



Network Functions Virtualisation (NFV) Release 2; Protocols and Data Models; RESTful protocols specification for the Ve-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.

ReferenceRGS/NFV-SOL002ed251

KeywordsAPI, 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/CommiteeSupportStaff.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	15
Foreword.....	15
Modal verbs terminology.....	15
1 Scope	16
2 References	16
2.1 Normative references	16
2.2 Informative references.....	17
3 Definitions and abbreviations.....	18
3.1 Definitions.....	18
3.2 Abbreviations	18
4 General aspects.....	19
4.1 Overview	19
4.2 URI structure and supported content formats.....	19
4.3 Common procedures.....	20
4.3.1 Introduction.....	20
4.3.2 Attribute-based filtering.....	20
4.3.2.1 Overview and example (informative).....	20
4.3.2.2 Specification.....	21
4.3.3 Attribute selectors.....	23
4.3.3.1 Overview and example (informative).....	23
4.3.3.2 Specification.....	23
4.3.3.2.1 GET request.....	23
4.3.3.2.2 GET response	24
4.3.4 Usage of HTTP header fields.....	24
4.3.4.1 Introduction.....	24
4.3.4.2 Request header fields.....	24
4.3.4.3 Response header fields.....	25
4.3.5 Error reporting	26
4.3.5.1 Introduction.....	26
4.3.5.2 General mechanism.....	26
4.3.5.3 Type: ProblemDetails.....	26
4.3.5.4 Common error situations.....	26
4.3.5.5 Overview of HTTP error status codes.....	28
4.4 Common data types.....	29
4.4.1 Structured data types.....	29
4.4.1.1 Introduction.....	29
4.4.1.2 Type: Object.....	29
4.4.1.3 Type: Link.....	29
4.4.1.3a Type: NotificationLink.....	30
4.4.1.4 Type: KeyValuePairs	30
4.4.1.5 Type: VnfInstanceSubscriptionFilter	30
4.4.1.6 Type: ApiVersionInformation.....	31
4.4.2 Simple data types and enumerations.....	32
4.4.2.1 Introduction.....	32
4.4.2.2 Simple data types	32
4.4.2.3 Enumerations	32
4.5 Authorization of API requests and notifications	32
4.5.1 Introduction.....	32
4.5.2 Flows (informative)	33
4.5.2.0 General	33
4.5.2.1 Authorization of API requests using OAuth 2.0 access tokens.....	33
4.5.2.1a Authorization of API requests using TLS certificates.....	35
4.5.2.2 Authorization of notifications using the HTTP Basic authentication scheme.....	36
4.5.2.3 Authorization of notifications using OAuth 2.0 access tokens.....	37

4.5.2.4	Authorization of notifications using TLS certificates	39
4.5.3	Specification	41
4.5.3.1	Introduction	41
4.5.3.2	General mechanism	41
4.5.3.3	Authorizing API requests	41
4.5.3.4	Authorizing the sending of notifications	42
4.5.3.5	Client roles	43
4.5.3.6	Negotiation of authorization method	44
4.5.3.6.1	Authorization of API requests	44
4.5.3.6.2	Authorization of notification requests	45
4.6	Version management	46
4.6.1	Version identifiers and parameters	46
4.6.1.1	Version identifiers	46
4.6.1.2	Version parameters	46
4.6.2	Rules for incrementing version identifier fields	47
4.6.2.1	General	47
4.6.2.2	Examples of backward and non-backward compatible changes	47
4.6.3	Version information retrieval	48
4.6.3.1	General	48
4.6.3.2	Resource structure and methods	49
4.6.3.3	Resource: API versions	49
4.6.3.3.1	Description	49
4.6.3.3.2	Resource definition	49
4.6.3.3.3	Resource methods	50
4.6.4	Version signalling	50
4.7	Handling of large query results	51
4.7.1	Description	51
4.7.2	Specification	51
4.7.2.1	Alternatives	51
4.7.2.2	Error response	52
4.7.2.3	Paged response	52
5	VNF Lifecycle Management interface	52
5.1	Description	52
5.1a	API version	53
5.2	Resource structure and methods	53
5.3	Sequence diagrams (informative)	55
5.3.1	Flow of the creation of a VNF instance resource	55
5.3.2	Flow of the deletion of a VNF instance resource	56
5.3.3	Flow of VNF lifecycle management operations triggered by task resources	57
5.3.4	Flow of automatic invocation of VNF scaling and VNF healing	60
5.3.5	Flow of the Query VNF operation	62
5.3.6	Flow of the Modify VNF Information operation	63
5.3.7	Flow of the Get Operation Status operation	64
5.3.8	Flow of managing subscriptions	65
5.3.9	Flow of sending notifications	67
5.3.10	Flow of retrying a VNF lifecycle management operation	68
5.3.11	Flow of rolling back a VNF lifecycle management operation	69
5.3.12	Flow of failing a VNF lifecycle management operation	70
5.3.13	Flow of cancelling a VNF lifecycle management operation	71
5.4	Resources	73
5.4.1	Introduction	73
5.4.1a	Resource: API versions	73
5.4.2	Resource: VNF instances	74
5.4.2.1	Description	74
5.4.2.2	Resource definition	74
5.4.2.3	Resource methods	74
5.4.2.3.1	POST	74
5.4.2.3.2	GET	74
5.4.2.3.3	PUT	75
5.4.2.3.4	PATCH	75
5.4.2.3.5	DELETE	75

