



## Network Functions Virtualisation (NFV) Release 2; Protocols and Data Models; RESTful protocols specification for the Or-Vnfm 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-SOL003ed281

---

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](http://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 .....	17
Foreword.....	17
Modal verbs terminology.....	17
1 Scope .....	18
2 References .....	18
2.1 Normative references .....	18
2.2 Informative references.....	19
3 Definition of terms, symbols and abbreviations.....	20
3.1 Terms.....	20
3.2 Symbols.....	20
3.3 Abbreviations .....	20
4 General aspects.....	21
4.1 Overview .....	21
4.2 Void.....	22
4.3 Void.....	22
4.4 Common data types .....	22
4.4.1 Structured data types.....	22
4.4.1.1 Introduction.....	22
4.4.1.2 Void.....	22
4.4.1.3 Void.....	22
4.4.1.3a Void.....	22
4.4.1.4 Void.....	22
4.4.1.5 Type: VnfInstanceSubscriptionFilter .....	22
4.4.1.6 Type: VimConnectionInfo .....	23
4.4.1.7 Type: ResourceHandle .....	24
4.4.1.8 Void.....	25
4.4.1.9 Void.....	25
4.4.1.10 Type: VnfExtCpData .....	25
4.4.1.10a Type: VnfExtCpConfig .....	25
4.4.1.10b Type: CpProtocolData.....	25
4.4.1.10c Type: IpOverEthernetAddressData .....	26
4.4.1.11 Type: ExtVirtualLinkData .....	26
4.4.1.12 Type: ExtManagedVirtualLinkData.....	27
4.4.1.13 Void.....	27
4.4.1.14 Type: ExtLinkPortData .....	27
4.4.2 Simple data types and enumerations.....	28
4.4.2.1 Introduction.....	28
4.4.2.2 Simple data types .....	28
4.4.2.3 Enumerations .....	28
4.5 Void.....	28
4.6 Void.....	28
4.7 Void.....	28
5 VNF Lifecycle Management interface .....	28
5.1 Description .....	28
5.1a API version.....	29
5.2 Resource structure and methods.....	29
5.3 Sequence diagrams (informative).....	32
5.3.1 Flow of the creation of a VNF instance resource.....	32
5.3.2 Flow of the deletion of a VNF instance resource.....	33
5.3.3 Flow of VNF lifecycle management operations triggered by task resources.....	33
5.3.4 Flow of automatic invocation of VNF scaling and VNF healing.....	36
5.3.5 Flow of the Query VNF operation .....	38
5.3.6 Flow of the Modify VNF Information operation .....	39
5.3.7 Flow of the Get Operation Status operation.....	41

5.3.8	Flow of managing subscriptions .....	42
5.3.9	Flow of sending notifications.....	43
5.3.10	Flow of retrying a VNF lifecycle management operation.....	44
5.3.11	Flow of rolling back a VNF lifecycle management operation .....	45
5.3.12	Flow of failing a VNF lifecycle management operation.....	46
5.3.13	Flow of cancelling a VNF lifecycle management operation.....	47
5.4	Resources .....	49
5.4.1	Introduction.....	49
5.4.1.1	Overview .....	49
5.4.1.2	Task resources that trigger VNF LCM operations .....	49
5.4.1a	Resource: API versions.....	51
5.4.2	Resource: VNF instances.....	51
5.4.2.1	Description .....	51
5.4.2.2	Resource definition .....	51
5.4.2.3	Resource methods .....	51
5.4.2.3.1	POST .....	51
5.4.2.3.2	GET .....	52
5.4.2.3.3	PUT .....	54
5.4.2.3.4	PATCH.....	54
