



Network Functions Virtualisation (NFV) Release 2; Protocols and Data Models; RESTful protocols specification for the Or-Vnm 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-SOL003ed251

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 only prevailing document is the print of the Portable Document Format (PDF) version kept on a specific network drive within ETSI Secretariat.

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 2018.

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 protected for the benefit of its Members.

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 Definitions and abbreviations.....	21
3.1 Definitions	21
3.2 Abbreviations.....	21
4 General aspects.....	22
4.1 Overview	22
4.2 URI structure and supported content formats	23
4.3 Common procedures	24
4.3.1 Introduction	24
4.3.2 Attribute-based filtering	24
4.3.2.1 Overview and example (informative)	24
4.3.2.2 Specification	25
4.3.3 Attribute selectors	26
4.3.3.1 Overview and example (informative)	26
4.3.3.2 Specification	27
4.3.3.2.1 GET request	27
4.3.3.2.2 GET response.....	27
4.3.4 Usage of HTTP header fields	28
4.3.4.1 Introduction	28
4.3.4.2 Request header fields	28
4.3.4.3 Response header fields	29
4.3.5 Error reporting.....	30
4.3.5.1 Introduction	30
4.3.5.2 General mechanism	30
4.3.5.3 Type: ProblemDetails	31
4.3.5.4 Common error situations	31
4.3.5.5 Overview of HTTP error status codes	33
4.4 Common data types	35
4.4.1 Structured data types	35
4.4.1.1 Introduction	35
4.4.1.2 Type: Object	35
4.4.1.3 Type: Link	35
4.4.1.3a Type: NotificationLink	35
4.4.1.4 Type: KeyValuePairs.....	35
4.4.1.5 Type: VnfInstanceSubscriptionFilter.....	35
4.4.1.6 Type: VimConnectionInfo.....	36
4.4.1.7 Type: ResourceHandle	37
4.4.1.8 void.....	38
4.4.1.9 void.....	38
4.4.1.10 Type: VnfExtCpData	38
4.4.1.10a Type: VnfExtCpConfig	38
4.4.1.10b Type: CpProtocolData	39
4.4.1.10c Type: IpOverEthernetAddressData	39
4.4.1.11 Type: ExtVirtualLinkData	40
4.4.1.12 Type: ExtManagedVirtualLinkData	41
4.4.1.13 Type: ApiVersionInformation	41
4.4.2 Simple data types and enumerations	42
4.4.2.1 Introduction	42