5.4.3	Resource: Individual VNF instance	76
5.4.3.1	Description	76
5.4.3.2	Resource definition	76
5.4.3.3	Resource methods	76
5.4.3.3.1	POST	76
5.4.3.3.2	GET	76
5.4.3.3.3	PUT	77
5.4.3.3.4	PATCH	77
5.4.3.3.5	DELETE	77
5.4.4	Resource: Instantiate VNF task	78
5.4.4.1	Description	78
5.4.4.2	Resource definition	78
5.4.4.3	Resource methods	78
5.4.4.3.1	POST	78
5.4.4.3.2	GET	79
5.4.4.3.3	PUT	79
5.4.4.3.4	PATCH	79
5.4.4.3.5	DELETE	79
5.4.5	Resource: Scale VNF task	79
5.4.5.1	Description	79
5.4.5.2	Resource definition	80
5.4.5.3	Resource methods	80
5.4.5.3.1	POST	80
5.4.5.3.2	GET	81
5.4.5.3.3	PUT	81
5.4.5.3.4	PATCH	81
5.4.5.3.5	DELETE	81
5.4.6	Resource: Scale VNF to Level task	82
5.4.6.1	Description	82
5.4.6.2	Resource definition	82
5.4.6.3	Resource methods	82
5.4.6.3.1	POST	82
5.4.6.3.2	GET	83
5.4.6.3.3	PUT	83
5.4.6.3.4	PATCH	83
5.4.6.3.5	DELETE	83
5.4.7	Resource: Change VNF Flavour task	84
5.4.7.1	Description	84
5.4.7.2	Resource definition	84
5.4.7.3	Resource methods	84
5.4.7.3.1	POST	84
5.4.7.3.2	GET	85
5.4.7.3.3	PUT	85
5.4.7.3.4	PATCH	85
5.4.7.3.5	DELETE	85
5.4.8	Resource: Terminate VNF task	86
5.4.8.1	Description	86
5.4.8.2	Resource definition	86
5.4.8.3	Resource methods	86
5.4.8.3.1	POST	86
5.4.8.3.2	GET	87
5.4.8.3.3	PUT	87
5.4.8.3.4	PATCH	87
5.4.8.3.5	DELETE	87
5.4.9	Resource: Heal VNF task	87
5.4.9.1	Description	87
5.4.9.2	Resource definition	87
5.4.9.3	Resource methods	88
5.4.9.3.1	POST	88
5.4.9.3.2	GET	89
5.4.9.3.3	PUT	89
5.4.9.3.4	PATCH	89