5.4.2.3.5	DELETE.....	54
5.4.3	Resource: Individual VNF instance .....	54
5.4.3.1	Description .....	54
5.4.3.2	Resource definition .....	54
5.4.3.3	Resource methods .....	55
5.4.3.3.1	POST .....	55
5.4.3.3.2	GET .....	55
5.4.3.3.3	PUT .....	55
5.4.3.3.4	PATCH.....	55
5.4.3.3.5	DELETE.....	57
5.4.4	Resource: Instantiate VNF task .....	58
5.4.4.1	Description .....	58
5.4.4.2	Resource definition .....	58
5.4.4.3	Resource methods .....	58
5.4.4.3.1	POST .....	58
5.4.4.3.2	GET .....	59
5.4.4.3.3	PUT .....	59
5.4.4.3.4	PATCH.....	59
5.4.4.3.5	DELETE.....	59
5.4.5	Resource: Scale VNF task .....	59
5.4.5.1	Description .....	59
5.4.5.2	Resource definition .....	60
5.4.5.3	Resource methods .....	60
5.4.5.3.1	POST .....	60
5.4.5.3.2	GET .....	61
5.4.5.3.3	PUT .....	61
5.4.5.3.4	PATCH.....	61
5.4.5.3.5	DELETE.....	61
5.4.6	Resource: Scale VNF to Level task .....	62
5.4.6.1	Description .....	62
5.4.6.2	Resource definition .....	62
5.4.6.3	Resource methods .....	62
5.4.6.3.1	POST .....	62
5.4.6.3.2	GET .....	63
5.4.6.3.3	PUT .....	63
5.4.6.3.4	PATCH.....	64
5.4.6.3.5	DELETE.....	64
5.4.7	Resource: Change VNF Flavour task .....	64
5.4.7.1	Description .....	64
5.4.7.2	Resource definition .....	64
5.4.7.3	Resource methods .....	64
5.4.7.3.1	POST .....	64
5.4.7.3.2	GET .....	66

5.4.7.3.3	PUT .....	66
5.4.7.3.4	PATCH.....	66
5.4.7.3.5	DELETE.....	66
5.4.8	Resource: Terminate VNF task.....	66
5.4.8.1	Description.....	66
5.4.8.2	Resource definition .....	66
5.4.8.3	Resource methods .....	66
5.4.8.3.1	POST .....	66
5.4.8.3.2	GET .....	67
5.4.8.3.3	PUT .....	67
5.4.8.3.4	PATCH.....	67
5.4.8.3.5	DELETE.....	68
5.4.9	Resource: Heal VNF task .....	68
5.4.9.1	Description.....	68
5.4.9.2	Resource definition .....	68
5.4.9.3	Resource methods .....	68
5.4.9.3.1	POST .....	68
5.4.9.3.2	GET .....	69
5.4.9.3.3	PUT .....	69
5.4.9.3.4	PATCH.....	70
5.4.9.3.5	DELETE.....	70
5.4.10	Resource: Operate VNF task .....	70
5.4.10.1	Description.....	70
5.4.10.2	Resource definition .....	70
5.4.10.3	Resource methods .....	71
5.4.10.3.1	POST .....	71
5.4.10.3.2	GET .....	72
5.4.10.3.3	PUT .....	72
5.4.10.3.4	PATCH.....	72
5.4.10.3.5	DELETE.....	72
5.4.11	Resource: Change external VNF connectivity task .....	72
5.4.11.1	Description.....	72
5.4.11.2	Resource definition .....	73
5.4.11.3	Resource methods .....	73
5.4.11.3.1	POST .....	73
5.4.11.3.2	GET .....	74
5.4.11.3.3	PUT .....	74
5.4.11.3.4	PATCH.....	74
5.4.11.3.5	DELETE.....	74
5.4.12	Resource: VNF LCM operation occurrences.....	74
5.4.12.1	Description.....	74
5.4.12.2	Resource definition .....	74
5.4.12.3	Resource methods .....	75
5.4.12.3.1	POST .....	75
5.4.12.3.2	GET .....	75
5.4.12.3.3	PUT .....	76
5.4.12.3.4	PATCH.....	76
5.4.12.3.5	DELETE.....	76
5.4.13	Resource: Individual VNF LCM operation occurrence.....	77
5.4.13.1	Description.....	77
5.4.13.2	Resource definition .....	77
5.4.13.3	Resource methods .....	77
5.4.13.3.1	POST .....	77
5.4.13.3.2	GET .....	77
5.4.13.3.3	PUT .....	78
5.4.13.3.4	PATCH.....	78