4.4.2.2	Simple data types.....	42
4.4.2.3	Void.....	43
4.5	Authorization of API requests and notifications.....	43
4.5.1	Introduction.....	43
4.5.2	Flows (informative).....	43
4.5.2.0	General.....	43
4.5.2.1	Authorization of API requests using OAuth 2.0 access tokens.....	43
4.5.2.1a	Authorization of API requests using TLS certificates.....	45
4.5.2.2	Authorization of notifications using the HTTP Basic authentication scheme.....	46
4.5.2.3	Authorization of notifications using OAuth 2.0 access tokens.....	47
4.5.2.4	Authorization of notifications using TLS certificates.....	49
4.5.3	Specification.....	51
4.5.3.1	Introduction.....	51
4.5.3.2	General mechanism.....	51
4.5.3.3	Authorizing API requests.....	51
4.5.3.4	Authorizing the sending of notifications.....	52
4.5.3.5	Client roles.....	53
4.5.3.6	Negotiation of the authorization method.....	54
4.5.3.6.1	Authorization of API requests.....	54
4.5.3.6.2	Authorization of notification requests.....	56
4.6	Version management.....	57
4.6.1	Version identifiers and parameters.....	57
4.6.1.1	Version identifiers.....	57
4.6.1.2	Version parameters.....	57
4.6.2	Rules for incrementing version identifier fields.....	57
4.6.2.1	General.....	57
4.6.2.2	Examples of backward and non-backward compatible changes.....	58
4.6.3	Version information retrieval.....	59
4.6.3.1	General.....	59
4.6.3.2	Resource structure and methods.....	59
4.6.3.3	Resource: API versions.....	60
4.6.3.3.1	Description.....	60
4.6.3.3.2	Resource definition.....	60
4.6.3.3.3	Resource methods.....	60
4.6.4	Version signaling.....	61
4.7	Handling of large query results.....	62
4.7.1	Description.....	62
4.7.2	Specification.....	62
4.7.2.1	Alternatives.....	62
4.7.2.2	Error response.....	62
4.7.2.3	Paged response.....	63
5	VNF Lifecycle Management interface.....	63
5.1	Description.....	63
5.1a	API version.....	64
5.2	Resource structure and methods.....	64
5.3	Sequence diagrams (informative).....	67
5.3.1	Flow of the creation of a VNF instance resource.....	67
5.3.2	Flow of the deletion of a VNF instance resource.....	68
5.3.3	Flow of VNF lifecycle management operations triggered by task resources.....	68
5.3.4	Flow of automatic invocation of VNF scaling and VNF healing.....	71
5.3.5	Flow of the Query VNF operation.....	73
5.3.6	Flow of the Modify VNF Information operation.....	74
5.3.7	Flow of the Get Operation Status operation.....	76
5.3.8	Flow of managing subscriptions.....	77
5.3.9	Flow of sending notifications.....	78
5.3.10	Flow of retrying a VNF lifecycle management operation.....	79
5.3.11	Flow of rolling back a VNF lifecycle management operation.....	80
5.3.12	Flow of failing a VNF lifecycle management operation.....	81
5.3.13	Flow of cancelling a VNF lifecycle management operation.....	82
5.4	Resources.....	84
5.4.1	Introduction.....	84

5.4.1a	Resource: API versions	84
5.4.2	Resource: VNF instances	85
5.4.2.1	Description	85
5.4.2.2	Resource definition.....	85
5.4.2.3	Resource methods.....	85
5.4.2.3.1	POST.....	85
5.4.2.3.2	GET.....	86
5.4.2.3.3	PUT.....	87
5.4.2.3.4	PATCH	87
5.4.2.3.5	DELETE	87
5.4.3	Resource: Individual VNF instance	87
5.4.3.1	Description	87
5.4.3.2	Resource definition.....	88
5.4.3.3	Resource methods.....	88
5.4.3.3.1	POST.....	88
5.4.3.3.2	GET.....	88
5.4.3.3.3	PUT.....	88
5.4.3.3.4	PATCH	89
5.4.3.3.5	DELETE	90
5.4.4	Resource: Instantiate VNF task.....	90
5.4.4.1	Description	90
5.4.4.2	Resource definition.....	90
5.4.4.3	Resource methods.....	91
5.4.4.3.1	POST.....	91
5.4.4.3.2	GET.....	91
5.4.4.3.3	PUT.....	91
5.4.4.3.4	PATCH	91
5.4.4.3.5	DELETE	92
5.4.5	Resource: Scale VNF task.....	92
5.4.5.1	Description	92
5.4.5.2	Resource definition.....	92
5.4.5.3	Resource methods.....	92
5.4.5.3.1	POST.....	92
5.4.5.3.2	GET.....	93
5.4.5.3.3	PUT.....	93
5.4.5.3.4	PATCH	93
5.4.5.3.5	DELETE	94
5.4.6	Resource: Scale VNF to Level task.....	94
5.4.6.1	Description	94
5.4.6.2	Resource definition.....	94
5.4.6.3	Resource methods.....	94
5.4.6.3.1	POST.....	94
5.4.6.3.2	GET.....	95
5.4.6.3.3	PUT.....	95
5.4.6.3.4	PATCH	95
5.4.6.3.5	DELETE	96
5.4.7	Resource: Change VNF Flavour task.....	96
5.4.7.1	Description	96
5.4.7.2	Resource definition.....	96
5.4.7.3	Resource methods.....	96
5.4.7.3.1	POST.....	96
5.4.7.3.2	GET.....	97
5.4.7.3.3	PUT.....	97
5.4.7.3.4	PATCH	98
5.4.7.3.5	DELETE	98
5.4.8	Resource: Terminate VNF task.....	98
5.4.8.1	Description	98
5.4.8.2	Resource definition.....	98
5.4.8.3	Resource methods.....	98
5.4.8.3.1	POST.....	98
5.4.8.3.2	GET.....	99
5.4.8.3.3	PUT.....	99