5.4.9.3.5	DELETE	89
5.4.10	Resource: Operate VNF task	90
5.4.10.1	Description	90
5.4.10.2	Resource definition	90
5.4.10.3	Resource methods	90
5.4.10.3.1	POST	90
5.4.10.3.2	GET	91
5.4.10.3.3	PUT	91
5.4.10.3.4	PATCH	91
5.4.10.3.5	DELETE	91
5.4.11	Resource: Change external VNF connectivity task	92
5.4.11.1	Description	92
5.4.11.2	Resource definition	92
5.4.11.3	Resource methods	92
5.4.11.3.1	POST	92
5.4.11.3.2	GET	93
5.4.11.3.3	PUT	93
5.4.11.3.4	PATCH	93
5.4.11.3.5	DELETE	93
5.4.12	Resource: VNF LCM operation occurrences	93
5.4.12.1	Description	93
5.4.12.2	Resource definition	93
5.4.12.3	Resource methods	94
5.4.12.3.1	POST	94
5.4.12.3.2	GET	94
5.4.12.3.3	PUT	95
5.4.12.3.4	PATCH	95
5.4.12.3.5	DELETE	95
5.4.13	Resource: Individual VNF LCM operation occurrence	95
5.4.13.1	Description	95
5.4.13.2	Resource definition	96
5.4.13.3	Resource methods	96
5.4.13.3.1	POST	96
5.4.13.3.2	GET	96
5.4.13.3.3	PUT	96
5.4.13.3.4	PATCH	97
5.4.13.3.5	DELETE	97
5.4.14	Resource: Retry operation task	97
5.4.14.1	Description	97
5.4.14.2	Resource definition	97
5.4.14.3	Resource methods	97
5.4.14.3.1	POST	97
5.4.14.3.2	GET	98
5.4.14.3.3	PUT	98
5.4.14.3.4	PATCH	98
5.4.14.3.5	DELETE	98
5.4.15	Resource: Rollback operation task	99
5.4.15.1	Description	99
5.4.15.2	Resource definition	99
5.4.15.3	Resource methods	99
5.4.15.3.1	POST	99
5.4.15.3.2	GET	100
5.4.15.3.3	PUT	100
5.4.15.3.4	PATCH	100
5.4.15.3.5	DELETE	100
5.4.16	Resource: Fail operation task	101
5.4.16.1	Description	101
5.4.16.2	Resource definition	101
5.4.16.3	Resource methods	101
5.4.16.3.1	POST	101
5.4.16.3.2	GET	102
5.4.16.3.3	PUT	102

5.4.16.3.4	PATCH	102
5.4.16.3.5	DELETE	102
5.4.17	Resource: Cancel operation task	103
5.4.17.1	Description	103
5.4.17.2	Resource definition	103
5.4.17.3	Resource methods	103
5.4.17.3.1	POST	103
5.4.17.3.2	GET	104
5.4.17.3.3	PUT	104
5.4.17.3.4	PATCH	104
5.4.17.3.5	DELETE	104
5.4.18	Resource: Subscriptions	105
5.4.18.1	Description	105
5.4.18.2	Resource definition	105
5.4.18.3	Resource methods	105
5.4.18.3.1	POST	105
5.4.18.3.2	GET	106
5.4.18.3.3	PUT	107
5.4.18.3.4	PATCH	107
5.4.18.3.5	DELETE	107
5.4.19	Resource: Individual subscription	107
5.4.19.1	Description	107
5.4.19.2	Resource definition	107
5.4.19.3	Resource methods	108
5.4.19.3.1	POST	108
5.4.19.3.2	GET	108
5.4.19.3.3	PUT	108
5.4.19.3.4	PATCH	108
5.4.19.3.5	DELETE	108
5.4.20	Resource: Notification endpoint	109
5.4.20.1	Description	109
5.4.20.2	Resource definition	109
5.4.20.3	Resource methods	109
5.4.20.3.1	POST	109
5.4.20.3.2	GET	110
5.4.20.3.3	PUT	111
5.4.20.3.4	PATCH	111
5.4.20.3.5	DELETE	111
5.5	Data model	111
5.5.1	Introduction	111
5.5.2	Resource and notification data types	111
5.5.2.1	Introduction	111
5.5.2.2	Type: VnfInstance	111
5.5.2.3	Type: CreateVnfRequest	113
5.5.2.4	Type: InstantiateVnfRequest	114
5.5.2.5	Type: ScaleVnfRequest	114
5.5.2.6	Type: ScaleVnfToLevelRequest	115
5.5.2.7	Type: ChangeVnfFlavourRequest	115
5.5.2.8	Type: TerminateVnfRequest	115
5.5.2.9	Type: HealVnfRequest	116
5.5.2.10	Type: OperateVnfRequest	116
5.5.2.11	Type: ChangeExtVnfConnectivityRequest	116
5.5.2.12	Type: VnfInfoModificationRequest	117
5.5.2.12a	Type: VnfInfoModifications	118
5.5.2.13	Type: VnfLcmOpOcc	119
5.5.2.14	Type: CancelMode	121
5.5.2.15	Type: LccnSubscriptionRequest	121
5.5.2.16	Type: LccnSubscription	121
5.5.2.17	Type: VnfLcmOperationOccurrenceNotification	121
5.5.2.18	Type: VnfIdentifierCreationNotification	124
5.5.2.19	Type: VnfIdentifierDeletionNotification	124
5.5.3	Referenced structured data types	124