5.4.13.3.5	DELETE.....	78
5.4.14	Resource: Retry operation task.....	78
5.4.14.1	Description.....	78
5.4.14.2	Resource definition .....	78
5.4.14.3	Resource methods .....	79
5.4.14.3.1	POST .....	79

5.4.14.3.2	GET .....	80
5.4.14.3.3	PUT .....	80
5.4.14.3.4	PATCH.....	80
5.4.14.3.5	DELETE.....	80
5.4.15	Resource: Rollback operation task.....	80
5.4.15.1	Description.....	80
5.4.15.2	Resource definition .....	80
5.4.15.3	Resource methods .....	80
5.4.15.3.1	POST .....	80
5.4.15.3.2	GET .....	81
5.4.15.3.3	PUT .....	81
5.4.15.3.4	PATCH.....	82
5.4.15.3.5	DELETE.....	82
5.4.16	Resource: Fail operation task.....	82
5.4.16.1	Description.....	82
5.4.16.2	Resource definition .....	82
5.4.16.3	Resource methods .....	82
5.4.16.3.1	POST .....	82
5.4.16.3.2	GET .....	83
5.4.16.3.3	PUT .....	83
5.4.16.3.4	PATCH.....	83
5.4.16.3.5	DELETE.....	84
5.4.17	Resource: Cancel operation task.....	84
5.4.17.1	Description.....	84
5.4.17.2	Resource definition .....	84
5.4.17.3	Resource methods .....	84
5.4.17.3.1	POST .....	84
5.4.17.3.2	GET .....	85
5.4.17.3.3	PUT .....	85
5.4.17.3.4	PATCH.....	86
5.4.17.3.5	DELETE.....	86
5.4.18	Resource: Subscriptions.....	86
5.4.18.1	Description.....	86
5.4.18.2	Resource definition .....	86
5.4.18.3	Resource methods .....	86
5.4.18.3.1	POST .....	86
5.4.18.3.2	GET .....	87
5.4.18.3.3	PUT .....	88
5.4.18.3.4	PATCH.....	89
5.4.18.3.5	DELETE.....	89
5.4.19	Resource: Individual subscription.....	89
5.4.19.1	Description.....	89
5.4.19.2	Resource definition .....	89
5.4.19.3	Resource methods .....	89
5.4.19.3.1	POST .....	89
5.4.19.3.2	GET .....	89
5.4.19.3.3	PUT .....	90
5.4.19.3.4	PATCH.....	90
5.4.19.3.5	DELETE.....	90
5.4.20	Resource: Notification endpoint .....	91
5.4.20.1	Description.....	91
5.4.20.2	Resource definition .....	91
5.4.20.3	Resource methods .....	91
5.4.20.3.1	POST .....	91
5.4.20.3.2	GET .....	91
5.4.20.3.3	PUT .....	92
5.4.20.3.4	PATCH.....	92
5.4.20.3.5	DELETE.....	92
5.5	Data model .....	92
5.5.1	Introduction.....	92
5.5.2	Resource and notification data types .....	92
5.5.2.1	Introduction.....	92

5.5.2.2	Type: VnfInstance .....	92
5.5.2.3	Type: CreateVnfRequest .....	97
5.5.2.4	Type: InstantiateVnfRequest .....	97
5.5.2.5	Type: ScaleVnfRequest .....	98
5.5.2.6	Type: ScaleVnfToLevelRequest .....	99
5.5.2.7	Type: ChangeVnfFlavourRequest .....	99
5.5.2.8	Type: TerminateVnfRequest .....	100
5.5.2.9	Type: HealVnfRequest .....	100
5.5.2.10	Type: OperateVnfRequest .....	101
5.5.2.11	Type: ChangeExtVnfConnectivityRequest .....	101
5.5.2.12	Type: VnfInfoModificationRequest .....	102
5.5.2.12a	Type: VnfInfoModifications .....	103
5.5.2.13	Type: VnfLcmOpOcc .....	104
5.5.2.14	Type: CancelMode .....	105
5.5.2.15	Type: LccnSubscriptionRequest .....	105
5.5.2.16	Type: LccnSubscription .....	106
5.5.2.17	Type: VnfLcmOperationOccurrenceNotification .....	106
5.5.2.18	Type: VnfIdentifierCreationNotification .....	108
5.5.2.19	Type: VnfIdentifierDeletionNotification .....	109