5.4.8.3.4	PATCH	99
5.4.8.3.5	DELETE	99
5.4.9	Resource: Heal VNF task	99
5.4.9.1	Description	99
5.4.9.2	Resource definition	100
5.4.9.3	Resource methods	100
5.4.9.3.1	POST	100
5.4.9.3.2	GET	101
5.4.9.3.3	PUT	101
5.4.9.3.4	PATCH	101
5.4.9.3.5	DELETE	102
5.4.10	Resource: Operate VNF task	102
5.4.10.1	Description	102
5.4.10.2	Resource definition	102
5.4.10.3	Resource methods	102
5.4.10.3.1	POST	102
5.4.10.3.2	GET	103
5.4.10.3.3	PUT	104
5.4.10.3.4	PATCH	104
5.4.10.3.5	DELETE	104
5.4.11	Resource: Change external VNF connectivity task	104
5.4.11.1	Description	104
5.4.11.2	Resource definition	104
5.4.11.3	Resource methods	104
5.4.11.3.1	POST	104
5.4.11.3.2	GET	105
5.4.11.3.3	PUT	105
5.4.11.3.4	PATCH	105
5.4.11.3.5	DELETE	106
5.4.12	Resource: VNF LCM operation occurrences	106
5.4.12.1	Description	106
5.4.12.2	Resource definition	106
5.4.12.3	Resource methods	106
5.4.12.3.1	POST	106
5.4.12.3.2	GET	106
5.4.12.3.3	PUT	108
5.4.12.3.4	PATCH	108
5.4.12.3.5	DELETE	108
5.4.13	Resource: Individual VNF LCM operation occurrence	108
5.4.13.1	Description	108
5.4.13.2	Resource definition	109
5.4.13.3	Resource methods	109
5.4.13.3.1	POST	109
5.4.13.3.2	GET	109
5.4.13.3.3	PUT	109
5.4.13.3.4	PATCH	110
5.4.13.3.5	DELETE	110
5.4.14	Resource: Retry operation task	110
5.4.14.1	Description	110
5.4.14.2	Resource definition	110
5.4.14.3	Resource methods	110
5.4.14.3.1	POST	110
5.4.14.3.2	GET	111
5.4.14.3.3	PUT	111
5.4.14.3.4	PATCH	111
5.4.14.3.5	DELETE	112
5.4.15	Resource: Rollback operation task	112
5.4.15.1	Description	112
5.4.15.2	Resource definition	112
5.4.15.3	Resource methods	112
5.4.15.3.1	POST	112
5.4.15.3.2	GET	113

