ETSI GS MEC 010-2 V3.1.1 (2023-06)

Multi-access Edge Computing (MEC); MEC Management; Part 2: Application lifecycle, rules and requirements management

Disclaimer

The present document has been produced and approved by the Multi-access Edge Computing (MEC) 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/MEC-0010-2v311LcRRM

Keywords
MEC

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 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

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 Definition of terms, symbols and abbreviations 17
3.1 Terms 17
3.2 Symbols 17
3.3 Abbreviations 18
4 Specification level requirements 18
4.1 Requirements for reference point Mm1 18
4.1.1 General requirements 18
4.1.2 Interface requirements 18
4.1.2.1 Application package management interface requirements 18
4.1.2.2 Application lifecycle management interface requirements 19
4.1.2.3 LCM Coordination interface requirements 19
4.2 Requirements for reference point Mm3 20
4.2.1 General requirements 20
4.2.2 Interface requirements 20
4.2.2.1 Application package management interface requirements 20
4.2.2.2 Application lifecycle management interface requirements 20
4.2.2.3 Application lifecycle change notification interface requirements 21
4.3 Requirements for application package 21
4.3.1 General requirements 21
4.3.2 Application descriptor requirements 22
4.4 Requirements for reference point Mm3* 22
4.4.1 General requirements 22
4.4.2 Application lifecycle management interface requirements 22
4.5 Requirements for reference point Mv1 23
4.5.1 General requirements 23
4.5.2 Interface requirements 23
4.6 Requirements for reference point Mv2 23
4.6.1 General requirements 23
4.6.2 Interface requirements 24
5 Message flows to support application package and lifecycle management 24
5.1 General 24
5.2 Application package management 24
5.2.1 General 24
5.2.2 On-board application package 25
5.2.3 Query application package information 25
5.2.4 Disable application package 26
5.2.5 Enable application package 26
5.2.6 Delete application package 27
5.2.7 Fetch onboarded application package 28
5.3 Application instance lifecycle management 28
5.3.1 Application instantiation 28
5.3.2 Application termination 30
5.3.3 Application operation 31
5.4 Invoking application LCM operations 31
5.4.1 Introduction 31
5.4.2 Lifecycle management flow with Granting Procedure 32
5.4.3 Synchronous granting procedure 34

5.4.4 Asynchronous granting procedure 35
5.4.5 Application LCM Operation Intervention 36
5.5 Coordination operations 39
5.5.1 Coordinate LCM operation 39
5.6 Application package management for NFV 41
5.6.1 General 41
5.6.2 On-board application package to NFV 41
5.6.3 Query application package information in NFV 43
5.6.4 Disable application package in NFV 45
5.6.5 Enable application package in NFV 46
5.6.6 Delete application package in NFV 47
5.6.7 Fetch on-boarded application package from NFV 49
5.7 Application lifecycle, rule and requirement management for NFV 50
5.7.1 General 50
5.7.2 Instantiate application in NFV 51
5.7.3 Terminate application in NFV 55
5.7.4 Operate application in NFV 59

6 Information models and interfaces 63
6.1 Applicable reference points 63
6.2 Information models 63
6.2.1 Application descriptor information model 63
6.2.1.1 Introduction 63
6.2.1.2 Type: AppD 63
6.2.1.2.1 Description 63
6.2.1.2.2 Attributes 63
6.2.1.3 Type: VirtualComputeDescriptor 65
6.2.1.3.1 Description 65
6.2.1.3.2 Attributes 65
6.2.1.4 Type: SwImageDescriptor 66
6.2.1.4.1 Description 66
6.2.1.4.2 Attributes 66
6.2.1.5 Type: VirtualStorageDescriptor 66
6.2.1.5.1 Description 66
6.2.1.5.2 Attributes 66
6.2.1.6 Type: AppExternalCpd 66
6.2.1.6.1 Description 66
6.2.1.6.2 Attributes 66
6.2.1.7 Type: ServiceDescriptor 67
6.2.1.7.1 Description 67
6.2.1.7.2 Attributes 67
6.2.1.8 Type: FeatureDependency 67
6.2.1.8.1 Description 67
6.2.1.8.2 Attributes 67
6.2.1.9 Type: TrafficRuleDescriptor 67
6.2.1.9.1 Description 67
6.2.1.9.2 Attributes 67
6.2.1.10 Type: TrafficFilter 68
6.2.1.10.1 Description 68
6.2.1.10.2 Attributes 68
6.2.1.11 Type: InterfaceDescriptor 69
6.2.1.11.1 Description 69
6.2.1.11.2 Attributes 69
6.2.1.12 Type: TunnelInfo 69
6.2.1.12.1 Description 69
6.2.1.12.2 Attributes 69
6.2.1.13 Type: DNSRuleDescriptor 70
6.2.1.13.1 Description 70
6.2.1.13.2 Attributes 70
6.2.1.14 Type: LatencyDescriptor 70
6.2.1.14.1 Description 70
6.2.1.14.2 Attributes 70

6.2.1.15 Type: TerminateAppInstanceOpConfig 70
6.2.1.15.1 Description 70
6.2.1.15.2 Attributes 71
6.2.1.16 Type: ChangeAppInstanceStateOpConfig 71
6.2.1.16.1 Description 71
6.2.1.16.2 Attributes 71
6.2.1.17 Type: ServiceDependency 71
6.2.1.17.1 Description 71
6.2.1.17.2 Attributes 71
6.2.1.18 Type: TransportDependency 71
6.2.1.18.1 Description 71
6.2.1.18.2 Attributes 71
6.2.1.19 Type: TransportDescriptor 72
6.2.1.19.1 Description 72
6.2.1.19.2 Attributes 72
6.2.1.20 Type: UserContextTransferCapability 73
6.2.1.20.1 Description 73
6.2.1.20.2 Attributes 73
6.2.1.21 Type: AppNetworkPolicy 73
6.2.1.21.1 Description 73
6.2.1.21.2 Attributes 73
6.2.1.22 Type: OsContainerDescriptor 74
6.2.1.23 Type: McioIdentificationData 74
6.2.1.23.1 Description 74
6.2.1.23.2 Attributes 74
6.2.1.24 Type: LogicalNodeRequirements 74
6.2.1.24.1 Description 74
6.2.1.24.2 Attributes 75
6.2.1.25 Type: RequestedAdditionalCapabilityData 75
6.2.1.25.1 Description 75
6.2.1.25.2 Attributes 75
6.2.2 Application lifecycle management information model 75
6.2.2.1 Introduction 75
6.2.2.2 Type: LocationConstraints 75
6.2.2.2.1 Description 75
6.2.2.2.2 Attributes 75
6.2.2.3 Type: CreateAppInstanceRequest 76
6.2.2.3.1 Description 76
6.2.2.3.2 Attributes 76
6.2.2.4 Type: AppInstanceInfo 76
6.2.2.4.1 Description 76
6.2.2.4.2 Attributes 76
6.2.2.5 Type: AppInstanceSubscriptionFilter 78
6.2.2.5.1 Description 78
6.2.2.5.2 Attributes 78
6.2.2.6 Type: AppLcmOpOccSubscriptionFilter 79
6.2.2.6.1 Description 79
6.2.2.6.2 Attributes 79
6.2.2.7 Type: InstantiateAppRequest 80
6.2.2.7.1 Description 80
6.2.2.7.2 Attributes 80
6.2.2.8 Type: OperateAppRequest 81
6.2.2.8.1 Description 81
6.2.2.8.2 Attributes 81
6.2.2.9 Type: TerminateAppRequest 81
6.2.2.9.1 Description 81
6.2.2.9.2 Attributes 82
6.2.2.10 Type: AppInstSubscriptionInfo 82
6.2.2.10.1 Description 82
6.2.2.10.2 Attributes 82
6.2.2.11 Type: AppInstNotification 83
6.2.2.11.1 Description 83

6.2.2.11.2 Attributes 83
6.2.2.12 Type: AppInstSubscriptionRequest 83
6.2.2.12.1 Description 83
6.2.2.12.2 Attributes 83
6.2.2.13 Type: AppLcmOpOcc 84
6.2.2.13.1 Description 84
6.2.2.13.2 Attributes 84
6.2.2.14 Type: AppLcmOpOccSubscriptionRequest 85
6.2.2.14.1 Description 85
6.2.2.14.2 Attributes 85
6.2.2.15 Type: AppLcmOpOccSubscriptionInfo 85
6.2.2.15.1 Description 85
6.2.2.15.2 Attributes 86
6.2.2.16 Type: AppLcmOpOccNotification 86
6.2.2.16.1 Description 86
6.2.2.16.2 Attributes 86
6.2.2.17 Type: MECHostInformation 87
6.2.2.17.1 Description 87
6.2.2.17.2 Attributes 87
6.2.2.18 Type: VimConnectionInfo 87
6.2.2.18.1 Description 87
6.2.2.18.2 Attributes 87
6.2.2.19 Type: AppInstanceSubscriptionLinkList 88
6.2.2.19.1 Description 88
6.2.2.19.2 Attributes 88
6.2.2.20 Referenced simple data types and enumerations 88
6.2.2.20.1 Introduction 88
6.2.2.20.2 Simple data types 88
6.2.2.20.3 Enumeration 89
6.2.2.21 Type: ConfigPlatformForAppRequest 89
6.2.2.21.1 Description 89
6.2.2.21.2 Attributes 89
6.2.2.22 Type: MepInformation 90
6.2.2.22.1 Description 90
6.2.2.22.2 Attributes 90
6.2.2.23 Type: AppTermCandsForCoord 90
6.2.2.23.1 Description 90
6.2.2.23.2 Attributes 90
6.2.2.24 Void 90
6.2.2.25 Type: AppInstIdCreationSubscriptionRequest 90
6.2.2.25.1 Description 90
6.2.2.25.2 Attributes 90
6.2.2.26 Type: AppInstIdCreationSubscriptionInfo 91
6.2.2.26.1 Description 91
6.2.2.26.2 Attributes 91
6.2.2.27 Type: AppInstanceIdentifierCreationNotification 91
6.2.2.27.1 Description 91
6.2.2.27.2 Attributes 91
6.2.2.28 Type: AppInstIdDeletionSubscriptionRequest 92
6.2.2.28.1 Description 92
6.2.2.28.2 Attributes 92
6.2.2.29 Type: AppInstIdDeletionSubscriptionInfo 92
6.2.2.29.1 Description 92
6.2.2.29.2 Attributes 92
6.2.2.30 Type: AppInstanceIdentifierDeletionNotification 93
6.2.2.30.1 Description 93
6.2.2.30.2 Attributes 93
6.2.2.31 Type: LocationInformation 93
6.2.2.31.1 Description 93
6.2.2.31.2 Attributes 93
6.2.2.32 Type: CancelMode 94
6.2.2.32.1 Description 94

6.2.2.32.2 Attributes 94
6.2.2.33 Type: McioInfo 94
6.2.2.33.1 Description 94
6.2.2.33.2 Attributes 94
6.2.3 Application package information model 94
6.2.3.1 Introduction 94
6.2.3.2 Type: CreateAppPkg 94
6.2.3.2.1 Description 94
6.2.3.2.2 Attributes 94
6.2.3.3 Type: AppPkgInfo 95
6.2.3.3.1 Description 95
6.2.3.3.2 Attributes 95
6.2.3.4 Type: AppPkgSubscriptionInfo 97
6.2.3.4.1 Description 97
6.2.3.4.2 Attributes 97
6.2.3.5 Type: AppPkgSubscriptionLinkList 97
6.2.3.5.1 Description 97
6.2.3.5.2 Attributes 97
6.2.3.6 Type: AppPkgNotification 97
6.2.3.6.1 Description 97
6.2.3.6.2 Attributes 97
6.2.3.7 Type: AppPkgSubscription 98
6.2.3.7.1 Description 98
6.2.3.7.2 Attributes 98
6.2.3.8 Type: AppPkgInfoModifications 98
6.2.3.8.1 Description 98
6.2.3.8.2 Attributes 98
6.2.3.9 Referenced simple data types and enumerations 99
6.2.3.9.1 Introduction 99
6.2.3.9.2 Simple data types 99
6.2.3.9.3 Enumeration 99
6.2.3.10 Type: AppPkgFilter 99
6.2.3.10.1 Description 99
6.2.3.10.2 Attributes 99
6.2.4 Granting information model 100
6.2.4.1 Introduction 100
6.2.4.2 Type: GrantRequest 100
6.2.4.2.1 Description 100
6.2.4.2.2 Attributes 100
6.2.4.3 Type: ResourceDefinition 101
6.2.4.3.1 Description 101
6.2.4.3.2 Attributes 101
6.2.4.4 Type: Grant 102
6.2.4.4.1 Description 102
6.2.4.4.2 Attributes 102
6.2.4.5 Type: GrantInfo 102
6.2.4.5.1 Description 102
6.2.4.5.2 Attributes 103
6.2.4.6 Type: ZoneInfo 103
6.2.4.6.1 Description 103
6.2.4.6.2 Attributes 103
6.2.4.7 Type: ZoneGroupInfo 104
6.2.4.7.1 Description 104
6.2.4.7.2 Attributes 104
6.2.4.8 Type: ExtVirtualLinkData 104
6.2.4.8.1 Description 104
6.2.4.8.2 Attributes 104
6.2.4.9 Type: ExtLinkPortData 105
6.2.4.9.1 Description 105
6.2.4.9.2 Attributes 105
6.2.4.10 Type: ResourceHandle 105
6.2.4.10.1 Description 105

6.2.4.10.2 Attributes 105
6.2.4.11 Type: VimSoftwareImage 105
6.2.4.11.1 Description 105
6.2.4.11.2 Attributes 106
6.2.4.12 Type: AppExtCpData 106
6.2.4.12.1 Description 106
6.2.4.12.2 Attributes 106
6.2.4.13 Type: AppExtCpConfig 106
6.2.4.13.1 Description 106
6.2.4.13.2 Attributes 106
6.2.4.14 Type: CpProtocolData 107
6.2.4.14.1 Description 107
6.2.4.14.2 Attributes 107
6.2.4.15 Type: IpOverEthernetAddressData 107
6.2.4.15.1 Description 107
6.2.4.15.2 Attributes 107
6.2.5 Common information model 108
6.2.5.1 Introduction 108
6.2.5.2 Type: LinkType 108
6.2.5.2.1 Description 108
6.2.5.2.2 Attributes 108
6.2.5.3 Type: KeyValuePairs 109
6.2.5.3.1 Description 109
6.2.5.4 Type: TimeStamp 109
6.2.5.4.1 Description 109
6.2.5.4.2 Attributes 109
6.2.5.5 Void 109
6.2.5.6 Type: Checksum 109
6.2.5.6.1 Description 109
6.2.5.6.2 Attributes 109
6.3 Interfaces 110
6.3.1 Application lifecycle management interface 110
6.3.1.1 Description 110
6.3.1.2 Create application instance identifier operation 110
6.3.1.2.1 Description 110
6.3.1.2.2 Input parameters 110
6.3.1.2.3 Output parameters 111
6.3.1.2.4 Operation results 111
6.3.1.3 Application instantiation operation 111
6.3.1.3.1 Definition 111
6.3.1.3.2 Input parameters 111
6.3.1.3.3 Output parameters 112
6.3.1.3.4 Operation results 112
6.3.1.4 Change application instance operational state operation 113
6.3.1.4.1 Description 113
6.3.1.4.2 Input parameters 113
6.3.1.4.3 Output parameters 114
6.3.1.4.4 Operation results 114
6.3.1.5 Query application instance information operation 115
6.3.1.5.1 Description 115
6.3.1.5.2 Input parameters 115
6.3.1.5.3 Output parameters 115
6.3.1.5.4 Operation results 115
6.3.1.6 Query application lifecycle operation status 116
6.3.1.6.1 Description 116
6.3.1.6.2 Input parameters 116
6.3.1.6.3 Output parameters 116
6.3.1.6.4 Operation results 116
6.3.1.7 Application instance terminate operation 117
6.3.1.7.1 Description 117
6.3.1.7.2 Input parameters 117
6.3.1.7.3 Output parameters 118