5.5.3	Referenced structured data types .....	109
5.5.3.1	Introduction .....	109
5.5.3.2	Type: ExtVirtualLinkInfo .....	109
5.5.3.3	Type: ExtManagedVirtualLinkInfo .....	109
5.5.3.4	Type: ScaleInfo .....	110
5.5.3.5	Type: VnfcResourceInfo .....	110
5.5.3.6	Type: VnfVirtualLinkResourceInfo .....	111
5.5.3.7	Type: VirtualStorageResourceInfo .....	111
5.5.3.8	Type: VnfLinkPortInfo .....	111
5.5.3.9	Type: ExtLinkPortInfo .....	112
5.5.3.9a	Void .....	113
5.5.3.9b	Type: CpProtocolInfo .....	113
5.5.3.10	Type: IpOverEthernetAddressInfo .....	113
5.5.3.11	Type: MonitoringParameter .....	114
5.5.3.12	Type: LifecycleChangeNotificationsFilter .....	114
5.5.3.13	Type: AffectedVnfc .....	115
5.5.3.14	Type: AffectedVirtualLink .....	116
5.5.3.14a	Type: AffectedExtLinkPort .....	117
5.5.3.15	Type: AffectedVirtualStorage .....	117
5.5.3.16	Type: LccnLinks .....	118
5.5.3.17	Type: VnfExtCpInfo .....	119
5.5.4	Referenced simple data types and enumerations .....	119
5.5.4.1	Introduction .....	119
5.5.4.2	Simple data types .....	119
5.5.4.3	Enumeration: VnfOperationalStateType .....	119
5.5.4.4	Enumeration: StopType .....	119
5.5.4.5	Enumeration: LcmOperationStateType .....	120
5.5.4.6	Enumeration: CancelModeType .....	120
5.5.4.7	Enumeration: LcmOperationType .....	120
5.6	Success and error states of VNF lifecycle management operations .....	121
5.6.1	Basic concepts for error handling (informative) .....	121
5.6.1.1	Motivation .....	121
5.6.1.2	Failure resolution strategies: Retry and Rollback .....	121
5.6.1.3	Error handling at VNFM and NFVO .....	122
5.6.2	States and state transitions of a VNF lifecycle management operation occurrence .....	123
5.6.2.1	General .....	123
5.6.2.2	States of a VNF lifecycle management operation occurrence .....	123
5.6.2.3	Error handling operations that change the state of a VNF lifecycle management operation occurrence .....	126
5.6.3	Detailed flows for error handling .....	127
5.6.3.1	Immediate failure .....	127
5.6.3.2	Failure in "STARTING" state .....	127
5.6.3.3	Failure during actual LCM operation execution .....	128



5.6.3.4	LCM operation cancellation.....	130
6	VNF Performance Management interface.....	130
6.1	Description.....	130
6.1a	API version.....	130
6.2	Resource structure and methods.....	130
6.3	Sequence diagrams (informative).....	132
6.3.1	Flow of creating a PM job.....	132
6.3.1a	Flow of updating the callback URI of a PM job.....	133
6.3.2	Flow of querying/reading PM jobs.....	133
6.3.3	Flow of deleting a PM job.....	134
6.3.4	Flow of obtaining performance reports.....	135
6.3.5	Flow of creating a threshold.....	136
6.3.5a	Flow of updating the callback URI of a threshold.....	137
6.3.6	Flow of querying/reading thresholds.....	137
6.3.7	Flow of deleting thresholds.....	138
6.3.8	Void.....	139
6.3.9	Flow of sending notifications.....	139
6.4	Resources.....	139
6.4.1	Introduction.....	139
6.4.1a	Resource: API versions.....	139
6.4.2	Resource: PM jobs.....	140
6.4.2.1	Description.....	140