5.4.15.3.3	PUT	113
5.4.15.3.4	PATCH	113
5.4.15.3.5	DELETE	113
5.4.16	Resource: Fail operation task	114
5.4.16.1	Description	114
5.4.16.2	Resource definition	114
5.4.16.3	Resource methods	114
5.4.16.3.1	POST	114
5.4.16.3.2	GET	115
5.4.16.3.3	PUT	115
5.4.16.3.4	PATCH	115
5.4.16.3.5	DELETE	116
5.4.17	Resource: Cancel operation task	116
5.4.17.1	Description	116
5.4.17.2	Resource definition	116
5.4.17.3	Resource methods	116
5.4.17.3.1	POST	116
5.4.17.3.2	GET	117
5.4.17.3.3	PUT	117
5.4.17.3.4	PATCH	117
5.4.17.3.5	DELETE	118
5.4.18	Resource: Subscriptions	118
5.4.18.1	Description	118
5.4.18.2	Resource definition	118
5.4.18.3	Resource methods	118
5.4.18.3.1	POST	118
5.4.18.3.2	GET	119
5.4.18.3.3	PUT	120
5.4.18.3.4	PATCH	120
5.4.18.3.5	DELETE	120
5.4.19	Resource: Individual subscription	120
5.4.19.1	Description	120
5.4.19.2	Resource definition	120
5.4.19.3	Resource methods	121
5.4.19.3.1	POST	121
5.4.19.3.2	GET	121
5.4.19.3.3	PUT	121
5.4.19.3.4	PATCH	121
5.4.19.3.5	DELETE	121
5.4.20	Resource: Notification endpoint	122
5.4.20.1	Description	122
5.4.20.2	Resource definition	122
5.4.20.3	Resource methods	122
5.4.20.3.1	POST	122
5.4.20.3.2	GET	123
5.4.20.3.3	PUT	123
5.4.20.3.4	PATCH	123
5.4.20.3.5	DELETE	124
5.5	Data model	124
5.5.1	Introduction	124
5.5.2	Resource and notification data types	124
5.5.2.1	Introduction	124
5.5.2.2	Type: VnfInstance	124
5.5.2.3	Type: CreateVnfRequest	127
5.5.2.4	Type: InstantiateVnfRequest	128
5.5.2.5	Type: ScaleVnfRequest	128
5.5.2.6	Type: ScaleVnfToLevelRequest	129
5.5.2.7	Type: ChangeVnfFlavourRequest	129
5.5.2.8	Type: TerminateVnfRequest	130
5.5.2.9	Type: HealVnfRequest	130
5.5.2.10	Type: OperateVnfRequest	131
5.5.2.11	Type: ChangeExtVnfConnectivityRequest	131

5.5.2.12	Type: VnfInfoModificationRequest	132
5.5.2.12a	Type: VnfInfoModifications.....	133
5.5.2.13	Type: VnfLcmOpOcc	133
5.5.2.14	Type: CancelMode	135
5.5.2.15	Type: LcmSubscriptionRequest.....	135
5.5.2.16	Type: LcmSubscription.....	136
5.5.2.17	Type: VnfLcmOperationOccurrenceNotification	136
5.5.2.18	Type: VnfIdentifierCreationNotification.....	137
5.5.2.19	Type: VnfIdentifierDeletionNotification.....	138
5.5.3	Referenced structured data types.....	138
5.5.3.1	Introduction	138
5.5.3.2	Type: ExtVirtualLinkInfo.....	138
5.5.3.3	Type: ExtManagedVirtualLinkInfo	139
5.5.3.4	Type: ScaleInfo	139
5.5.3.5	Type: VnfcResourceInfo	139
5.5.3.6	Type: VnfVirtualLinkResourceInfo	140
5.5.3.7	Type: VirtualStorageResourceInfo	140
5.5.3.8	Type: VnfLinkPortInfo	140
5.5.3.9	Type: ExtLinkPortInfo	141
5.5.3.9a	Type: ExtLinkPortData.....	141
5.5.3.9b	Type: CpProtocolInfo	142
5.5.3.10	Type: IpOverEthernetAddressInfo	142
5.5.3.11	Type: MonitoringParameter	143
5.5.3.12	Type: LifecycleChangeNotificationsFilter	143
5.5.3.13	Type: AffectedVnfc.....	144
5.5.3.14	Type: AffectedVirtualLink	145
5.5.3.15	Type: AffectedVirtualStorage	145
5.5.3.16	Type: LcmLinks.....	146
5.5.3.17	Type: VnfExtCpInfo.....	146
5.5.4	Referenced simple data types and enumerations	147
5.5.4.1	Introduction	147
5.5.4.2	Simple data types.....	147
5.5.4.3	Enumeration: VnfOperationalStateType	147
5.5.4.4	Enumeration: StopType.....	147
5.5.4.5	Enumeration: LcmOperationStateType	148
5.5.4.6	Enumeration: CancelModeType.....	148
5.5.4.7	Enumeration: LcmOperationType.....	148
5.6	Handling of errors during VNF lifecycle management operations	149
5.6.1	Basic concepts (informative).....	149
5.6.1.1	Motivation	149
5.6.1.2	Failure resolution strategies: Retry and Rollback.....	149
5.6.1.3	Error handling at VNFM and NFVO	150
5.6.2	States and state transitions of a VNF lifecycle management operation occurrence	151
5.6.2.1	General	151
5.6.2.2	States of a VNF lifecycle management operation occurrence	151
5.6.2.3	Error handling operations that change the state of a VNF lifecycle management operation occurrence.....	154
5.6.3	Detailed flows	155
5.6.3.1	Immediate failure.....	155
5.6.3.2	Failure in "STARTING" state	155
5.6.3.3	Failure during actual LCM operation execution	156
5.6.3.4	LCM operation cancellation	158
6	VNF Performance Management interface.....	158
6.1	Description.....	158
6.1a	API version	158
6.2	Resource structure and methods	158
6.3	Sequence diagrams (informative)	160
6.3.1	Flow of creating a PM job.....	160
6.3.2	Flow of querying/reading PM jobs.....	160
6.3.3	Flow of deleting a PM job.....	161
6.3.4	Flow of obtaining performance reports	162