5.5.3.1	Introduction	124
5.5.3.2	Type: ExtVirtualLinkData	125
5.5.3.3	Type: ExtVirtualLinkInfo	125
5.5.3.4	Type: ExtManagedVirtualLinkData	125
5.5.3.5	Type: ExtManagedVirtualLinkInfo	126
5.5.3.6	Type: VnfExtCpData	126
5.5.3.6a	Type: VnfExtCpConfig	126
5.5.3.6b	Type: CpProtocolData	127
5.5.3.6c	Type: IpOverEthernetAddressData	127
5.5.3.7	Type: ScaleInfo	128
5.5.3.8	Type: VnfcResourceInfo	129
5.5.3.9	Type: VnfVirtualLinkResourceInfo	129
5.5.3.10	Type: VirtualStorageResourceInfo	129
5.5.3.11	Type: VnfLinkPortInfo	130
5.5.3.12	Type: ExtLinkPortInfo	130
5.5.3.12a	Type: ExtLinkPortData	131
5.5.3.13	Type: ResourceHandle	131
5.5.3.14	VOID	131
5.5.3.15	VOID	131
5.5.3.15a	Type: CpProtocolInfo	131
5.5.3.16	Type: IpOverEthernetAddressInfo	132
5.5.3.17	Type: MonitoringParameter	132
5.5.3.18	Type: LifecycleChangeNotificationsFilter	133
5.5.3.19	Type: AffectedVnfc	133
5.5.3.20	Type: AffectedVirtualLink	134
5.5.3.21	Type: AffectedVirtualStorage	135
5.5.3.22	Type: LcnLinks	135
5.5.3.23	Type: VnfcInfo	136
5.5.3.24	Type: VnfcInfoModifications	136
5.5.3.25	Type: VnfExtCpInfo	136
5.5.4	Referenced simple data types and enumerations	137
5.5.4.1	Introduction	137
5.5.4.2	Simple data types	137
5.5.4.3	Enumeration: VnfOperationalStateType	137
5.5.4.4	VOID	137
5.5.4.5	Enumeration: LcmOperationType	137
5.5.4.6	Enumeration: LcmOperationStateType	138
5.5.4.7	Enumeration: CancelModeType	138
5.6	Handling of errors during VNF lifecycle management operations	139
5.6.1	Basic concepts (informative)	139
5.6.1.1	Motivation	139
5.6.1.2	Failure resolution strategies: Retry and Rollback	139
5.6.1.3	Error handling at VNFM and EM	139
5.6.2	States and state transitions of a VNF lifecycle management operation occurrence	141
5.6.2.1	General	141
5.6.2.2	States of a VNF lifecycle management operation occurrence	141
5.6.2.3	Error handling operations that change the state of a VNF lifecycle management operation occurrence	144
5.6.3	Detailed flows	145
5.6.3.1	Immediate failure	145
5.6.3.2	Failure in "STARTING" state	145
5.6.3.3	Failure during actual LCM operation execution	146
5.6.3.4	LCM operation cancellation	148
6	VNF Performance Management interface	148
6.1	Description	148
6.1a	API version	148
6.2	Resource structure and methods	149
6.3	Sequence diagrams (informative)	150
6.3.1	Flow of creating a PM job	150
6.3.2	Flow of querying/reading PM jobs	151
6.3.3	Flow of deleting a PM job	151