6.4.2.2	Resource definition.....	140
6.4.2.3	Resource methods.....	140
6.4.2.3.1	POST.....	140
6.4.2.3.2	GET.....	141
6.4.2.3.3	PUT.....	142
6.4.2.3.4	PATCH.....	142
6.4.2.3.5	DELETE.....	142
6.4.3	Resource: Individual PM job.....	143
6.4.3.1	Description.....	143
6.4.3.2	Resource definition.....	143
6.4.3.3	Resource methods.....	143
6.4.3.3.1	POST.....	143
6.4.3.3.2	GET.....	143
6.4.3.3.3	PUT.....	144
6.4.3.3.4	PATCH.....	144
6.4.3.3.5	DELETE.....	145
6.4.4	Resource: Individual performance report.....	145
6.4.4.1	Description.....	145
6.4.4.2	Resource definition.....	145
6.4.4.3	Resource methods.....	146
6.4.4.3.1	POST.....	146
6.4.4.3.2	GET.....	146
6.4.4.3.3	PUT.....	146
6.4.4.3.4	PATCH.....	146
6.4.4.3.5	DELETE.....	146
6.4.5	Resource: Thresholds.....	146
6.4.5.1	Description.....	146
6.4.5.2	Resource definition.....	147
6.4.5.3	Resource methods.....	147
6.4.5.3.1	POST.....	147
6.4.5.3.2	GET.....	148
6.4.5.3.3	PUT.....	149
6.4.5.3.4	PATCH.....	149
6.4.5.3.5	DELETE.....	149
6.4.6	Resource: Individual threshold.....	149
6.4.6.1	Description.....	149
6.4.6.2	Resource definition.....	150
6.4.6.3	Resource methods.....	150
6.4.6.3.1	POST.....	150



6.4.6.3.2	GET .....	150
6.4.6.3.3	PUT .....	150
6.4.6.3.4	PATCH.....	151
6.4.6.3.5	DELETE.....	152
6.4.7	Void.....	152
6.4.8	Void.....	152
6.4.9	Resource: Notification endpoint .....	152
6.4.9.1	Description.....	152
6.4.9.2	Resource definition .....	152
6.4.9.3	Resource methods .....	153
6.4.9.3.1	POST .....	153
6.4.9.3.2	GET .....	153
6.4.9.3.3	PUT .....	154
6.4.9.3.4	PATCH.....	154
6.4.9.3.5	DELETE.....	154
6.5	Data Model.....	154
6.5.1	Introduction.....	154
6.5.2	Resource and notification data types .....	154
6.5.2.1	Introduction.....	154
6.5.2.2	Void.....	154
6.5.2.3	Void.....	154
6.5.2.4	Type: ThresholdCrossedNotification .....	154
6.5.2.5	Type: PerformanceInformationAvailableNotification .....	155
6.5.2.6	Type: CreatePmJobRequest .....	156
6.5.2.7	Type: PmJob .....	157
6.5.2.8	Type: CreateThresholdRequest .....	157
6.5.2.9	Type: Threshold .....	158
6.5.2.10	Type: PerformanceReport .....	159
6.5.2.11	Type: ThresholdModifications .....	159
6.5.2.12	Type: PmJobModifications .....	160
6.5.3	Referenced structured data types.....	160
6.5.3.1	Introduction.....	160
6.5.3.2	Void.....	160
6.5.3.3	Type: PmJobCriteria .....	160
6.5.3.4	Type: ThresholdCriteria .....	161
6.5.4	Referenced simple data types and enumerations .....	161
6.5.4.1	Introduction.....	161
6.5.4.2	Simple data types .....	161
6.5.4.3	Enumeration: CrossingDirectionType.....	161
7	VNF Fault Management interface.....	162
7.1	Description .....	162
7.1a	API version.....	162
7.2	Resource structure and methods.....	162
7.3	Sequence diagrams (informative).....	163
7.3.1	Flow of the Get Alarm List operation.....	163
7.3.2	Flow of acknowledging alarm .....	164
7.3.3	Flow of managing subscriptions .....	165
7.3.4	Flow of sending notifications.....	166
7.4	Resources .....	167
7.4.1	Introduction.....	167
7.4.2	Resource: Alarms.....	167
