>The response body shall contain a representation of the new individual VNF package resource, as defined in clause 9.5.2.5.</p> <p>The HTTP response shall include a "Location" HTTP header that contains the resource URI of the individual VNF package resource.</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

9.4.2.3.2 GET

The GET method queries the information of the VNF packages matching the filter.

This method shall follow the provisions specified in the Tables 9.4.2.3.2-1 and 9.4.2.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 9.4.2.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
filter	0..1	<p>Attribute-based filtering expression according to clause 5.2 of ETSI GS NFV-SOL 013 [16].</p> <p>The NFVO shall support receiving this parameter as part of the URI query string. The OSS/BSS may supply this parameter.</p> <p>All attribute names that appear in the VnfPkgInfo and in data types referenced from it shall be supported by the NFVO in the filter expression.</p>
all_fields	0..1	Include all complex attributes in the response. See clause 5.3 of ETSI GS NFV-SOL 013 [16] for details. The NFVO shall support this parameter.
fields	0..1	Complex attributes to be included into the response. See clause 5.3 of ETSI GS NFV-SOL 013 [16] for details. The NFVO should support this parameter.
exclude_fields	0..1	Complex attributes to be excluded from the response. See clause 5.3 of ETSI GS NFV-SOL 013 [16] for details. The NFVO should support this parameter.
exclude_default	0..1	<p>Indicates to exclude the following complex attributes from the response. See clause 5.3 of ETSI GS NFV-SOL 013 [16] for details.</p> <p>The NFVO shall support this parameter.</p> <p>The following attributes shall be excluded from the VnfPkgInfo structure in the response body if this parameter is provided, or none of the parameters "all_fields", "fields", "exclude_fields", "exclude_default" are provided:</p> <ul style="list-style-type: none"> - softwareImages - additionalArtifacts - userDefinedData - checksum - onboardingFailureDetails
nextpage_opaque_marker	0..1	Marker to obtain the next page of a paged response. Shall be supported by the NFVO if the NFVO supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource.

Table 9.4.2.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	VnfPkgInfo	0..N	200 OK	<p>Shall be returned when information about zero or more VNF packages has been queried successfully.</p> <p>The response body shall contain in an array the VNF package info representations that match the attribute filter, i.e. zero or more VNF package info representations as defined in clause 9.5.2.5.</p> <p>If the "filter" URI parameter or one of the "all_fields", "fields", "exclude_fields" or "exclude_default" URI parameters was supplied in the request and is supported, the data in the response body shall have been transformed according to the rules specified in clauses 5.2.2 and 5.3.2 of ETSI GS NFV-SOL 013 [16], respectively.</p> <p>If the NFVO supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource, inclusion of the Link HTTP header in this response shall follow the provisions in clause 5.4.2.3 of ETSI GS NFV-SOL 013 [16].</p>
	ProblemDetails	1	400 Bad Request	<p>Shall be returned upon the following error: Invalid attribute-based filtering expression.</p> <p>In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.</p>
	ProblemDetails	1	400 Bad Request	<p>Shall be returned upon the following error: Invalid attribute selector.</p> <p>In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.</p>
	ProblemDetails	1	400 Bad Request	<p>Shall be returned upon the following error: Response too big.</p> <p>If the NFVO supports alternative 1 (error) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource, this error response shall follow the provisions in clause 5.4.2.2 of ETSI GS NFV-SOL 013 [16].</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	<p>In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.</p>

9.4.2.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

9.4.2.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

9.4.2.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

9.4.3 Resource: Individual VNF package

9.4.3.1 Description

This resource represents an individual VNF package. The API consumer can use this resource to read information of the VNF package, update information of the VNF package, or delete a VNF package.

9.4.3.2 Resource definition

The resource URI is:

{apiRoot}/vnfpgm/{apiMajorVersion}/vnf_packages/{vnfPkgId}

This resource shall support the resource URI variables defined in Table 9.4.3.2-1.

Table 9.4.3.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 9.1a.
vnfPkgId	Identifier of the VNF package. The identifier is allocated by the NFVO. See note.
NOTE:	This identifier can be retrieved from the "VnfPkgId" attribute in the VnfPackageOnboardingNotification or VnfPackageChangeNotification.

9.4.3.3 Resource methods

9.4.3.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

9.4.3.3.2 GET

The GET method reads the information of an individual VNF package.

This method shall follow the provisions specified in the Tables 9.4.3.3.2-1 and 9.4.3.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 9.4.3.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
none supported		

Table 9.4.3.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	VnfPkgInfo	1	200 OK	Shall be returned when information of the VNF package has been read successfully. The response body shall contain the VNF package info representation defined in clause 9.5.2.5.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

9.4.3.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

9.4.3.3.4 PATCH

The PATCH method updates the information of a VNF package.

This method shall follow the provisions specified in the Tables 9.4.3.3.4-1 and 9.4.3.3.4-2 for URI query parameters, request and response data structures, and response codes.

Table 9.4.3.3.4-1: URI query parameters supported by the PATCH method on this resource

Name	Cardinality	Description
none supported		

Table 9.4.3.3.4-2: Details of the PATCH request/response on this resource

Request body	Data type	Cardinality	Description	
	VnfPkgInfoModifications	1	Parameters for VNF package information modifications. The Content-Type header shall be set to "application/merge-patch+json" according to IETF RFC 7396 [11].	
Response body	Data type	Cardinality	Response Codes	Description
	VnfPkgInfoModifications	1	200 OK	Shall be returned when the operation has been accepted and completed successfully. The response body shall contain attribute modifications for an "Individual VNF package" resource.
	ProblemDetails	1	409 Conflict	Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource. Typically, this is due to any of the following scenarios: <ul style="list-style-type: none"> - Disable a VNF package resource of which the operational state is not "ENABLED". - Enable a VNF package resource of which the operational state is not "DISABLED". The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

9.4.3.3.5 DELETE

The DELETE method deletes an individual VNF package resource.

This method shall follow the provisions specified in the Tables 9.4.3.3.5-1 and 9.4.3.3.5-2 for URI query parameters, request and response data structures, and response codes.

Table 9.4.3.3.5-1: URI query parameters supported by the DELETE method on this resource

Name	Cardinality	Description
none supported		

Table 9.4.3.3.5-2: Details of the DELETE request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	n/a		204 No Content	Shall be returned when the VNF package has been deleted successfully. The response body shall be empty.
	ProblemDetails	1	409 Conflict	Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource. Typically, this is due to the fact that the operational state of the VNF package resource is "ENABLED", or the usage state is "IN_USE" (i.e. the "Individual VNF instance" resource created from the concerned VNF package exists). The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

9.4.4 Resource: VNFD in an individual VNF package

9.4.4.1 Description

This resource represents the VNFD contained in an on-boarded VNF package. The API consumer can use this resource to obtain the content of the VNFD.

9.4.4.2 Resource definition

The resource URI is:

{apiRoot}/vnfpkgm/{apiMajorVersion}/vnf_packages/{vnfPkgId}/vnfd

This resource shall support the resource URI variables defined in Table 9.4.4.2-1.

Table 9.4.4.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 9.1a.
vnfPkgId	Identifier of the VNF package. The identifier is allocated by the NFVO. See note.
NOTE:	This identifier can be retrieved from the "vnfPkgId" attribute in the VnfPackageOnboardingNotification or VnfPackageChangeNotification.

9.4.4.3 Resource methods

9.4.4.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

9.4.4.3.2 GET

The GET method reads the content of the VNFD within a VNF package.

The VNFD is implemented as a collection of one or more files. A ZIP archive embedding these files shall be returned when reading this resource.

The default format of the ZIP archive shall comply with the CSAR format as specified in ETSI GS NFV-SOL 004 [5] where only the files representing the VNFD and information needed to navigate the ZIP archive and to identify the file that is the entry point for parsing the VNFD and (if requested) further security information are included, and software images as well as other artifacts referenced from the YAML files are excluded. This means that the structure of the ZIP archive shall correspond to the directory structure used in the VNF package and that the archive shall contain the following files from the package:

- TOSCA.meta (if available in the package).
- The main TOSCA definitions YAML file (either as referenced from TOSCA.meta or available as a file with the extension ".yaml" or ".yml" from the root of the archive).
- Every component of the VNFD referenced (recursively) from the main TOSCA definitions YAML file.

NOTE 1: For a VNFD based on TOSCA, it includes all the imported type definition files as indicated in the top level service template and in any of the lower level service template if it has any as described in ETSI GS NFV-SOL 001 [i.1].

NOTE 2: For a VNFD based on YANG, it includes the file as indicated by the "yang_definitions" keyname in the metadata section of the main yaml file as described in ETSI GS NFV-SOL 004 [5].

- The related security information, if the "include_signatures" URI parameter is provided, as follows:
 - the manifest file;
 - the singleton certificate file in the root of the VNF package (if available in the package);
 - the signing certificates of the individual files included in the ZIP archive (if available in the package);
 - the signatures of the individual files (if available in the package).

Three examples are provided below.

NOTE 3: These examples do not show the security related files.

EXAMPLE 1: Assuming a request is sent for the following VNF package (as described in clause A.1 in ETSI GS NFV-SOL 004 [5]):

```
!-----TOSCA-Metadata
!-----TOSCA.meta (metadata for navigating the ZIP file)

!-----Definitions
!----- MRF.yaml (main VNFD file)
!----- OtherTemplates (e.g. type definitions referenced by the main VNFD file)

!-----Files
!----- ChangeLog.txt
!----- image(s)
!----- other artifacts

!-----Tests
!----- file(s)

!-----Licenses
!----- file(s)

!-----Scripts
!----- install.sh
!----- MRF.mf
```

The NFVO will return a ZIP file of the following format:

```
!-----TOSCA-Metadata
```



```

!-----TOSCA.meta
!-----Definitions
!----- MRF.yaml
!----- OtherTemplates

```

EXAMPLE 2: Assuming a request is sent for the following VNF package (a VNF package without a TOSCA-Metadata directory, as described in clause A.2 in ETSI GS NFV-SOL 004 [5]):

```

!-----MRF.yaml (main VNFD file)
!-----MRF.mf
!-----ChangeLog.txt

!-----Tests
!----- file(s)

!-----Licenses
!----- file(s)

!-----Artifacts
!----- install.sh
!----- start.yang

```

The NFVO will return a ZIP file of the following format:

```
!-----MRF.yaml
```

EXAMPLE 3: Assuming a request is sent for the following VNF package (a VNF package with the YANG VNFD without a TOSCA-Metadata directory, as described in clause A.3 in ETSI GS NFV-SOL 004 [5]):

```

!----CompanyVNFD.yaml
!----CompanyVNFD.xml
!----CompanyVNFD.mf
!----ChangeLog.txt

!-----Files

!-----Instance Data Files
!----start.xml

!-----Licenses

!-----Scripts
!-----install.sh

```

The NFVO will return a ZIP file of the following format:

```

!----CompanyVNFD.yaml
!----CompanyVNFD.xml (indicated in the yang_definitions metadata in CompanyVNFD.yaml)

```

This method shall follow the provisions specified in the Tables 9.4.4.3.2-1 and 9.4.4.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 9.4.4.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
include_signatures	0..1	<p>If this parameter is provided, the NFVO shall include in the ZIP archive the security information as specified above.</p> <p>This URI query parameter is a flag, i.e. it shall have no value.</p> <p>The NFVO shall support this parameter.</p>

Table 9.4.4.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
n/a				
Response body	Data type	Cardinality	Response Codes	Description
	n/a	1	200 OK	<p>Shall be returned when the content of the VNFD has been read successfully.</p> <p>The payload body shall contain a ZIP archive that contains the files representing the VNFD, as specified above.</p> <p>The "Content-Type" HTTP header shall be set to "application/zip".</p>
	ProblemDetails	1	409 Conflict	<p>Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource.</p> <p>Typically, this is due to the fact that "onboardingState" of the VNF package has a value different from "ONBOARDED".</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

9.4.4.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

9.4.4.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

9.4.4.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

9.4.4a Resource: Access configuration for external artifacts

9.4.4a.1 Description

This resource represents access configuration information for the download of external artifacts. The OSS/BSS can use this resource to provide required credentials and address information for the downloading of external artifacts to the NFVO, enabling to retrieve the artifacts. The OSS/BSS can also read the information and can overwrite wrong information to trigger a retry of unsuccessful downloads.

9.4.4a.2 Resource definition

The resource URI is:

{apiRoot}/vnfpkgm/{apiMajorVersion}/vnf_packages/{vnfPkgId}/ext_artifacts_access

This resource shall support the resource URI variables defined in Table 9.4.4a.2-1.

Table 9.4.4a.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 9.1a.
vnfPkgId	Identifier of the VNF package. The identifier is allocated by the NFVO. See note.
NOTE:	This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request creating a new Individual VNF package resource.

9.4.4a.3 Resource methods

9.4.4a.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

9.4.4a.3.2 GET

The GET method reads the access configuration information that is used by the NFVO to get the content of external VNF package artifacts.

This method shall follow the provisions specified in the Tables 9.4.4a.3.2-1 and 9.4.4a.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 9.4.4a.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
none supported		

Table 9.4.4a.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	ExternalArtifactsAccessConfig	1	200 OK	Shall be returned when the access configuration information has been read successfully. If no information has been provided previously, an empty array shall be returned.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

9.4.4a.3.3 PUT

The PUT method provides the access configuration information for the NFVO to download the content of external VNF package artifacts.

As precondition for invoking this method, the individual VNF package resource shall have been created, and the value of "onboardingState" attribute shall equal to "CREATED" or "ERROR".

The resource representation in the payload body of the PUT request shall replace the current state of the resource.

This method shall follow the provisions specified in the Tables 9.4.4a.3.3-1 and 9.4.4a.3.3-2 for URI query parameters, request and response data structures, and response codes.

Table 9.4.4a.3.3-1: URI query parameters supported by the PUT method on this resource

Name	Cardinality	Description
none supported		

Table 9.4.4a.3.3-2: Details of the PUT request/response on this resource

Request body	Data type	Cardinality	Description	
	ExternalArtifactsAccessConfig	1	The payload body contains the access configuration information based on which the NFVO can obtain the external VNF package artifact files.	
Response body	Data type	Cardinality	Response Codes	Description
	ExternalArtifactsAccessConfig	1	200 OK	<p>Shall be returned when the attribute "onboardingState" = "CREATED" or when the attribute "onboardingState" = "ERROR" and the VNF package has not been uploaded successfully previously, to indicate that the access configuration information has been stored successfully by the NFVO.</p> <p>The response body shall contain a representation of the state of the resource.</p>
	n/a		202 Accepted	<p>Shall be returned when the attribute "onboardingState" = "ERROR" and the VNF package has been uploaded successfully previously, to indicate that the access configuration information has been stored successfully by the NFVO and the NFVO now starts downloading external artifacts.</p> <p>The response body shall be empty.</p> <p>Prior to returning the 202 response, the NFVO shall set the "onboardingState" attribute to "UPLOADING".</p> <p>Subsequently to returning the 202 response, the NFVO shall retry the download of those external artifacts that have failed downloading, or that were downloaded successfully previously and for which modified access configuration information has been provided in the request.</p>
	ProblemDetails	1	409 Conflict	<p>Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource.</p> <p>Typically, this is due to the fact that the "onboardingState" attribute contains a value different from "CREATED" or "ERROR".</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</p>
ProblemDetails	See clause 6.4 of [16]		4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

9.4.4a.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

9.4.4a.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

9.4.4b Resource: Manifest in an individual VNF package

9.4.4b.1 Description

This resource represents the manifest contained in an on-boarded VNF package. The API consumer can use this resource to obtain the content of the manifest.

9.4.4b.2 Resource definition

The resource URI is:

{apiRoot}/vnfpkgm/{apiMajorVersion}/vnf_packages/{vnfPkgId}/manifest

This resource shall support the resource URI variables defined in Table 9.4.4b.2-1.

Table 9.4.4b.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 9.1a.
vnfPkgId	Identifier of the VNF package. The identifier is allocated by the NFVO. See note.
NOTE:	This identifier can be retrieved from the "vnfPkgId" attribute in the VnfPackageOnboardingNotification or VnfPackageChangeNotification.

9.4.4b.3 Resource methods

9.4.4b.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

9.4.4b.3.2 GET

The GET method reads the content of the manifest within a VNF package.