6.3.5	Flow of creating a threshold.....	163
6.3.6	Flow of querying/reading thresholds.....	163
6.3.7	Flow of deleting thresholds.....	164
6.3.8	Flow of managing subscriptions.....	165
6.3.9	Flow of sending notifications.....	166
6.4	Resources.....	167
6.4.1	Introduction.....	167
6.4.1a	Resource: API versions.....	167
6.4.2	Resource: PM jobs.....	167
6.4.2.1	Description.....	167
6.4.2.2	Resource definition.....	167
6.4.2.3	Resource methods.....	167
6.4.2.3.1	POST.....	167
6.4.2.3.2	GET.....	168
6.4.2.3.3	PUT.....	169
6.4.2.3.4	PATCH.....	169
6.4.2.3.5	DELETE.....	169
6.4.3	Resource: Individual PM job.....	169
6.4.3.1	Description.....	169
6.4.3.2	Resource definition.....	170
6.4.3.3	Resource methods.....	170
6.4.3.3.1	POST.....	170
6.4.3.3.2	GET.....	170
6.4.3.3.3	PUT.....	170
6.4.3.3.4	PATCH.....	171
6.4.3.3.5	DELETE.....	171
6.4.4	Resource: Individual performance report.....	171
6.4.4.1	Description.....	171
6.4.4.2	Resource definition.....	171
6.4.4.3	Resource methods.....	172
6.4.4.3.1	POST.....	172
6.4.4.3.2	GET.....	172
6.4.4.3.3	PUT.....	172
6.4.4.3.4	PATCH.....	172
6.4.4.3.5	DELETE.....	172
6.4.5	Resource: Thresholds.....	172
6.4.5.1	Description.....	172
6.4.5.2	Resource definition.....	173
6.4.5.3	Resource methods.....	173
6.4.5.3.1	POST.....	173
6.4.5.3.2	GET.....	173
6.4.5.3.3	PUT.....	174
6.4.5.3.4	PATCH.....	174
6.4.5.3.5	DELETE.....	175
6.4.6	Resource: Individual threshold.....	175
6.4.6.1	Description.....	175
6.4.6.2	Resource definition.....	175
6.4.6.3	Resource methods.....	175
6.4.6.3.1	POST.....	175
6.4.6.3.2	GET.....	175
6.4.6.3.3	PUT.....	176
6.4.6.3.4	PATCH.....	176
6.4.6.3.5	DELETE.....	176
6.4.7	Resource: Subscriptions.....	176
6.4.7.1	Description.....	176
6.4.7.2	Resource definition.....	177
6.4.7.3	Resource methods.....	177
6.4.7.3.1	POST.....	177
6.4.7.3.2	GET.....	178
6.4.7.3.3	PUT.....	179
6.4.7.3.4	PATCH.....	179
6.4.7.3.5	DELETE.....	179