7.4.2.1	Description.....	167
7.4.2.2	Resource definition .....	167
7.4.2.3	Resource methods .....	167
7.4.2.3.1	POST .....	167
7.4.2.3.2	GET .....	167
7.4.2.3.3	PUT .....	168
7.4.2.3.4	PATCH.....	169
7.4.2.3.5	DELETE.....	169
7.4.3	Resource: Individual alarm .....	169
7.4.3.1	Description.....	169

7.4.3.2	Resource definition .....	169
7.4.3.3	Resource methods .....	169
7.4.3.3.1	POST .....	169
7.4.3.3.2	GET .....	169
7.4.3.3.3	PUT .....	170
7.4.3.3.4	PATCH .....	170
7.4.3.3.5	DELETE .....	171
7.4.4	Resource: Subscriptions .....	171
7.4.4.1	Description .....	171
7.4.4.2	Resource definition .....	171
7.4.4.3	Resource methods .....	171
7.4.4.3.1	POST .....	171
7.4.4.3.2	GET .....	173
7.4.4.3.3	PUT .....	174
7.4.4.3.4	PATCH .....	174
7.4.4.3.5	DELETE .....	174
7.4.5	Resource: Individual subscription .....	174
7.4.5.1	Description .....	174
7.4.5.2	Resource definition .....	174
7.4.5.3	Resource methods .....	174
7.4.5.3.1	POST .....	174
7.4.5.3.2	GET .....	174
7.4.5.3.3	PUT .....	175
7.4.5.3.4	PATCH .....	175
7.4.5.3.5	DELETE .....	175
7.4.6	Resource: Notification endpoint .....	176
7.4.6.1	Description .....	176
7.4.6.2	Resource definition .....	176
7.4.6.3	Resource methods .....	176
7.4.6.3.1	POST .....	176
7.4.6.3.2	GET .....	177
7.4.6.3.3	PUT .....	177
7.4.6.3.4	PATCH .....	177
7.4.6.3.5	DELETE .....	177
7.5	Data Model .....	177
7.5.1	Introduction .....	177
7.5.2	Resource and notification data types .....	177
7.5.2.1	Introduction .....	177
7.5.2.2	Type: FmSubscriptionRequest .....	178
7.5.2.3	Type: FmSubscription .....	178
7.5.2.4	Type: Alarm .....	178
7.5.2.5	Type: AlarmNotification .....	179
7.5.2.6	Type: AlarmClearedNotification .....	179
7.5.2.7	Type: AlarmListRebuiltNotification .....	180
7.5.2.8	Type: AlarmModifications .....	180
7.5.3	Referenced structured data types .....	181
7.5.3.1	Introduction .....	181
7.5.3.2	Type: FmNotificationsFilter .....	181
7.5.3.3	Type: FaultyResourceInfo .....	181
7.5.4	Referenced simple data types and enumerations .....	181
7.5.4.1	Introduction .....	181
7.5.4.2	Simple data types .....	182
7.5.4.3	Enumeration: PerceivedSeverityType .....	182
7.5.4.4	Enumeration: EventType .....	182
7.5.4.5	Enumeration: FaultyResourceType .....	182
8	VNF Indicator interface .....	183
8.1	Description .....	183
8.1a	API version .....	183
8.2	Resource structure and methods .....	183
8.3	Sequence diagrams (informative) .....	184
8.3.1	Flow of querying VNF indicators .....	184

8.3.2	Flow of reading a VNF indicator .....	185
8.3.3	Flow of managing subscriptions .....	186
8.3.4	Flow of sending notifications.....	187
8.4	Resources .....	188
8.4.1	Introduction.....	188
8.4.1a	Resource: API versions.....	188
8.4.2	Resource: VNF indicators.....	188
8.4.2.1	Description .....	188
8.4.2.2	Resource definition .....	188
8.4.2.3	Resource methods .....	188
8.4.2.3.1	POST .....	188
8.4.2.3.2	GET .....	188
8.4.2.3.3	PUT .....	189
8.4.2.3.4	PATCH.....	190
8.4.2.3.5	DELETE.....	190
8.4.3	Resource: VNF indicators related to a VNF instance .....	190