This method shall follow the provisions specified in the Tables 9.4.4b.3.2-1 and 9.4.4b.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 9.4.4b.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
include_signatures	0..1	<p>If this parameter is provided, the NFVO shall return the manifest and related security information (such as certificate) in a ZIP archive.</p> <p>If this parameter is not given, the NFVO shall provide only a copy of the manifest file.</p> <p>This URI query parameter is a flag, i.e. it shall have no value.</p> <p>The NFVO shall support this parameter.</p>

Table 9.4.4b.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
n/a				
Response body	Data type	Cardinality	Response Codes	Description
	n/a	1	200 OK	<p>Shall be returned when the content of the manifest has been read successfully.</p> <p>If the "include_signatures" URI query parameter was absent in the request, or if the manifest file has all security-related information embedded (i.e. there is no separate certificate file), the payload body shall contain a copy of the manifest file of the VNF package, and the "Content-Type" HTTP header shall be set to "text/plain".</p> <p>If the "include_signatures" URI query parameter was present in the related request and the manifest file does not have all the security-related information embedded (i.e. there is a separate certificate file), the "Content-Type" HTTP header shall be set to "application/zip" and the payload body shall contain a ZIP archive which includes:</p> <ul style="list-style-type: none"> - a copy of the manifest file of the VNF package - a copy of the related individual certificate file.
	ProblemDetails	0..1	406 Not Acceptable	<p>If the related request contained an "Accept" header not compatible with the Content type "application/zip" but the "include_signatures" flag was provided, the NFVO shall respond with this response code.</p> <p>The "ProblemDetails" structure may be included with the "detail" attribute providing more information about the error.</p>
	ProblemDetails	1	409 Conflict	<p>Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource.</p> <p>Typically, this is due to the fact that "onboardingState" of the VNF package has a value different from "ONBOARDED".</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	<p>In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.</p>

9.4.4b.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

9.4.4b.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

9.4.4b.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

9.4.5 Resource: VNF package content

9.4.5.1 Description

This resource represents the content of VNF package identified by the VNF package identifier allocated by the NFVO. The API consumer can use this resource to fetch and to upload the content of the VNF package.

9.4.5.2 Resource definition

The resource URI is:

{apiRoot}/vnfpkgm/{apiMajorVersion}/vnf_packages/{vnfPkgId}/package_content

This resource shall support the resource URI variables defined in Table 9.4.5.2-1.

Table 9.4.5.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 9.1a.
vnfPkgId	Identifier of the VNF package. The identifier is allocated by the NFVO. See note.
NOTE:	This identifier can be retrieved from the "vnfPkgId" attribute in the VnfPackageOnboardingNotification or VnfPackageChangeNotification.

9.4.5.3 Resource methods

9.4.5.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

9.4.5.3.2 GET

The GET method fetches the content of a VNF package identified by the VNF package identifier allocated by the NFVO.

The content of the package is provided as onboarded, i.e. depending on the security option used, the CSAR or the CSAR wrapped in a ZIP archive together with an external signature is returned, as defined in clause 5.1 of ETSI GS NFV-SOL 004 [5].

NOTE: Information about the applicable security option can be obtained by evaluating the "packageSecurityOption" attribute in the "VnfPkgInfo" structure.

This method shall follow the provisions specified in the Tables 9.4.5.3.2-1 and 9.4.5.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 9.4.5.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
none supported		

Table 9.4.5.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			<p>The request may contain a "Range" HTTP header to obtain single range of bytes from the VNF package file. This can be used to continue an aborted transmission.</p> <p>If the Range header is present in the request and the NFVO does not support responding to range requests with a 206 response, it shall return a 200 OK response instead as defined below.</p>
Response body	Data type	Cardinality	Response Codes	Description
	n/a	1	200 OK	<p>Shall be returned when the whole content of the VNF package file has been read successfully.</p> <p>The response body shall include a copy of the VNF package file.</p> <p>The "Content-Type" HTTP header shall be set according to the type of the file, i.e. to "application/zip" for a VNF Package as defined in ETSI GS NFV-SOL 004 [5].</p>
	n/a	1	206 Partial Content	<p>If the NFVO supports range requests, this response shall be returned when a single consecutive byte range from the content of the VNF package file has been read successfully according to the request.</p> <p>The response body shall contain the requested part of the VNF package file.</p> <p>The "Content-Range" HTTP header shall be provided according to IETF RFC 7233 [10].</p> <p>The "Content-Type" HTTP header shall be set as defined above for the "200 OK" response.</p>
	ProblemDetails	1	409 Conflict	<p>Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource.</p> <p>Typically, this is due to the fact that "onboardingState" of the VNF package has a value different from "ONBOARDED".</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</p>
	ProblemDetails	0..1	416 Range Not Satisfiable	<p>Shall be returned upon the following error: The byte range passed in the "Range" header did not match any available byte range in the VNF package file (e.g. "access after end of file").</p> <p>The response body may contain a ProblemDetails structure.</p>
ProblemDetails	See clause 6.4 of [16]	4xx/5xx	<p>In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.</p>	

9.4.5.3.3 PUT

The PUT method uploads the content of a VNF package.

This method shall follow the provisions specified in the Tables 9.4.5.3.3-1 and 9.4.5.3.3-2 for URI query parameters, request and response data structures, and response codes.

Upon start of the upload of the package, the NFVO shall set the "onboardingState" attribute in the "VnfPkgInfo" structure to "UPLOADING". Upon successful upload of the package, if the package references external artifacts, the NFVO shall obtain the external artifacts. Subsequently, upon success, the NFVO shall set that attribute to "PROCESSING" and shall process the package, which shall include checking package consistency. Upon successful processing, the NFVO shall set the "onboardingState" attribute to "ONBOARDED", the "operationalState" attribute to "ENABLED", and the "usageState" attribute to "NOT_IN_USE". In addition, the NFVO shall set the value of the attributes in the "VnfPkgInfo" that are copied from the VNFD (refer to clause 9.5.2.5). If an error occurs during uploading the package, downloading the external artifacts or processing the package, the NFVO shall set the "onboardingState" attribute to "ERROR" and shall populate the "onboardingFailureDetails" attribute in "VnfPkgInfo".

Table 9.4.5.3.3-1: URI query parameters supported by the PUT method on this resource

Name	Cardinality	Description
none supported		

Table 9.4.5.3.3-2: Details of the PUT request/response on this resource

	Data type	Cardinality	Description	
	Request body	n/a	1	<p>The payload body contains a ZIP file that represents the VNF package.</p> <p>The "Content-Type" HTTP header shall be set according to the type of the file, i.e. to "application/zip" for a VNF Package as defined in ETSI GS NFV-SOL 004 [5].</p>
Response body	Data type	Cardinality	Response Codes	Description
	n/a		202 Accepted	<p>Shall be returned when the VNF package has been accepted for uploading, but the processing has not been completed. It is expected to take some time for processing.</p> <p>The response body shall be empty. See note.</p>
	ProblemDetails	1	409 Conflict	<p>Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource.</p> <p>Typically, this is due to the fact that the onboarding state of the VNF package resource is not "CREATED" or "ERROR".</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.
NOTE: The API consumer can track the uploading progress by receiving the "VnfPackageOnBoardingNotification" from the NFVO or by reading the status of the individual VNF package resource using the GET method.				

9.4.5.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

9.4.5.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

9.4.5a Resource: VNF package artifacts

9.4.5a.1 Description

This resource represents the artifacts contained in a VNF package. The API consumer can use this resource to bulk-fetch the artifacts.

Optional filters allow to restrict the set of artifacts included in the resource representation. In the present version of the present document, image artifacts are excluded from the representation of this resource.

9.4.5a.2 Resource definition

The possible resource URI is:

{apiRoot}/vnfpkgm/{apiMajorVersion}/vnf_packages/{vnfPkgId}/artifacts

This resource shall support the resource URI variables defined in Table 9.4.5a.2-1.

Table 9.4.5a.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 9.1a.
vnfPkgId	Identifier of the VNF package. The identifier is allocated by the NFVO. See note.
NOTE:	This identifier can be retrieved from the "vnfPkgId" attribute in the VnfPackageOnboardingNotification or VnfPackageChangeNotification.

9.4.5a.3 Resource methods

9.4.5a.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

9.4.5a.3.2 GET

The GET method shall return an archive that contains a set of artifacts according to the provisions for inclusion/exclusion defined below, embedded in a directory structure being the same as in the VNF package.

The criteria for exclusion/inclusion of an artifact in the archive are defined as follows:

- Artifacts that are software images shall be excluded from the archive.
- Artifacts that are not software images and that are external to the VNF package shall be excluded from the archive unless the URI query parameter "include_external_artifacts" has been provided. External artifacts shall be included in the archive using the content of the "artifactPath" attribute as the path.
- All additional artifacts included in the VNF package that are MANO artifacts shall be included in the archive, unless the URI query parameter "exclude_all_mano_artifacts" has been provided, in which case such artifacts shall be excluded.
- All additional artifacts included in the VNF package that are non-MANO artifacts shall be included in the archive, unless:
 - the URI query parameter "exclude_all_non_mano_artifacts" has been provided, in which case such artifacts shall be excluded;
 - the URI query parameter "select_non_mano_artifact_sets" has been provided and is supported by the NFVO, in which case only those non-MANO artifacts shall be included whose non-MANO artifact set identifier matches one of the values of the query parameter.

Package metadata such as manifest file or VNFD shall not be included in the archive.

This method shall follow the provisions specified in the Tables 9.4.5a.3.2-1 and 9.4.5a.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 9.4.5a.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
exclude_all_mano_artifacts	0..1	Flag (i.e. parameter without value) that instructs the NFVO to exclude the set of additional MANO artifacts (i.e. those that are not images) from the response payload body. The NFVO shall support this parameter. The OSS/BSS may supply this parameter.
exclude_all_non_mano_artifacts	0..1	Flag (i.e. parameter without value) that instructs the NFVO to exclude the set of non-MANO artifacts from the response payload body. The NFVO shall support this parameter. The OSS/BSS may supply this parameter.
include_external_artifacts	0..1	Flag (i.e. parameter without value) that instructs the NFVO to include external artifacts in the response payload body. It shall not be treated as an error if this flag is provided but there is no external artifact to include in the result. If this parameter is missing, no external artifacts shall be included. The NFVO shall support this parameter. The OSS/BSS may supply this parameter.
select_non_mano_artifact_sets	0..1	Comma-separated list of non-MANO artifact set identifiers for which the artifacts are to be included in the response body. The NFVO should support this parameter. If the NFVO does not support this parameter, it shall ignore it, i.e. provide a response as if no parameter was provided. The OSS/BSS may supply this parameter.
include_signatures	0..1	If this parameter is provided, the NFVO shall include in the ZIP archive the individual signatures and, if provided, related certificates for the included artifacts, in the format in which they are provided in the VNF package. If this parameter is not given, the NFVO shall only provide copies of the artifact files. This URI query parameter is a flag, i.e. it shall have no value. The NFVO shall support this parameter.

Table 9.4.5a.3.2-2: Details of the GET request/response on this resource

	Data type	Cardinality	Description
Request body	n/a		The "Accept" HTTP header shall be set to "application/zip". The request may contain a "Range" HTTP header to obtain single range of bytes from the archive containing the artifacts. This can be used to continue an aborted transmission. If the "Range" header is present in the request and the NFVO does not support responding to range requests with a 206 response, it shall return a 200 OK response instead as defined below.

	Data type	Cardinality	Response Codes	Description
Response body	n/a	1	200 OK	<p>Shall be returned when the whole content of the archive containing the artifact files has been read successfully.</p> <p>The payload body shall be a ZIP archive containing the requested set of artifacts selected according to the provisions specified above in this clause, and, if the flag "include_signatures" was provided in the related request, the applicable signature files and, if available, the separate certificate files from the VNF package.</p> <p>The "Content-Type" HTTP header shall be set to "application/zip".</p>
	n/a	1	206 Partial Content	<p>If the NFVO supports range requests, this response shall be returned when a single consecutive byte range from the content of the archive that would have been returned in a "200 OK" response has been read successfully according to the request.</p> <p>The response body shall contain the requested part of the archive.</p> <p>The "Content-Type" HTTP header shall be set to "application/zip".</p> <p>The "Content-Range" HTTP header shall be provided according to IETF RFC 7233 [10].</p>
	ProblemDetails	1	400 Bad Request	<p>Shall be returned upon the following error: Invalid URI query parameters, including the following situations:</p> <ul style="list-style-type: none"> - "exclude_all_non_mano_artifacts" and "select_non_mano_artifact_sets" are both present in the request; - "exclude_all_non_mano_artifacts" and "exclude_all_mano_artifacts" are both present in the request; - one or more of the values provided in "select_non_mano_artifact_sets" are not defined in the manifest of the VNF package. <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.</p>
	ProblemDetails	1	409 Conflict	<p>Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource.</p> <p>Typically, this is due to the fact that "onboardingState" of the VNF package has a value different from "ONBOARDED".</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</p>
	ProblemDetails	0..1	416 Range Not Satisfiable	<p>Shall be returned upon the following error: The byte range passed in the "Range" header did not match any available byte range in the archive file (e.g. "access after end of file").</p> <p>The response body may contain a ProblemDetails structure.</p>
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	<p>In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.</p>

9.4.5a.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

9.4.5a.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

9.4.5a.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

9.4.6 Resource: Upload VNF package from URI task

9.4.6.1 Description

This task resource represents the "Upload VNF package from URI" operation. The API consumer can use this resource to request the uploading of a VNF package by providing address information to the NFVO for retrieving the content of the VNF package.

9.4.6.2 Resource definition

The resource URI is:

{apiRoot}/vnfpgm/{apiMajorVersion}/vnf_packages/{vnfPkgId}/package_content/upload_from_uri

This resource shall support the resource URI variables defined in Table 9.4.6.2-1.

Table 9.4.6.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 9.1a.
vnfPkgId	Identifier of the VNF package. The identifier is allocated by the NFVO. See note.
NOTE:	This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request creating a new Individual VNF package resource.

9.4.6.3 Resource methods

9.4.6.3.1 POST

The POST method provides the information for the NFVO to get the content of a VNF package.

This method shall follow the provisions specified in the Tables 9.4.6.3.1-1 and 9.4.6.3.1-2 for URI query parameters, request and response data structures, and response codes.

Upon start of obtaining the package, the NFVO shall set the "onboardingState" attribute in the "VnfPkgInfo" structure to "UPLOADING". Upon successfully obtaining the package, if the package references external artifacts, the NFVO shall obtain the external artifacts. Subsequently, upon success, the NFVO shall set that attribute to "PROCESSING" and shall process the package, which shall include checking package consistency. Upon successful processing, the NFVO shall set the "onboardingState" attribute to "ONBOARDED", the "operationalState" attribute to "ENABLED", and the "usageState" attribute to "NOT_IN_USE". In addition, the NFVO shall set the value of the attributes in the "VnfPkgInfo" that are copied from the VNFD (refer to clause 9.5.2.5). If an error occurs during obtaining the package, downloading the external artifacts or processing the package, the NFVO shall set the "onboardingState" attribute to "ERROR" and shall populate the "onboardingFailureDetails" attribute in "VnfPkgInfo".

Table 9.4.6.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
none supported		

Table 9.4.6.3.1-2: Details of the POST request/response on this resource

Request body	Data type	Cardinality	Description	
	UploadVnfPkgFromUri Request		1	The payload body contains the address information based on which the NFVO can obtain the content of the VNF package.
Response body	Data type	Cardinality	Response Codes	Description
	n/a		202 Accepted	<p>Shall be returned when the information about the VNF package has been received successfully, but the on-boarding has not been completed. It is expected to take some time for processing.</p> <p>The response body shall be empty. See note.</p>
	ProblemDetails	1	409 Conflict	<p>Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource.</p> <p>Typically, this is due to the fact that the on-boarding state of the VNF package resource is not "CREATED" or "ERROR".</p> <p>The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</p>
ProblemDetails	See clause 6.4 of [16]		4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.
NOTE: The API consumer can track the uploading progress by receiving the "VnfPackageOnBoardingNotification" from the NFVO or by reading the status of the individual on-boarded VNF package resource using the GET method.				

9.4.6.3.2 GET

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

9.4.6.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

9.4.6.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

9.4.6.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

9.4.7 Resource: Individual VNF package artifact

9.4.7.1 Description

This resource represents an individual artifact contained in a VNF package. The API consumer can use this resource to fetch the content of the artifact.

9.4.7.2 Resource definition

The resource URI is:

{apiRoot}/vnfpkgm/{apiMajorVersion}/vnf_packages/{vnfPkgId}/artifacts/{artifactPath}

This resource shall support the resource URI variables defined in Table 9.4.7.2-1.