6.4.8	Resource: Individual subscription	179
6.4.8.1	Description	179
6.4.8.2	Resource definition.....	179
6.4.8.3	Resource methods.....	179
6.4.8.3.1	POST.....	179
6.4.8.3.2	GET.....	179
6.4.8.3.3	PUT.....	180
6.4.8.3.4	PATCH	180
6.4.8.3.5	DELETE	180
6.4.9	Resource: Notification endpoint.....	181
6.4.9.1	Description	181
6.4.9.2	Resource definition.....	181
6.4.9.3	Resource methods.....	181
6.4.9.3.1	POST.....	181
6.4.9.3.2	GET.....	181
6.4.9.3.3	PUT.....	182
6.4.9.3.4	PATCH	182
6.4.9.3.5	DELETE	182
6.5	Data Model	182
6.5.1	Introduction	182
6.5.2	Resource and notification data types.....	182
6.5.2.1	Introduction	182
6.5.2.2	Type: PmSubscriptionRequest.....	182
6.5.2.3	Type: PmSubscription	183
6.5.2.4	Type: ThresholdCrossedNotification.....	183
6.5.2.5	Type: PerformanceInformationAvailableNotification	184
6.5.2.6	Type: CreatePmJobRequest.....	185
6.5.2.7	Type: PmJob.....	185
6.5.2.8	Type: CreateThresholdRequest	185
6.5.2.9	Type: Threshold.....	185
6.5.2.10	Type: PerformanceReport.....	186
6.5.3	Referenced structured data types	186
6.5.3.1	Introduction	186
6.5.3.2	Type: PmNotificationsFilter	186
6.5.3.3	Type: PmJobCriteria.....	187
6.5.3.4	Type: ThresholdCriteria	188
6.5.4	Referenced simple data types and enumerations	188
6.5.4.1	Introduction	188
6.5.4.2	Simple data types.....	188
6.5.4.3	Enumeration: CrossingDirectionType.....	188
7	VNF Fault Management interface.....	189
7.1	Description.....	189
7.1a	API version	189
7.2	Resource structure and methods	189
7.3	Sequence diagrams (informative)	190
7.3.1	Flow of the Get Alarm List operation	190
7.3.2	Flow of acknowledging alarm.....	191
7.3.3	Flow of managing subscriptions.....	191
7.3.4	Flow of sending notifications	193
7.4	Resources.....	194
7.4.1	Introduction.....	194
7.4.2	Resource: Alarms	194
7.4.2.1	Description	194
7.4.2.2	Resource definition.....	194
7.4.2.3	Resource methods.....	194
7.4.2.3.1	POST.....	194
7.4.2.3.2	GET.....	194
7.4.2.3.3	PUT.....	195
7.4.2.3.4	PATCH	195
7.4.2.3.5	DELETE	195
7.4.3	Resource: Individual alarm	195