6.3.1.7.4 Operation results 118
6.3.1.8 Delete application instance identifier operation 118
6.3.1.8.1 Description 118
6.3.1.8.2 Input parameters 118
6.3.1.8.3 Output parameters 119
6.3.1.8.4 Operation results 119
6.3.1.9 Subscribe to application lifecycle management notifications 119
6.3.1.9.1 Description 119
6.3.1.9.2 Subscribe 119
6.3.1.9.3 Notify 121
6.3.1.9.4 Query subscription 121
6.3.1.9.5 Delete subscription operation 122
6.3.1.10 Configure platform for application instance operation 123
6.3.1.10.1 Description 123
6.3.1.10.2 Input parameters 123
6.3.1.10.3 Output parameters 123
6.3.1.10.4 Operation results 124
6.3.2 Void 124
6.3.3 Application package management interface 124
6.3.3.1 Fetch onboarded application package operation 124
6.3.3.1.1 Definition 124
6.3.3.1.2 Input parameters 124
6.3.3.1.3 Output parameters 124
6.3.3.1.4 Operation results 125
6.3.3.2 Query application package information operation 125
6.3.3.2.1 Definition 125
6.3.3.2.2 Input parameters 125
6.3.3.2.3 Output parameters 125
6.3.3.2.4 Operation results 125
6.3.3.3 Subscribe operation 126
6.3.3.3.1 Definition 126
6.3.3.3.2 Input parameters 126
6.3.3.3.3 Output parameters 126
6.3.3.3.4 Operation results 126
6.3.3.4 Notify application package operation 126
6.3.3.4.1 Definition 126
6.3.3.5 Onboarding operation 127
6.3.3.5.1 Definition 127
6.3.3.5.2 Input parameters 127
6.3.3.5.3 Output parameters 127
6.3.3.5.4 Operation results 128
6.3.3.6 Enable operation 128
6.3.3.6.1 Definition 128
6.3.3.6.2 Input parameters 128
6.3.3.6.3 Output parameters 128
6.3.3.6.4 Operation results 128
6.3.3.7 Disable operation 128
6.3.3.7.1 Definition 128
6.3.3.7.2 Input parameters 129
6.3.3.7.3 Output parameters 129
6.3.3.7.4 Operation results 129
6.3.3.8 Void 129
6.3.3.9 Delete operation 129
6.3.3.9.1 Definition 129
6.3.3.9.2 Input parameters 129
6.3.3.9.3 Output parameters 130
6.3.3.9.4 Operation results 130
6.3.3.10 Abort application package deletion operation 130
6.3.3.11 Query subscription operation 130
6.3.3.11.1 Definition 130
6.3.3.11.2 Input parameters 130
6.3.3.11.3 Output parameters 130

6.3.3.11.4 Operation results 130
6.3.4 Granting interface 131
6.3.4.1 Introduction 131
6.3.4.2 Granting request 131
6.3.4.2.1 Definition 131
6.3.4.2.2 Input parameters 131
6.3.4.2.3 Output parameters 131
6.3.4.2.4 Operation results 131
6.3.5 LCM Coordination interface 131
6.3.5.1 Description 131
6.3.5.2 Coordinate application lifecycle operation produced by OSS 132
6.3.5.2.1 Description 132
6.3.5.3 Coordinate application lifecycle operation produced by MEAO 132
6.3.5.3.1 Description 132
6.3.5.4 Coordinate application lifecycle operation produced by MEPM-V 132
6.3.5.4.1 Description 132
6.3.5.5 Standardized coordination actions 132
6.3.5.5.1 Introduction 132
6.3.5.5.2 Select application instances to terminate 132
7 API definitions 133
7.1 Introduction 133
7.2 Global definitions and resource structure 133
7.3 Resources of application package management on Mm1 and Mm3 140
7.3.1 Resource: application packages 140
7.3.1.1 Description 140
7.3.1.2 Resource definition 140
7.3.1.3 Resource methods 140
7.3.1.3.1 POST 140
7.3.1.3.2 GET 141
7.3.1.3.3 PUT 143
7.3.1.3.4 DELETE 143
7.3.1.3.5 PATCH 143
7.3.2 Resource: individual application package 143
7.3.2.1 Description 143
7.3.2.2 Resource definition 143
7.3.2.3 Resource methods 143
7.3.2.3.1 POST 143
7.3.2.3.2 GET 144
7.3.2.3.3 PUT 144
7.3.2.3.4 DELETE 145
7.3.2.3.5 PATCH 146
7.3.3 Resource: subscriptions 147
7.3.3.1 Description 147
7.3.3.2 Resource definition 147
7.3.3.3 Resource methods 147
7.3.3.3.1 POST 147
7.3.3.3.2 GET 148
7.3.3.3.3 PUT 149
7.3.3.3.4 DELETE 149
7.3.3.3.5 PATCH 149
7.3.4 Resource: individual subscription 149
7.3.4.1 Description 149
7.3.4.2 Resource definition 150
7.3.4.3 Resource methods 150
7.3.4.3.1 POST 150
7.3.4.3.2 GET 150
7.3.4.3.3 PUT 151
7.3.4.3.4 DELETE 151
7.3.4.3.5 PATCH 152
7.3.5 Resource: notification endpoint 152
7.3.5.1 Description 152

7.3.5.2 Resource definition 152
7.3.5.3 Resource methods 152
7.3.5.3.1 POST 152
7.3.5.3.2 GET 153
7.3.5.3.3 PUT 153
7.3.5.3.4 DELETE 153
7.3.5.3.5 PATCH 153
7.3.6 Resource: application descriptor 154
7.3.6.1 Description 154
7.3.6.2 Resource definition 154
7.3.6.3 Resource methods 154
7.3.6.3.1 POST 154
7.3.6.3.2 GET 154
7.3.6.3.3 PUT 156
7.3.6.3.4 DELETE 156
7.3.6.3.5 PATCH 156
7.3.7 Resource: application package content 156
7.3.7.1 Description 156
7.3.7.2 Resource definition 156
7.3.7.3 Resource methods 157
7.3.7.3.1 POST 157
7.3.7.3.2 GET 157
7.3.7.3.3 PUT 158
7.3.7.3.4 DELETE 159
7.3.7.3.5 PATCH 160
7.4 Resources of application lifecycle management on Mm1 160
7.4.1 Resource: application instances 160
7.4.1.1 Description 160
7.4.1.2 Resource definition 160
7.4.1.3 Resource methods 160
7.4.1.3.1 POST 160
7.4.1.3.2 GET 161
7.4.1.3.3 PUT 163
7.4.1.3.4 DELETE 163
7.4.1.3.5 PATCH 163
7.4.2 Resource: individual application instance 164
7.4.2.1 Description 164
7.4.2.2 Resource definition 164
7.4.2.3 Resource methods 164
7.4.2.3.1 POST 164
7.4.2.3.2 GET 164
7.4.2.3.3 PUT 165
7.4.2.3.4 DELETE 165
7.4.2.3.5 PATCH 166
7.4.3 Resource: subscriptions 167
7.4.3.1 Description 167
7.4.3.2 Resource definition 167
7.4.3.3 Resource methods 167
7.4.3.3.1 POST 167
7.4.3.3.2 GET 169
7.4.3.3.3 PUT 169
7.4.3.3.4 DELETE 169
7.4.3.3.5 PATCH 170
7.4.4 Resource: individual subscription 170
7.4.4.1 Description 170
7.4.4.2 Resource definition 170
7.4.4.3 Resource methods 170
7.4.4.3.1 POST 170
7.4.4.3.2 GET 170
7.4.4.3.3 PUT 171
7.4.4.3.4 DELETE 171
7.4.4.3.5 PATCH 172

7.4.5 Resource: notification endpoint 172
7.4.5.1 Description 172
7.4.5.2 Resource definition 172
7.4.5.3 Resource methods 173
7.4.5.3.1 POST 173
7.4.5.3.2 GET 173
7.4.5.3.3 PUT 173
7.4.5.3.4 DELETE 173
7.4.5.3.5 PATCH 174
7.4.6 Resource: instantiate application instance task 174
7.4.6.1 Description 174
7.4.6.2 Resource definition 174
7.4.6.3 Resource methods 174
7.4.6.3.1 POST 174
7.4.6.3.2 GET 175
7.4.6.3.3 PUT 176
7.4.6.3.4 DELETE 176
7.4.6.3.5 PATCH 176
7.4.7 Resource: terminate application instance task 176
7.4.7.1 Description 176
7.4.7.2 Resource definition 176
7.4.7.3 Resource methods 176
7.4.7.3.1 POST 176
7.4.7.3.2 GET 177
7.4.7.3.3 PUT 178
7.4.7.3.4 DELETE 178
7.4.7.3.5 PATCH 178
7.4.8 Resource: operate application instance task 178
7.4.8.1 Description 178
7.4.8.2 Resource definition 178
7.4.8.3 Resource methods 178
7.4.8.3.1 POST 178
7.4.8.3.2 GET 180
7.4.8.3.3 PUT 180
7.4.8.3.4 DELETE 180
7.4.8.3.5 PATCH 180
7.4.9 Resource: application LCM operation occurrences 180
7.4.9.1 Description 180
7.4.9.2 Resource definition 180
7.4.9.3 Resource methods 180
7.4.9.3.1 POST 180
7.4.9.3.2 GET 180
7.4.9.3.3 PUT 182
7.4.9.3.4 DELETE 182
7.4.9.3.5 PATCH 182
7.4.10 Resource: individual application LCM operation occurrence 182
7.4.10.1 Description 182
7.4.10.2 Resource definition 182
7.4.10.3 Resource methods 182
7.4.10.3.1 POST 182
7.4.10.3.2 GET 183
7.4.10.3.3 PUT 183
7.4.10.3.4 DELETE 184
7.4.10.3.5 PATCH 184
7.4.11 Resource: application LCM operation occurrence cancel task 184
7.4.11.1 Description 184
7.4.11.2 Resource definition 184
7.4.11.3 Resource methods 184
7.4.11.3.1 POST 184
7.4.11.3.2 GET 185
7.4.11.3.3 PUT 185
7.4.11.3.4 DELETE 186

7.4.11.3.5 PATCH 186
7.4.12 Resource: application LCM operation occurrence fail task 186
7.4.12.1 Description 186
7.4.12.2 Resource definition 186
7.4.12.3 Resource methods 186
7.4.12.3.1 POST 186
7.4.12.3.2 GET 187
7.4.12.3.3 PUT 187
7.4.12.3.4 DELETE 188
7.4.12.3.5 PATCH 188
7.4.13 Resource: application LCM operation occurrence retry task 188
7.4.13.1 Description 188
7.4.13.2 Resource definition 188
7.4.13.3 Resource methods 188
7.4.13.3.1 POST 188
7.4.13.3.2 GET 189
7.4.13.3.3 PUT 189
7.4.13.3.4 DELETE 189
7.4.13.3.5 PATCH 190
7.5 Resources of granting on Mm3 190
7.5.1 Resource: grants 190
7.5.1.1 Description 190
7.5.1.2 Resource definition 190
7.5.1.3 Resource methods 190
7.5.1.3.1 POST 190
7.5.1.3.2 GET 191
7.5.1.3.3 PUT 192
7.5.1.3.4 DELETE 192
7.5.1.3.5 PATCH 192
7.5.2 Resource: individual grant 192
7.5.2.1 Description 192
7.5.2.2 Resource definition 192
7.5.2.3 Resource methods 192
7.5.2.3.1 POST 192
7.5.2.3.2 GET 192
7.5.2.3.3 PUT 193
7.5.2.3.4 DELETE 193
7.5.2.3.5 PATCH 194
7.6 Resources of MEPM's application lifecycle management on Mm3 194
7.7 Resources of MEPM-V's application lifecycle management on Mm3* 194
7.7.1 Resource: application instances 194
7.7.2 Resource: individual application instance 194
7.7.3 Resource: subscriptions 194
7.7.4 Resource: individual subscription 194
7.7.5 Resource: notification endpoint 194
7.7.6 Resource: configure_platform_for_app task 194
7.7.6.1 Description 194
7.7.6.2 Resource definition 194
7.7.6.3 Resource methods 195
7.7.6.3.1 POST 195
7.7.6.3.2 GET 196
7.7.6.3.3 PUT 196
7.7.6.3.4 DELETE 196
7.7.6.3.5 PATCH 196
7.7.7 Resource: terminate application instance task 196
7.7.8 Resource: operate application instance task 197
7.7.9 Resource: application LCM operation occurrences 197
7.7.10 Resource: individual application LCM operation occurrence 197
7.7.11 Resource: application LCM operation occurrence cancel task 197
7.7.12 Resource: application LCM operation occurrence fail task 197
7.7.13 Resource: application LCM operation occurrence retry task 197

Annex A (informative): Application package state model 198
A.1 Introduction 198
A.2 State model 198
Annex B (informative): Bibliography 199
History 200

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are 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 Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, 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.

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.

Foreword

This Group Specification (GS) has been produced by ETSI Industry Specification Group (ISG) Multi-access Edge Computing (MEC).

The present document is part 2 of a multi-part deliverable covering MEC Management, as identified below:

Part 1: "System, host and platform management";

Part 2: "Application lifecycle, rules and requirements management".

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 provides information flows for lifecycle management of MEC applications, and describes interfaces over the reference points to support application lifecycle management. It also describes application rules and requirements, application-related events, mobility handling and MEC service availability tracking. The present document specifies the necessary data model, data format and operation format when applicable.

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] ETSI GS NFV-IFA 011: "Network Functions Virtualisation (NFV); Management and Orchestration; VNF Packaging Specification".

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

[3] ISO 3166: "Codes for the representation of names of countries and their subdivisions".

[4] ETSI GS MEC 009: "Multi-access Edge Computing (MEC); General principles, patterns and common aspects of MEC Service APIs".

[5] IETF RFC 8259: "The JavaScript Object Notation (JSON) Data Interchange Format".

[6] IETF RFC 9110: "HTTP Semantics".

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

[8] IETF RFC 7946: "The GeoJSON Format".

[9] Void.

[10] IETF RFC 5246: "The Transport Layer Security (TLS) Protocol Version 1.2".

[11] IETF RFC 6749: "The OAuth 2.0 Authorization Framework".

[12] IETF RFC 6750: "The OAuth 2.0 Authorization Framework: Bearer Token Usage".

[13] ETSI GS MEC 021: "Multi-access Edge Computing (MEC); Application Mobility Service API".

[14] IETF RFC 8446: "The Transport Layer Security (TLS) Protocol Version 1.3".

[15] ETSI GS NFV-IFA 013: "Network Functions Virtualisation (NFV) Release 4; Management and Orchestration; Os-Ma-nfvo reference point - Interface and Information Model Specification".

[16] ETSI GS NFV-IFA 008: "Network Functions Virtualisation (NFV) Release 4; Management and Orchestration; Ve-Vnfm reference point - Interface and Information Model Specification".

[17] ETSI GS MEC 011: "Multi-access Edge Computing (MEC); Edge Platform Application Enablement".

[18] ETSI GS NFV-SOL 004: "Network Functions Virtualisation (NFV) Release 4; Protocols and Data Models; VNF Package and PNFD Archive specification".

[19] ETSI GS NFV-SOL 001: "Network Functions Virtualisation (NFV) Release 4; Protocols and Data Models; NFV descriptors based on TOSCA 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 GR MEC 001: "Multi-access Edge Computing (MEC); Terminology".

[i.2] ETSI GS MEC 002: "Multi-access Edge Computing (MEC); Use Cases and Requirements".

[i.3] ETSI GS NFV-IFA 007: "Network Functions Virtualisation (NFV) Release 4; Management and Orchestration; Or-Vnfm reference point - Interface and Information Model Specification".

[i.4] Void.

[i.5] ETSI GR NFV 003: "Network Functions Virtualisation (NFV); Terminology for Main Concepts in NFV".

[i.6] Void.

[i.7] ETSI GS NFV-SOL 005: "Network Functions Virtualisation (NFV) Release 4; Protocols and Data Models; RESTful protocols specification for the Os-Ma-nfvo Reference Point".

[i.8] Void.

[i.9] ETSI GS NFV-SOL 016: "Network Functions Virtualisation (NFV) Release 4; Protocols and Data Models; NFV-MANO procedures specification".

[i.10] Void.

3 Definition of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the terms given in ETSI GR MEC 001 [i.1] and ETSI GR NFV 003 [i.5] apply.

3.2 Symbols

Void.

3.3 Abbreviations

For the purposes of the present document, the abbreviations given in ETSI GR MEC 001 [i.1], ETSI GR NFV 003 [i.5] and the following apply:

AMS	Application Mobility Service
FQDN	Fully Qualified Domain Name
GRE	Generic Routing Encapsulation

4 Specification level requirements

4.1 Requirements for reference point Mm1

4.1.1 General requirements

The Mm1 reference point between the MEC Orchestrator/MEC application orchestrator and the OSS is used for on-boarding application packages, triggering the instantiation and the termination of MEC applications in the MEC system. Table 4.1.1-1 specifies requirements related to application lifecycle management applicable to the Mm1 reference point.

Table 4.1.1-1: Mm1 reference point requirements

Numbering	Functional requirement description
Mm1.001	The Mm1reference point shall support the application package management interface produced by the MEC Orchestrator/MEC application orchestrator.
Mm1.002	The Mm1reference point shall support the application lifecycle management interface produced by the MEC Orchestrator/MEC application orchestrator.
Mm1.003	The Mm1 reference point may support the application LCM Coordination interface produced by the OSS (see note).
NOTE: LCM Coordination interface dependency information declared in an instantiate application request overrides pre-provisioned coordination dependency information.	