6.3.4	Flow of obtaining performance reports.....	152
6.3.5	Flow of creating a threshold	153
6.3.6	Flow of querying/reading thresholds	153
6.3.7	Flow of deleting thresholds.....	154
6.3.8	Flow of managing subscriptions	155
6.3.9	Flow of sending notifications.....	156
6.4	Resources	157
6.4.1	Introduction.....	157
6.4.1a	Resource: API versions.....	157
6.4.2	Resource: PM jobs.....	157
6.4.2.1	Description	157
6.4.2.2	Resource definition	157
6.4.2.3	Resource methods	157
6.4.2.3.1	POST	157
6.4.2.3.2	GET	158
6.4.2.3.3	PUT	159
6.4.2.3.4	PATCH.....	159
6.4.2.3.5	DELETE.....	159
6.4.3	Resource: Individual PM job	159
6.4.3.1	Description	159
6.4.3.2	Resource definition	159
6.4.3.3	Resource methods	160
6.4.3.3.1	POST	160
6.4.3.3.2	GET	160
6.4.3.3.3	PUT	160
6.4.3.3.4	PATCH.....	160
6.4.3.3.5	DELETE.....	160
6.4.4	Resource: Individual performance report	161
6.4.4.1	Description	161
6.4.4.2	Resource definition	161
6.4.4.3	Resource methods	161
6.4.4.3.1	POST	161
6.4.4.3.2	GET	162
6.4.4.3.3	PUT	162
6.4.4.3.4	PATCH.....	162
6.4.4.3.5	DELETE.....	162
6.4.5	Resource: Thresholds.....	162
6.4.5.1	Description	162
6.4.5.2	Resource definition	162
6.4.5.3	Resource methods	163
6.4.5.3.1	POST	163
6.4.5.3.2	GET	163
6.4.5.3.3	PUT	164
6.4.5.3.4	PATCH.....	164
6.4.5.3.5	DELETE.....	164
6.4.6	Resource: Individual threshold	164
6.4.6.1	Description	164
6.4.6.2	Resource definition	164
6.4.6.3	Resource methods	165
6.4.6.3.1	POST	165
6.4.6.3.2	GET	165
6.4.6.3.3	PUT	165
6.4.6.3.4	PATCH.....	165
6.4.6.3.5	DELETE.....	165
6.4.7	Resource: Subscriptions.....	166
6.4.7.1	Description	166
6.4.7.2	Resource definition	166
6.4.7.3	Resource methods	166
6.4.7.3.1	POST	166
6.4.7.3.2	GET	167
6.4.7.3.3	PUT	168
6.4.7.3.4	PATCH.....	168

6.4.7.3.5	DELETE	168
6.4.8	Resource: Individual subscription.....	169
6.4.8.1	Description	169
6.4.8.2	Resource definition	169
6.4.8.3	Resource methods	169
6.4.8.3.1	POST	169
6.4.8.3.2	GET	169
6.4.8.3.3	PUT	170
6.4.8.3.4	PATCH.....	170
6.4.8.3.5	DELETE.....	170
6.4.9	Resource: Notification endpoint	170
6.4.9.1	Description	170
6.4.9.2	Resource definition	171
6.4.9.3	Resource methods	171
6.4.9.3.1	POST	171
6.4.9.3.2	GET	171
6.4.9.3.3	PUT	172
6.4.9.3.4	PATCH.....	172
6.4.9.3.5	DELETE.....	172
6.5	Data Model.....	172
6.5.1	Introduction.....	172
6.5.2	Resource and notification data types	172
6.5.2.1	Introduction	172
6.5.2.2	Type: PmSubscriptionRequest	172
6.5.2.3	Type: PmSubscription.....	173
6.5.2.4	Type: ThresholdCrossedNotification	173
6.5.2.5	Type: PerformanceInformationAvailableNotification	174
6.5.2.6	Type: CreatePmJobRequest	175
6.5.2.7	Type: PmJob	175
6.5.2.8	Type: CreateThresholdRequest.....	176
6.5.2.9	Type: Threshold	176
6.5.2.10	Type: PerformanceReport	177
6.5.3	Referenced structured data types	178
6.5.3.1	Introduction.....	178
6.5.3.2	Type: PmNotificationsFilter.....	178
6.5.3.3	Type: PmJobCriteria	178
6.5.3.4	Type: ThresholdCriteria	179
6.5.4	Referenced simple data types and enumerations	179
6.5.4.1	Introduction.....	179
6.5.4.2	Simple data types	179
6.5.4.3	Enumeration: CrossingDirectionType.....	179
7	VNF Fault Management interface.....	180
7.1	Description	180
7.1a	API version.....	180
7.2	Resource structure and methods.....	180
7.3	Sequence diagrams (informative).....	181
7.3.1	Flow of the Get Alarm List operation.....	181
7.3.2	Escalate perceived severity task flow	182
7.3.3	Flow of acknowledging alarm	183
7.3.4	Flow of managing subscriptions	183
7.3.5	Flow of sending notifications.....	185
7.4	Resources	186
7.4.1	Introduction.....	186
7.4.1a	Resource: API versions.....	186
7.4.2	Resource: Alarms.....	186
7.4.2.1	Description	186
7.4.2.2	Resource definition	186
7.4.2.3	Resource methods	186
7.4.2.3.1	POST	186
7.4.2.3.2	GET	186
7.4.2.3.3	PUT	187