7.4.3.1	Description	195
7.4.3.2	Resource definition.....	195
7.4.3.3	Resource methods.....	196
7.4.3.3.1	POST.....	196
7.4.3.3.2	GET.....	196
7.4.3.3.3	PUT.....	196
7.4.3.3.4	PATCH	196
7.4.3.3.5	DELETE	197
7.4.4	Resource: Subscriptions	197
7.4.4.1	Description	197
7.4.4.2	Resource definition.....	198
7.4.4.3	Resource methods.....	198
7.4.4.3.1	POST.....	198
7.4.4.3.2	GET.....	199
7.4.4.3.3	PUT.....	200
7.4.4.3.4	PATCH	200
7.4.4.3.5	DELETE	200
7.4.5	Resource: Individual subscription	200
7.4.5.1	Description	200
7.4.5.2	Resource definition.....	200
7.4.5.3	Resource methods.....	200
7.4.5.3.1	POST.....	200
7.4.5.3.2	GET.....	200
7.4.5.3.3	PUT.....	201
7.4.5.3.4	PATCH	201
7.4.5.3.5	DELETE	201
7.4.6	Resource: Notification endpoint.....	201
7.4.6.1	Description	201
7.4.6.2	Resource definition.....	202
7.4.6.3	Resource methods.....	202
7.4.6.3.1	POST.....	202
7.4.6.3.2	GET.....	202
7.4.6.3.3	PUT.....	203
7.4.6.3.4	PATCH	203
7.4.6.3.5	DELETE	203
7.5	Data Model	203
7.5.1	Introduction.....	203
7.5.2	Resource and notification data types.....	203
7.5.2.1	Introduction	203
7.5.2.2	Type: FmSubscriptionRequest.....	203
7.5.2.3	Type: FmSubscription	204
7.5.2.4	Type: Alarm.....	204
7.5.2.5	Type: AlarmNotification	205
7.5.2.6	Type: AlarmClearedNotification.....	206
7.5.2.7	Type: AlarmListRebuiltNotification	206
7.5.2.8	Type: AlarmModifications	206
7.5.3	Referenced structured data types.....	207
7.5.3.1	Introduction	207
7.5.3.2	Type: FmNotificationsFilter	207
7.5.3.3	Type: FaultyResourceInfo	207
7.5.4	Referenced simple data types and enumerations	207
7.5.4.1	Introduction	207
7.5.4.2	Simple data types.....	208
7.5.4.3	Enumeration: PerceivedSeverityType	208
7.5.4.4	Enumeration: EventType.....	208
7.5.4.5	Enumeration: FaultyResourceType	208
8	VNF Indicator interface	209
8.1	Description.....	209
8.1a	API version	209
8.2	Resource structure and methods	209
8.3	Sequence diagrams (informative)	210

8.3.1	Flow of querying VNF indicators	210
8.3.2	Flow of reading a VNF indicator	211
8.3.3	Flow of managing subscriptions.....	212
8.3.4	Flow of sending notifications	213
8.4	Resources.....	214
8.4.1	Introduction	214
8.4.1a	Resource: API versions	214
8.4.2	Resource: VNF indicators	214
8.4.2.1	Description	214
8.4.2.2	Resource definition.....	214
8.4.2.3	Resource methods.....	214
8.4.2.3.1	POST.....	214
8.4.2.3.2	GET.....	214
8.4.2.3.3	PUT.....	215
8.4.2.3.4	PATCH	215
8.4.2.3.5	DELETE	215
8.4.3	Resource: VNF indicators related to a VNF instance.....	215
8.4.3.1	Description	215
8.4.3.2	Resource definition.....	215
8.4.3.3	Resource methods.....	216
8.4.3.3.1	POST.....	216
8.4.3.3.2	GET.....	216
8.4.3.3.3	PUT.....	217
8.4.3.3.4	PATCH	217
8.4.3.3.5	DELETE	217
8.4.4	Resource: Individual VNF indicator	217
8.4.4.1	Description	217
8.4.4.2	Resource definition.....	217
8.4.4.3	Resource methods.....	218
8.4.4.3.1	POST.....	218
8.4.4.3.2	GET.....	218
8.4.4.3.3	PUT.....	218
8.4.4.3.4	PATCH	218
8.4.4.3.5	DELETE	219
8.4.5	Resource: Subscriptions	219
8.4.5.1	Description	219
8.4.5.2	Resource definition.....	219
8.4.5.3	Resource methods.....	219
8.4.5.3.1	POST.....	219
8.4.5.3.2	GET.....	220
8.4.5.3.3	PUT.....	221
8.4.5.3.4	PATCH	221
8.4.5.3.5	DELETE	221
8.4.6	Resource: Individual subscription	221
8.4.6.1	Description	221
8.4.6.2	Resource definition.....	221
8.4.6.3	Resource methods.....	222
8.4.6.3.1	POST.....	222
8.4.6.3.2	GET.....	222
8.4.6.3.3	PUT.....	222
8.4.6.3.4	PATCH	222
8.4.6.3.5	DELETE	223
8.4.7	Resource: Notification endpoint.....	223
8.4.7.1	Description	223
8.4.7.2	Resource definition.....	223
8.4.7.3	Resource methods.....	223
8.4.7.3.1	POST.....	223
8.4.7.3.2	GET.....	224
8.4.7.3.3	PUT.....	224
8.4.7.3.4	PATCH	225
8.4.7.3.5	DELETE	225
8.5	Data model.....	225