4.1.2 Interface requirements

4.1.2.1 Application package management interface requirements

Table 4.1.2.1-1 specifies requirements applicable to the application package management interface produced by the MEC Orchestrator/MEC application orchestrator on the Mm1 reference point. Those requirements are derived from clause 5.2 of ETSI GS MEC 002 [i.2].

Table 4.1.2.1-1: Application package management interface requirements

Numbering	Functional requirement description
Mm1.AppPkgm.001	The Application Package Management interface produced by the MEO/MEAO on the Mm1 reference point shall support on-boarding an Application Package.
Mm1.AppPkgm.002	The Application Package Management interface produced by the MEO/MEAO on the Mm1 reference point shall support querying Application Package information (see note).
Mm1.AppPkgm.003	The Application Package Management interface produced by the MEO/MEAO on the Mm1 reference point shall support deleting an Application Package.
Mm1.AppPkgm.004	The Application Package Management interface produced by the MEO/MEAO on the Mm1 reference point shall support enabling an application package.
Mm1.AppPkgm.005	The Application Package Management interface produced by the MEO/MEAO on the Mm1 reference point shall support disabling an application package.
Mm1.AppPkgm.006	The Application Package Management interface produced by the MEO/MEAO on the Mm1 reference point shall support providing notifications as a result of changes on application package states.
Mm1.AppPkgm.007	The Application Package Management interface produced by the MEO/MEAO on the Mm1 reference point shall support providing notifications about the on-boarding of application packages.
Mm1.AppPkgm.008	The Application Package Management interface produced by the MEO/MEAO on the Mm1 reference point shall support fetching an application package, or selected files contained in a package.
NOTE: Application package information may include information such as release date, vendor info, manifest, application descriptor, and other files contained in the application package, etc.	

4.1.2.2 Application lifecycle management interface requirements

Table 4.1.2.2-1 specifies requirements applicable to the application lifecycle management interface produced by the MEC Orchestrator/MEC application orchestrator on the Mm1 reference point.

Table 4.1.2.2-1: Application lifecycle management interface requirements

Numbering	Functional requirement description
Mm1.AppLcm.001	The Application Lifecycle Management interface produced by the MEO/MEAO on the Mm1 reference point shall support instantiating an Application instance.
Mm1.AppLcm.002	The Application Lifecycle Management interface produced by the MEO/MEAO on the Mm1 reference point shall support terminating an Application instance.
Mm1.AppLcm.003	The Application Lifecycle Management interface produced by the MEO/MEAO on the Mm1 reference point shall support requesting to change the state of an application instance (see note).
Mm1.AppLcm.004	The Application Lifecycle Management interface produced by the MEO/MEAO on the Mm1 reference point shall support providing notifications as a result of lcm operations.
NOTE: Changing the state of an application instance refers to starting or stopping an application instance. These operations are complementary to instantiating or terminating an application.	

4.1.2.3 LCM Coordination interface requirements

Table 4.1.2.3-1 specifies requirements applicable to the LCM Coordination interface produced by the OSS on the Mm1 reference point.

Table 4.1.2.3-1: LCM Coordination interface requirements

Numbering	Functional requirement description
Mm1.LcmCoord.001	The LCM Coordination interface produced by the OSS on the Mm1 reference point shall support receiving requests to coordinate LCM operations affecting an application instance (see note).
NOTE: Examples where coordination is required include requests by the MEO or MEAO to the OSS to request permission to terminate application instances.	

4.2 Requirements for reference point Mm3

4.2.1 General requirements

The Mm3 reference point between the MEC Orchestrator and the MEC Platform Manager is used for the management of the application lifecycle, application rules and requirements and keeping track of available MEC services, etc. Table 4.2.1-1 specifies requirements related to application lifecycle management applicable to the Mm3 reference point.

Table 4.2.1-1: Mm3 reference point requirements

Numbering	Functional requirement description
Mm3.001	The Mm3 reference point shall support the application package management interface produced by the MEC Orchestrator.
Mm3.002	The Mm3 reference point shall support the application Lifecycle Management interface produced by the MEC Platform Manager.
Mm3.003	The Mm3 reference point shall support the application Lifecycle Change Notification interface produced by the MEC Platform Manager.

4.2.2 Interface requirements

4.2.2.1 Application package management interface requirements

Table 4.2.2.1-1 specifies requirements applicable to the Application Package Management interface produced by the MEO on the Mm3 reference point.

Table 4.2.2.1-1: Application Package Management interface requirements

Numbering	Functional requirement description
Mm3.AppPkgm.001	The Application Package Management interface produced by the MEO on the Mm3 reference point shall support querying application package information (see note).
Mm3.AppPkgm.002	The Application Package Management interface produced by the MEO on the Mm3 reference point shall support providing notifications as a result of changes on application package states.
Mm3.AppPkgm.003	The Application Package Management interface produced by the MEO on the Mm3 reference point shall support providing notifications about the on-boarding of application packages.
Mm3.AppPkgm.004	The Application Package Management interface produced by the MEO on the Mm3 reference point shall support fetching an application package, or selected files contained in a package.
NOTE: Application package information may include information such as release date, vendor info, manifest, application descriptor, and other files contained in the application package, etc.	

4.2.2.2 Application lifecycle management interface requirements

Table 4.2.2.2-1 specifies requirements applicable to the Application Lifecycle Management interface produced by the MEC Platform Manager on the Mm3 reference point.

Table 4.2.2.2-1: Application Lifecycle Management interface requirements

Numbering	Functional requirement description
Mm3.AppLcm.001	The Application Lifecycle Management interface produced by the MEC Platform Manager on the Mm3 reference point shall support instantiating an Application.
Mm3.AppLcm.002	The Application Lifecycle Management interface produced by the MEC Platform Manager on the Mm3 reference point shall support terminating an application instance.
Mm3.AppLcm.003	The Application Lifecycle Management interface produced by the MEC Platform Manager on the Mm3 reference point shall support querying information about an application instance.
Mm3.AppLcm.004	The Application Lifecycle Management interface produced by the MEC Platform Manager on the Mm3 reference point shall support requesting to change the state of an application instance (see note).
Mm3.AppLcm.005	The Application Lifecycle Management interface produced by the MEC Platform Manager on the Mm3 reference point shall support querying the status of an ongoing application lifecycle management operation.
NOTE: Changing the state of an application instance refers to starting or stopping an application instance. These operations are complementary to instantiating or terminating an application.	

4.2.2.3 Application lifecycle change notification interface requirements

Table 4.2.2.3-1 specifies requirements applicable to the Application Lifecycle Change Notification interface produced by the MEC Platform manager on the Mm3 reference point.

Table 4.2.2.3-1: Application Lifecycle Change Notification interface requirements

Numbering	Functional requirement description
Mm3.AppLccn.001	The Application Lifecycle Change Notification interface produced by the MEC Platform Manager on the Mm3 reference point shall support providing to the MEO notifications about changes of an application instance that are related to application lifecycle management operations.
Mm3.AppLccn.002	Notifications provided on the Application Lifecycle Change Notification interface produced by the MEC Platform Manager on the Mm3 reference point shall contain information about the type of application lifecycle operation, the identification of the application instance.
Mm3.AppLccn.003	Notifications provided on the Application Lifecycle Change Notification interface produced by the MEC Platform Manager on the Mm3 reference point shall support indicating the start of the lifecycle operation, the end and the results of the lifecycle operation including any error produced from the lifecycle operation.
Mm3.AppLccn.004	The Application Lifecycle Change Notification interface produced by the MEC Platform Manager on the Mm3 reference point shall support notifying the result (successful or failed) of application instantiation with indicating the application instance identifier, and the consumed, modified or released resources.

4.3 Requirements for application package

4.3.1 General requirements

Table 4.3.1-1 specifies requirements related to application lifecycle management applicable to the application package.

Table 4.3.1-1: Application package requirements

Numbering	Functional requirement description
AppPkt.001	The application package shall contain software image(s) or link(s) to software image(s). See note.
AppPkt.002	The application package shall contain an application descriptor that describes the application requirements and rules which are required or preferred by the MEC application. See note.
AppPkt.003	The application package shall be signed by the application provider. The digest and the public key of the entity signing shall be included in the package along with the corresponding certificate.
AppPkt.004	Files in the package may be individually signed. For each signed file, the corresponding public key, algorithm and certificate used shall be stored in a well-known location within the application package.
AppPkt.005	The application package shall contain a manifest file which lists files that the package contains and a hash of their content. See note.

Numbering	Functional requirement description
AppPkt.006	The application package shall contain one or more Managed Container Infrastructure Object Packages (MCIOP) representing aggregated containerized workload structures, when the application is realized by a set of OS containers.
NOTE: The application package, the software image(s), the manifest file and the application descriptor are provided by the application provider.	

4.3.2 Application descriptor requirements

Table 4.3.2-1 specifies requirements related to application lifecycle management applicable to the application descriptor.

Table 4.3.2-1: Application descriptor requirements

Numbering	Functional requirement description
AppDesc.001	The application descriptor shall contain a description of minimum computation resources required by the application, e.g. amount, characteristics and capabilities for virtual compute.
AppDesc.002	The application descriptor shall contain a description of minimum virtual storage resources the required by application.
AppDesc.003	The application descriptor shall contain a description of minimum virtual network resources required by the application.
AppDesc.004	The application descriptor shall support describing a list of services a MEC application requires to run.
AppDesc.005	The application descriptor shall support describing a list of additional services that a MEC application may use if available.
AppDesc.006	The application descriptor shall support describing a list of features a MEC application requires to run.
AppDesc.007	The application descriptor shall support describing a list of additional features a MEC application may use if available.
AppDesc.008	The application descriptor shall support a description of Traffic Rules.
AppDesc.009	The application descriptor shall support a description of DNS Rules which provide specific FQDNs to be registered into the MEC system (e.g. for redirection of traffic to local host).
AppDesc.010	The application descriptor shall support a description of latency required by the MEC application.

4.4 Requirements for reference point Mm3*

4.4.1 General requirements

The Mm3* reference point between the MEC Application Orchestrator and the MEC Platform Manager - NFV is used for the management of the application lifecycle, application rules and requirements and keeping track of available MEC services, etc. Table 4.4.1-1 specifies requirements related to application lifecycle management applicable to the Mm3* reference point.

Table 4.4.1-1: Mm3* reference point requirements

Numbering	Functional requirement description
Mm3*.001	The Mm3* reference point shall support the application Lifecycle Management interface produced by the MEC Platform Manager - NFV.

4.4.2 Application lifecycle management interface requirements

Table 4.4.2-1 specifies requirements applicable to the Application Lifecycle Management interface produced by the MEC Platform Manager - NFV on the Mm3* reference point.

Table 4.4.2-1: Application Lifecycle Management interface requirements

Numbering	Functional requirement description
Mm3*.AppLcm.001	The Application Lifecycle Management interface produced by the MEPM-V on the Mm3* reference point shall support configuration an application instance. (see note 1).
Mm3*.AppLcm.002	The Application Lifecycle Management interface produced by the MEPM-V on the Mm3* reference point shall support terminating an application instance.
Mm3*.AppLcm.003	The Application Lifecycle Management interface produced by the MEPM-V on the Mm3* reference point shall support querying information about an application instance.
Mm3*.AppLcm.004	The Application Lifecycle Management interface produced by the MEPM-V on the Mm3* reference point shall support requesting to change the state of an application instance (see note 2).
NOTE 1: Configuration an application instance refers to send configuration to MEPM-V, this configuration includes the traffic rules, DNS rules, the required and optional services, and services produced by the application instance. NOTE 2: Changing the state of an application instance refers to starting or stopping an instance at application level without virtualised resources operations. These operations are complementary to instantiating or terminating an application as a VNF instance.	

4.5 Requirements for reference point Mv1

4.5.1 General requirements

The Mv1 reference point between the MEC application Orchestrator and the NFV Orchestrator is used for on-boarding application packages as a VNF package, triggering the application lifecycle of MEC applications as VNF instances in the MEC system. Table 4.5.1-1 specifies requirements applicable to the Mv1 reference point.

Table 4.5.1-1: Mv1 reference point requirements

Numbering	Functional requirement description
Mv1.001	The Mv1 reference point shall support the application package as a VNF package on the VNF package management interface produced by the NFV Orchestrator (see note 1).
Mv1.002	The Mv1 reference point shall support the application Lifecycle Management interface produced by the NFV Orchestrator (see note 1).
Mv1.003	The Mv1 reference point shall support the application Lifecycle Change Notification interface produced by the NFV Orchestrator (see note 1).
Mv1.004	The Mv1 reference point shall support the LCM Coordination interface produced by the MEC application Orchestrator (see note 1 and note 2).
NOTE 1: Refer to ETSI GS NFV-IFA 013 [15] for details. NOTE 2: The dependency on the LCM Coordination interface is declared in the application package.	

4.5.2 Interface requirements

Table 4.5.2-1 specifies requirements applicable to the LCM Coordination interface produced by the MEAO on the Mv1 reference point.

Table 4.5.2-1: LCM Coordination interface requirements

Numbering	Functional requirement description
Mv1.LcmCoord.001	The LCM Coordination interface produced by the MEAO on the Mv1 reference point shall support receiving requests to coordinate LCM operations affecting an application instance as a VNF instances.

4.6 Requirements for reference point Mv2

4.6.1 General requirements

The Mv2 reference point between the MEC Platform Manager - NFV and the VNF Manager is used for the management of the application lifecycle of MEC applications as VNF instances in the MEC system. Table 4.6.1-1 specifies requirements applicable to the Mv2 reference point.

Table 4.6.1-1: Mv2 reference point requirements

Numbering	Functional requirement description
Mv2.001	The Mv2 reference point shall support the application Lifecycle Change Notification interface produced by the VNF Manager (see note).
Mv2.002	The Mv2 reference point shall support the LCM Coordination interface produced by the MEC Platform Manager - NFV.
NOTE: Refer to ETSI GS NFV-IFA 008 [16] for details.	

4.6.2 Interface requirements

Table 4.6.2-1 specifies requirements applicable to the LCM Coordination interface produced by the MEAO on the Mv2 reference point.

Table 4.6.2-1: LCM Coordination interface requirements

Numbering	Functional requirement description
Mv2.LcmCoord.001	The LCM Coordination interface produced by the MEPM-V on the Mv2 reference point shall support receiving requests to coordinate LCM operations affecting an application instance as a VNF instances.

5 Message flows to support application package and lifecycle management

5.1 General

Message flows in this clause are informative.

5.2 Application package management

5.2.1 General

Message flows for application package management are used to make application package available to the MEC system, delete the application package from the MEC system, or query information of one or more application packages. The series of message flows include:

- On-board application package.
- Query application package information.
- Disable application package.
- Enable application package.
- Delete application package.
- Fetch application package.

This interface is available on the reference points Mm1 and Mm3. On Mm1, the full functionality of this interface is supported. On Mm3, only read access to the packages and their content is supported.

5.2.2 On-board application package

The message flow of on-board application package is used to make application package available to the MEC system. The on-board application package message flow is executed before an application is instantiated; the actual time to execute this message flow is dependent on implementation. The detailed description of the flow is in figure 5.2.2-1.

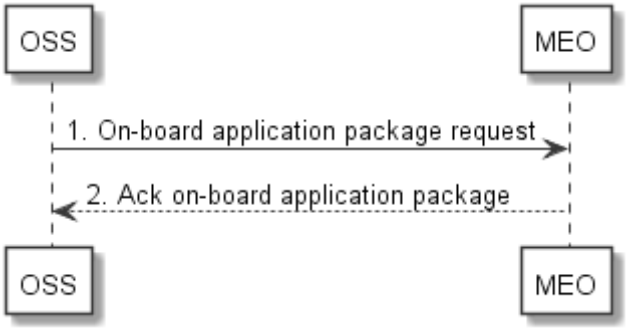

Figure 5.2.2-1: On-board application package

1) The OSS sends an on-board application package request to the MEC Orchestrator, in which, the MEC application package is included.

   The MEC Orchestrator checks the application package, for example the MEC Orchestrator checks for the existence of mandatory elements within the application package, validates the authenticity and integrity of the application package; and the MEC Orchestrator checks the format of application image and format of the application rules and requirements.

   The MEC Orchestrator allocates a unique application package ID for the on-boarded MEC application package and related status information, and keeps a record of on-boarded application packages. Optionally the MEC Orchestrator prepares the virtualisation infrastructure manager(s) with the application image (e.g. by sending the application image to appropriate virtualisation infrastructure managers), alternatively, such preparation may be done later, but needs to be finished before the application is instantiated. The MEC Orchestrator notifies the subscribers to AppPackageOnBoardingNotification of the on-boarding of the MEC application package.