Table 9.4.7.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 9.1a.
vnfPkgId	Identifier of the VNF package. The identifier is allocated by the NFVO. See note 1.
artifactPath	<p>For an artifact contained as a file in the VNF package, this variable shall contain a sequence of one or more path segments representing the path of the artifact within the VNF package, relative to the root of the package. See note 3.</p> <p>EXAMPLE: foo/bar/m%40ster.sh</p> <p>For an external artifact represented as a URI in the VNF package manifest, this variable shall contain a sequence of one or more path segments as synthesized by the NFVO (see clause 9.5.3.3), representing this artifact.</p> <p>See note 2 and note 3.</p>
NOTE 1:	This identifier can be retrieved from the "vnfPkgId" attribute in the VnfPackageOnboardingNotification or VnfPackageChangeNotification.
NOTE 2:	This identifier can be retrieved from the "artifactPath" attribute of the applicable "additionalArtifacts" entry in the body of the response to a GET request querying the "Individual VNF package" or the "VNF packages" resource.
NOTE 3:	Since multiple path segments are allowed to be contained in this variable, the "/" character that separates these segments is not percent-encoded. Each individual segment is percent-encoded if necessary as defined in clause 4.1 of ETSI GS NFV-SOL 013 [16].

9.4.7.3 Resource methods

9.4.7.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

9.4.7.3.2 GET

The GET method fetches the content of an artifact within a VNF package.

This method shall follow the provisions specified in the Tables 9.4.7.3.2-1 and 9.4.7.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 9.4.7.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
include_signatures	0..1	<p>If this parameter is provided, the NFVO shall return the artifact and related security information (such as signature and optional certificate) in a ZIP archive.</p> <p>If this parameter is not given, the NFVO shall provide only a copy of the artifact file.</p> <p>This URI query parameter is a flag, i.e. it shall have no value.</p> <p>The NFVO shall support this parameter.</p>

Table 9.4.7.3.2-2: Details of the GET request/response on this resource

	Data type	Cardinality	Description	
	Request body	n/a		<p>The request may contain a "Range" HTTP header to obtain single range of bytes from an artifact file. This can be used to continue an aborted transmission.</p> <p>If the Range header is present in the request and the NFVO does not support responding to range requests with a 206 response, it shall return a 200 OK response instead as defined below.</p>
Response body	Data type	Cardinality	Response Codes	Description
	n/a	1	200 OK	<p>Shall be returned when the whole content of the artifact file has been read successfully.</p> <p>If the "include_signatures" request URI parameter was not provided in the related request, the payload body shall contain a copy of the artifact file from the VNF package, as defined by ETSI GS NFV-SOL 004 [5], and the "Content-Type" HTTP header shall be set according to the content type of the artifact file. If the artifact is encrypted, the header shall be set to the value "application/cms" (IETF RFC 7193 [17]). If the content type cannot be determined, the header shall be set to the value "application/octet-stream".</p> <p>If the "include_signatures" request URI parameter was provided in the related request, the "Content-Type" HTTP header shall be set to "application/zip" and the payload body shall contain a ZIP archive which includes:</p> <ul style="list-style-type: none"> - a copy of the artifact file from the VNF package, as defined by ETSI GS NFV-SOL 004 [5]; - the related security information (individual signature file and optional related individual certificate file).
	n/a	1	206 Partial Content	<p>If the NFVO supports range requests and the "include_signatures" request URI parameter was not present in the related request, this response shall be returned when a single consecutive byte range from the content of the artifact file has been read successfully according to the request.</p> <p>The response body shall contain the requested part of the artifact file from the VNF package, as defined by ETSI GS NFV-SOL 004 [5].</p> <p>The "Content-Type" HTTP header shall be set according to the content type of the artifact file. If the content type cannot be determined, the header shall be set to the value "application/octet-stream". The "Content-Range" HTTP header shall be provided according to IETF RFC 7233 [10].</p>

	ProblemDetails	0..1	406 Not Acceptable	If the related request contained an "Accept" header not compatible with the Content type "application/zip" but the "include_signatures" flag was provided, the NFVO shall respond with this response code. The "ProblemDetails" structure may be included with the "detail" attribute providing more information about the error.
	ProblemDetails	1	409 Conflict	Shall be returned upon the following error: The operation cannot be executed currently, due to a conflict with the state of the resource. Typically, this is due to the fact that "onboardingState" of the VNF package has a value different from "ONBOARDED". The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.
	ProblemDetails	0..1	416 Range Not Satisfiable	Shall be returned upon the following error: The byte range passed in the "Range" header did not match any available byte range in the artifact file (e.g. "access after end of file"). The response body may contain a ProblemDetails structure.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

9.4.7.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

9.4.7.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

9.4.7.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

9.4.8 Resource: Subscriptions

9.4.8.1 Description

This resource represents subscriptions. The API consumer can use this resource to subscribe to notifications related to the VNF package management, and to query its subscriptions.

9.4.8.2 Resource definition

The resource URI is:

{apiRoot}/vnfpgm/{apiMajorVersion}/subscriptions

This resource shall support the resource URI variables defined in Table 9.4.8.2-1.

Table 9.4.8.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 9.1a.

9.4.8.3 Resource methods

9.4.8.3.1 POST

The POST method creates a new subscription.

This method shall follow the provisions specified in the Tables 9.4.8.3.1-1 and 9.4.8.3.1-2 for URI query parameters, request and response data structures, and response codes.

As the result of successfully executing this method, a new "Individual subscription" resource as defined in clause 9.4.9 shall have been created. This method shall not trigger any notification.

Creation of two "Individual subscription" resources with the same callback URI and the same filter can result in performance degradation and will provide duplicates of notifications to the OSS/BSS, and might make sense only in very rare use cases. Consequently, the NFVO may either allow creating a new "Individual subscription" resource if another "Individual subscription" resource with the same filter and callback URI already exists (in which case it shall return the "201 Created" response code), or may decide to not create a duplicate "Individual subscription" resource (in which case it shall return a "303 See Other" response code referencing the existing "Individual subscription" resource with the same filter and callback URI).

Table 9.4.8.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
none supported		

Table 9.4.8.3.1-2: Details of the POST request/response on this resource

Request body	Data type	Cardinality	Description	
	PkgmSubscriptionRequest	1	Details of the subscription to be created.	
Response body	Data type	Cardinality	Response Codes	Description
	PkgmSubscription	1	201 Created	<p>Shall be returned when the subscription has been created successfully.</p> <p>The response body shall contain a representation of the created "Individual subscription" resource.</p> <p>The HTTP response shall include a "Location" HTTP header that points to the created resource.</p>
	n/a		303 See Other	<p>Shall be returned when a subscription with the same callback URI and the same filter already exists and the policy of the NFVO is to not create redundant subscriptions.</p> <p>The HTTP response shall include a "Location" HTTP header that contains the resource URI of the existing "Individual subscription" resource.</p> <p>The response body shall be empty.</p>
	ProblemDetails	1	422 Unprocessable Entity	<p>Shall be returned upon the following error: The content type of the payload body is supported and the payload body of a request contains syntactically correct data but the data cannot be processed.</p> <p>The general cause for this error and its handling is specified in clause 6.4 of ETSI GS NFV-SOL 013 [16], including rules for the presence of the response body.</p> <p>Specifically in case of this resource, the response code 422 shall also be returned if the NFVO has tested the Notification endpoint as described in clause 9.4.10.3.2 and the test has failed. In this case, the "detail" attribute in the "ProblemDetails" structure shall convey more information about the error.</p>
ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.	

9.4.8.3.2 GET

The GET method queries the list of active subscriptions of the functional block that invokes the method. It can be used e.g. for resynchronization after error situations.

This method shall follow the provisions specified in the Tables 9.4.8.3.2-1 and 9.4.8.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 9.4.8.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
filter	0..1	Attribute-based filtering expression according to clause 5.2 of ETSI GS NFV-SOL 013 [16]. The NFVO shall support receiving this parameter as part of the URI query string. The OSS/BSS may supply this parameter. All attribute names that appear in the PkgmSubscription and in data types referenced from it shall be supported by the NFVO in the filter expression.
nextpage_opaque_marker	0..1	Marker to obtain the next page of a paged response. Shall be supported by the NFVO if the NFVO supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource.

Table 9.4.8.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	PkgmSubscription	0..N	200 OK	Shall be returned when the list of subscriptions has been queried successfully. The response body shall contain in an array the representations of all active subscriptions of the functional block that invokes the method, i.e. zero or more representations of VNF package management subscriptions, as defined in clause 9.5.2.7. If the "filter" URI parameter was supplied in the request, the data in the response body shall have been transformed according to the rules specified in clause 5.2.2 of ETSI GS NFV-SOL 013 [16]. If the NFVO supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource, inclusion of the Link HTTP header in this response shall follow the provisions in clause 5.4.2.3 of ETSI GS NFV-SOL 013 [16].
	ProblemDetails	1	400 Bad Request	Shall be returned upon the following error: Invalid attribute-based filtering expression. The response body shall contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error.
	ProblemDetails	1	400 Bad Request	Shall be returned upon the following error: Response too big. If the NFVO supports alternative 1 (error) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 [16] for this resource, this error response shall follow the provisions in clause 5.4.2.2 of ETSI GS NFV-SOL 013 [16].
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

9.4.8.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

9.4.8.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

9.4.8.3.5 DELETE

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

9.4.9 Resource: Individual subscription

9.4.9.1 Description

This resource represents an individual subscription. The API consumer can use this resource to read and to terminate a subscription to notifications related to the VNF package management.

9.4.9.2 Resource definition

The resource URI is:

{apiRoot}/vnfpkgm/{apiMajorVersion}/subscriptions/{subscriptionId}

This resource shall support the resource URI variables defined in Table 9.4.9.2-1.

Table 9.4.9.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 4.1 of ETSI GS NFV-SOL 013 [16].
apiMajorVersion	See clause 9.1a.
subscriptionId	Identifier of this subscription. See note.
NOTE: This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request creating a new "Individual subscription" resource. It can also be retrieved from the "id" attribute in the payload body of that response.	

9.4.9.3 Resource methods

9.4.9.3.1 POST

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

9.4.9.3.2 GET

The GET method reads an individual subscription.

This method shall follow the provisions specified in the Tables 9.4.9.3.2-1 and 9.4.9.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 9.4.9.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
none supported		

Table 9.4.9.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	PkgmSubscription	1	200 OK	Shall be returned when information about an individual subscription has been read successfully. The response body shall contain a representation of the "Individual subscription" resource.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

9.4.9.3.3 PUT

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

9.4.9.3.4 PATCH

This method is not supported. When this method is requested on this resource, the NFVO shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

9.4.9.3.5 DELETE

The DELETE method terminates an individual subscription.

This method shall follow the provisions specified in the Tables 9.4.9.3.5-1 and 9.4.9.3.5-2 for URI query parameters, request and response data structures, and response codes.

As the result of successfully executing this method, the "Individual subscription" resource shall not exist any longer. This means that no notifications for that subscription shall be sent to the formerly-subscribed API consumer.

NOTE: Due to race conditions, some notifications might still be received by the formerly-subscribed API consumer for a certain time period after the deletion.

Table 9.4.9.3.5-1: URI query parameters supported by the DELETE method on this resource

Name	Cardinality	Description
none supported		

Table 9.4.9.3.5-2: Details of the DELETE request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	n/a		204 No Content	Shall be returned when the "Individual subscription" resource has been deleted successfully.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

9.4.10 Resource: Notification endpoint

9.4.10.1 Description

This resource represents a notification endpoint.

The API producer can use this resource to send notifications related to VNF package management events to a subscribed API consumer, which has provided the URI of this resource during the subscription process.

9.4.10.2 Resource definition

The resource URI is provided by the API consumer when creating the subscription.

This resource shall support the resource URI variables defined in Table 9.4.10.2-1.

Table 9.4.10.2-1: Resource URI variables for this resource

Name	Definition
n/a	

9.4.10.3 Resource methods

9.4.10.3.1 POST

The POST method delivers a notification from the API producer to an API consumer. The API consumer shall have previously created an "individual subscription resource" with a matching filter.

This method shall follow the provisions specified in the Tables 9.4.10.3.1-1 and 9.4.10.3.1-2 for URI query parameters, request and response data structures, and response codes.

Table 9.4.10.3.1-1: URI query parameters supported by the POST method on this resource

Name	Cardinality	Description
none supported		

Each notification request body shall include exactly one of the alternatives defined in Table 9.4.10.3.1-2.

Table 9.4.10.3.1-2: Details of the POST request/response on this resource

Request body	Data type	Cardinality	Description	
	VnfPackageOnboardingNotification	1	A notification about on-boarding of a VNF package.	
VnfPackageChangeNotification	1	A notification about changes of status in a VNF package.		
Response body	Data type	Cardinality	Response Codes	Description
	n/a		204 No Content	Shall be returned when the notification has been delivered successfully.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

9.4.10.3.2 GET

The GET method allows the API producer to test the notification endpoint that is provided by the API consumer, e.g. during subscription.

This method shall follow the provisions specified in the Tables 9.4.10.3.2-1 and 9.4.10.3.2-2 for URI query parameters, request and response data structures, and response codes.

Table 9.4.10.3.2-1: URI query parameters supported by the GET method on this resource

Name	Cardinality	Description
none supported		

Table 9.4.10.3.2-2: Details of the GET request/response on this resource

Request body	Data type	Cardinality	Description	
	n/a			
Response body	Data type	Cardinality	Response Codes	Description
	n/a		204 No Content	Shall be returned to indicate that the notification endpoint has been tested successfully. The response body shall be empty.
	ProblemDetails	See clause 6.4 of [16]	4xx/5xx	In addition to the response codes defined above, any common error response code as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16] may be returned.

9.4.10.3.3 PUT

This method is not supported. When this method is requested on this resource, the OSS/BSS shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

9.4.10.3.4 PATCH

This method is not supported. When this method is requested on this resource, the OSS/BSS shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

9.4.10.3.5 DELETE

This method is not supported. When this method is requested on this resource, the OSS/BSS shall return a "405 Method Not Allowed" response as defined in clause 6.4 of ETSI GS NFV-SOL 013 [16].

9.5 Data model

9.5.1 Introduction

This clause defines the request and response data structures of the VNF package management interface. If a request or response contains attributes not defined in the present document, a receiving functional block that does not understand these attributes shall not treat their presence as an error, and may choose to ignore them.

9.5.2 Resource and notification data types

9.5.2.1 Introduction

This clause defines data structures to be used in resource representations and notifications.

9.5.2.2 Type: CreateVnfPkgInfoRequest

This type represents the request parameters for creating a new individual VNF package resource. It shall comply with the provisions defined in Table 9.5.2.2-1.

Table 9.5.2.2-1: Definition of the CreateVnfPkgInfoRequest data type

Attribute name	Data type	Cardinality	Description
userDefinedData	KeyValuePairs	0..1	User defined data for the VNF package.

9.5.2.3 Type: VnfPkgInfoModifications

This type represents modifications to the information of a VNF package. It shall comply with the provisions defined in Table 9.5.2.3-1.

Table 9.5.2.3-1: Definition of the VnfPkgInfoModifications data type

Attribute name	Data type	Cardinality	Description
operationalState	PackageOperationalStateType	0..1	New value of the operational state of the on-boarded instance of the VNF package. See note.
userDefinedData	KeyValuePairs	0..1	User defined data to be updated. For existing keys, the value is replaced. See note.

NOTE: At least one of the two parameters shall be present. If the VNF package is not on-boarded, the operation is used only to update existing or add additional user defined data using the userDefinedData attribute.

9.5.2.4 Type: UploadVnfPackageFromUriRequest

This type represents the request parameters for uploading the content of a VNF package. The NFVO can obtain the VNF package content through the information provided in the request parameters. It shall comply with the provisions defined in Table 9.5.2.4-1.

Table 9.5.2.4-1: Definition of the UploadVnfPackageFromUriRequest data type

Attribute name	Data type	Cardinality	Description
addressInformation	Uri	1	Address information of the VNF package content. The NFVO can use this address to obtain the VNF package.
authType	Enum (inlined)	0..1	Defines the type of authentication / authorization for downloading the VNF package. Permitted values: <ul style="list-style-type: none"> - BASIC: Only the "username" and "password" attributes shall be present. - OAUTH2_CLIENT_CREDENTIALS: Only the "paramsOauth2ClientCredentials" attribute shall be present. This attribute shall not be present if the credentials were provisioned out of band.
userName	String	0..1	User name to be used for authentication.
password	String	0..1	Password to be used for authentication.
paramsOauth2ClientCredentials	Structure (inlined)	0..1	Parameters for authentication/authorization using OAuth 2.0.
>clientId	String	1	Client identifier to be used in the access token request of the OAuth 2.0 client credentials grant type.
>clientPassword	String	1	Client password to be used in the access token request of the OAuth 2.0 client credentials grant type.
>tokenEndpoint	Uri	1	The token endpoint from which the access token can be obtained.

9.5.2.5 Type: VnfPkgInfo

This type represents the information of a VNF package. It shall comply with the provisions defined in Table 9.5.2.5-1.