8.5.1	Introduction	225
8.5.2	Resource and notification data types	225
8.5.2.1	Introduction	225
8.5.2.2	Type: VnfIndicator	225
8.5.2.3	Type: VnfIndicatorSubscriptionRequest	225
8.5.2.4	Type: VnfIndicatorSubscription	226
8.5.2.5	Type: VnfIndicatorValueChangeNotification	226
8.5.3	Referenced structured data types	227
8.5.3.1	Introduction	227
8.5.3.2	Type: VnfIndicatorNotificationsFilter	227
8.5.4	Referenced simple data types and enumerations	227
9	VNF Lifecycle Operation Granting interface	227
9.1	Description	227
9.1a	API version	228
9.2	Resource structure and methods	228
9.3	Sequence diagrams (informative)	229
9.3.1	Flow of grant request with synchronous response	229
9.3.2	Flow of grant request with asynchronous response	230
9.4	Resources	230
9.4.1	Introduction	230
9.4.1a	Resource: API versions	231
9.4.2	Resource: Grants	231
9.4.2.1	Description	231
9.4.2.2	Resource definition	231
9.4.2.3	Resource methods	231
9.4.2.3.1	POST	231
9.4.2.3.2	GET	232
9.4.2.3.3	PUT	232
9.4.2.3.4	PATCH	232
9.4.2.3.5	DELETE	232
9.4.3	Resource: Individual grant	232
9.4.3.1	Description	232
9.4.3.2	Resource definition	233
9.4.3.3	Resource methods	233
9.4.3.3.1	POST	233
9.4.3.3.2	GET	233
9.4.3.3.3	PUT	234
9.4.3.3.4	PATCH	234
9.4.3.3.5	DELETE	234
9.5	Data model	234
9.5.1	Introduction	234
9.5.2	Resource and notification data types	234
9.5.2.1	Introduction	234
9.5.2.2	Type: GrantRequest	235
9.5.2.3	Type: Grant	236
9.5.3	Referenced structured data types	238
9.5.3.1	Introduction	238
9.5.3.2	Type: ResourceDefinition	238
9.5.3.3	Type: GrantInfo	239
9.5.3.4	Type: ZoneInfo	240
9.5.3.5	Type: ZoneGroupInfo	241
9.5.3.6	Type: PlacementConstraint	241
9.5.3.7	Type: VimConstraint	241
9.5.3.8	Type: ConstraintResourceRef	242
9.5.3.9	Type: VimComputeResourceFlavour	243
9.5.3.10	Type: VimSoftwareImage	243
9.5.4	Referenced simple data types and enumerations	244
9.5.4.1	Introduction	244
9.5.4.2	Simple data types	244
9.5.4.3	Enumeration: GrantedLcmOperationType	244