2) The MEC Orchestrator acknowledges the application package on-boarding to the OSS. The application package is now available in the MEC system and enabled.

This message flow is only supported on Mm1.

5.2.3 Query application package information

Query application package allows returning the information contained in an application package (such as the location of the application descriptor) to the OSS and the MEPM. The detailed description of the flow is in figure 5.2.3-1.

This flow can be used by the MEPM to obtain the application requirements, traffic redirection rules and DNS rules that it needs to configure the MEC platform to run the application. This information is provided as defined by the application provider in the application package. Modification of these rules by the MEO prior to sending it to the MEPM is not supported in the present version of the present document.

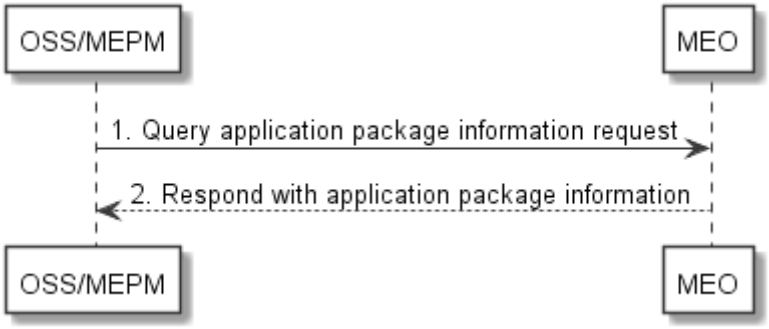

Figure 5.2.3-1: Query application package information

1) The OSS or MEPM sends a query application package information request (including a filter) to the MEC Orchestrator to query information of application package(s).

   The MEC Orchestrator authorizes the request.

2) The MEC Orchestrator returns the information related to the application package(s) that matches the filter in the query request.

This message flow is supported on Mm1 and Mm3.

5.2.4 Disable application package

Disabling application package refers to the process of marking an application package as disabled in the MEC system, so that it is not possible to be used for application instantiation. The detailed description of the flow is in figure 5.2.4-1.

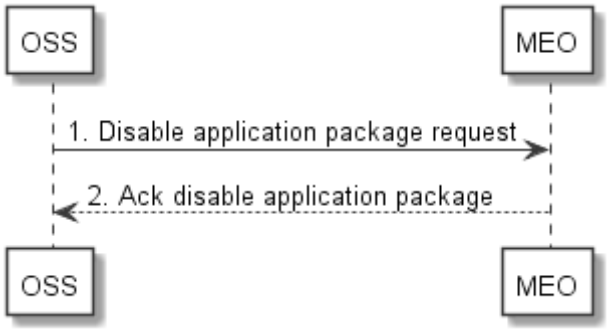

Figure 5.2.4-1: Disable application package

1) The OSS sends a request to disable application package to the MEC Orchestrator.

   The MEC Orchestrator processes the request and checks if the application package exists, and is enabled. If the application package is enabled, the MEC Orchestrator marks the application package as disabled in the MEC system.

   The MEC Orchestrator notifies the subscribers to AppPackageStateChangeNotification of the state change of the MEC application package.

2) The MEC Orchestrator acknowledges the disable application package request.

This message flow is only supported on Mm1.

5.2.5 Enable application package

Enabling an application package refers to the process of marking an application package as enabled in the MEC system, so that it may be used for application instantiation again. The detailed description of the flow is in figure 5.2.5-1.

Figure 5.2.5-1: Enable application package

1) The OSS sends an enable application package request to the MEC Orchestrator.

   The MEC Orchestrator processes the request and checks if the application package exists, and is disabled. If the application package is disabled, the MEC Orchestrator marks the application package as enabled in the MEC system.

   The MEC Orchestrator notifies the subscribers to AppPackageStateChangeNotification of the state change of the MEC application package.

2) The MEC Orchestrator acknowledges the application package enable request.

This message flow is only supported on Mm1.

5.2.6 Delete application package

Delete application package refers to the process of removing an application package from the MEC system. The OSS initiates the delete application package message flow. The detailed description of the flow is in figure 5.2.6-1.

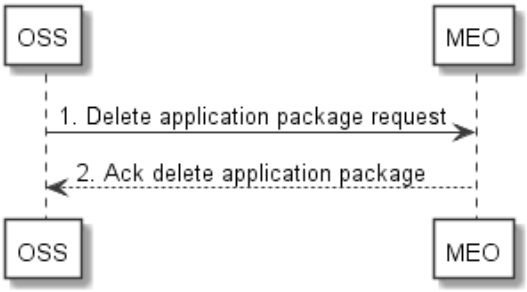

Figure 5.2.6-1: Delete application package

1) The OSS sends a delete application package request to the MEC Orchestrator.

   The MEC Orchestrator checks whether the application package is onboarded, disabled and in use. If the application package is not onboarded, not disabled or is in use, this operation will return an error. If the application package is disabled and is not in use, the MEC Orchestrator removes the application package and its associated AppPkgInfo from the MEC system.

   The MEC Orchestrator notifies the subscribers to AppPackageStateChangeNotification of the state change of the MEC application package.

2) The MEC Orchestrator acknowledges application package deletion request.

This message flow is only supported on Mm1.

5.2.7 Fetch onboarded application package

Fetch onboarded application package allows retrieving an application package, or selected files contained in a package to the OSS and the MEPM. The detailed description of the flow is in figure 5.2.7-1.

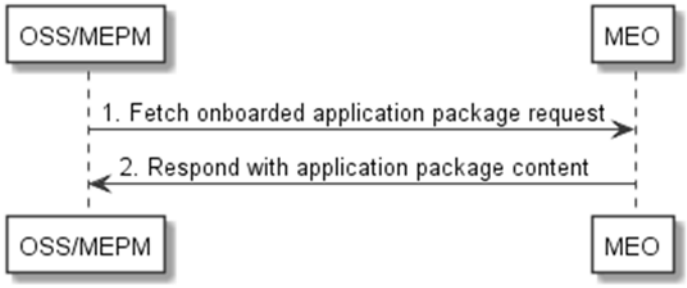

Figure 5.2.7-1: Fetch onboarded application package

1) The OSS or MEPM sends a fetch onboarded application package request to the MEC Orchestrator to fetch the content of an application package.

   The MEC Orchestrator authorizes the request.

2) The MEC Orchestrator returns the application package.

5.3 Application instance lifecycle management

5.3.1 Application instantiation

The message flow of application instantiation is used to instantiate an application instance in the MEC system. The detailed description of the flow is in figure 5.3.1-1.

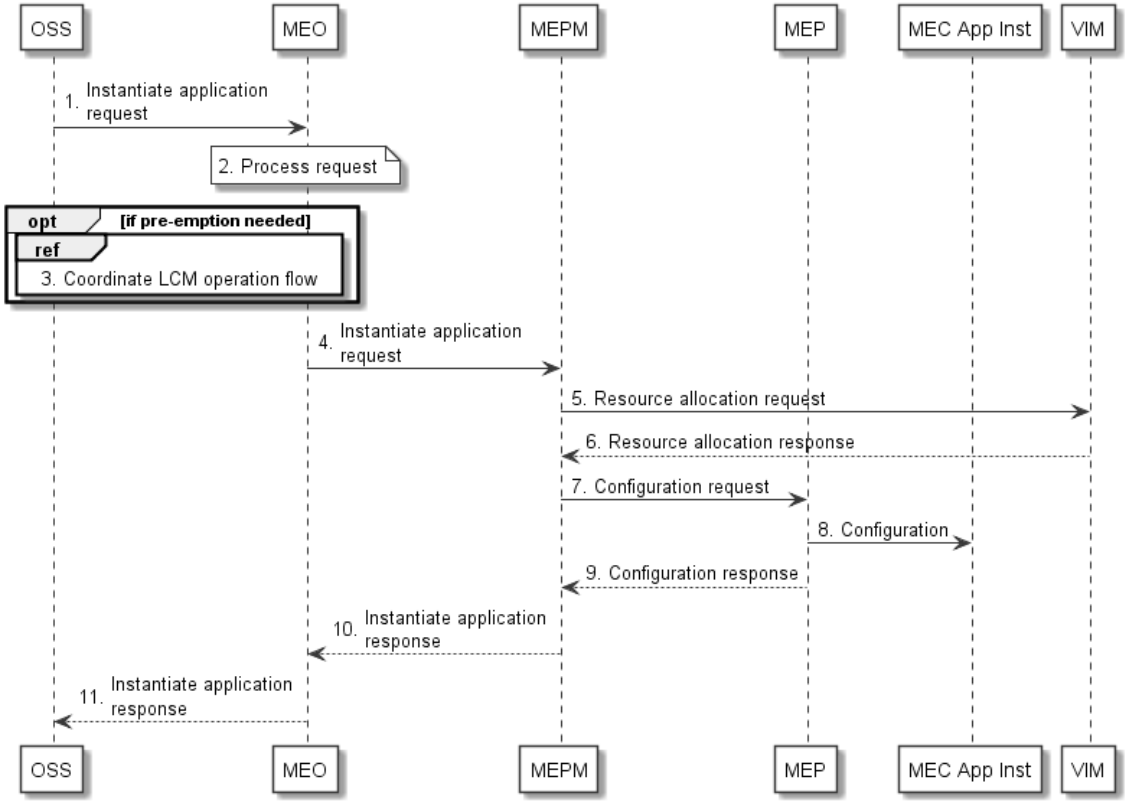

Figure 5.3.1-1: Application instantiation flow

1) The OSS sends an instantiate application request to the MEC Orchestrator.

2) MEC Orchestrator checks the application instance configuration data, and authorizes the request. The MEC Orchestrator selects the MEC host (and corresponding MEC Platform Manager).

3) If the MEC Orchestrator determines that the resources required to instantiate the new application instance are insufficient at the MEC host, the MEO may suggest terminating already instantiated application instances (sets of candidate alternatives for such termination may be provided in the instantiate application request, and will otherwise be chosen by the MEO) through the coordinate LCM operation (using pre-provisioned coordination endpoint information) so that the necessary resources could be made available, see clause 5.5.1.

NOTE 1: Any application instance termination resulting from use of the coordinate LCM operation will be performed using a separate LCM operation initiated by the MEO, with associated notifications provided to the OSS.

4) The MEC Orchestrator sends an instantiate application request to the MEC Platform Manager.

5) The MEC Platform Manager sends a resource allocation request to the Virtualisation Infrastructure Manager (VIM), with the requested resource including compute, storage, and network resources. The MEC Platform Manager will include application image information (e.g. a link to the image or an ID of the application image) in the request.

6) The VIM allocates the resources according to the request of the MEC Platform Manager. And if the application image is available, the VIM loads the virtualisation container with the application image, and runs the virtualisation container and the application instance. The VIM sends resource allocation response to the MEC Platform Manager.

NOTE 2: Optionally, if traffic redirection is realized based on NFVI mechanisms controlled by the VIM, the MEC Platform Manager further determines the need to create or update a forwarding path based on the traffic rule(s) (associated with the instantiated application), and sends a forwarding path creation or update request to the VIM. The VIM creates or updates the forwarding path according to the indication of the request and sends a forwarding path creation or update response to the MEC Platform Manager. The MEC Platform Manager associates the forwarding path with the traffic rule(s) of the instantiated application.

NOTE 3: If traffic redirection is realized based on mechanisms internal to the data plane and controlled via Mp2, it is out of scope of the present document how traffic redirection via Mp2 during application instance creation, and during enabling, disabling or modifying a traffic rule are controlled.

7) The MEC Platform Manager sends configuration request to the MEC platform. In this message, the MEC Platform Manager includes the traffic rules to be configured, DNS rules to be configured, the required and optional services, and services produced by the application instance, etc.

NOTE 4: This step is only for the completeness of the flow, and will not be specified in the present document.

8) The MEC platform configures the Traffic rules and DNS rules for the application instance. The MEC platform needs to wait until the application instance runs normally (e.g. the application instance state turns into the running state) to activate the traffic and DNS rules. After the application instance runs normally, the MEC platform provides the available service information to the application. The details of how the MEC application instance interacts with MEC platform and how the MEC platform handles services can be found in ETSI GS MEC 011 [17].

9) The MEC platform sends a configuration response to the MEC Platform Manager.

NOTE 5: This step is only for the completeness of the flow, and will not be specified in the present document.

10) The MEC Platform Manager sends an instantiate application response to the MEC Orchestrator. The MEC Platform Manager includes the information of the resources allocated to the application instance to the MEC Orchestrator.

11) The MEC Orchestrator sends an instantiate application response to the OSS, returning the results of the instantiation operation. The MEC Orchestrator also returns the application instance ID to the OSS if the flow is successful. Meanwhile a notification is sent to the subscribers to the application instance operational state change notification and this application package is marked in use.

5.3.2 Application termination

The message flow of application instance termination is used to terminate an application instance in the MEC system. The detailed description of the flow is in figure 5.3.2-1.

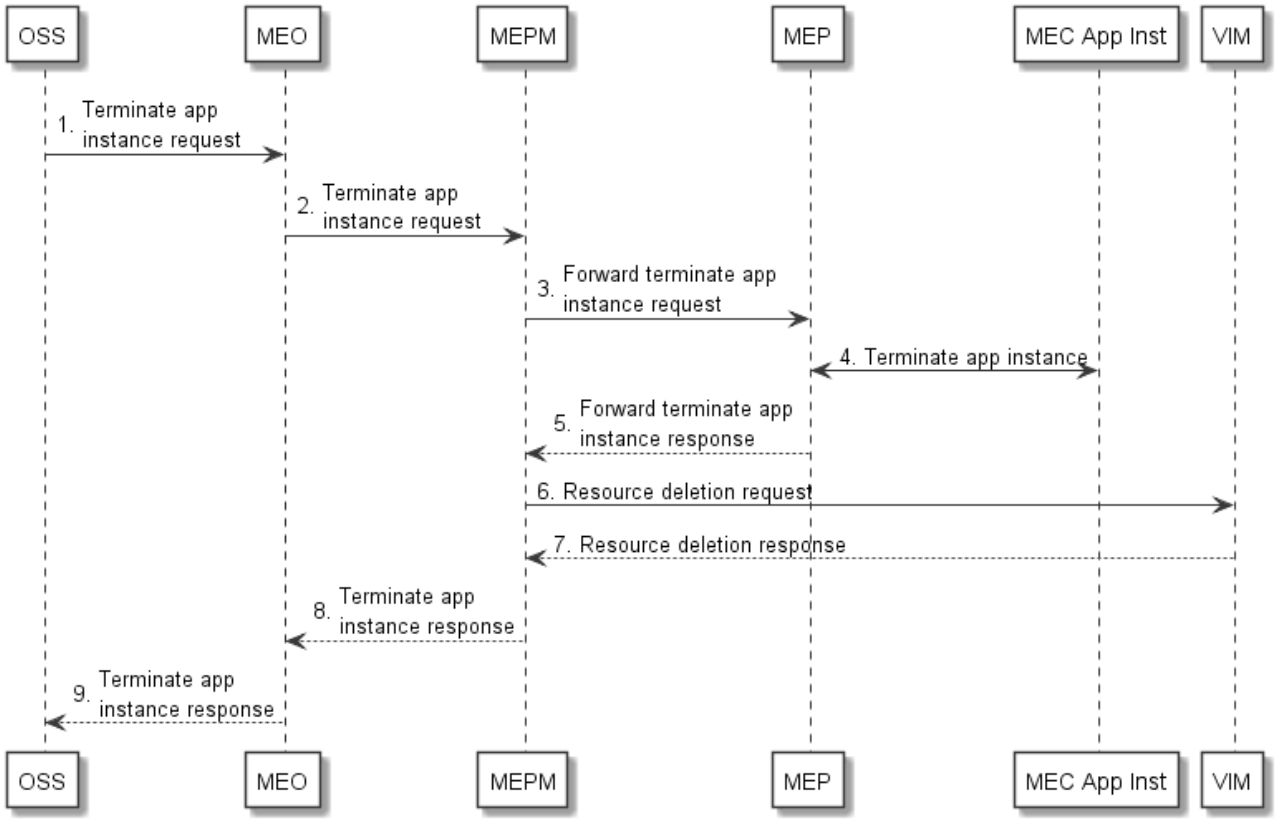

Figure 5.3.2-1: Instance Termination information flow

1) The OSS sends a terminate application instance request to the MEC Orchestrator. The message includes the MEC application instance ID to be terminated.

2) The MEC Orchestrator authorizes the request, and verifies the existence of requested instance. The MEC Orchestrator sends a terminate application instance request to the MEC Platform Manager that manages the MEC application instance to be terminated.

NOTE 1: The application instance can be terminated independent of its operational state.