Table 9.5.2.5-1: Definition of the VnfPkgInfo data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of the VNF package. This identifier is allocated by the NFVO.
vnfdId	Identifier	0..1	This identifier, which is managed by the VNF provider, identifies the VNF package and the VNFD in a globally unique way. It is copied from the VNFD of the on-boarded VNF package. It shall be present after the VNF package content has been on-boarded and absent otherwise.
vnfProvider	String	0..1	Provider of the VNF package and the VNFD. This information is copied from the VNFD. It shall be present after the VNF package content has been on-boarded and absent otherwise.
vnfProductName	String	0..1	Name to identify the VNF product. Invariant for the VNF product lifetime. This information is copied from the VNFD. It shall be present after the VNF package content has been on-boarded and absent otherwise.
vnfSoftwareVersion	Version	0..1	Software version of the VNF. This is changed when there is any change to the software included in the VNF package. This information is copied from the VNFD. It shall be present after the VNF package content has been on-boarded and absent otherwise.
vnfdVersion	Version	0..1	The version of the VNFD. This information is copied from the VNFD. It shall be present after the VNF package content has been on-boarded and absent otherwise.
compatibleSpecificationVersions	Version	0..N	Indicates which versions of the ETSI GS NFV-SOL 004 [5] specification the package complies to, as defined in the manifest of the package. Each entry shall be formatted as defined in clause 4.3.2 of ETSI GS NFV-SOL 004 [5].
checksum	Checksum	0..1	Checksum of the on-boarded VNF package. It shall be present after the VNF package content has been on-boarded and absent otherwise.
packageSecurityOption	Enum (inlined)	1	Signals the security option used by the package as defined in clause 5.1 of ETSI GS NFV-SOL 004 [5]. Valid values: <u>OPTION_1</u> , <u>OPTION_2</u>
signingCertificate	String	0..1	The singleton signing certificate if it is included as a file in the VNF package.
softwareImages	VnfPackageSoftwareImageInfo	0..N	Information about VNF package artifacts that are software images. Every local and external software image referenced from the VNFD shall be included. No other artifacts shall be included. This attribute shall not be present before the VNF package content is on-boarded. Otherwise, this attribute shall be present unless it has been requested to be excluded per attribute selector.

Attribute name	Data type	Cardinality	Description
additionalArtifacts	VnfPackageArtifactInfo	0..N	<p>Information about VNF package artifacts contained in the VNF package that are not software images.</p> <p>Every local and external artifact declared in the manifest shall be included, except the software images and the files that make up the parts of the VNFD (see clause 9.4.4.3.2).</p> <p>Signature files and certificate files are not considered as artifacts, however, the content of the "Licenses" and "Testing" directories in the VNF package is.</p> <p>This attribute shall not be present before the VNF package content is on-boarded. Otherwise, this attribute shall be present if the VNF package contains additional artifacts.</p>
onboardingState	PackageOnboardingStateType	1	On-boarding state of the VNF package. See note 3.
operationalState	PackageOperationalStateType	1	Operational state of the VNF package. See note 1 and note 3.
usageState	PackageUsageStateType	1	Usage state of the VNF package. See note 2 and note 3.
vnfmInfo	String	1..N	Specifies VNFM compatible with the VNF. This information is copied from the VNFD. See note 4.
userDefinedData	KeyValuePairs	0..1	User defined data for the VNF package.
onboardingFailureDetails	ProblemDetails	0..1	<p>Failure details of current onboarding procedure. See clause 6.3 of ETSI GS NFV-SOL 013 [16] for the details of "ProblemDetails" structure.</p> <p>If "onboardingState" is "ERROR", this attribute shall be present and contain error information (such as failed onboarding or processing operation, affected artifact, etc.), unless it has been requested to be excluded via an attribute selector.</p>
_links	Structure (inlined)	1	Links to resources related to this resource.
>self	Link	1	URI of this resource.
>vnfd	Link	1	Link to the "VNFD in an individual VNF package" resource.
>packageContent	Link	1	Link to the "VNF package content" resource.
<p>NOTE 1: If the value of the onboardingState attribute is not equal to "ONBOARDED", the value of the operationalState attribute shall be equal to "DISABLED".</p> <p>NOTE 2: If the value of the onboardingState attribute is not equal to "ONBOARDED", the value of the usageState attribute shall be equal to "NOT_IN_USE".</p> <p>NOTE 3: State changes of a VNF package are illustrated in clause B.2.</p> <p>NOTE 4: ETSI GS NFV-SOL 001 [i.1] specifies the structure and format of the VNFD based on TOSCA specifications.</p>			

9.5.2.6 Type: PkgmSubscriptionRequest

This type represents a subscription request related to VNF package management notifications about VNF package on-boarding or changes. It shall comply with the provisions defined in Table 9.5.2.6-1.

Table 9.5.2.6-1: Definition of the PkgmSubscriptionRequest data type

Attribute name	Data type	Cardinality	Description
filter	PkgmNotificationsFilter	0..1	Filter settings for this subscription, to define the subset of all notifications this subscription relates to. A particular notification is sent to the subscriber if the filter matches, or if there is no filter.
callbackUri	Uri	1	The URI of the endpoint to send the notification to.
authentication	SubscriptionAuthentication	0..1	Authentication parameters to configure the use of authorization when sending notifications corresponding to this subscription, as defined in clause 8.3.4 of ETSI GS NFV-SOL 013 [16]. This attribute shall only be present if the subscriber requires authorization of notifications.

9.5.2.7 Type: PkgmSubscription

This type represents a subscription related to notifications about VNF package management. It shall comply with the provisions defined in Table 9.5.2.7-1.

Table 9.5.2.7-1: Definition of the PkgmSubscription data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this "Individual subscription" resource.
filter	PkgmNotificationsFilter	0..1	Filter settings for this subscription, to define the subset of all notifications this subscription relates to. A particular notification is sent to the subscriber if the filter matches, or if there is no filter.
callbackUri	Uri	1	The URI of the endpoint to send the notification to.
_links	Structure (inlined)	1	Links to resources related to this resource.
>self	Link	1	URI of this resource.

9.5.2.8 Type: VnfPackageOnboardingNotification

This type represents a VNF package management notification, which informs the receiver that the onboarding process of a VNF package is complete and the package is ready for use. A change of the on-boarding state before the VNF package is on-boarded is not reported. It shall comply with the provisions defined in Table 9.5.2.8-1. The support of this notification is mandatory. The notification shall be triggered by the NFVO when the value of the "onboardingState" attribute of a new VNF package has changed to "ONBOARDED".

Table 9.5.2.8-1: Definition of the VnfPackageOnboardingNotification data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this notification. If a notification is sent multiple times due to multiple subscriptions, the "id" attribute of all these notifications shall have the same value.
notificationType	String	1	Discriminator for the different notification types. Shall be set to "VnfPackageOnboardingNotification" for this notification type.
subscriptionId	Identifier	1	Identifier of the subscription that this notification relates to.
timeStamp	DateTime	1	Date-time of the generation of the notification.
vnfPkgId	Identifier	1	Identifier of the VNF package. This identifier is allocated by the NFVO. Its value is the same as the value of the "id" attribute of the related "Individual VNF package" resource.

Attribute name	Data type	Cardinality	Description
vnfdId	Identifier	1	This identifier, which is managed by the VNF provider, identifies the VNF package and the VNFD in a globally unique way. It is copied from the VNFD of the on-boarded VNF package.
vnfmInfo	String	1..N	Specifies VNFMs compatible with the VNF. This information is copied from the VNFD. See Table 9.5.2.5-1.
_links	PkgmLinks	1	Links to resources related to this notification.

9.5.2.9 Type: VnfPackageChangeNotification

This type represents a VNF package management notification, which informs the receiver of a change of the status in an on-boarded VNF package. Only changes in the "operationalState" attribute of an on-boarded VNF package and the deletion of the VNF package will be reported. Changes in the "usageState" and "onboardingState" attributes are not reported. The notification shall comply with the provisions defined in Table 9.5.2.9-1. The support of this notification is mandatory. The notification shall be triggered by the NFVO when there is a change in the status of an onboarded VNF package, as follows.

- The "operationalState" attribute of a VNF package has changed, and the "onboardingState" attribute of the package has the value "ONBOARDED" (i.e. the package has been onboarded previously).
- The on-boarded VNF package has been deleted, and the "onboardingState" attribute of the deleted package had the value "ONBOARDED".

Table 9.5.2.9-1: Definition of the VnfPackageChangeNotification data type

Attribute name	Data type	Cardinality	Description
id	Identifier	1	Identifier of this notification. If a notification is sent multiple times due to multiple subscriptions, the "id" attribute of all these notifications shall have the same value.
notificationType	String	1	Discriminator for the different notification types. Shall be set to "VnfPackageChangeNotification" for this notification type.
subscriptionId	Identifier	1	Identifier of the subscription that this notification relates to.
timeStamp	DateTime	1	Date-time of the generation of the notification.
vnfPkgId	Identifier	1	Identifier of the VNF package. This identifier is allocated by the NFVO. Its value is the same as the value of the "id" attribute of the related "Individual VNF package" resource.
vnfdId	Identifier	1	Identifier of the VNFD contained in the VNF package, which also identifies the VNF package. This identifier is allocated by the VNF provider and copied from the VNFD.
changeType	PackageChangeType	1	The type of change of the VNF package.
operationalState	PackageOperationalStateType	0..1	New operational state of the VNF package. Only present when changeType is OP_STATE_CHANGE.
_links	PkgmLinks	1	Links to resources related to this notification.

9.5.2.10 Type: ExternalArtifactsAccessConfig

This type represents the access configuration information for downloading external VNF package artifacts. The NFVO can obtain the external VNF package artifact file through the information provided in this structure, together with information provided in the manifest / VNFD. The data structure shall comply with the provisions defined in Table 9.5.2.10-1. If the data structure is part of a response body, security-sensitive attributes shall be excluded as specified in Table 9.5.2.10-1.

Table 9.5.2.10-1: Definition of the ExternalArtifactsAccessConfig data type

Attribute name	Data type	Cardinality	Description
artifact	Structure (inlined)	0..N	Access configuration information for an external artifact.
>artifactUri	Uri	1	The artifactUri attribute as defined in the VNF package manifest that identifies the external artifact for which the access configuration information is provided, and that provides the default download location for this artifact.
>overrideUri	Uri	0..1	The Uri from which the external artifact can be downloaded. If this attribute is present, the NFVO shall download from this address. If this attribute is not present, the NFVO shall download from the address provided in "artifactUri".
>authType	Enum (inlined)	0..1	Defines the type of authentication / authorization for downloading the VNF package. Permitted values: <ul style="list-style-type: none"> - BASIC: Only the "username" and "password" attributes shall be present. - OAUTH2_CLIENT_CREDENTIALS: Only the "paramsOauth2ClientCredentials" attribute shall be present. This attribute shall not be present if no credentials are provided for the artifact.
>userName	String	0..1	User name to be used for authentication.
>password	String	0..1	Password to be used for authentication. Shall not be present in response bodies.
>paramsOauth2ClientCredentials	Structure (inlined)	0..1	Parameters for authentication/authorization using OAUTH2_CLIENT_CREDENTIALS.
>>clientId	String	1	Client identifier to be used in the access token request of the OAuth 2.0 client credentials grant type.
>>clientPassword	String	1	Client password to be used in the access token request of the OAuth 2.0 client credentials grant type. Shall not be present in response bodies.
>>tokenEndpoint	Uri	1	The token endpoint from which the access token can be obtained.

9.5.3 Referenced structured data types

9.5.3.1 Introduction

This clause defines data structures that can be referenced from data structures defined in the previous clauses, but are not resource representations.

9.5.3.2 Type: VnfPackageSoftwareImageInfo

This type represents an artifact contained in or external to a VNF package which represents a software image. It shall comply with provisions defined in Table 9.5.3.2-1.

Table 9.5.3.2-1: Definition of the VnfPackageSoftwareImageInfo data type

Attribute name	Data type	Cardinality	Description
id	IdentifierInVnfd	1	Identifier of the software image.
name	String	1	Name of the software image.
provider	String	1	Provider of the software image.
version	Version	1	Version of the software image.
checksum	Checksum	1	Checksum of the software image file.
isEncrypted	Boolean	1	Reflects whether the image is encrypted (true) or not (false).
containerFormat	Enum (inlined)	1	<p>Container format indicates whether the software image is in a file format that also contains metadata about the actual software.</p> <p>Permitted values:</p> <ul style="list-style-type: none"> - AKI: a kernel image format - AMI: a machine image format - ARI: a ramdisk image format - BARE: the image does not have a container or metadata envelope - DOCKER: docker container format - OVA: OVF package in a tarfile - OVF: OVF container format <p>See note 1.</p>
diskFormat	Enum (inlined)	1	<p>Disk format of a software image is the format of the underlying disk image.</p> <p>Permitted values:</p> <ul style="list-style-type: none"> - AKI: a kernel image format - AMI: a machine image format - ARI: a ramdisk image format - ISO: an archive format for the data contents of an optical disc, such as CD-ROM - QCOW2: a common disk image format, which can expand dynamically and supports copy on write - RAW: an unstructured disk image format - VDI: a common disk image format - VHD: a common disk image format - VHDX: enhanced version of VHD format - VMDK: a common disk image format <p>See note 2.</p>
createdAt	DateTime	1	Time when this software image was created.
minDisk	UnsignedInt	1	The minimal disk for this software image in bytes.
minRam	UnsignedInt	1	The minimal RAM for this software image in bytes.
size	UnsignedInt	1	Size of this software image in bytes.
userMetadata	KeyValuePairs	0..1	User-defined data.
imagePath	String	0..1	<p>Path which identifies the image artifact and also allows to access a copy of the image artifact.</p> <p>For a software image contained as a file in the VNF package, this attribute shall be present, and the value of this attribute shall start with the name of the first segment in the path in the package, i.e., it shall not be prefixed by path separator characters such as "." and "/".</p> <p>EXAMPLE: foo/bar/m%40ster.vhd</p>

Attribute name	Data type	Cardinality	Description
			For an external software image represented as a URI in the VNF descriptor, this attribute shall be present if the image artifact has been downloaded by the NFVO and shall be absent otherwise. If present, it shall contain the artifactPath under which the image artifact can be obtained using the "Individual artifact in a VNF package" resource defined in clause 9.4.7. It is the responsibility of the NFVO to synthesize this path in a manner that avoids any collision of the synthesized artifact path with the paths and names of image artifacts included in the package.
imageUri	Uri	0..1	URI of the image artifact as defined in the VNF package manifest. Shall be present if the image artifact is external to the VNF package and shall be absent otherwise. EXAMPLE: https://example.com/m%40ster.vhd
NOTE 1: The list of permitted values was taken from "Container formats" in [i.2].			
NOTE 2: The list of permitted values was adapted from "Disk formats" in [i.2].			

9.5.3.3 Type: VnfPackageArtifactInfo

This type represents an artifact other than a software image which is contained in or external to a VNF package. It shall comply with provisions defined in Table 9.5.3.3-1.

Table 9.5.3.3-1: Definition of the VnfPackageArtifactInfo data type

Attribute name	Data type	Cardinality	Description
artifactPath	String	0..1	Path which identifies the artifact and also allows to access a copy of the artifact. For an artifact contained as a file in the VNF package, this attribute shall be present, and the value of this attribute shall start with the name of the first segment in the path in the package, i.e. it shall not be prefixed by path separator characters such as "." and "/". EXAMPLE: foo/bar/m@ster.sh For an external artifact represented as a URI in the VNF descriptor, this attribute shall be present if the artifact has been downloaded by the NFVO and shall be absent otherwise. If present, it shall contain the artifactPath under which the artifact can be obtained using the "Individual artifact in a VNF package" resource defined in clause 9.4.7. It is the responsibility of the NFVO to synthesize this path in a manner that avoids any collision of the synthesized artifact path with the paths and names of artifacts included in the package.
artifactURI	Uri	0..1	URI of the artifact as defined in the VNF package manifest. Shall be present if the artifact is external to the package and shall be absent otherwise. EXAMPLE: https://example.com/m%40ster.sh
checksum	Checksum	1	Checksum of the artifact file.
isEncrypted	Boolean	1	Reflects whether the artifact is encrypted (true) or not (false).
nonManoArtifactSetId	String	0..1	Non-MANO artifact set identifier of the non-MANO artifact set to which the artifact belongs, as defined in clause 4.3.7 of ETSI GS NFV-SOL 004 [5]. Shall be provided if the artifact is a non-MANO artifact, and shall be omitted otherwise.