8.4.3.1	Description .....	190
8.4.3.2	Resource definition .....	190
8.4.3.3	Resource methods .....	190
8.4.3.3.1	POST .....	190
8.4.3.3.2	GET .....	190
8.4.3.3.3	PUT .....	191
8.4.3.3.4	PATCH.....	192
8.4.3.3.5	DELETE.....	192
8.4.4	Resource: Individual VNF indicator.....	192
8.4.4.1	Description .....	192
8.4.4.2	Resource definition .....	192
8.4.4.3	Resource methods .....	192
8.4.4.3.1	POST .....	192
8.4.4.3.2	GET .....	192
8.4.4.3.3	PUT .....	193
8.4.4.3.4	PATCH.....	193
8.4.4.3.5	DELETE.....	193
8.4.5	Resource: Subscriptions.....	193
8.4.5.1	Description .....	193
8.4.5.2	Resource definition .....	193
8.4.5.3	Resource methods .....	193
8.4.5.3.1	POST .....	193
8.4.5.3.2	GET .....	195
8.4.5.3.3	PUT .....	196
8.4.5.3.4	PATCH.....	196
8.4.5.3.5	DELETE.....	196
8.4.6	Resource: Individual subscription.....	196
8.4.6.1	Description .....	196
8.4.6.2	Resource definition .....	196
8.4.6.3	Resource methods .....	197
8.4.6.3.1	POST .....	197
8.4.6.3.2	GET .....	197
8.4.6.3.3	PUT .....	197
8.4.6.3.4	PATCH.....	197
8.4.6.3.5	DELETE.....	197
8.4.7	Resource: Notification endpoint .....	198
8.4.7.1	Description .....	198
8.4.7.2	Resource definition .....	198
8.4.7.3	Resource methods .....	198
8.4.7.3.1	POST .....	198
8.4.7.3.2	GET .....	199
8.4.7.3.3	PUT .....	199
8.4.7.3.4	PATCH.....	199
8.4.7.3.5	DELETE.....	199
8.5	Data model .....	200
8.5.1	Introduction.....	200

8.5.2	Resource and notification data types .....	200
8.5.2.1	Introduction .....	200
8.5.2.2	Type: VnfIndicator .....	200
8.5.2.3	Type: VnfIndicatorSubscriptionRequest .....	200
8.5.2.4	Type: VnfIndicatorSubscription .....	200
8.5.2.5	Type: VnfIndicatorValueChangeNotification .....	201
8.5.3	Referenced structured data types .....	201
8.5.3.1	Introduction .....	201
8.5.3.2	Type: VnfIndicatorNotificationsFilter .....	201
8.5.4	Referenced simple data types and enumerations .....	202
9	VNF Lifecycle Operation Granting interface .....	202
9.1	Description .....	202
9.1a	API version .....	203
9.2	Resource structure and methods .....	203
9.3	Sequence diagrams (informative) .....	204
9.3.1	Flow of grant request with synchronous response .....	204
9.3.2	Flow of grant request with asynchronous response .....	204
9.4	Resources .....	205
9.4.1	Introduction .....	205
9.4.1a	Resource: API versions .....	205
9.4.2	Resource: Grants .....	206
9.4.2.1	Description .....	206
9.4.2.2	Resource definition .....	206
9.4.2.3	Resource methods .....	206
9.4.2.3.1	POST .....	206
9.4.2.3.2	GET .....	207
9.4.2.3.3	PUT .....	207
9.4.2.3.4	PATCH .....	207
9.4.2.3.5	DELETE .....	207
9.4.3	Resource: Individual grant .....	207
9.4.3.1	Description .....	207
9.4.3.2	Resource definition .....	207
9.4.3.3	Resource methods .....	208
9.4.3.3.1	POST .....	208
9.4.3.3.2	GET .....	208
9.4.3.3.3	PUT .....	208
9.4.3.3.4	PATCH .....	209
9.4.3.3.5	DELETE .....	209
9.5	Data model .....	209
9.5.1	Introduction .....	209
9.5.2	Resource and notification data types .....	209
9.5.2.1	Introduction .....	209