3) The MEC Platform Manager sends a forward terminate application instance request to the MEC platform.

NOTE 2: This step is only for the completeness of the flow, and will not be specified in the present document.

4) If supported by the MEC application, and graceful termination is requested, the MEC platform notifies the MEC application instance of the termination event. The MEC application instance executes the actions needed before it has been terminated by the MEC platform, the actual action(s) the MEC application instance will perform for the application level termination is up to MEC application, and is out of scope of the present document. After the MEC application instance finishes application level termination, it may inform the MEC platform that it is ready to be terminated. The MEC platform may set a timer for the application level termination. After the timer expires, the MEC platform will shut down the application regardless of the progress of application level termination.

5) The MEC platform sends a forward terminate application instance response to the MEC Platform Manager.

NOTE 3: This step is only for the completeness of the flow, and will not be specified in the present document.

6) The MEC Platform Manager sends a resource deletion request to the corresponding virtualisation infrastructure manager, to terminate the virtualisation container and release the resources.

7) The virtualisation infrastructure manager releases the resources allocated for the application instance to be terminated. And the virtualisation infrastructure manager sends resource a deletion response to the MEC Platform Manager.

8) The MEC Platform Manager sends a terminate application instance response to the MEC Orchestrator.

9) The MEC Orchestrator sends a terminate application instance response to the OSS. Meanwhile a notification is sent to the subscribers to the application instance operational state change notification.

5.3.3 Application operation

This message flow is to operate (i.e. start or stop) an application instance in the MEC system. The detailed description of the flow is in figure 5.3.3-1.

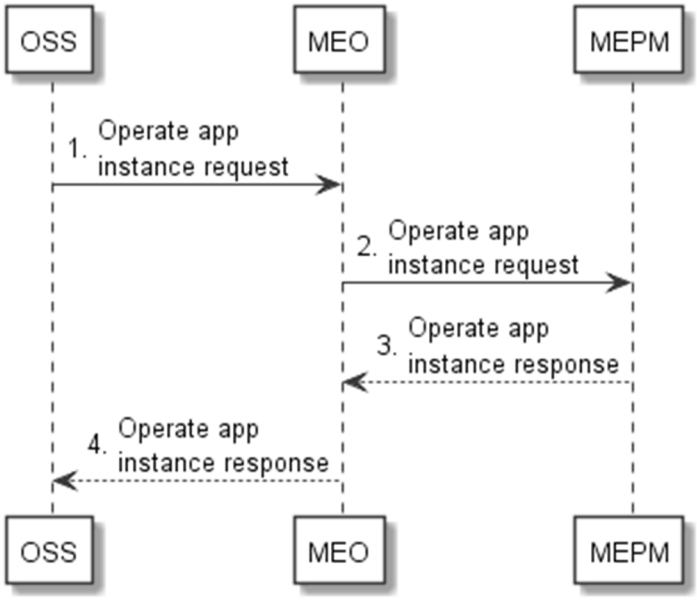

Figure 5.3.3-1: Application operation flow

1) The OSS sends an operate (i.e. start or stop) application request to the MEC Orchestrator.

2) The MEC Orchestrator forwards the operate application request to the MEC Platform Manager.

3) The MEC Platform Manager processes this request, and sends the result of operation on the application instance once the operation completes.

4) The MEC Orchestrator sends the result of application operation to the OSS.

5.4 Invoking application LCM operations

5.4.1 Introduction

The following clauses describe the general sequence for application instance lifecycle management operations that operate on application instance resource and are triggered by task resources. The operations of application instance lifecycle management are:

- Instantiate.
- Operate.
- Terminate.

These operations are provided by both, MEO and MEAO.

The MEO and MEAO can use a resource that represents an application instance LCM operation occurrence to monitor the progress of the operation.

The MEO can also subscribe to notifications sent by the MEPM when an application instance LCM operation occurrence changes its state. Further, each application LCM operation invocation triggers a granting procedure between MEPM and MEO.

The MEAO can subscribe to notifications sent by the NFVO in a MEC on NFV environment. The NFVO then notifies the MEAO on LCM operations invoked at the NFVO and on state changes. This also enables the MEC specific actions to be performed when MEC applications are managed directly via the NFVO.

Table 5.4.1-1 shows parameterization associated to above application instance lifecycle management operations.

Table 5.4.1-1: Parameterization for application instance lifecycle management operations

Operation	Precondition	Task	RequestStructure	Postcondition
Instantiate	Application instance created and in NOT_INSTANTIATED state	instantiate	InstantiateAppRequest	Application instance in INSTANTIATED state. Once an application is instantiated, the application instance is in the operational state STARTED.
Operate	Application instance in INSTANTIATED state	operate	OperateAppRequest	Application instance still in INSTANTIATED state and application instance operational state changed.
Terminate	Application instance in INSTANTIATED state. See note.	terminate	TerminateAppRequest	Application instance in NOT_INSTANTIATED state.
NOTE: The application instance can be terminated independently to its operational state (STARTED or STOPPED).				

5.4.2 Lifecycle management flow with Granting Procedure

Figure 5.4.2-1 illustrates the general lifecycle management flow with granting procedure for the application lifecycle management. Placeholders in this flow allow for differentiating among the operations and are marked with double angular brackets "<<…>>".

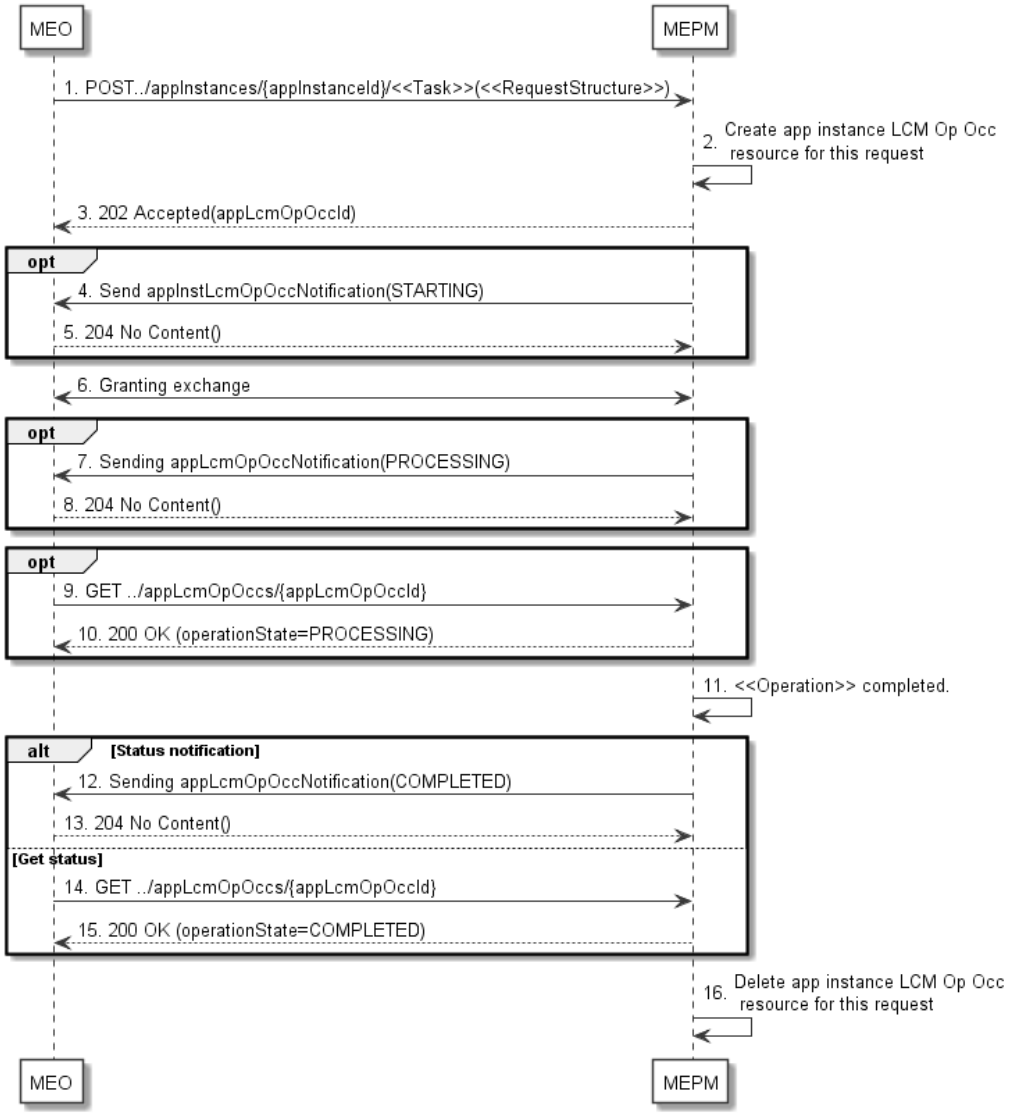

Figure 5.4.2-1: Application instance LCM operation with granting procedure

An application instance lifecycle operation, as illustrated in figure 5.4.2-1, consists of the following steps:

1) The MEO sends a POST request to the <<Task>> resource that represents the lifecycle operation to be executed on the application instance, and includes in the message content a data structure of type <<RequestStructure>>. The <<Task>> of the task resource and the <<RequestStructure>> depend on the operation and are described in table 5.4.1-1.

2) The MEPM creates an "application instance LCM operation occurrence" resource for the request.

3) The MEPM returns a "202 Accepted" response with an empty message content and a "Location" HTTP header that points to the new "application instance LCM operation occurrence" resource, i.e. it includes the URI of that resource which is "…/app_lcm_op_occs/{appLcmOpOccId}".

If the MEO has subscribed to receive notifications sent by the MEPM when an application instance LCM operation occurrence changes its state to "STARTING":

4) The MEPM sends to the MEO an application instance lifecycle management operation occurrence notification to indicate the start of the lifecycle management operation occurrence with the "STARTING" state.

5) The MEO returns a "204 No Content" response with an empty message content.

6) The MEPM and MEO exchange granting information, via either synchronous mode (see clause 5.4.3) or asynchronous mode (see clause 5.4.4). Asynchronous mode is typically used if it is expected to take some time to create the grant.

If the MEO has subscribed to receive notifications sent by the MEPM when an application instance LCM operation occurrence changes its state to "PROCESSING":

7) The MEPM sends to the MEO an application instance lifecycle management operation occurrence notification to indicate that the application instance LCM operation occurrence enters the "PROCESSING" state.

8) The MEO returns a "204 No Content" response with an empty message content.

Whilst the <<Operation>> is ongoing:

9) The MEO may query the "application instance LCM operation occurrence" resource to obtain information about the ongoing operation by sending a GET request to the resource that represents the application instance LCM operation occurrence.

10) In the response to that optional request, the MEPM returns to the MEO information of the operation, including the "PROCESSING" operation state in the "AppLcmOpOcc" resource representation.

11) The MEPM finishes the <<Operation>>.

If the MEO has subscribed to receive notifications sent by the MEPM when an application instance LCM operation occurrence changes its state to "COMPLETED".

12) The MEPM sends an application instance lifecycle management operation occurrence notification to indicate the completion of the lifecycle management operation occurrence with the success state "COMPLETED".

13) The MEO returns a "204 No Content" response with an empty message content.

Alternatively:

14) The MEO may query the "application instance LCM operation occurrence" resource to obtain information about the ongoing operation by sending a GET request to the resource that represents the application instance LCM operation occurrence.

15) In the response to that request, the MEPM returns to the MEO information of the operation, including the "COMPLETED" operation state in the "AppLcmOpOcc" resource representation.

In clean-up:

16) The MEPM deletes the "application instance LCM operation occurrence" resource for this <<Task>>.

5.4.3 Synchronous granting procedure

Figure 5.4.3-1 illustrates the synchronous granting procedure for an application lifecycle operation.

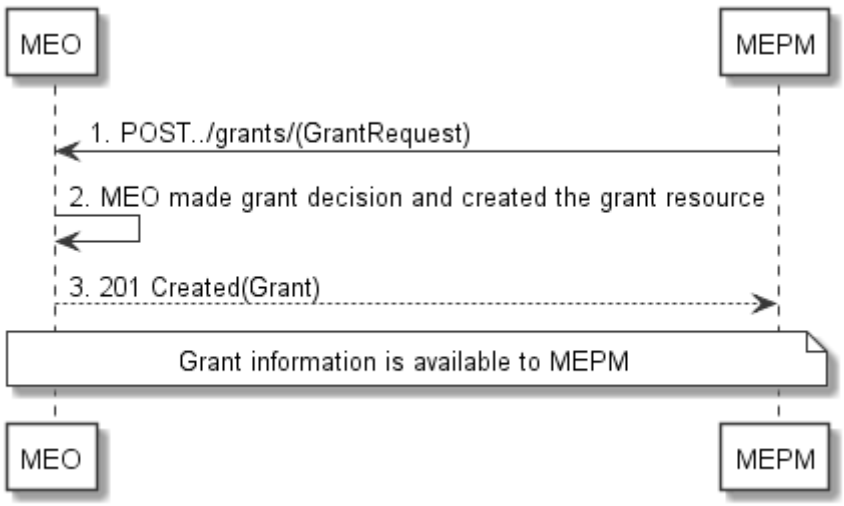

Figure 5.4.3-1: Synchronized granting procedure for application instance LCM operation

Granting with synchronous response, as illustrated in the figure 5.4.3-1, consists of the following steps:

1) The MEPM sends a POST request to the "Grants" resource with a "GrantRequest" data structure in the message content.

2) The MEO makes the granting decision, and creates a new "Individual grant" resource.

3) The MEO returns to the MEPM a "201 Created" response with a "Grant" data structure in the message content and a "Location" HTTP header that points to the new "Individual grant" resource.

5.4.4 Asynchronous granting procedure

Figure 5.4.4-1 illustrates the asynchronous granting procedure for an application lifecycle operation.

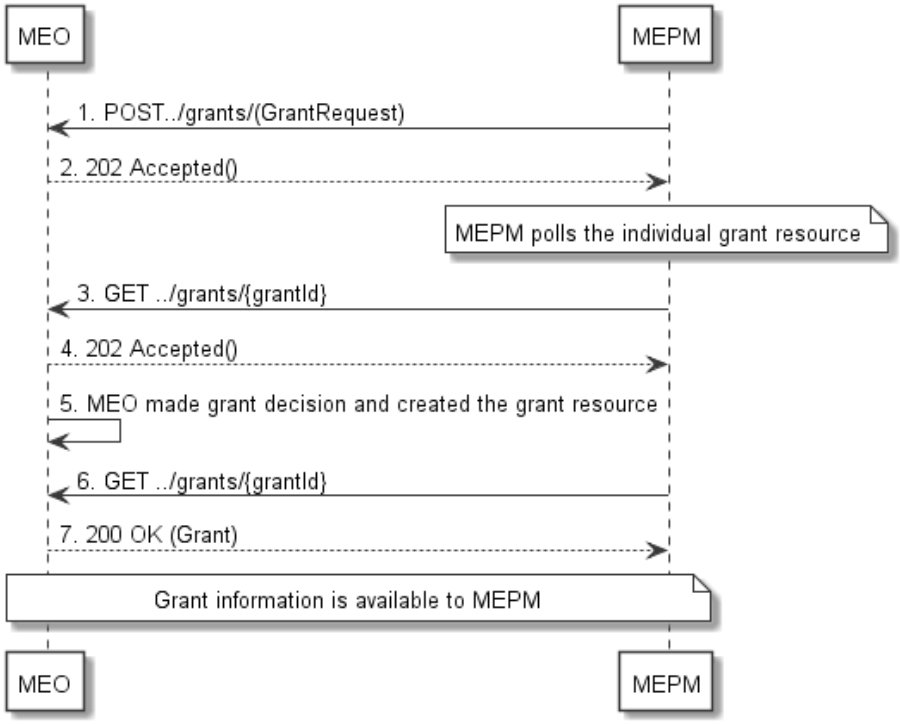

Figure 5.4.4-1: Asynchronous granting procedure for application instance LCM operation

Granting with asynchronous response, as illustrated in figure 5.4.4-1, consists of the following steps:

1) The MEPM sends a POST request to the "Grants" resource with a "GrantRequest" data structure in the message content.

2) The MEO sends an Accepted response with return code 202 that contains the resource URI of the to-be-created "Individual grant" resource in the "Location" header.

3) MEPM may query the status of granting process via the GET method.

4) MEO sends an accepted response with return code 202 to indicate that the granting decision has not been made.

5) The MEO makes the granting decision, and creates a new "Individual grant" resource.

6) The MEPM may query the status of granting process via GET method.

7) This time the MEO responds with the return code 200 to confirm the grant.