Attribute name	Data type	Cardinality	Description
artifactClassification	Enum (inlined)	0..1	Marks specific types of artifacts as defined in the VNF package. If none of the specific classes listed below applies, the attribute shall not be present. Valid values: <ul style="list-style-type: none"> - HISTORY: a history artifact as per clause 4.3.3 in ETSI GS NFV-SOL 004 [5] - TESTING: a testing artifact as per clause 4.3.4 in ETSI GS NFV-SOL 004 [5] - LICENSE: a license artifact as per clause 4.3.5 in ETSI GS NFV-SOL 004 [5]
metadata	KeyValuePairs	0..1	The metadata of the artifact that are available in the VNF package, such as Content type, size, creation date, etc.

9.5.3.4 Type: PkgmNotificationsFilter

This type represents a subscription filter related to notifications related to VNF package management. It shall comply with the provisions defined in Table 9.5.3.4-1.

At a particular nesting level in the filter structure, the following applies: All attributes shall match in order for the filter to match (logical "and" between different filter attributes). If an attribute is an array, the attribute shall match if at least one of the values in the array matches (logical "or" between the values of one filter attribute).

Table 9.5.3.4-1: Definition of the PkgmNotificationsFilter data type

Attribute name	Data type	Cardinality	Description
notificationTypes	Enum (inlined)	0..N	Match particular notification types. Permitted values: <ul style="list-style-type: none"> - VnfPackageOnboardingNotification - VnfPackageChangeNotification See note 1.
vnfProductsFromProviders	Structure (inlined)	0..N	If present, match VNF packages that contain VNF products from certain providers. See note 2.
>vnfProvider	String	1	Name of the VNF provider to match.
>vnfProducts	Structure (inlined)	0..N	If present, match VNF packages that contain VNF products with certain product names, from one particular provider.
>>vnfProductName	String	1	Name of the VNF product to match.
>>versions	Structure (inlined)	0..N	If present, match VNF packages that contain VNF products with certain versions and a certain product name, from one particular provider.
>>>vnfSoftwareVersion	Version	1	VNF software version to match.
>>>vnfdVersions	Version	0..N	If present, match VNF packages that contain VNF products with certain VNFD versions, a certain software version and a certain product name, from one particular provider.
vnfdId	Identifier	0..N	Match VNF packages with a VNFD identifier listed in the attribute. See note 2.
vnfPkgId	Identifier	0..N	Match VNF packages with a package identifier listed in the attribute. May be present if the "notificationTypes" attribute contains the value "VnfPackageChangeNotification", and shall be absent otherwise. See note 2.

Attribute name	Data type	Cardinality	Description
operationalState	PackageOperationalStateType	0..N	Match particular operational states of the VNF package. May be present if the "notificationTypes" attribute contains the value "VnfPackageChangeNotification", and shall be absent otherwise.
usageState	PackageUsageStateType	0..N	Match particular usage states of the VNF package. May be present if the "notificationTypes" attribute contains the value "VnfPackageChangeNotification", and shall be absent otherwise.
vnfmInfo	String	0..N	Match strings that specify VNFMs compatible with the VNF. See Table 9.5.2.5-1.
NOTE 1: The permitted values of the "notificationTypes" attribute are spelled exactly as the names of the notification types to facilitate automated code generation systems.			
NOTE 2: The attributes "vnfProductsFromProviders", "vnfdId", and "vnfPkgId" are alternatives to reference particular VNF packages in a filter. They should not be used both in the same filter instance, but one alternative should be chosen.			

9.5.3.5 Type: PkgmLinks

This type represents the links to resources that a VNF package management notification can contain. It shall comply with the provisions defined in Table 9.5.3.5-1.

Table 9.5.3.5-1: Definition of the PkgmLinks data type

Attribute name	Data type	Cardinality	Description
vnfPackage	NotificationLink	1	Link to the resource representing the VNF package to which the notified change applies, i.e. the "Individual VNF package" resource that represents the VNF package.
subscription	NotificationLink	1	Link to the related subscription.

9.5.3.6 Type: Checksum

This type represents the checksum of a VNF package or an artifact file. It shall comply with the provisions defined in Table 9.5.3.6-1.

Table 9.5.3.6-1: Definition of the Checksum data type

Attribute name	Data type	Cardinality	Description
algorithm	String	1	Name of the algorithm used to generate the checksum, as defined in ETSI GS NFV-SOL 004 [5]. For example, SHA-256, SHA-512.
hash	String	1	The hexadecimal value of the checksum.

9.5.4 Referenced simple data types and enumerations

9.5.4.1 Introduction

This clause defines simple data types and enumerations that can be referenced from data structures defined in the previous clauses.

9.5.4.2 Simple data types

No particular simple data types are defined for this interface, in addition to those defined in clause 4.4.2.

9.5.4.3 Enumeration: PackageOnboardingStateType

The enumeration PackageOnboardingStateType shall comply with the provisions defined in Table 9.5.4.3-1.

Table 9.5.4.3-1: Enumeration PackageOnboardingStateType

Enumeration value	Description
CREATED	The VNF package resource has been created.
UPLOADING	The associated VNF package content is being uploaded.
PROCESSING	The associated VNF package content is being processed, e.g. validation.
ONBOARDED	The associated VNF package content has been on-boarded successfully.
ERROR	There was an error during upload of the VNF package content or external artifacts, or during VNF package processing.

9.5.4.4 Enumeration: PackageOperationalStateType

The enumeration PackageOperationalStateType shall comply with the provisions defined in Table 9.5.4.4-1.

Table 9.5.4.4-1: Enumeration PackageOperationalStateType

Enumeration value	Description
ENABLED	The VNF package is enabled, i.e. it can be used for the creation of new "Individual VNF instance" resources.
DISABLED	The VNF package is disabled, i.e. it shall not be used for the creation of further "Individual VNF instance" resources (unless and until the VNF package is re-enabled).

9.5.4.5 Enumeration: PackageUsageStateType

The enumeration PackageUsageStateType shall comply with the provisions defined in Table 9.5.4.5-1.

Table 9.5.4.5-1: Enumeration PackageUsageStateType

Enumeration value	Description
IN_USE	"Individual VNF instance" resources created from this VNF package exist.
NOT_IN_USE	No "Individual VNF instance" resource created from this VNF package exists.

9.5.4.6 Enumeration: PackageChangeType

The enumeration PackageChangeType shall comply with the provisions defined in Table 9.5.4.6-1.

Table 9.5.4.6-1: Enumeration PackageChangeType

Enumeration value	Description
OP_STATE_CHANGE	The "operationalState" attribute has been changed.
PKG_DELETE	The VNF package has been deleted.

Annex A (informative): Mapping operations to protocol elements

A.1 Overview

This annex provides the mapping between operations as defined in ETSI GS NFV-IFA 013 [3] and the corresponding resources and HTTP methods defined in the present document.

A.2 NSD Management interface

The mapping of NSD management interface operations, defined in ETSI GS NFV-IFA 013 [3], to the resources and HTTP methods defined in the present document can be found in Table A.2-1.

Table A.2-1: Mapping of ETSI GS NFV-IFA 013 [3] NSD Management interface operations with resources and HTTP methods

ETSI GS NFV-IFA 013 [3] NSD Management interface operation	HTTP method	Resource	Direction
Create NSD Info	POST	ns_descriptors	OSS/BSS → NFVO
Upload NSD	PUT	ns_descriptors/{nsdInfold}/nsd_content	OSS/BSS → NFVO
Update NSD Info	PATCH	ns_descriptors/{nsdInfold}	OSS/BSS → NFVO
Delete NSD	DELETE	ns_descriptors/{nsdInfold}	OSS/BSS → NFVO
Query NSD	GET	ns_descriptors	OSS/BSS → NFVO
	GET	ns_descriptors/{nsdInfold}	OSS/BSS → NFVO
Fetch NSD	GET	ns_descriptors/{nsdInfold}/nsd_content	OSS/BSS → NFVO
Create PNFD Info	POST	pnf_descriptors	OSS/BSS → NFVO
Upload PNFD	PUT	pnf_descriptors/{pnfdInfold}/pnfd_content	OSS/BSS → NFVO
Update PNFD Info	POST	pnf_descriptors/{pnfdInfold}	OSS/BSS → NFVO
Delete PNFD	DELETE	pnf_descriptors/{pnfdInfold}	OSS/BSS → NFVO
Query PNFD Info	GET	pnf_descriptors	OSS/BSS → NFVO
	GET	pnf_descriptors/{pnfdInfold}	OSS/BSS → NFVO
Fetch PNFD	GET	pnf_descriptors/{pnfdInfold}/pnfd_content	OSS/BSS → NFVO
Subscribe	POST	subscriptions	OSS/BSS → NFVO
Query Subscription Information	GET	subscriptions	OSS/BSS → NFVO
	GET	subscriptions/{subscriptionId}	OSS/BSS → NFVO
Terminate Subscription	DELETE	subscriptions/{subscriptionId}	OSS/BSS → NFVO
Notify	POST	(provided by API consumer)	NFVO → OSS/BSS

A.3 NS Lifecycle Management interface

The mapping of NS lifecycle management operations, defined in ETSI GS NFV-IFA 013 [3], to the resources and HTTP methods defined in the present document can be found in Table A.3-1.

Table A.3-1: Mapping of ETSI GS NFV-IFA 013 [3] operations with NS Lifecycle Management interface resources and methods

ETSI GS NFV-IFA 013 [3] NS Lifecycle Management interface operation	HTTP method	Resource	Direction
Create NS Identifier	POST	ns_instances	OSS/BSS → NFVO
Instantiate NS	POST	ns_instances/{nsInstanceId}/instantiate	OSS/BSS → NFVO
Scale NS	POST	ns_instances/{nsInstanceId}/scale	OSS/BSS → NFVO
Update NS	POST	ns_instances/{nsInstanceId}/update	OSS/BSS → NFVO
Terminate NS	POST	ns_instances/{nsInstanceId}/terminate	OSS/BSS → NFVO
Delete NS Identifier	DELETE	ns_instances/{nsInstanceId}	OSS/BSS → NFVO
Query NS	GET	ns_instances/{nsInstanceId}	OSS/BSS → NFVO
	GET	ns_instances	OSS/BSS → NFVO
Heal NS	POST	ns_instances/{nsInstanceId}/heal	OSS/BSS → NFVO
Get Operation Status	GET	ns_lcm_op_occs	OSS/BSS → NFVO
	GET	ns_lcm_op_occs/{nsLcmOpOcclId}	OSS/BSS → NFVO
Subscribe	POST	subscriptions	OSS/BSS → NFVO
Query Subscription Information	GET	subscriptions	OSS/BSS → NFVO
	GET	subscriptions/{subscriptionId}	OSS/BSS → NFVO
Terminate Subscription	DELETE	subscriptions/{subscriptionId}	OSS/BSS → NFVO
Notify	POST	(provided by API consumer)	NFVO → OSS/BSS

A.4 NS Performance Management interface

The mapping of NS performance management operations, defined in ETSI GS NFV-IFA 013 [3], to the resources and HTTP methods defined in the present document can be found in Table A.4-1.

Table A.4-1: Mapping of ETSI GS NFV-IFA 013 [3] NS Performance Management interface operations with resources and HTTP methods

ETSI GS NFV-IFA 013 [3] NS Performance Management operation	HTTP method	Resource	Direction
Create PM Job	POST	pm_jobs	OSS/BSS → NFVO
Delete PM Job	DELETE	pm_jobs/{pmJobId}	OSS/BSS → NFVO
	GET	pm_jobs	OSS/BSS → NFVO
Query PM Job	GET	pm_jobs/{pmJobId}	OSS/BSS → NFVO
	POST	thresholds	OSS/BSS → NFVO
Delete Threshold	DELETE	thresholds/{thresholdId}	OSS/BSS → NFVO
Query Threshold	GET	thresholds	OSS/BSS → NFVO
	GET	thresholds/{thresholdId}	OSS/BSS → NFVO
Subscribe	n/a	see note	n/a
Query Subscription Information	n/a	see note	n/a
	n/a	see note	n/a
Terminate Subscription	n/a	see note	n/a
Notify	POST	(provided by API consumer)	NFVO → OSS/BSS
NOTE:	In the NS Performance Management interface, support for subscriptions has been dropped in version 2.7.1 of the present document in favour of controlling the delivery of notifications directly by the "Thresholds" and "PM jobs" resources.		

A.5 NS Fault Management interface

The mapping of NS fault management operations, defined in ETSI GS NFV-IFA 013 [3], to the resources and HTTP methods defined in the present document can be found in Table A.5-1.

Table A.5-1: Mapping of ETSI GS NFV-IFA 013 [3] NS Fault Management interface operations with resources and HTTP methods

ETSI GS NFV-IFA 013 [3] NS Fault Management interface operation	HTTP method	Resource	Direction
Get Alarm List	GET	alarms	OSS/BSS → NFVO
	GET	alarms/{alarmId}	OSS/BSS → NFVO
Acknowledge Alarm	PATCH	alarms/{alarmId}	OSS/BSS → NFVO
Subscribe	POST	subscriptions	OSS/BSS → NFVO
Query Subscription Information	GET	subscriptions	OSS/BSS → NFVO
	GET	subscriptions/{subscriptionId}	OSS/BSS → NFVO
Terminate Subscription	DELETE	subscriptions/{subscriptionId}	OSS/BSS → NFVO
Notify	POST	(provided by API consumer)	NFVO → OSS/BSS

A.6 VNF Package Management interface

The mapping of VNF package management operations, defined in ETSI GS NFV-IFA 013 [3], to the resources and HTTP methods defined in the present document can be found in Table A.6-1.

Table A.6-1: Mapping of ETSI GS NFV-IFA 013 [3] operations with VNF Package Management interface resources and methods

ETSI GS NFV-IFA 013 [3] VNF Package Management interface operation	HTTP method	Resource	Direction
Create VNF Package Info	POST	vnf_packages	OSS/BSS → NFVO
Update VNF Package Info	PATCH	vnf_packages/{vnfPkgId}	OSS/BSS → NFVO
Delete VNF Package	DELETE	vnf_packages/{vnfPkgId}	OSS/BSS → NFVO
Query VNF Package Info	GET	vnf_packages	OSS/BSS → NFVO
	GET	vnf_packages/{vnfPkgId}	OSS/BSS → NFVO
	GET	vnf_packages/{vnfPkgId}/vnfd	OSS/BSS → NFVO
	GET	/vnf_packages/{vnfPkgId}/manifest	OSS/BSS → NFVO
Upload VNF Package	PUT	vnf_packages/{vnfPkgId}/package_content	OSS/BSS → NFVO
	POST	vnf_packages/{vnfPkgId}/package_content/upload_from_uri	OSS/BSS → NFVO
Fetch VNF Package	GET	vnf_packages/{vnfPkgId}/package_content	OSS/BSS → NFVO
Fetch VNF Package Artifacts	GET	/vnf_packages/{vnfPkgId}/artifacts	OSS/BSS → NFVO
	GET	vnf_packages/{vnfPkgId}/artifacts/{artifactPath}	OSS/BSS → NFVO
Subscribe	POST	subscriptions	OSS/BSS → NFVO
Query Subscription Information	GET	subscriptions	OSS/BSS → NFVO
	GET	subscriptions/{subscriptionId}	OSS/BSS → NFVO
Terminate subscription	DELETE	subscriptions/{subscriptionId}	OSS/BSS → NFVO
Notify	POST	(provided by API consumer)	NFVO → OSS/BSS

Annex B (informative): State models

B.1 NSD state model

B.1.1 Introduction

This clause describes the state model of NSD in the NFVO. It includes the state models for two phases, i.e. onboarding phase of NSD and operational phase of NSD.

B.1.2 State model

A given NSD has three states, i.e. on-boarding state, operational state and usage state.

The on-boarding state is represented by the "nsdOnboardingState" attribute in the "NsdInfo" data type with below values:

- **CREATED:** The NSD information object has been created.
- **UPLOADING:** The NSD is being uploaded.
- **PROCESSING:** The NSD is being processed, e.g. validation.
- **ONBOARDED:** The NSD has been on-boarded successfully.
- **ERROR:** There was an error during NSD uploading or processing.

The operational state is represented by the "nsdOperationalState" attribute in the "NsdInfo" data type with below values:

- **ENABLED:** The NSD is enabled.
- **DISABLED:** The NSD is disabled.

The usage state is represented by the "nsdUsageState" attribute in the "NsdInfo" data type with below values:

- **IN_USE:** The NSD is in use.
- **NOT_IN_USE:** The NSD is not in use.

The state model of on-boarding phase in Figure B.1.2-1 applies to a given NSD being on-boarded. Besides the operations and conditions specified in the figure, below operations are also considered as available during the on-boarding phase:

- Query NSD Info
- Update NSD Info (with user defined data only)

The state model of operational phase in Figure B.1.2-1 applies to an on-boarded NSD. Besides the operations and conditions specified in the figure, below operations are also considered as available during the operational phase:

- Query NSD Info
- Update NSD Info (with user defined data only)
- Fetch NSD

At the end of the on-boarding phase, the "nsdOnboardingState" value transitions to "ONBOARDED" and the "nsdOperationalState" value transitions from "DISABLED" to "ENABLED", and the operational phase is entered.