7.4.2.3.4	PATCH.....	187
7.4.2.3.5	DELETE.....	187
7.4.3	Resource: Individual alarm.....	188
7.4.3.1	Description.....	188
7.4.3.2	Resource definition.....	188
7.4.3.3	Resource methods.....	188
7.4.3.3.1	POST.....	188
7.4.3.3.2	GET.....	188
7.4.3.3.3	PUT.....	189
7.4.3.3.4	PATCH.....	189
7.4.3.3.5	DELETE.....	190
7.4.4	Resource: Escalate Perceived Severity task.....	190
7.4.4.1	Description.....	190
7.4.4.2	Resource definition.....	190
7.4.4.3	Resource Methods.....	191
7.4.4.3.1	POST.....	191
7.4.4.3.2	GET.....	191
7.4.4.3.3	PUT.....	191
7.4.4.3.4	PATCH.....	191
7.4.4.3.5	DELETE.....	191
7.4.5	Resource: Subscriptions.....	192
7.4.5.1	Description.....	192
7.4.5.2	Resource definition.....	192
7.4.5.3	Resource methods.....	192
7.4.5.3.1	POST.....	192
7.4.5.3.2	GET.....	193
7.4.5.3.3	PUT.....	194
7.4.5.3.4	PATCH.....	194
7.4.5.3.5	DELETE.....	194
7.4.6	Resource: Individual subscription.....	194
7.4.6.1	Description.....	194
7.4.6.2	Resource definition.....	194
7.4.6.3	Resource methods.....	195
7.4.6.3.1	POST.....	195
7.4.6.3.2	GET.....	195
7.4.6.3.3	PUT.....	195
7.4.6.3.4	PATCH.....	195
7.4.6.3.5	DELETE.....	195
7.4.7	Resource: Notification endpoint.....	196
7.4.7.1	Description.....	196
7.4.7.2	Resource definition.....	196
7.4.7.3	Resource methods.....	196
7.4.7.3.1	POST.....	196
7.4.7.3.2	GET.....	197
7.4.7.3.3	PUT.....	197
7.4.7.3.4	PATCH.....	197
7.4.7.3.5	DELETE.....	198
7.5	Data Model.....	198
7.5.1	Introduction.....	198
7.5.2	Resource and notification data types.....	198
7.5.2.1	Introduction.....	198
7.5.2.2	Type: FmSubscriptionRequest.....	198
7.5.2.3	Type: FmSubscription.....	198
7.5.2.4	Type: Alarm.....	199
7.5.2.5	Type: AlarmNotification.....	199
7.5.2.6	Type: AlarmClearedNotification.....	200
7.5.2.7	Type: PerceivedSeverityRequest.....	200
7.5.2.8	Type: AlarmListRebuiltNotification.....	200
7.5.2.9	Type: AlarmModifications.....	201
7.5.3	Referenced structured data types.....	201
7.5.3.1	Introduction.....	201
7.5.3.2	Type: FmNotificationsFilter.....	201