5.4.5 Application LCM Operation Intervention

This clause describes how a client can intervene with an application LCM operation. For instance a cancel intervention can be initiated by the MEO if, for example, a MEPM is deemed to be making insufficient progress with an ongoing application instantiation LCM operation. Specifically three client invoked operations are supported:

- Cancel: to stop the execution of an ongoing application LCM operation, which is in PROCESSING state, before it can be successfully completed. Once stopped, the application LCM operation transits into the FAILED_TEMP state which allows root cause analysis, possible fixing of the root cause, followed by retrying, or finally failing of the operation.
- Retry: to retry an application LCM operation that has entered FAILED_TEMP state, either through an explicit cancel, or because automatic retry attempts by the server have failed.
- Fail: to permanently fail an application LCM operation that has entered FAILED_TEMP state, since it is deemed that the operation cannot be recovered.

The state and state transitions supported by the MEO/MEPM are illustrated in figure 5.4.5-1, noting that STARTING state and the Grant operation are only applicable to Mm3. Transitions labelled with underlined text represent error handling operations; other transitions represent conditions.

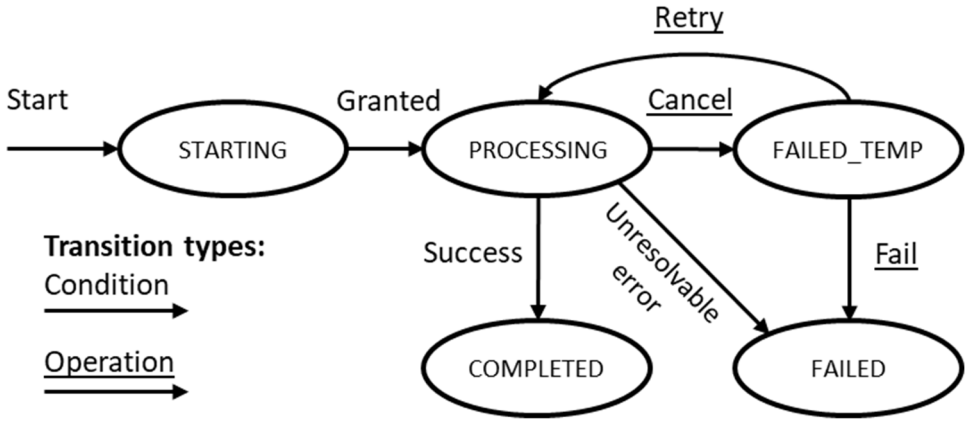

Figure 5.4.5-1: States of an application lifecycle management operation occurrence

Table 5.4.5-1 shows parameterization associated to above application instance lifecycle management operation interventions.

Table 5.4.5-1: Parameterization for application instance LCM operation interventions

Operation	Precondition	Task	RequestStructure	Postcondition
Cancel	Application LCM operation in PROCESSING state.	cancel	CancelMode	Application LCM operation in FAILED_TEMP state.
Retry	Application LCM operation in FAILED_TEMP state.	retry	n/a	If successful, application LCM operation transitions to COMPLETED state, otherwise it remains in FAILED_TEMP state.
Fail	Application LCM operation in FAILED state.	fail	n/a	Application instance in FAILED state.

The cancel application LCM operation flow is illustrated in figure 5.4.5-2.

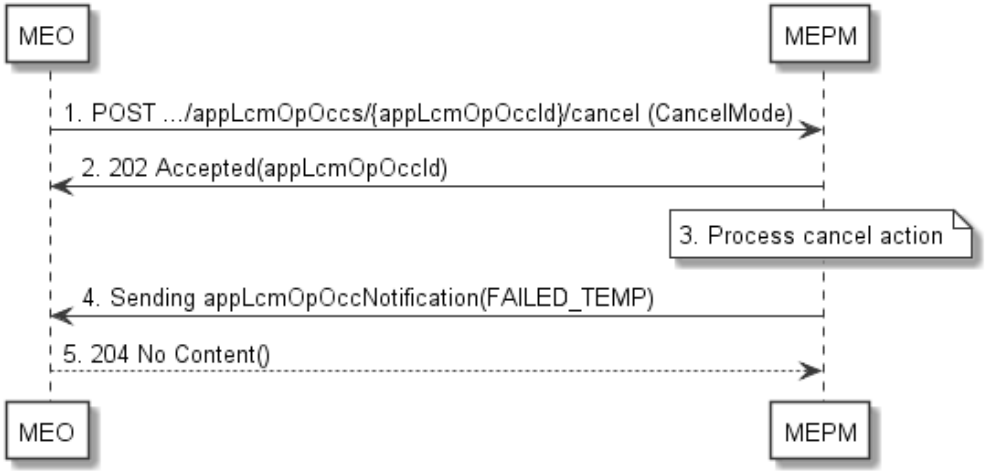

Figure 5.4.5-2: Flow of cancelling an application LCM operation in "PROCESSING" state

The application LCM operation cancellation process for an operation in "PROCESSING" state, as illustrated in figure 5.4.5-2, consists of the following steps in the case of the MEO acting as client and the MEPM as the server.

1) The MEO sends a cancel request to the "application instance LCM operation occurrence" resource at the MEPM. The request includes indication of whether forceful or graceful cancellation is to be performed.

2) The MEPM returns a "202 Accepted" response.

3) The MEPM processes the cancellation request either gracefully or forcefully, which may take some time.

4) Once the cancellation has completed, the MEPM sends an application instance lifecycle management operation occurrence notification to indicate that the application instance LCM operation occurrence has entered the FAILED_TEMP state.

5) The MEPM returns a "204 No Content" response.

The retry application LCM operation flow is illustrated in figure 5.4.5-3.

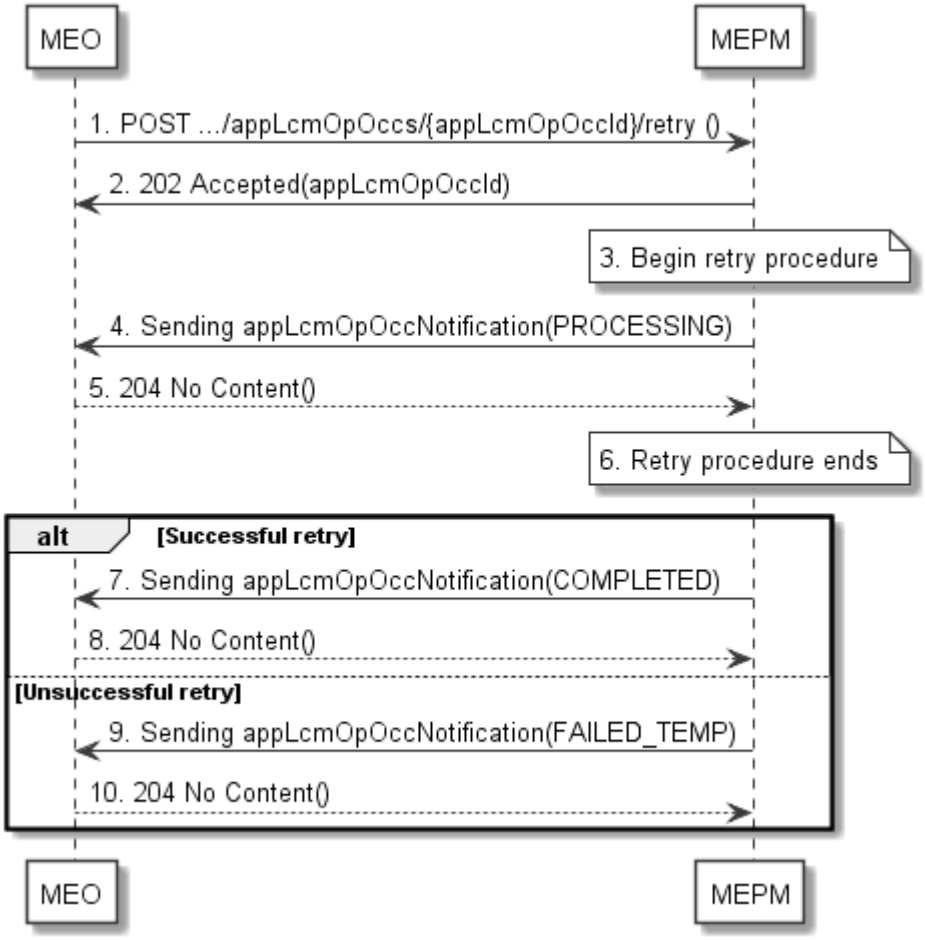

Figure 5.4.5-3: Flow of retrying an application LCM operation in "FAILED_TEMP" state

The application LCM operation retry process for an operation in "FAILED_TEMP" state, as illustrated in figure 5.4.5-3, consists of the following steps in the case of the MEO acting as client and the MEPM as the server:

1) The MEO sends a retry request to the "application instance LCM operation occurrence" resource at the MEPM.

2) The MEPM returns a "202 Accepted" response.

3) The MEPM processes the retry request, which may take some time.
When performed over Mm3, or Mm3*, retry shall operate with the bounds of the Grant for the LCM operation.

4) The MEPM sends an application instance lifecycle management operation occurrence notification to indicate that the application instance LCM operation occurrence has entered the PROCESSING state.

5) The MEO returns a "204 No Content" response.

6) The MEPM finishes the retry procedure.

7) On successful retry, the MEPM sends an application LCM operation occurrence notification to indicate successful completion of the operation.

8) The MEO returns a "204 No Content" response.

9) On unsuccessful retry, MEPM sends an application LCM operation occurrence notification to indicate an intermediate error (retry failed) of the operation.

10) The MEO returns a "204 No Content" response.

The fail application LCM operation flow is illustrated in figure 5.4.5-4.

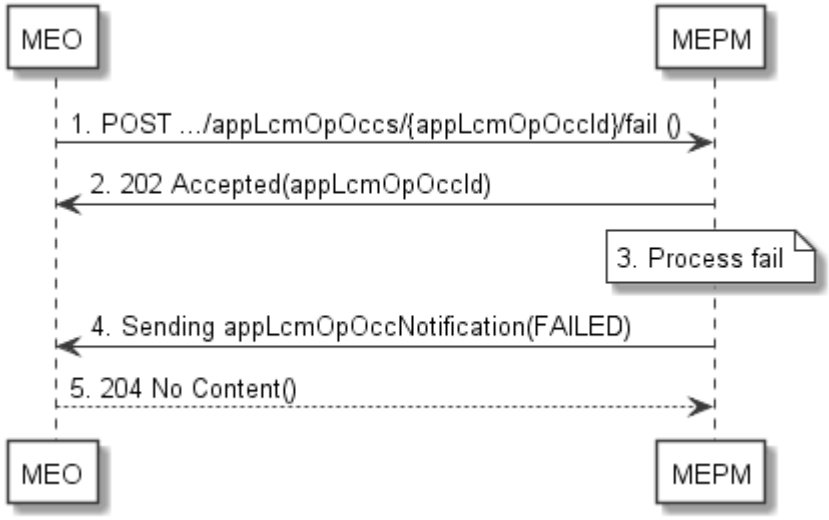

Figure 5.4.5-4: Flow of failing an application LCM operation in "FAILED_TEMP" state

The application LCM operation failure process for an operation in "FAILED_TEMP" state, as illustrated in figure 5.4.5-4, consists of the following steps in the case of the MEO acting as client and the MEPM as the server:

1) The MEO sends a fail to the "application instance LCM operation occurrence" resource at the MEPM.

2) The MEPM returns a "202 Accepted" response.

3) The MEPM processes the fail request and marks the operation as failed.

4) The MEPM sends an application instance lifecycle management operation occurrence notification to indicate the final failure of the application instance LCM operation occurrence, which enters FAILED state.

5) The MEPM returns a "204 No Content" response.

5.5 Coordination operations

5.5.1 Coordinate LCM operation

This message flow is used to request coordination of an LCM operation in the MEC system. The detailed description of the flow is in figure 5.5.1-1.

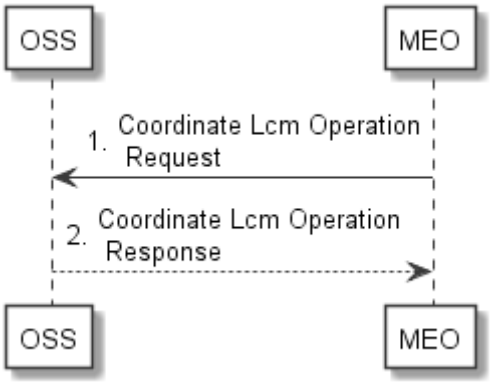

Figure 5.5.1-1: Coordinate LCM operation flow

1) The MEO sends the CoordinateLcmOperation request to the OSS identifying an application instance and the LCM operation that is requested.

2) The OSS sends the CoordinateLcmOperation response with an indication of whether or not to resume the LCM operation execution.

As an example of the use of the coordination of an LCM operation in the MEC system, figure 5.5.1-2 provides the message flow to select a set of application instances that can be terminated in order to allow the instantiation of a new, typically higher-priority application during a resource shortage.

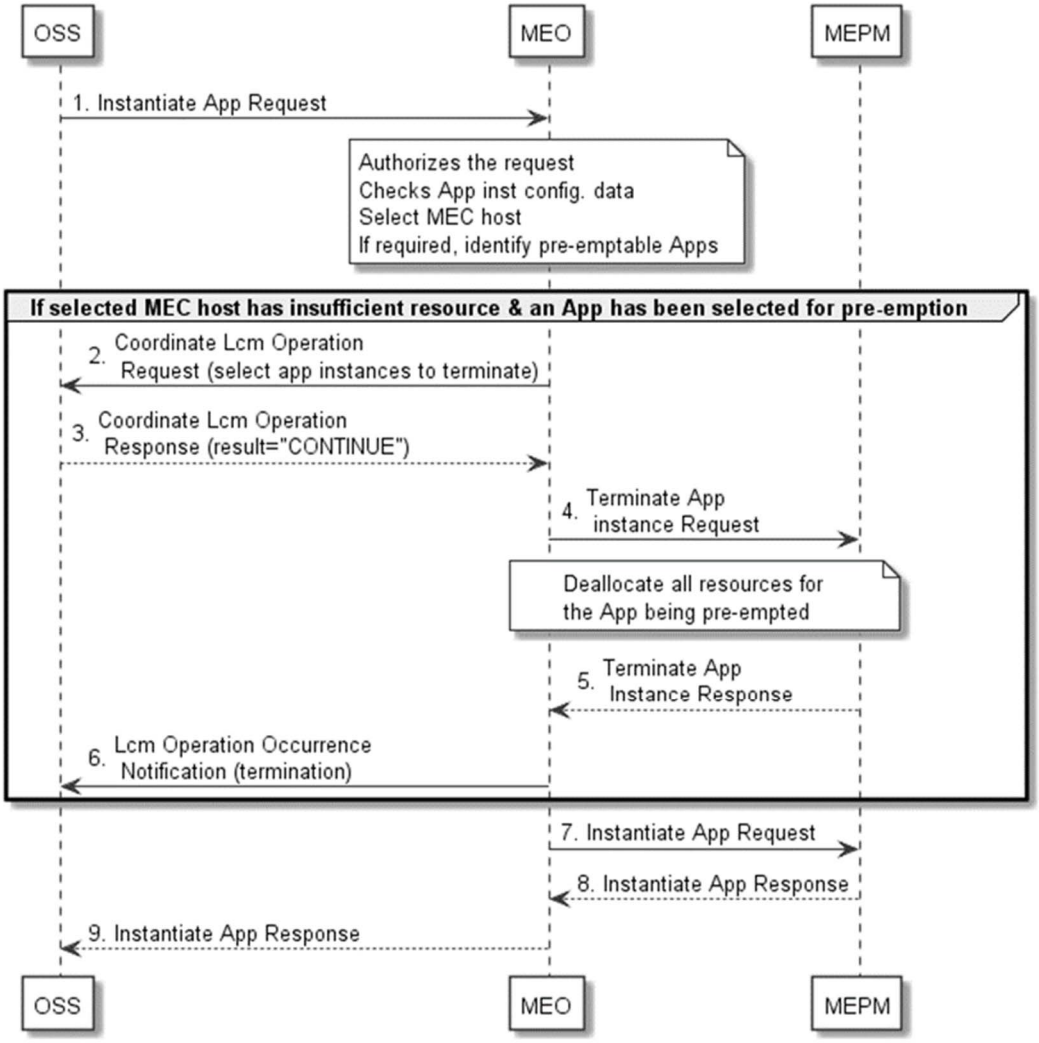

Figure 5.5.1-2: Application instantiation flow with pre-emption

1) The OSS sends an instantiate application request to the MEC Orchestrator, reference clause 5.3.1 step 1, optionally including pre-emption information, reference clause 5.3.1 step 2.