The "nsdOperationalState" and "nsdUsageState" detail the state changes during the NSD operational phase. During the NSD on-boarding phase, the value of the "nsdOperationalState" is "DISABLED" and the value of the "nsdUsageState" is "NOT_IN_USE". Right after the NSD becomes on-boarded, the value of the "nsdOperationalState" is changed to "ENABLED" and the value of the "nsdUsageState" is kept as "NOT_IN_USE".

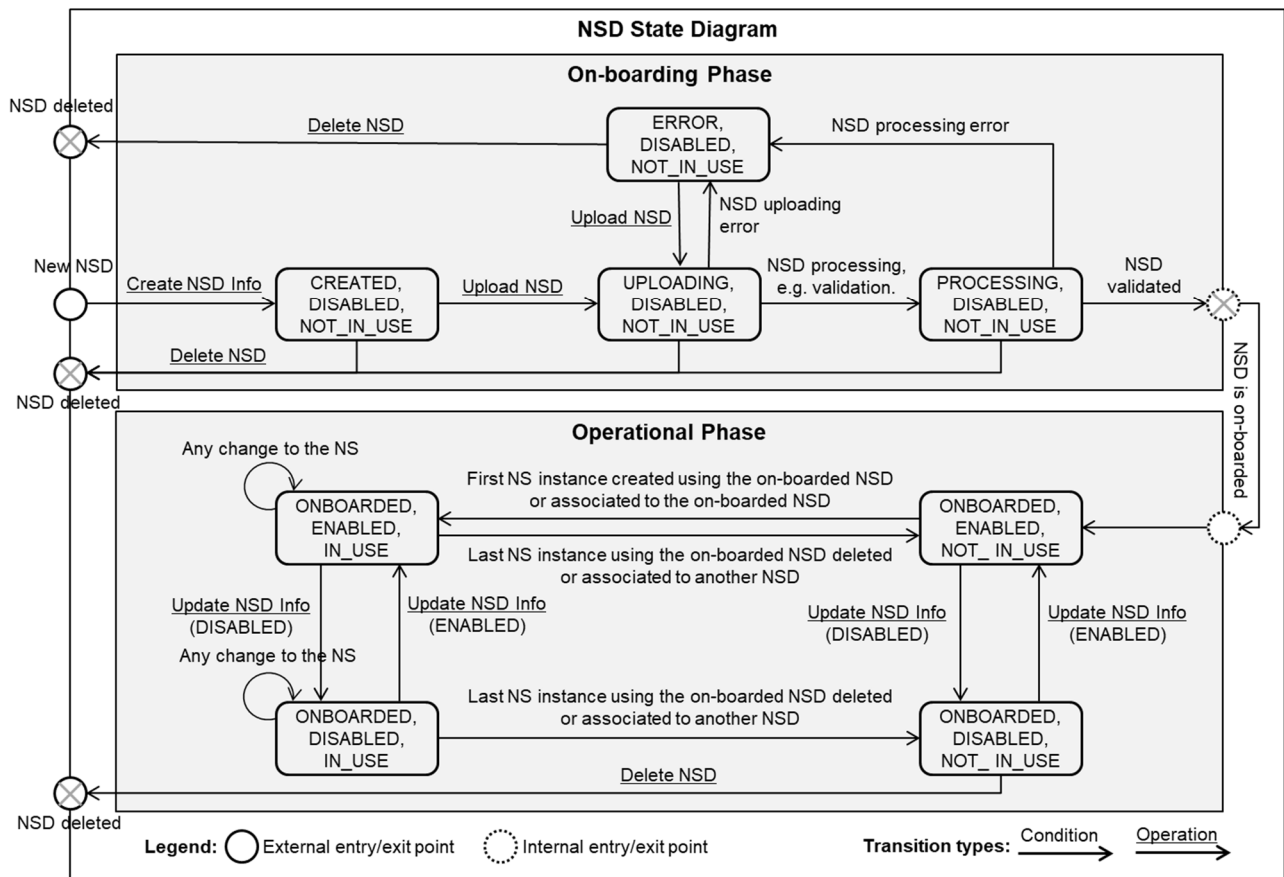


Figure B.1.2-1: NSD state model

B.2 VNF package state model

B.2.1 Introduction

This clause describes the state model of VNF Package in the NFVO. It includes the state models for two phases, i.e. on-boarding phase and operational phase.

B.2.2 State model

A given VNF Package has three states, i.e. on-boarding state, operational state and usage state.

The on-boarding state is represented by the "onboardingState" attribute in the "VnfPkgInfo" information element with below values:

- **CREATED:** The VNF Package information object has been created.
- **UPLOADING:** The VNF Package or the external artifacts are being uploaded.
- **PROCESSING:** The VNF Package is being processed, e.g. validation.
- **ONBOARDED:** The VNF Package has been on-boarded successfully.

- **ERROR:** There was an error during uploading of the VNF package or the external artifacts, or during VNF package processing.

The operational state is represented by the "operationalState" attribute in the "VnfPkgInfo" information element with below values:

- **ENABLED:** The VNF Package is enabled.
- **DISABLED:** The VNF Package is disabled.

The usage state is represented by the "usageState" attribute in the "VnfPkgInfo" information element with below values:

- **IN_USE:** The VNF Package is in use.
- **NOT_IN_USE:** The VNF Package is not in use.

The state model of on-boarding phase in Figure B.2.2-1 applies to a given VNF Package being on-boarded. Besides the operations and conditions specified in the figure, below operations are also considered as available during the on-boarding phase:

- Query VNF Package Info
- Update VNF Package Info (with user defined data only)

The state model of operational phase in Figure B.2.2-1 applies to an on-boarded VNF Package. Besides the operations and conditions specified in the figure, below operations are also considered as available during the operational phase:

- Query VNF Package Info
- Update VNF Package Info (with user defined data only)
- Fetch VNF Package
- Fetch VNF Package Artifacts

The "onboardingState" details the state changes during the VNF Package on-boarding phase. The value of this attribute during the VNF Package operational phase is "ONBOARDED".

The "operationalState" and "usageState" detail the state changes during the VNF Package operational phase. During the VNF Package on-boarding phase, the value of the "operationalState" is "DISABLED" and the value of the "usageState" is "NOT_IN_USE". Right after the VNF Package becomes on-boarded, the value of the "operationalState" is changed to "ENABLED" and the value of the "usageState" is kept as "NOT_IN_USE", as shown in Figure B.2.2-1.

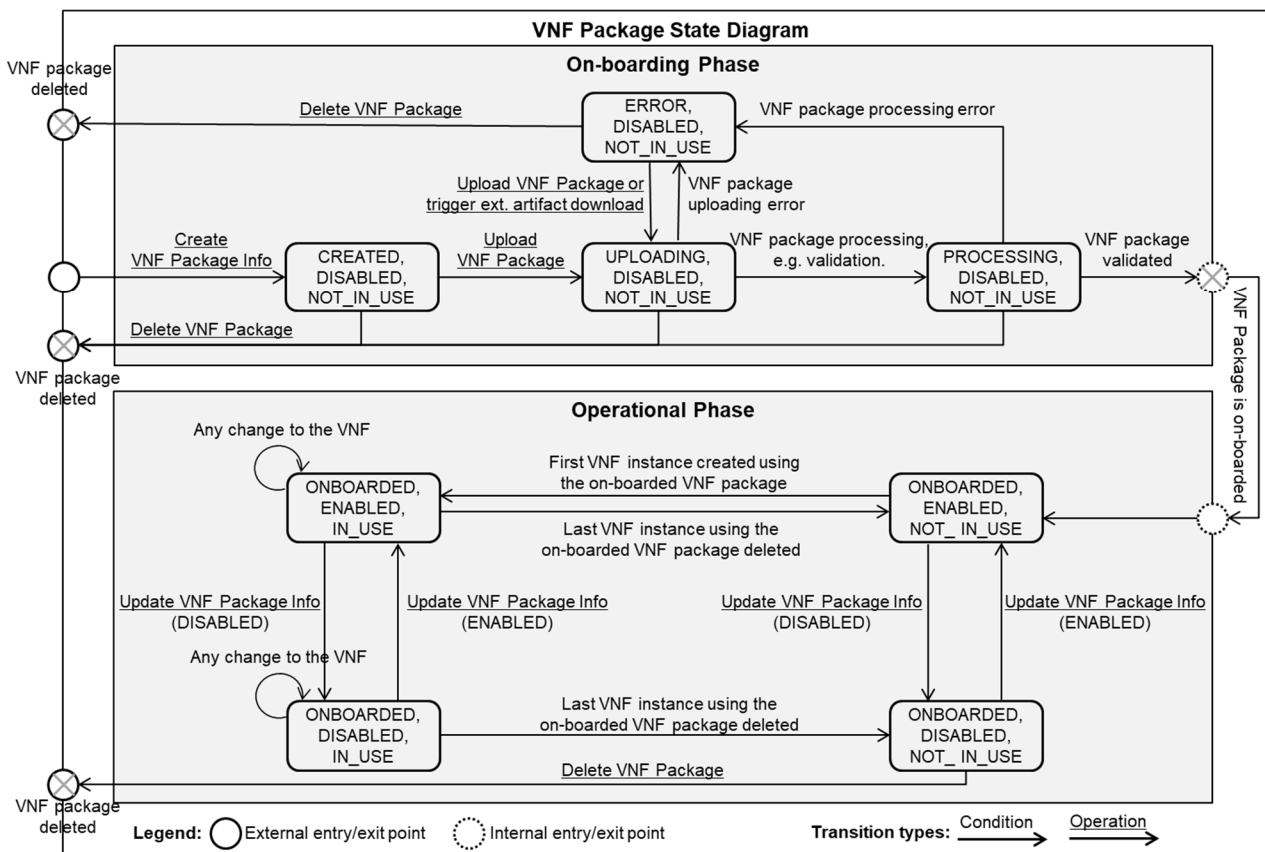


Figure B.2.2-1: VNF Package state model

Annex C (informative): Complementary material for API utilization

To complement the definitions of each method, resource, and data type defined in the main body of the present document, the ETSI NFV ISG is providing supplementary description files, compliant to the OpenAPI Specification [i.3], for the Os-Ma-nfvo reference point. These supplementary description files, containing the OpenAPI Specification for each API defined in the present document, are located at <https://forge.etsi.org/rep/nfv/NFV-SOL005>.

In case of discrepancies between the supplementary files and the related data structure definitions in the main body of the present document, the data structure definitions take precedence.

The OpenAPI representations referenced above:

- 1) use the MAJOR.MINOR.PATCH version fields to signal the version of the API as defined in the present document; and
- 2) use the "impl" version parameter (see clause 9.1.2 of ETSI GS NFV-SOL 013 [16]) to represent changes to the OpenAPI representation without changing the present document.

It is specified in clause 6 of ETSI GS NFV-SOL 015 [i.7] how the OpenAPI specification references the present document and signals the version information.

Annex D (informative): Bibliography

- IETF RFC 2818: "HTTP Over TLS".
- IETF RFC 3339: "Date and Time on the Internet: Timestamps".
- IETF RFC 3986: "Uniform Resource Identifier (URI): Generic Syntax".
- IETF RFC 4291: "IP Version 6 Addressing Architecture".
- IETF RFC 4632: "Classless Inter-Domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan".
- IETF RFC 4918: "HTTP Extensions for Web Distributed Authoring and Versioning (WebDAV)".
- IETF RFC 5246: "The Transport Layer Security (TLS) Protocol Version 1.2".
- IETF RFC 6585: "Additional HTTP Status Codes".
- IETF RFC 6749: "The OAuth 2.0 Authorization Framework".
- IETF RFC 6750: "The OAuth 2.0 Authorization Framework: Bearer Token Usage".
- IETF RFC 8259: "The JavaScript Object Notation (JSON) Data Interchange Format".
- IETF RFC 7231: "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content".
- IETF RFC 7232: "Hypertext Transfer Protocol (HTTP/1.1): Conditional Requests".
- IETF RFC 7235: "Hypertext Transfer Protocol (HTTP/1.1): Authentication".
- IETF RFC 7617: "The 'Basic' HTTP Authentication Scheme".
- IETF RFC 7807: "Problem Details for HTTP APIs".
- IETF RFC 6901: "JavaScript Object Notation (JSON) Pointer".
- IETF RFC 4229: "HTTP Header Field Registrations".
- IETF RFC 8288: "Web Linking".
- Semantic Versioning 2.0.0.
- ETSI TS 133 310: "Universal Mobile Telecommunications System (UMTS); LTE; Network Domain Security (NDS); Authentication Framework (AF) (3GPP TS 33.310)".
- Hypertext Transfer Protocol (HTTP) Status Code Registry at IANA.

Annex E (informative): Change History

Date	Version	Information about changes
December 2016	0.0.1	Initial version based on contributions that were agreed at the NFVSOL#15 meeting <ul style="list-style-type: none"> - NFVSOL(16)000169_SOL005_Scope_Statement - NFVSOL(16)000171_SOL005_Proposed_Table_of_Contents
February 2017	0.1.0	Version 0.1.0 based on contributions that were agreed at the NFVSOL#20 meeting <ul style="list-style-type: none"> - NFVSOL(17)000061_SOL005_URI_structure_and_supported_content_formats - NFVSOL(17)000064R1_SOL002_SOL003_SOL005_Labeling_of_API_names - NFVSOL(17)000095R1_SOL005_Error_reporting - NFVSOL(17)000106R1_Conventions_document_NFVSOL(17)000050_swagger_representation_of_the_API - NFVSOL(17)000107R1_SOL005_Common_procedures - NFVSOL(17)000111_SOL003_Conventions_move_Resource_structure_up_in_the_TOC
March 2017	0.2.0	Version 0.2.0 based on contributions that were agreed at the NFVSOL#22 meeting <ul style="list-style-type: none"> - NFVSOL(17)000196R1_SOL005_Add_SOL003_Normative_Reference - NFVSOL(17)000198_SOL005_Clause_4.1_Overview - NFVSOL(17)000199_SOL002_SOL003_Renaming_attribute_selectors - NFVSOL(17)000200_SOL002_SOL003_Attribute_filter_equality - NFVSOL(17)000179_SOL005_Simple_Data_Types - NFVSOL(17)000138_SOL005_NS_LCM_Description_Clause_6.1 - NFVSOL(17)000142R1_SOL005_NS_LCM_Resource_Structure_and_Methods_Clause_6.2 - NFVSOL(17)000123R2_SOL005_Flow_of_the_Creation_of_a_NS_Instance_Resource - NFVSOL(17)000129R1_SOL005_Flow_of_the_Deletion_of_a_NS_Instance_Resource - NFVSOL(17)000130_SOL005_Flow_of_the_Get_Operations_Status_Operation - NFVSOL(17)000137_SOL005_Flow_of_Managing_Subscriptions_to_Notifications_Related_to_NS_Lifecycle_Management - NFVSOL(17)000136_SOL005_Flow_of_Sending_Notifications_Related_to_NS_Lifecycle_management - NFVSOL(17)000156R2_SOL005_NS_Lifecycle_Change_Resource_Definitions_Methods_and_Data_Types - NFVSOL(17)000160_SOL005_NS_Lifecycle_Management_Resource_Definitions_Methods_and_Data_Types
April 2017	0.3.0	Version 0.3.0 based on contributions that were agreed at the NFVSOL#26 meeting <ul style="list-style-type: none"> - NFVSOL(17)000263_SOL005_Instantiate_NS_Lifecycle_Management_Resource_Definition_Methods_and_Data_Types - NFVSOL(17)000265_SOL005_Heal_NS_Lifecycle_Management_Resource_Definition_Methods_and_Data_Types
May 2017	0.4.0	Version 0.4.0 based on contributions that were agreed at the NFVSOL#28 meeting <ul style="list-style-type: none"> - NFVSOL(17)000264R1_SOL005_Update_NS_Lifecycle_Management_Resource_Methods_and_Data_Types - NFVSOL(17)000266_SOL005_Scale_NS_Operation_Resource_Definition_Methods_and_Data_Types - NFVSOL(17)000267_SOL005_Terminate_NS_Operation_Resource_Definition_Methods_and_Data Type - NFVSOL(17)000268_SOL005_Merge_NS_LCCN_interface_into_the_NS_LCM_interface - NFVSOL(17)000348R4_SOL005_NS_D_Management_Interface - NFVSOL(17)000378_SOL005_NS_Lifecycle_Change_Occurrence_Resource_Definitions_Methods_and_Data_Types - NFVSOL(17)000383_SOL005_Flow_of_NS_Lifecycle_Management_Operations_Triggered_by_Task_Resources - NFVSOL(17)000385R1_SOL005_Individual_NS_Descriptor_Resource_Descriptor_Methods_and_Data_Types - NFVSOL(17)000390R1_SOL005_Handling_of_Errors_During_NS_Lifecycle_Management_Operations