9.5.2.2	Type: GrantRequest .....	209
9.5.2.3	Type: Grant .....	210
9.5.3	Referenced structured data types .....	212
9.5.3.1	Introduction .....	212
9.5.3.2	Type: ResourceDefinition .....	212
9.5.3.3	Type: GrantInfo .....	213
9.5.3.4	Type: ZoneInfo .....	214
9.5.3.5	Type: ZoneGroupInfo .....	214
9.5.3.6	Type: PlacementConstraint .....	215
9.5.3.7	Type: VimConstraint .....	215
9.5.3.8	Type: ConstraintResourceRef .....	216
9.5.3.9	Type: VimComputeResourceFlavour .....	216
9.5.3.10	Type: VimSoftwareImage .....	217
9.5.4	Referenced simple data types and enumerations .....	217
9.5.4.1	Introduction .....	217
9.5.4.2	Simple data types .....	217
9.5.4.3	Enumeration: GrantedLcmOperationType .....	217
10	VNF Package Management interface .....	218

10.1	Description .....	218
10.1a	API version.....	218
10.2	Resource structure and methods.....	218
10.3	Sequence diagrams (informative).....	220
10.3.1	Flow of querying/reading VNF package information.....	220
10.3.2	Flow of reading the VNFD of an on-boarded VNF package.....	221
10.3.2a	Flow of fetching the VNF package manifest.....	221
10.3.3	Flow of fetching an on-boarded VNF package.....	222
10.3.4	Flow of fetching a VNF package artifact.....	223
10.3.4a	Flow of bulk-fetching VNF package artifacts that are not images.....	224
10.3.5	Flow of managing subscriptions.....	224
10.3.6	Flow of sending notifications.....	226
10.4	Resources .....	227
10.4.1	Introduction.....	227
10.4.1a	Resource: API versions.....	227
10.4.2	Resource: VNF packages.....	227
10.4.2.1	Description.....	227
10.4.2.2	Resource definition.....	227
10.4.2.3	Resource methods.....	227
10.4.2.3.1	POST.....	227
10.4.2.3.2	GET.....	227
10.4.2.3.3	PUT.....	229
10.4.2.3.4	PATCH.....	229
10.4.2.3.5	DELETE.....	229
10.4.3	Resource: Individual VNF package.....	230
10.4.3.1	Description.....	230
10.4.3.2	Resource definition.....	230
10.4.3.3	Resource methods.....	230
10.4.3.3.1	POST.....	230
10.4.3.3.2	GET.....	230
10.4.3.3.3	PUT.....	231
10.4.3.3.4	PATCH.....	231
10.4.3.3.5	DELETE.....	231
10.4.4	Resource: VNFD in an individual VNF package.....	231
10.4.4.1	Description.....	231
10.4.4.2	Resource definition.....	231
10.4.4.3	Resource methods.....	232
10.4.4.3.1	POST.....	232
10.4.4.3.2	GET.....	232
10.4.4.3.3	PUT.....	234
10.4.4.3.4	PATCH.....	234
10.4.4.3.5	DELETE.....	234
10.4.4a	Resource: Manifest in an individual VNF package.....	234
10.4.4a.1	Description.....	234
10.4.4a.2	Resource definition.....	234
10.4.4a.3	Resource methods.....	235
10.4.4a.3.1	POST.....	235
10.4.4a.3.2	GET.....	235
10.4.4a.3.3	PUT.....	236
10.4.4a.3.4	PATCH.....	236
10.4.4a.3.5	DELETE.....	236
10.4.5	Resource: VNF package content.....	236
10.4.5.1	Description.....	236
10.4.5.2	Resource definition.....	236
10.4.5.3	Resource methods.....	237
10.4.5.3.1	POST.....	237
10.4.5.3.2	GET.....	237
10.4.5.3.3	PUT.....	238
10.4.5.3.4	PATCH.....	238
10.4.5.3.5	DELETE.....	238
10.4.5a	Resource: VNF package artifacts.....	238
10.4.5a.1	Description.....	238