2) MEC Orchestrator processes the request, reference clause 5.3.1 step 2. Then if the selected MEC host has insufficient resources and one or more applications have been selected for pre-emption from the candidates in the instantiate application request (if provided), or from candidates deemed suitable by the MEO, the MEO sends to the OSS coordination endpoint a CoordinateLcmOperation request that solicits the OSS to select the application instance(s) for termination.

NOTE: Coordination endpoint information optionally provided in the instantiation request overrides pre-provisioned coordination endpoint information.

3) The OSS sends the CoordinateLcmOperation response with an indication to continue the termination.

4) The MEC Orchestrator sends a terminate application instance request to the MEC Platform Manager that manages the MEC application instance to be terminated, reference clause 5.3.2 step 2.

5) The MEC Platform Manager sends a terminate application instance response to the MEC Orchestrator having completed steps 3 to 7 of clause 5.3.2.

6) The MEC Orchestrator notifies the OSS of the successful termination.

7) The MEC Orchestrator sends an instantiate application request to the MEC Platform Manager.

8) The MEC Platform Manager sends an instantiate application response to the MEC Orchestrator having completed steps 3 to 7 of clause 5.3.1.

9) The MEC Orchestrator sends an instantiate application response to the OSS and performs the additional steps indicated in step 9 of clause 5.3.1.

5.6 Application package management for NFV

5.6.1 General

Message flows of application package management for NFV are used to make application package available to the NFV, delete the application package from the NFV, query information of one or more application packages on-boarded in the NFV, or fetch an on-boarded application package from the NFV. The series of message flows include:

- On-board application package to NFV
- Query application package information in NFV
- Disable application package in NFV
- Enable application package in NFV
- Fetch application package in NFV

The message flows for these operations are all presented in two variants:

- Trigger from OSS to MEAO, where MEAO calls NFVO via Mv1 reference point (e.g. on-boarding indirectly from OSS through MEAO to NFVO).
- Trigger from OSS to NFVO, where MEAO receives a notification from NFVO via Mv1 reference point (e.g. on-boarding from OSS to NFVO directly).

5.6.2 On-board application package to NFV

The message flow of on-board application package to NFV is used to make MEC application package available to the NFV, which is executed before an application is instantiated. The actual time to execute this message flow is dependent on implementation. On-board application package to NFV utilizes the API defined in the specifications of ETSI GS NFV-IFA 013 [15] and ETSI GS NFV-SOL 005 [i.7] to on-board an application package as a VNF package to the NFV.

The detailed description of the message flows are in figures 5.6.2-1 and 5.6.2-2.

Figure 5.6.2-1 illustrates the message flow of on-boarding application package from OSS through MEAO to NFVO.

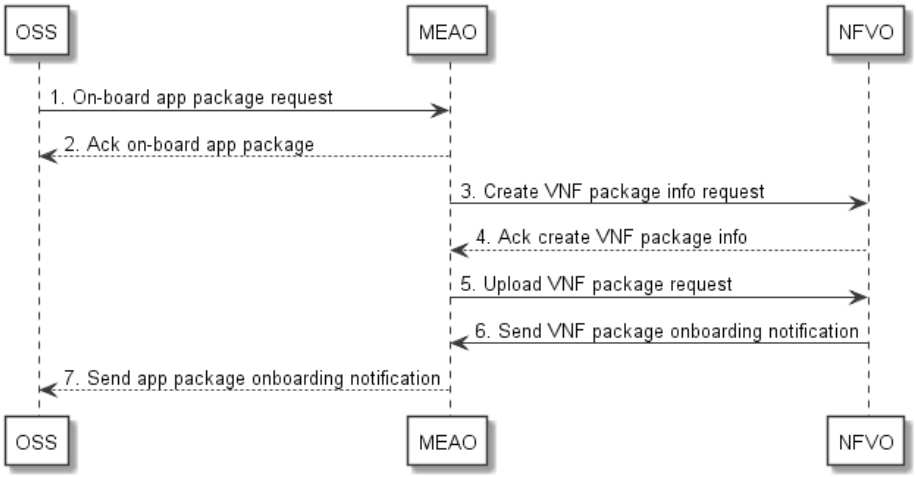

Figure 5.6.2-1: On-board application package through MEAO

1) In order to on-board an application package to NFV, the OSS sends an on-board application package request to the MEAO using the API defined for OSS to on-board an application package on Mm1 reference point.

2) After receiving the on-boarding application from the OSS, the MEAO sends an acknowledgement to the OSS and starts the process of on-boarding application package to NFV.

3) The MEAO may create an appPkgId and sends a create VNF package info request to the NFVO.

4) The NFVO creates the VNF package info, and sends an acknowledgement to the create VNF package info request back to the MEAO with a unique VNF package ID.

5) In order for NFVO to understand application package as a VNF package, either a complex conversion including signature would be necessary or the formats need to be identical. It is out of scope of the present document how this is achieved. The MEAO sends an upload VNF package request to the NFVO to on-board the package according to the formats specified in ETSI GS NFV-SOL 004 [18].

6) The NFVO checks the VNF package and related status information. The NFVO sends the VNF package onboarding notification to the MEAO if it subscribes to receiving such notification. Once the upload VNF software image process completes, the NFVO sends to the MEAO another VNF package onboarding notification to indicate the completion of uploading procedure.

   The MEAO maps the vnfPkgId to the appPkgId. It is possible for MEAO to use the vnfPkgId as the appPkgId directly.

   States of the onboarded application package are volatile values and may need to change due to LCM operations. The MEAO may maintain the state values of an onboarded package. The MEAO may optionally cache VnfPkgInfo and maps the states of the VnfPkgInfo to the corresponding state attributes of AppPkgInfo. It may update the cached AppPkgInfo accordingly.

   The MEAO maps VnfPkgInfo to the AppPkgInfo and may update and cache AppPkgInfo locally.

7) The MEAO sends to the OSS the application package onboarding notification with the onboarding status information.

Figure 5.6.2-2 illustrates the message flow of on-boarding application package as VNF package from OSS to NFVO directly.

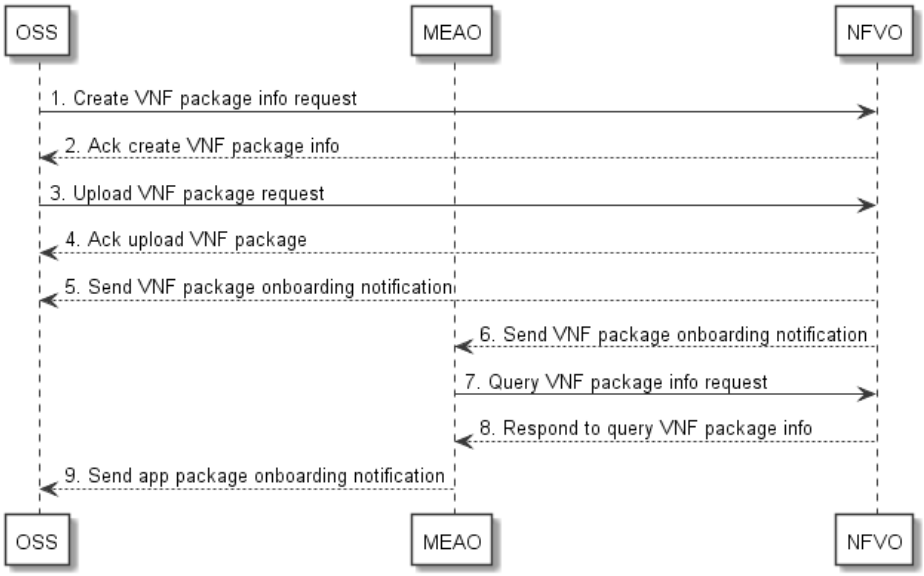

Figure 5.6.2-2: On-board application package as VNF package to NFVO directly

1) The OSS sends a create VNF package info request to the NFVO on Os-Ma-nfvo reference point.

2) After the VNF package info is created, the NFVO sends an acknowledgement of the create VNF package info request to the OSS.

3) The OSS sends an upload VNF package request to the NFVO to on-board an application package to the NFV, in which the MEC application package is attached as a VNF package.

4) The NFVO acknowledges to the OSS the upload VNF package request on Os-Ma-nfvo reference point and prepares uploading NFV package.

5) The NFVO sends the VNF package onboarding notification to the OSS on Os-Ma-nfvo reference point.

6) The NFVO sends the VNF package onboarding notification to the MEAO on Mv1 reference point if MEAO subscribes to receiving such notification.

   The NFVO checks the VNF package and other attributes, and then upload VNF package. The NFVO allocates a unique VNF package ID for the on-boarded VNF package and related status information.

   Once the upload VNF software image process completes, the NFVO sends to the MEAO another VNF package onboarding notification to indicate the completion of uploading procedure.

7) The MEAO may send a query VNF package information request to NFVO to acquire more information about the application package onboarded as VNF package.

8) The NFVO sends a response to the query VNF package information request.

9) The MEAO reads the VNF package information including VNF package states, and other attributes. The MEAO may query the NFVO for MEC-related non-MANO artifacts attached to the VNF package.

   The MEAO then maps them to the AppPkgInfo and AppD. The MEAO may update and cache AppPkgInfo and AppD locally.

   The MEAO sends to the OSS the application package onboarding notification with the onboarding status information.

5.6.3 Query application package information in NFV

Query application package allows returning the information contained in an application package, such as the location of the application descriptor (i.e. a vnfd in the vnf package), the state information of application package onboarded as VNF package, etc.

The detailed description of message flows are shown in Figures 5.6.3-1 and 5.6.3-2.

Figure 5.6.3-1 illustrates the message flow of query application package information by the OSS through MEAO to NFVO. This message flow can be used for OSS to obtain the application package information, such as onboarding state, usage state, etc. of the application package being onboarded as a VNF package. It can also be used for OSS to acquire the application requirements, traffic redirection rules and DNS rules in the AppD that it needs to configure the MEC platform to run the application as a VNF instance.

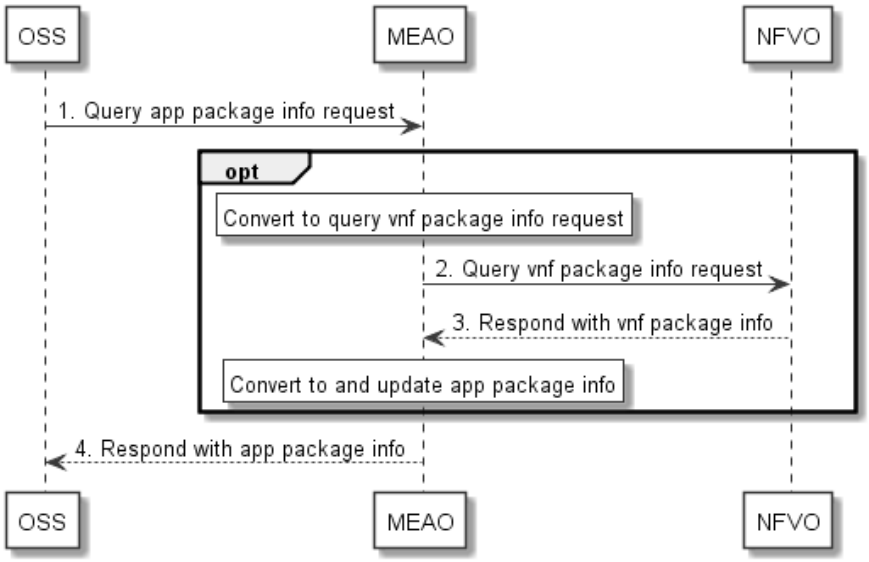

Figure 5.6.3-1: Query application package information through MEAO

1) The OSS sends the query application package information request, which may include a filter, to the MEAO via the APM API of Mm1 reference point to acquire the application package information including AppD.

Optionally, if the MEAO does not cache the latest application package information, the following steps are executed:

2) The MEAO converts this query application package information request to the query VNF package information request format, and then sends a query VNF package information request to the NFVO on the API of Mv1 reference point to query the information of an application package on-boarded as a VNF package. This API is defined in the specifications of ETSI GS NFV-IFA 013 [15] and ETSI GS NFV-SOL 005 [i.7]. Otherwise, the MEAO responds to the OSS with the cached application package information.

3) The NFVO authorizes the request, and returns to the MEAO the information related to the on-boarded VNF package(s) that matches the filter in the query VNF package information request.

   If the MEAO receives the VNF package information in the response to query VNF package information request, it converts the VNF package information according to the AppPkgInfo format, updates and caches the application package information.

Finally, the following step is executed:

4) The MEAO responds to the OSS with the application package information.

Figure 5.6.3-2 illustrates the message flow of query application package (as VNF package) information directly through NFVO on the API over the Os-Ma-nfvo reference point. Refer to ETSI GS NFV-IFA 013 [**15**] for query VNF package information operation in detail.

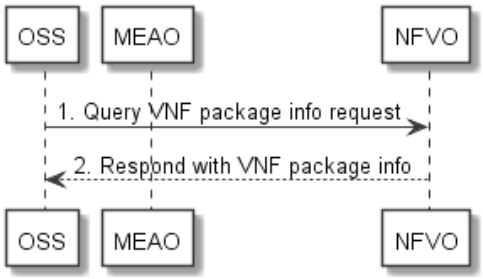

Figure 5.6.3-2: Query application package information in VNF package through NFVO directly

5.6.4 Disable application package in NFV

Disabling application package refers to the process of marking an application package on-boarded as a VNF package to be disabled in the NFVO, so that it is not possibly used for new VNF instance creation. Disable application package in NFV utilizes the API defined in the specifications of ETSI GS NFV-IFA 013 [15] and ETSI GS NFV-SOL 005 [i.7] to disable an application package on-boarded as a VNF package.

The detailed description of message flows are shown in Figures 5.6.4-1 and 5.6.4-2.

Figure 5.6.4-1 illustrates the message flow of disable application package on-boarded as VNF package through the MEAO.

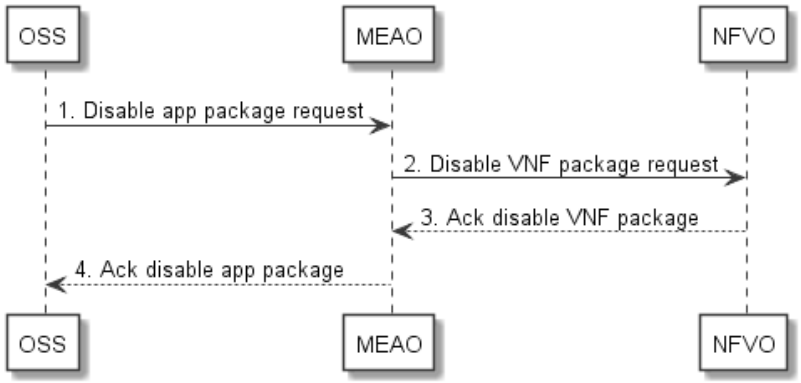

Figure 5.6.4-1: Disable application package through MEAO

1) The OSS sends a disable application package request to the MEAO.

2) The MEAO sends the disable VNF package request to the NFVO.

   The NFVO processes the request of disable VNF package, checking if the VNF package exists and is enabled. If the VNF package is enabled, the NFVO marks this VNF package as disabled. The NFVO notifies the subscribers with VnfPackageStateChangeNotification about the state change of this VNF package.

3) The NFVO sends to the MEAO an acknowledgement to the disable VNF package request.

4) After receiving the acknowledgement, the MEAO updates the AppPkgInfo, and sends to the OSS an acknowledgement to the disable application package request. The MEAO notifies the subscribers with AppPackageStateChangeNotification about the state change of this application package.

Figure 5.6.4-2 illustrates the message flow of disable application package onboarded as VNF package through the NFVO directly.

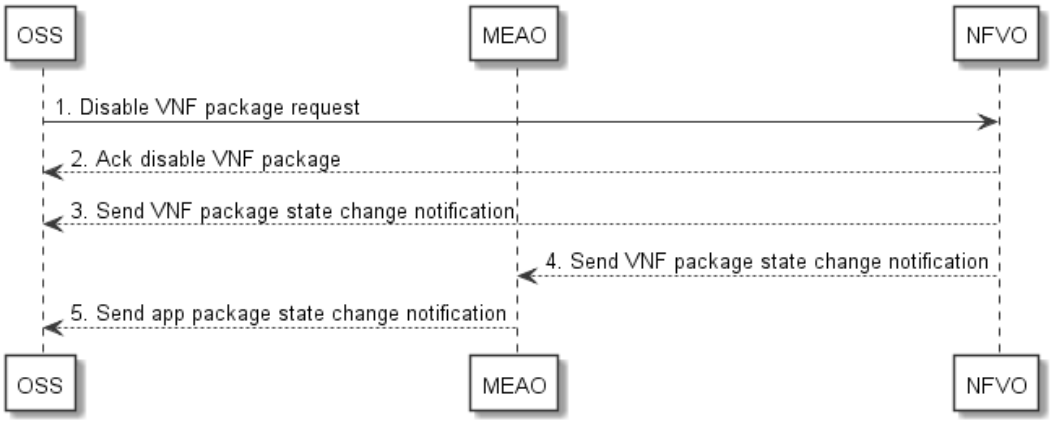

Figure 5.6.4-2: Disable application package through NFVO directly

1) The OSS sends a disable VNF package request to the NFVO.