Date	Version	Information about changes
June 2017	0.5.0	<p>Version 0.5.0 based on contributions that were agreed at the NFVSOL#30 and NFVSOL#31 meetings</p> <ul style="list-style-type: none"> - NFVSOL(17)000371R1_SOL005:_Data_Type_NsInstance - NFVSOL(17)000411R1_SOL005:_Clause_6.6_Update,_ "Handling_of_Errors_During_NS_Lifecycle_Management" - NFVSOL(17)000420R1_SOL005:_Update_ScaleVnfData_and_ScaleByStepData_Data_Types_consistent_with_NFVIFA(17)000382R1 - NFVSOL(17)000421R1_SOL005:_Update_"additionalParam..."_attribute_names, KeyValuePair data type_and_its_cardinality_consistent_with_SOL003 - NFVSOL(17)000422_SOL005:_Update_AffinityOrAntiAffinityRule_Data_Type_Consistent_with_NFVIFA(17)000534 - NFVSOL(17)000424_SOL005:_Modify_VNF_Configuration_in_Update_NS_operation_consistent_with_NFVIFA(17)000527 - NFVSOL(17)000427R2_SOL005:_Update_NSD_Interface_Resource_Tree,_Resources,_and_Methods_(Clause_5.2) - NFVSOL(17)000428_SOL005:_ParamsForVnf_and_Resolution_of_the_Associated_Editor's_Note - NFVSOL(17)000431R3_SOL005:_Error_Handling_for_NS_Lifecycle_operation,_Resource_Definition_and_Methods - NFVSOL(17)000432R2_SOL005:_Update_Flow_of_Error_Handling_for_NS_LCM_Operations - NFVSOL(17)000433R3_SOL005:_Update_"address"_attribute's_data_type_in_PnfExtCplInfo_data_type - NFVSOL(17)000434R3_SOL005:_Update_LocationConstraints_data_type_in_Instantiate_NS_operation - NFVSOL(17)000436R2_SOL005:_Filter_design_for_NS_Instances - NFVSOL(17)000438R2_SOL005_VNF_Package_management_interface_-_resource_structure_and_methods - NFVSOL(17)000439R3_SOL005_VNF_package_management_interface_-_resources - NFVSOL(17)000440R3_SOL005:_Updates_to_the_Lifecycle_Change_Notifications_Filter - NFVSOL(17)000441R1_SOL005:_Authorizations_of_API_requests_and_notifications - NFVSOL(17)000446R1_SOL005:_Update_Clause_4.3_"Common_Procedures"_Consistent_with_SOL003 - NFVSOL(17)000451_SOL005:_Correct_various_references_to_NS_Lifecycle_operation_occurrences - NFVSOL(17)000464_SOL005_-_Modifications_on_sequence_diagrams_in_clauses_5.3.2_and_5.3.3 - NFVSOL(17)000466_SOL005:_Add_IdentifierInVim_simple_data_type
August 2017	0.6.0	<p>Version 0.6.0 based on contributions that were agreed at the NFVSOL#34 and NFVSOL#35,meetings</p> <ul style="list-style-type: none"> - NFVSOL(17)00426R6_SOL005:_Changes_to_NSD_Resources,_Methods,_and_Data_Types - NFVSOL(17)00493_SOL005:_Update_Clause_3,_Abbreviations - NFVSOL(17)000494R1_SOL005:_NS_PM_Interface_description,_resource_structure,_and_methods - NFVSOL(17)000508_SOL005:_Remove_Editor's_Note_in_Clause_6.5.3.8 - NFVSOL(17)000509_SOL005:_Remove_Editor's_Note_in_Clauses_5.5.3.1, 5.5.4.1, and 5.5.4.2 - NFVSOL(17)000510R1_SOL005:_NS_Performance_Management_Interface_resources_and_data_model - NFVSOL(17)000511R1_SOL005:_VNF_package_management_interface_flows - NFVSOL(17)000512R1_SOL005:_VNF_package_management_interface_data_types - NFVSOL(17)000513R2_SOL005:_NSD_Management_Interface_operations,_resource_structure,_and_methods - NFVSOL(17)000514R1_SOL005:_NS_LCM_Updates

Date	Version	Information about changes
September 2017	0.7.0	Version 0.7.0 based on contributions that were agreed at the NFVSOL#36 and NFVSOL#37 meetings <ul style="list-style-type: none"> - NFVSOL(17)000386R3_SOL005: PNFD_resources,_methods,_and_data_types - NFVSOL(17)00478R1_SOL005: VL_and_CP_consistency_(mirror_of_403r3_and_423) - NFVSOL(17)000519R4_SOL005: NS_fault_management_interface - NFVSOL(17)000520R1_SOL005: New_Annex_B_Mapping_operations_to_protocol_elements - NFVSOL(17)000524R1_SOL005: Refactoring_of_NSD_management_interface - NFVSOL(17)000526R1_SOL005: Definition_of_the_HealNsData_data_type_(clause_6.5.3.33) - NFVSOL(17)000529_SOL005: Apply_the_agreed_design_for_VNF_package_on-boarding_operation - NFVSOL(17)000530R1_SOL005: Remaining_VNF_package_management_interface_flows - NFVSOL(17)000531R1_SOL005: Remaining_resource_design_for_VNF_package_management_interface - NFVSOL(17)000532R1_SOL005: Filters_and_selectors_for_VNF_package_management_interface - NFVSOL(17)000533R2_SOL005: Additional_data_model_for_VNF_package_management_interface - NFVSOL(17)000534_SOL005: VNF_package_management_interface_data_type_align_with_SOL003 - NFVSOL(17)000554_SOL005: Align_with_IFA013_on_vimld
October 2017	0.8.0	Version 0.8.0 based on contributions that were agreed at the NFVSOL#38 meeting. <ul style="list-style-type: none"> - NFVSOL(17)000521_SOL005: NS_performance_management_interface_sequence_diagrams - NFVSOL(17)000558R1_SOL005: NSD_management_interface_-_notification_and_state_diagram - NFVSOL(17)000573_SOL005: NSD_management_interface_subscription_resources
November 2017	0.9.0	Version 0.9.0 based on contributions that were agreed at the NFVSOL#40 and NFVSOL#41 meetings. <ul style="list-style-type: none"> - NFVSOL(17)000535r2_SOL005: Modification_of_data_types_due_to_the_separation_of_on-boarding_VNF_package_operation - NFVSOL(17)000546R3_SOL005: Define_the_nfpRule_attribute - NFVSOL(17)000578R1_SOL005: Refactor_PNFD_management - NFVSOL(17)000579_SOL005: Annex_B.5_-_Add_NS_fault_management_mapping_operations_to_protocol_elements - NFVSOL(17)000604r1_SOL005: NS_LCM_interface_-_edit_note_fix - NFVSOL(17)000607r1_SOL005: NSD_management_interface_-_consistency_fix - NFVSOL(17)000608r1_SOL005: NSD_management_interface_-_partial_download - NFVSOL(17)000610r1_SOL005: NS_LCM_interface_-_edit_note_fix2 - NFVSOL(17)000613_SOL005: VNF_package_management_interface_delete_VNF_package_flow - NFVSOL(17)000623R1_SOL005: Resolve_editor's_note_in_clause_6.5.3.33_(HealNsData) - NFVSOL(17)000625_SOL005: Remove_"pnfdName"-related_editor's_note_in_clause_5.5.2.5 - NFVSOL(17)000627r1_SOL005: Use_of_verbal_forms_for_the_expression_of_provisions - NFVSOL(17)000628R1_SOL005: Miscellaneous technical improvements - NFVSOL(17)000640_SOL005: Update_sequence_diagram_for_the_get_operations_status_operations
November 2017	0.9.1	Clean-up done by editHelp! E-mail: mailto:edithelp@etsi.org

Date	Version	Information about changes
November 2017	0.10.0	<p>Version 0.10.0 based on contributions that were agreed at the NFVSOL#42 and NFVSOL#43 meetings and during Email Approval (EA) resulting from the NFVSOL#41 meeting.</p> <ul style="list-style-type: none"> - NFVSOL(17)000612R2_SOL005:_VNF_package_management_interface_refactoring_operations_and_data_types - NFVSOL(17)000632R2_SOL005_-_Editorial_changes - NFVSOL(17)000633R3_SOL005:_Sequence_diagram_for_the_deletion_of_an_individual_PNF_descriptor_resource - NFVSOL(17)000639R2_SOL005:_Sequence_diagram_for_the_deletion_of_an_individual_NS_descriptor_resource - NFVSOL(17)000645R1_SOL005:_Adding_405_response - NFVSOL(17)000648_SOL005:_Addition_of_the_notes_for_identifier_in_the_resource_URI - NFVSOL(17)000657_SOL005:_Flow_of_the_creation_of_an_individual_NS_descriptor_resource - NFVSOL(17)000658R1_SOL005:_Flow_of_the_creation_of_an_individual_PNF_descriptor_resource - NFVSOL(17)000659_SOL005:_Resolution_of_Rapporteur's_Notes_in_Annex_A - NFVSOL(17)000660_SOL005_Flow_of_the_querying_reading_of_NS_descriptor_resources - NFVSOL(17)000662_SOL005_Flow_of_the_querying_reading_of_PNF_descriptor_resources - NFVSOL(17)000664_SOL005_Proposed_resolution_of_clause_4_editor's_note - NFVSOL(17)000665_SOL005_Proposed_resolution_of_clause_4.2_editor's_note - NFVSOL(17)000678R1_SOL005_Update_NFP_related_data_types - NFVSOL(17)000679_SOL005_Add_VNF_package_state_model_to_annex_C - NFVSOL(17)000682_SOL005_Flow_of_the_update_of_an_individual_NS_descriptor_resource <p>Version 0.10.0 also reflects additional clean-up done by editHelp!</p> <ul style="list-style-type: none"> - E-mail: mailto:edithelp@etsi.org
December 2017	0.11.0	<p>Version 0.11.0 based on contributions that were agreed at the NFVSOL#45 meeting.</p> <ul style="list-style-type: none"> - NFVSOL(17)000606R3_SOL005:_Update_data_type_NsLcmOpOcc_and_NsLcmOperationOccurrenceNotification - NFVSOL(17)000609R3_SOL005:_NS_LCM_interface_-_Error_handling_enhancement - NFVSOL(17)000614R3_SOL005:_VNF_package_management_interface_upload_VNF_content_through_external_link - NFVSOL(17)000649R2:_SOL005:_6.5.3.2_OperateVnfData - NFVSOL(17)000672R2_SOL005:_Resolution_of_editor's_note_on_structure_of_theNSD_zip_file_in_clause_5.4.4.3.2 - NFVSOL(17)000680R2:_SOL005:_Update_ResourceHandle_datatype - NFVSOL(17)000683R2_SOL005:_Remove_pnfInfold_and_related_editor's_note_from_clause_5.5.2.6 - NFVSOL(17)000701R1_SOL005:_NS_LCM_interface_-_Sequence_diagram_for_continue_operation - NFVSOL(17)000702R1_SOL005:_Update_the_NSD_state_diagram_in_annex - NFVSOL(17)000703R1_SOL005:_NS_LCL_interface_-_resolve_the_editor's_note_about_resource_changes_in_NsLcmOperationOccurrenceNotification - NFVSOL(17)000704R1_SOL005:_Resolve_the_editor's_notes_on_NS_PM_interface - NFVSOL(17)000705_SOL005:_Resolve_the_editor's_notes_on_ExtVirtualLinkData_and_ExtManagedVirtualLinkData - NFVSOL(17)000707R2_SOL005:_Sequence_diagram_for_the_uploading_of_NSD_content - NFVSOL(17)000708R1_SOL005:_Sequence_diagram_for_the_uploading_of_PNFD_content - NFVSOL(17)000709R1_SOL005:_Resolution_of_editor's_note_on_the_NsLcmOpOcc_data_type_in_clause_6.4.9.3.2 - NFVSOL(17)000710_SOL005:_Resolution_of_editor's_note_in_clause_5.4.4.3.3_on_partial/chunking/resumable_upload - NFVSOL(17)000712R1:_SOL005_-_Additional_fields_and_values_for_NfpRules - NFVSOL(17)000713_SOL005_4.2_Consistency_of_URI_and_OAuth

Date	Version	Information about changes
		<ul style="list-style-type: none"> - NFVSOL(17)000714R1:_SOL005:_Resolve_the_editor's_notes_on_ExtLinkPort and NsLinkPort - NFVSOL(17)000716R1_SOL005:_Double_subscriptions_for_notifications - NFVSOL(17)000720_SOL005:_Resolution_of_two_editor's_notes_on_pnfdInvariantId - NFVSOL(17)000721R1_SOL005:_Sequences_of_responses_and_notifications - NFVSOL(17)000727_SOL005:_Flow_of_the_fetching_of_the_content_of_a_NSD - NFVSOL(17)000728_SOL005:_Flow_of_the_fetching_of_the_content_of_a_PNFD - NFVSOL(17)000729_SOL005:_Update_to_the_flow_of_the_creation_of_the_individual_NS_descriptor_resource - NFVSOL(17)000731R1_SOL005:_Fix_description_of_unsupported_method_for_notification_endpoint - NFVSOL(17)000732R1_SOL005:_Resolve_editor's_note_on_checksum - NFVSOL(17)000738_SOL005_Add_Update_PNFD_Info_operation_to_the_list_of_NSD_management_interface_operations_in_clause_5.1 - NFVSOL(17)000741_SOL005:_Fix_the_inconsistency_related_to_NS_LCM_operation_state - NFVSOL(17)000742_SOL005:_Add_PNFD_Notifications - NFVSOL(17)000743R1_SOL005:_Adding_normative_category_to_resource_and_methods_Tables - NFVSOL(17)000744_SOL005:_Align_PkgmNotificationsFilter_with_VnfInstanceSubscriptionFilter - NFVSOL(17)000748R1_SOL005:_Add_NS_change_notification_on_NS_LCM_interface - NFVSOL(17)000750:_SOL005:_Add_error_code_for_fetching_package_content,_vnfd_and_artifact_operations - NFVSOL(17)000755:_SOL005:_Remove_redundant_description_of_vnfConfigurableProperties - NFVSOL(17)000759R2_SOL005_Add_ChangeVnfFlavourData - NFVSOL(17)000760_SOL005_6.5.3.2_Add_changedInfo_to_AffectedVnf_data_type - NFVSOL(17)000762R1_SOL005:_Authorization_of_API_requests_and_notifications
December 2017	0.12.0	<p>Version 0.12.0 based on contributions that were agreed at the NFVSOL#46 meeting and during Email Approval (EA) following the NFVSOL#45 meeting.</p> <ul style="list-style-type: none"> - NFVSOL(17)000603R1_SOL005:_NS_LCM_interface_-_network_address - NFVSOL(17)000745_SOL005:_Change_the_name_of_the "ScaleInfo" data type to "VnfScaleInfo" - NFVSOL(17)000747R2_SOL005:_Add_and_modify_PNF_on_NS_LCM_interface - NFVSOL(17)000752_SOL005:_Change_"pnfdInfoStateType"_to_"pnfdOnboardingStateType"_to_resolve_editor's_note_in_clause_5.5.4.6 - NFVSOL(17)000761_SOL005:_Resolve_NsInstanceSubscriptionFilter_editor's_note_(Issue_Gen.3) - NFVSOL(17)000763_SOL005:_Align_normative_statements_in_trigger_conditions_mirror_734 - NFVSOL(17)000767_SOL005:_Authorization_method_negotiation - NFVSOL(17)000769_SOL005:_Annex_A.6_operation_name_and_resource_URI_alignment - NFVSOL(17)000770_SOL005:Add/remove_notes_about_race_conditions - NFVSOL(17)000772_SOL005:_Add_get_method_support_on_the_notification_endpoint_resource_in_the_NSD_and_NS_LCM_interfaces - NFVSOL(17)000776_SOL005:_Alignment_of_timeStamp_attribute_name - NFVSOL(17)000777_SOL005:_Change "NsLinkPort" to "NsLinkPortInfo" - NFVSOL(17)000779_SOL005:_Precondition_for_VNF_package_deletion - NFVSOL(17)000784_SOL005:_Add_a_pointer_to_annex_B_for_the_NSD_state_model_in_NsdInfo
February 2018	2.4.1	Publication
March 2018	2.4.2	<p>Version 2.4.2 based on contributions that were agreed at the NFVSOL#55 meeting.</p> <ul style="list-style-type: none"> - NFVSOL(18)000047_SOL005ed251_API_authorization_clarification - NFVSOL(18)000084_SOL005ed251_Making_authorization_negotiation_more_flexible
April 2018	2.4.3	Version 2.4.3 based on contributions that were agreed at the NFVSOL#59, NFVSOL#60, and NFVSOL#61 meetings (including EA)