7.5.3.3	Type: FaultyResourceInfo.....	202
7.5.4	Referenced simple data types and enumerations	202
7.5.4.1	Introduction.....	202
7.5.4.2	Simple data types	202
7.5.4.3	Enumeration: PerceivedSeverityType.....	202
7.5.4.4	Enumeration: EventType	203
7.5.4.5	Enumeration: FaultyResourceType.....	203
8	VNF Indicator interface.....	204
8.1	Description	204
8.1a	API version.....	204
8.2	Resource structure and methods.....	204
8.3	Sequence diagrams (informative).....	205
8.3.1	Flow of querying VNF indicators	205
8.3.2	Flow of reading a VNF indicator	206
8.3.3	Flow of managing subscriptions	207
8.3.4	Flow of sending notifications.....	209
8.4	Resources	210
8.4.1	Introduction.....	210
8.4.1a	Resource: API versions.....	210
8.4.2	Resource: VNF indicators.....	210
8.4.2.1	Description	210
8.4.2.2	Resource definition	210
8.4.2.3	Resource methods	210
8.4.2.3.1	POST	210
8.4.2.3.2	GET	210
8.4.2.3.3	PUT	211
8.4.2.3.4	PATCH.....	211
8.4.2.3.5	DELETE.....	211
8.4.3	Resource: VNF indicators related to a VNF instance	211
8.4.3.1	Description.....	211
8.4.3.2	Resource definition	211
8.4.3.3	Resource methods	212
8.4.3.3.1	POST	212
8.4.3.3.2	GET	212
8.4.3.3.3	PUT	213
8.4.3.3.4	PATCH.....	213
8.4.3.3.5	DELETE.....	213
8.4.4	Resource: Individual VNF indicator.....	213
8.4.4.1	Description	213
8.4.4.2	Resource definition	213
8.4.4.3	Resource methods	214
8.4.4.3.1	POST	214
8.4.4.3.2	GET	214
8.4.4.3.3	PUT	215
8.4.4.3.4	PATCH.....	215
8.4.4.3.5	DELETE.....	215
8.4.5	Resource: Subscriptions.....	215
8.4.5.1	Description	215
8.4.5.2	Resource definition	215
8.4.5.3	Resource methods	215
8.4.5.3.1	POST	215
8.4.5.3.2	GET	216
8.4.5.3.3	PUT	217
8.4.5.3.4	PATCH.....	217
8.4.5.3.5	DELETE.....	217
8.4.6	Resource: Individual subscription.....	218
8.4.6.1	Description	218
8.4.6.2	Resource definition	218
8.4.6.3	Resource methods	218
8.4.6.3.1	POST	218
8.4.6.3.2	GET	218

8.4.6.3.3	PUT	219
8.4.6.3.4	PATCH	219
8.4.6.3.5	DELETE	219
8.4.7	Resource: Notification endpoint	220
8.4.7.1	Description	220
8.4.7.2	Resource definition	220
8.4.7.3	Resource methods	220
8.4.7.3.1	POST	220
8.4.7.3.2	GET	220
8.4.7.3.3	PUT	221
8.4.7.3.4	PATCH	221
8.4.7.3.5	DELETE	221
8.5	Data model	221
8.5.1	Introduction	221
8.5.2	Resource and notification data types	221
8.5.2.1	Introduction	221
8.5.2.2	Type: VnfIndicator	221
8.5.2.3	Type: VnfIndicatorSubscriptionRequest	222
8.5.2.4	Type: VnfIndicatorSubscription	222
8.5.2.5	Type: VnfIndicatorValueChangeNotification	222
8.5.3	Referenced structured data types	223
8.5.3.1	Introduction	223
8.5.3.2	Type: VnfIndicatorNotificationsFilter	223
8.5.4	Referenced simple data types and enumerations	223
9	VNF Configuration interface	224
9.1	Description	224
9.1a	API version	224
9.2	Resource structure and methods	224
9.3	Sequence diagrams (informative)	224
9.3.1	Flow of setting the VNF configuration	224
9.4	Resources	225
9.4.1	Introduction	225
9.4.1a	Resource: API versions	225
9.4.2	Resource: Configuration	225
9.4.2.1	Description	225
9.4.2.2	Resource definition	225
9.4.2.3	Resource methods	226
9.4.2.3.1	POST	226
9.4.2.3.2	GET	226
9.4.2.3.3	PUT	226
9.4.2.3.4	PATCH	226
9.4.2.3.5	DELETE	227
9.5	Data model	227
9.5.1	Introduction	227
9.5.2	Resource and notification data types	227
9.5.2.1	Introduction	227
9.5.2.2	Type: VnfConfigModifications	227
9.5.3	Referenced structured data types	228
9.5.3.1	Introduction	228
9.5.3.2	Type: VnfConfiguration	228
9.5.3.3	Type: VnfConfigurationData	229
9.5.3.4	Type: VnfcConfigurationData	229
9.5.3.5	Type: CpConfiguration	229
9.5.3.6	Type: CpAddress	230
9.5.4	Referenced simple data types and enumerations	230
Annex A (informative):	Mapping operations to protocol elements	231
A.1	Overview	231
A.2	VNF Lifecycle Management interface	231