Date	Version	Information about changes
		<ul style="list-style-type: none"> - NFVSOL(18)000098_SOL005ed251_Empty_collections_clarification_addressing_Plugtest™ issue - NFVSOL(18)000137_SOL005ed251_Disambiguating_artifactPath - NFVSOL(18)000155R1_SOL005ed251_Fix_cardinality_of_the_operationParams_attribute_in_the_NsLcmOpOcc_data_type
May 2018	2.4.4	<p>Version 2.4.4 based on contributions that were agreed at the NFVSOL#62, NFVSOL#64, and NFVSOL#65 meetings (including EA)</p> <ul style="list-style-type: none"> - NFVSOL(18)000167_SOL005ed251:_fixing_tracker_issue_007748 - NFVSOL(18)000177_SOL005ed251:_Correct_description_of_POST_method_on_NS_descriptors_resourc - NFVSOL(18)000184_SOL005ed251:_Change_the_cardinality_of_the_subscriptionId_attribute_in_the_NSD_management,_NS_LCM,_and_VNF_package_management_notifications - NFVSOL(18)000233_SOL005ed251:_Define_userDefinedData_attribute_consistently - NFVSOL(18)000235_SOL005ed251:_Remove_reference_to_the_note_in_the_description_of_the_pnfdInvariantId_attribute_in_clause_5.5.2.5 - NFVSOL(18)000238_SOL005ed251:_Updating_JSON_RFC_reference - NFVSOL(18)000243_SOL005ed251:_VnfPkgm_small_fix - NFVSOL(18)000248_SOL005ed251:_Version_Management - NFVSOL(18)000251R1_SOL005ed251:_Move_ResourceHandle_to_common_data_types_in_clause_4.4.1.6
July 2018	2.4.5	<p>Version 2.4.5 based on contributions that were agreed at the NFVSOL#66 and NFVSOL#67 (including EA)</p> <ul style="list-style-type: none"> - NFVSOL(18)000244_SOL005ed251:_Different_names_for_virtual_link_descriptor_id - NFVSOL(18)000245_SOL005ed251:_Adding_status_codes - NFVSOL(18)000259R1_SOL005ed251:_Add_two... monitoringParameter_data_types - NFVSOL(18)000279_SOL005ed251:_Attribute_selectors - NFVSOL(18)000280_SOL005ed251:_Fixing_the_sequence_of_400_response_code_definitions - NFVSOL(18)000281_SOL005ed251:_Small_fix_to_the_description_of_the_400_error_code - NFVSOL(18)000311_SOL005ed251:_Small_fix_replace_queried_by_read - NFVSOL(18)000318_SOL005ed251:_Add_IFA027_reference - NFVSOL(18)000319_SOL005ed251:_Attribute_filters - NFVSOL(18)000320_SOL005ed251:_String_and_number_data_types - NFVSOL(18)000321_SOL005ed251:_Mirror_of_NFVSOL(18)000341r2 - NFVSOL(18)000332R2_SOL005ed251:_Add_annex_with_a_reference_to_OpenAPI_repository - NFVSOL(18)000334_SOL005ed251:_Add_VL_profile_id_in_NsrtualLinkInfo_data_type - NFVSOL(18)000341_SOL005ed251:_Clarifying_association_from_VnfLinkPort_to_VnfcCp_and_VnfExtCp
July 2018	2.4.6	<p>Version 2.4.6 based on contributions that were agreed at the NFVSOL#68 and NFVSOL#69 (including EA)</p> <ul style="list-style-type: none"> - NFVSOL(18)000347_SOL005ed251:_Normative_attribute_filters_support - NFVSOL(18)000350_SOL005ed251:_Metadata_for_CPs_and_extCPs - NFVSOL(18)000355_SOL005ed251:_Add_relative_URIs_for_links_in_notifications - NFVSOL(18)000356_SOL005ed251:_Retry_as_reaction_to_error_responses_during_notification_delivery - NFVSOL(18)000357_SOL005ed251:_NestedNsInstanceData_for_the_InstaniateNs_and_UpdateNs_operations - NFVSOL(18)000358_SOL005ed251:_Add_ParamsForNestedNS_to_the_InstaniateNs_operations - NFVSOL(18)000359_SOL005ed251:_Clarify_linkage_between_vnfcCP_and_vnfcExtCP - NFVSOL(18)000360_SOL005ed251:_Attach_metadata_to_extCPs - NFVSOL(18)000391_SOL005ed251:_Define_minor_version_number - NFVSOL(18)000392_SOL005ed251:_Attribute_selector_attribute_filter_small_fixes
July 2018	2.4.7	<p>Version 2.4.7 based on contributions that were agreed at NFVSOL#71 (including EA)</p> <ul style="list-style-type: none"> - NFVSOL(18)000361_SOL005ed251:_Fix_NFP_Management - NFVSOL(18)000438_SOL005ed251:_Ensure_consistency_with_SOL003_on_VnfLinkPortInfo_and_VnfExtCpInfo

Date	Version	Information about changes
August 2018	2.4.8	Version 2.4.8 based on contributions that were agreed at NFVSOL#72 and NFVSOL#72a (including EA) <ul style="list-style-type: none"> - NFVSOL(18)000456R1_SOL005ed251:_Version_management - NFVSOL(18)000458_SOL005ed251:_Version_signaling - NFVSOL(18)000461R1_SOL005ed251: Define_patch_version_number - NFVSOL(18)000462_SOL005ed251: Closing_pagination_gap - NFVSOL(18)000473R1_SOL005ed251: Add_note_to_MAJOR_version_field
September 2018	2.5.1	Publication
October 2018	2.5.2	Version 2.5.2 based on contributions that were agreed at NFVSOL#78, NFVSOL#79 and NFVSOL#80 (including EA) <ul style="list-style-type: none"> - NFVSOL(18)000559r1_SOL005ed261_Mirror_of_552r5 - NFVSOL(18)000569_Proposal_to_solve_inconsistencies_on_ModifyVnflInfoData_data_type - NFVSOL(18)000588r3_SOL005_Corrections Editorials <ul style="list-style-type: none"> - Updated Rapporteur name
February 2019	2.5.3	Version 2.5.3 based on contributions that were agreed at NFVSOL#89 (EA) <ul style="list-style-type: none"> - NFVSOL(19)000045_SOL005ed261_Mirror_of_581r2 - NFVSOL(19)000048_SOL005ed261_Mirror_of_016 - NFVSOL(19)000049_SOL005ed261_Mirror_of_040
February 2019	2.5.4	Version 2.5.4 based on contributions that were agreed at NFVSOL#90 (EA) <ul style="list-style-type: none"> - NFVSOL(19)000044_SOL005ed261_Mirror_of_584r2 - NFVSOL(19)000047_SOL005ed261_Mirror_of_015 - NFVSOL(19)000092r2_SOL005ed261_Alignment_of_Attributes_of_the_table_GET_method_of_the_PM_jobs - NFVSOL(19)000094r2_SOL005ed261_Mirror_of_057 - NFVSOL(19)000095_SOL005ed261_Mirror_of_022r3 - NFVSOL(19)000123r2_SOL005ed261_API_versions_update
April 2019	2.6.1	Publication
July 2019	2.6.2	Version 2.6.2 based on contributions that were agreed at NFVSOL#96, NFVSOL #101 and NFVSOL #105 (EA) <ul style="list-style-type: none"> - NFVSOL(19)000220r1_SOL005ed271_Small_fix_to_PnfExtCpData_in_Clause_6_5_3_16-1 - NFVSOL(19)000264r2_SOL005Ed271_-_Clause_6_5_3_22_-_ParamsForVnf - NFVSOL(19)000403_SOL005ed271_Update_on_Alarm_data_type_descriptions - NFVSOL(19)000404r1_SOL005ed271_Alignment_of_clauses_4_4_2 - NFVSOL(19)000405_SOL005ed271_Addition_of_standard_text_on_methods_of_clause_6_4_13 - NFVSOL(19)000409_SOL005ed271_Addition_periodicity_of_triggering_to_Performance_Notification_data_type - NFVSOL(19)000417r1_SOL005ed271_General_normative_update - NFVSOL(19)000418_SOL005ed271_Inclusion_of_Content-Type_header_to_PATCH_method - NFVSOL(19)000419r1_SOL005ed271_Clarification_of_success_processing_of_the_asynchronous_request_in_POST_methods - NFVSOL(19)000420r2_SOL005ed271_Updating_of_the_Cancel_operation_description Editorial fixes
October 2019	2.6.3	Version 2.6.3 based on contributions that were agreed at NFVSOL#108, NFVSOL#109, NFVSOL#113, NFVSOL#114 and NFVSOL#115 <ul style="list-style-type: none"> - NFVSOL(19)000511r1_SOL005ed271_Aligning_version_indication_with_SOL015 - NFVSOL(19)000530_SOL005ed271_mirror_of_change_5_in_454 - NFVSOL(19)000540_SOL005ed271_Update_of_the_interfaces_descriptions_with_API_version_information_retrieval - NFVSOL(19)000542r1_SOL005ed271_Alignment_of_resources_descriptions - NFVSOL(19)000547_SOL005ed271_Addition_of_note_in_IpOverEthernetAddressInfo_datatype - NFVSOL(19)000549_SOL005ed271_Addition_of_the_id_in_Notifications_datatypes - NFVSOL(19)000565r1_SOL005ed271_Minor_updates_on_PM_and_FM_interfaces - NFVSOL(19)000575r2_SOL005ed271_Addressing_Ns_Update_operation_data_type_definition_gap

Date	Version	Information about changes
November 2019	2.6.4	<p>Version 2.6.4 based on contributions that were agreed at NFVSOL#116, NFVSOL#117 and NFVSOL#118</p> <ul style="list-style-type: none"> - NFVSOL(19)000468_SOL005_fixes_related_to_IFA027 - NFVSOL(19)000524_SOL005ed271_mirror_of_525__Marking_non-MANO_artifacts_in_VNF - NFVSOL(19)000526r2_SOL005ed271_Mirror_of_482r1_Fetching_the_manifest - NFVSOL(19)000533r2_SOL005ed271_Mirror_of_477r2_Bulk_fetch_of_package_artifacts - NFVSOL(19)000535_SOL005ed271_Mirror_of_483_Exposing_MaxScaleLevel - NFVSOL(19)000539r3_SOL005ed271_Clarify_passing_of_external_connectivity_information - NFVSOL(19)000545r2_SOL005ed271_Alignment_of_LCM_flows - NFVSOL(19)000546r1_SOL005ed271_Alignment_of_resources_and_datatypes_normative_descriptions - NFVSOL(19)000567r5_SOL005ed271_NSD_archive_and_PNFD_archive_management_changes - NFVSOL(19)000571r2_SOL005ed271_support_for_downloading_external_artifacts_from_URI - NFVSOL(19)000583r1_SOL005ed271_mirror_of_581_fixing_the_PM_interface_wrt_subscriptions - NFVSOL(19)000590r2_SOL005ed271_Moving_pre_and_post-conditions_into_normative_clauses - NFVSOL(19)000601_SOL005ed271_fixes_to_FM_interface - NFVSOL(19)000604_SOL005ed271_PATCH_alarm_acknowledge_status - NFVSOL(19)000637_SOL005ed271_mirror_of_523_example_of_artifactPath_in_GET - NFVSOL(19)000649r4_SOL005ed271_Missing_error_state_in_VNF_Package_onboarding - NFVSOL(19)000651_SOL005ed271_Correcting_description_of_individual_VNF_package_content_resource - NFVSOL(19)000652r1_SOL005ed271_Normative_postconditions_after_uploading_VNF_package_content - NFVSOL(19)000668_SOL005ed271_Mirror_of_659_Fixing_non-support_of_range_request - NFVSOL(19)000672r2_SOL005Ed271_-_Initial_configurable_properties_values__Mirror_of_576 - NFVSOL(19)000676r1_SOL005ed271_mirror_of_675r3_Obtaining_artifact_security_info_via_the_API - NFVSOL(19)000678_SOL005ed271_mirror_of_667_fixing_VnfPackageChangeNotification_condition - NFVSOL(19)000681_SOL005ed271_mirror_of_679_adding_error_response_for_failed_notification_endpoint_test - NFVSOL(19)000694_SOL005ed217_adding_support_for_OAuth2_for_VNF_Package_Download - NFVSOL(19)000698_SOL005ed271_mirror_of_667_fixes_to_IFA_mapping_an nex_related_to_dropping_subscriptions_from_PM_API - NFVSOL(19)000709_SOL005ed271_Mirror_of_708_-_clarify_the_zip_file_structure_returned_by_the_fetch_VNFD_operation - NFVSOL(19)000715_SOL005ed271_mirror_of_712r1_mark_for_testing_and_license_artifacts - NFVSOL(19)000723r1_SOL005ed271_mirror_of_722_VNF_package_metadata_in_VnfPkgInfo - NFVSOL(19)000733_SOL005ed271_fixed_multiple_small_bugs - NFVSOL(19)000750_SOL005ed271_mirror_of_474r1_version - NFVSOL(19)000769_SOL005ed271_mirror_of_755_Fix_for_the_disabled_VNF_packages - NFVSOL(19)000778r1_SOL005ed271_mirror_of_752r1_rapporteur_s_cleanup - NFVSOL(19)000782r1_SOL005ed271_API_versions_for_V2_7_1 - NFVSOL(19)000786r1_SOL005ed271_mirror_of_653r2_Representing_the_artifact_path_of_external_artifacts <p>This version contains the changes of NFVSOL(19)000661 as a rapporteur's action</p>
January 2020	2.7.1	Publication
March 2020	2.7.2	<p>Contributions incorporated on version 2.7.2:</p> <ul style="list-style-type: none"> - BWC:NFVSOL(20)000035r3_SOL005ed281_mirror_of_NFVSOL_20_000034_fixing_enumeration_of_PackageUsageStateType - BWC:NFVSOL(20)000111_SOL005ed281_Fixes_of_the_sequence_diagrams_postconditions

Date	Version	Information about changes
April 2020	2.7.3	Contributions incorporated on version 2.7.3: <ul style="list-style-type: none"> - BWC:NFVSOL(20)000173r1_SOL005ed281_Bulk_fetch_of_external_artifacts - BWC:NFVSOL(20)000187_SOL005ed281_Fix_a_Bug_of_the_Data_Type_VirtualStorageResourceInfo - BWC:NFVSOL(20)000229r2_SOL005ed281_SOL016_review_alignments - BWC:NFVSOL(20)000253_SOL005ed281_Small_fixes
June 2020	2.7.4	Contributions incorporated on version 2.7.4: <ul style="list-style-type: none"> - BWC:NFVSOL(20)000283_SOL005ed281_mirror_of_42r5_VnfcResourceInfo_bugfix - BWC:NFVSOL(20)000317_SOL005ed281_Mirror_of_095r1_Fixing_notifying_information_about_extLinkPort - BWC:NFVSOL(20)000361r1_SOL005ed281_Mirror_of_294_Guidelines_link_ports_notification - BWC:NFVSOL(20)000371_SOL005ed281_mirror_of_349_adding_PM_job_id_to_notification - BWC:NFVSOL(20)000411r1_SOL005ed281_backport_mirror_of_14_adding_missing_extensions_and_vnfConfigurableProperties_to_ChangeVnfFlavour - BWC:NFVSOL(20)000422r1_SOL005ed281_Mirror_of_421_Adding_422_error_code_to_the_response_bodies - BWC:NFVSOL(20)000433r1_SOL005ed281_Correction_external_images_VNF_package - BWC:NFVSOL(20)000437_SOL005ed281_Mirror_of_419_Changes_related_to_the_testing_of_notification_endpoint - BWC:NFVSOL(20)000484r2_SOL005ed281_API_versions_update - BWC:NFVSOL(20)000525_SOL005ed281_API_versions_correction_on_top_of_484r2 - BWC:NFVSOL(20)000537r2_SOL005ed281_rel2_mirror_clarification_for_VNF_D_NSD_content - BWC:NFVSOL(20)000561r1_SOL005ed281_Rel2_mirror_of_560 - BWC:NFVSOL(20)000571r1_SOL005ed281_mirror_of_570_LCM-related_fixes with the clarifications in NFVSOL(20)000582_SOL005ed281_clarifications_on_implementing_571r1

History

Document history		
V2.4.1	February 2018	Publication
V2.5.1	September 2018	Publication
V2.6.1	April 2019	Publication
V2.7.1	January 2020	Publication
V2.8.1	September 2020	Publication