2) The NFVO processes the request of disable VNF package, checking if the VNF package exists and is enabled, and then marking this VNF package as disabled. The NFVO sends to the OSS an acknowledgement to the disable VNF package request. The NFVO notifies the subscribers about the state change of this VNF package.

3) The NFVO sends to the OSS the VnfPackageStateChangeNotification for the state change of the VNF package.

4) The NFVO sends to the MEAO the VnfPackageStateChangeNotification for the state change of the VNF package.

5) After receiving this notification, the MEAO sends to the OSS a notification about the application package state change.

5.6.5 Enable application package in NFV

Enabling an application package refers to the process of marking an application package on-boarded as a VNF package to be enabled in the NFVO, so that it may be used for instance creation again. Enable application package in NFV utilizes the API defined in the specifications of ETSI GS NFV-IFA 013 [15] and ETSI GS NFV-SOL 005 [i.7] to enable an application package on-boarded as a VNF package.

The detailed description of message flows are shown in Figures 5.6.5-1 and 5.6.5-2.

Figure 5.6.5-1 illustrates the message flow of enable application package on-boarded as VNF package through the MEAO.

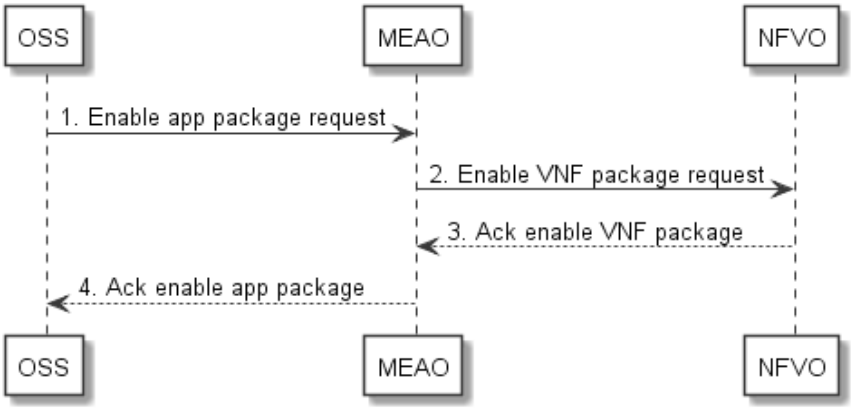

Figure 5.6.5-1: Enable application package through MEAO

1) The OSS sends an enable application package request to the MEAO.

2) The MEAO sends the enable VNF package request to the NFVO.

   The NFVO processes the request of enable VNF package, checking if the VNF package exists, and is disabled. If the VNF package is disabled, the NFVO marks the VNF package as enabled. The NFVO notifies the subscribers to VnfPackageStateChangeNotification about the state change of this VNF package.

3) The NFVO sends to the MEAO an acknowledgement to the enable VNF package request.

4) After receiving this acknowledgement, the MEAO updates the AppPkgInfo, and sends to the OSS an acknowledgement to the enable application package request. The MEAO notifies the subscribers to AppPackageStateChangeNotification about the state change of this application package.

Figure 5.6.5-2 illustrates the message flow of enable application package onboarded as VNF package through the NFVO directly.

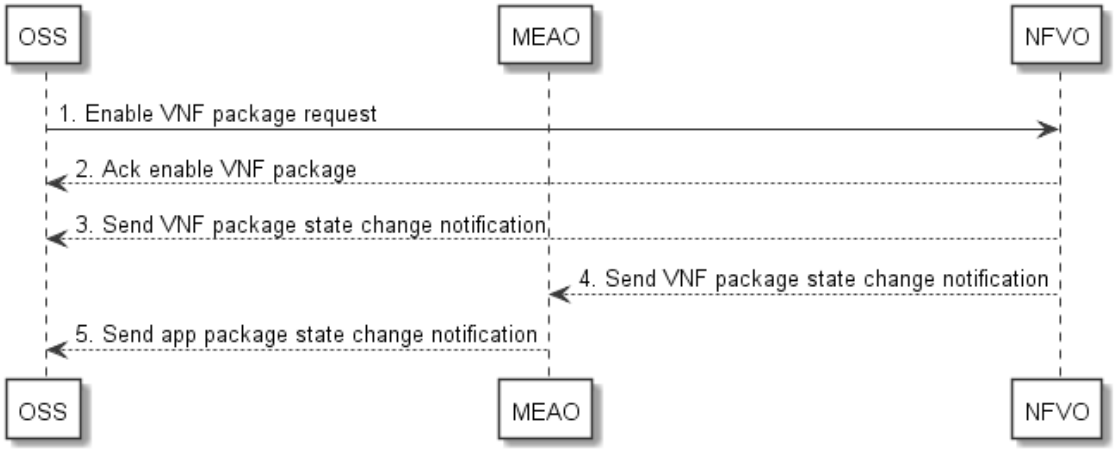

Figure 5.6.5-2: Enable application package through NFVO directly

1) The OSS sends an enable VNF package request to the NFVO.

2) The NFVO processes the request of enable VNF package, checking if the VNF package exists and is disabled, and then marking this VNF package as enabled. The NFVO sends to the OSS an acknowledgement to the enable VNF package request. The NFVO notifies the subscribers about the state change of this VNF package.

3) The NFVO sends to the OSS the VnfPackageStateChangeNotification for the state change of the VNF package.

4) The NFVO sends to the MEAO the VnfPackageStateChangeNotification for the state change of the VNF package.

5) After receiving this notification, the MEAO sends to the OSS a notification about the application package state change.

5.6.6 Delete application package in NFV

Delete application package in NFV refers to the process of removing an application package on-boarded as a VNF package in NFV. Delete application package in NFV utilizes the API defined in the specifications of ETSI GS NFV-IFA 013 [15] and ETSI GS NFV-SOL 005 [i.7] to delete an application package on-boarded as a VNF package to the NFV.

The detailed description of message flows are shown in Figures 5.6.6-1 and 5.6.6-2.

Figure 5.6.6-1 illustrates the message flow of delete application package initiated by OSS through MEAO.

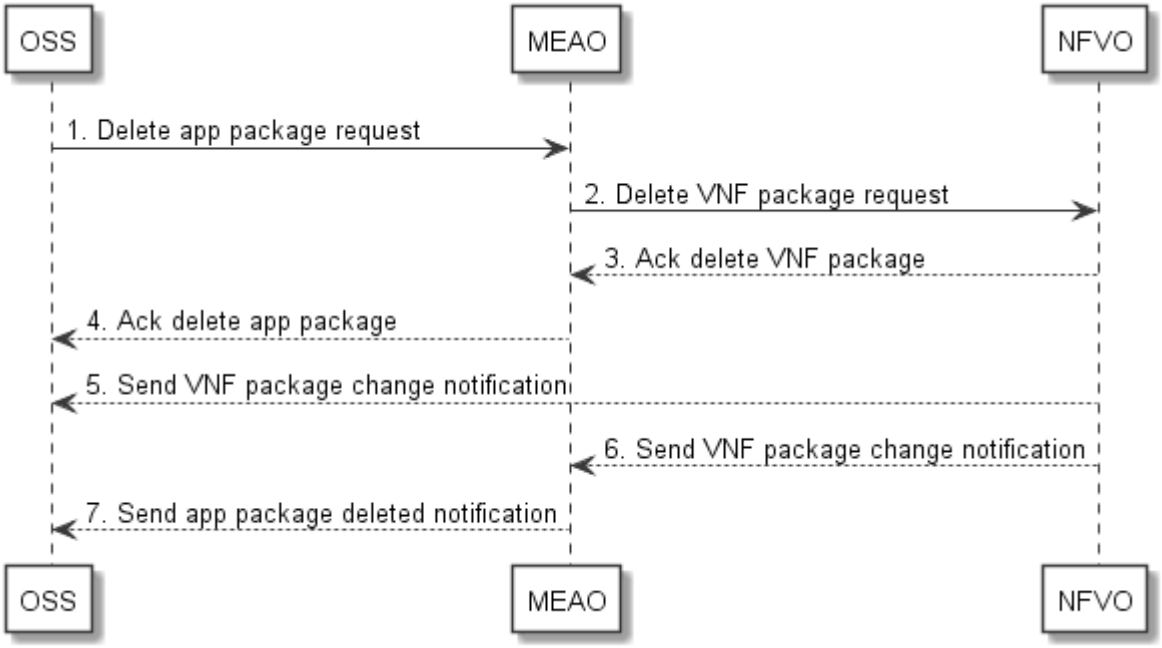

Figure 5.6.6-1: Delete application package through MEAO

1) The OSS sends a delete application package request to the MEAO.

2) After receiving the delete application package request, the MEAO checks whether the application package is onboarded, disabled and in use. If the application package is not disabled or is in use, the MEAO returns to the OSS an error for this operation.
If the application package is disabled and is not in use, the MEAO sends a delete VNF package request to the NFVO to delete the application package on-boarded as a VNF package.

3) The NFVO processes this delete request.

The NFVO checks whether the VNF package is on-boarded, disabled and in use. If the VNF package is not disabled or is in use, the NFVO returns to the MEAO an error for this operation. If the VNF package is disabled and is not in use, the NFVO sends to the MEAO an acknowledgement to the request and removes the VNF package from the NFV.

4) After receiving the acknowledgement, the MEAO sends to the OSS an acknowledgement to the delete application package request.

5) After the VNF package has been deleted, the NFVO sends to the OSS a VnfPackageChangeNotification about the state change of this VNF package.

6) The NFVO sends to the MEAO a VnfPackageChangeNotification about the state change of this VNF package.

7) The MEAO deletes the AppPkgInfo associated to this application package, and sends to the OSS an application package deleted notification.

Figure 5.6.6-2 illustrates the message flow of delete application package onboarded as VNF package through the NFVO directly.

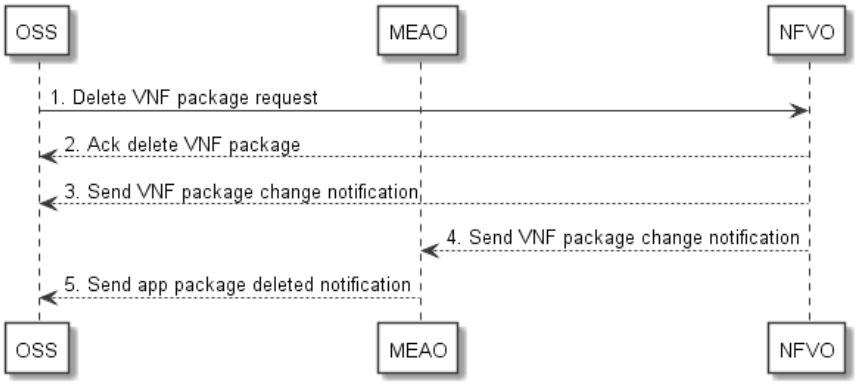

Figure 5.6.6-2: Delete application package through NFVO

1) The OSS sends a delete VNF package request to the NFVO.

2) The NFVO sends to the OSS an acknowledgement to the delete VNF package request. Meanwhile the NFVO processes this delete request.

   The NFVO checks whether the VNF package is on-boarded, disabled and in use. If the VNF package is not disabled or is in use, this operation will return an error. If the VNF package is disabled and is not in use, the NFVO sends to the OSS an acknowledgement to the request and removes the VNF package from the NFV.

3) After the VNF package has been deleted, the NFVO sends to the OSS a VnfPackageChangeNotification about the state change of this VNF package.

4) The NFVO sends to the MEAO a VnfPackageChangeNotification about the state change of this VNF package.

5) The MEAO may delete the AppPkgInfo for this application package if it exists, and sends to the OSS an application package deleted notification.

5.6.7 Fetch on-boarded application package from NFV

Fetch on-boarded application package in NFV to the MEAO allows retrieving an application package, or selected files contained in a package on-boarded as a NFV package in NFVO. This procedure utilizes the API defined in the specifications of ETSI GS NFV-IFA 013 [15] and ETSI GS NFV-SOL 005 [i.7] to fetch an application package on-boarded as a VNF package.

The detailed description of the flow is in figure 5.6.7-1.

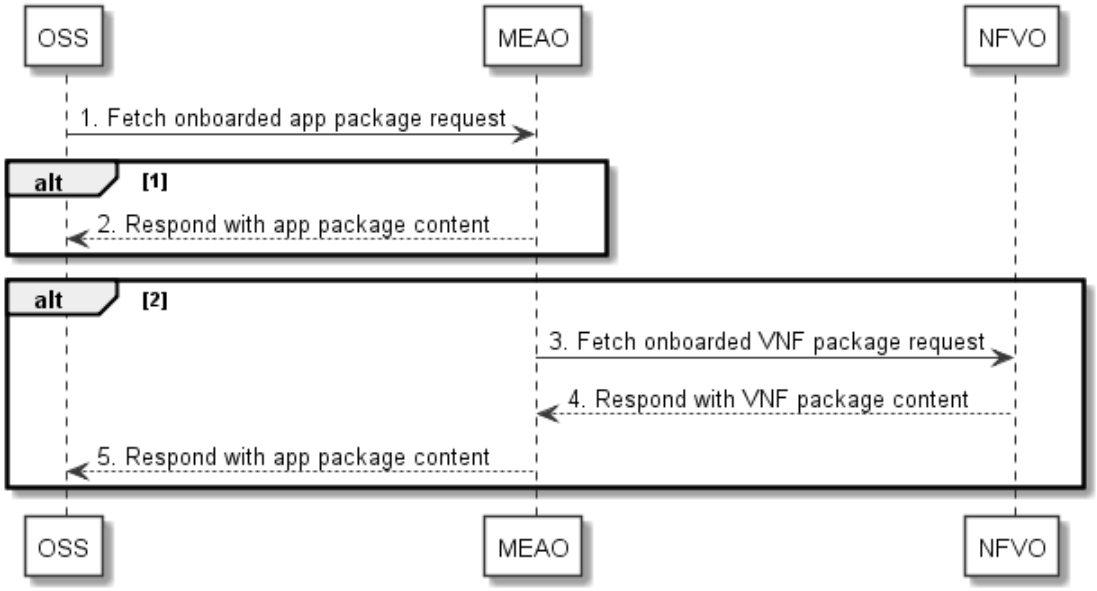

Figure 5.6.7-1: Fetch on-boarded application package through MEAO

1) The OSS sends to the MEAO a fetch on-boarded application package request.

If the application package content is cached locally, the following step is executed:

2) After receiving a fetch on-boarded application package request, the MEAO returns to the OSS the application package content which is cached locally.

Alternatively, if the application content is not cached locally, the following steps are executed:

3) The MEAO sends a fetch on-boarded VNF package request to the NFVO to fetch the content of a VNF package.

4) The NFVO authorizes the request, and returns the VNF package to the MEAO.

5) Possibly necessary conversion steps and signing of the package are out of scope for the present document. The MEAO returns the application package to the OSS.

Figure 5.6.7-2 illustrates the message flow of fetching an onboarded VNF package of MEC application directly from NFVO on the API over the Os-Ma-nfvo reference point. This procedure is only between the OSS and the NFVO without involving the MEAO. Refer to ETSI GS NFV-IFA 013 [15] and ETSI GS NFV-SOL 005 [i.7] details of the fetch VNF package operation and its steps.

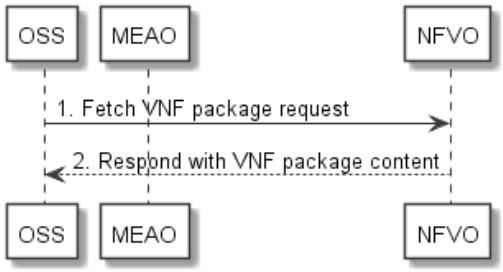

Figure 5.6.7-2: Fetch on-boarded VNF package of MEC application through NFVO directly

5.7 Application lifecycle, rule and requirement management for NFV

5.7.1 General

Message flows of application lifecycle, rule and requirement management for MEC in NFV are used to instantiate application as a VNF, terminate application instance as a VNF, operate application instance as a VNF and configure an application instance as a VNF in NFV. The series of message flows include:

- Instantiate application in NFV.
- Terminate application in NFV.
- Operate application in NFV.
- Configure application in NFV.

The message flows for these operations are all presented in two variants:

- Trigger from OSS to MEAO, where MEAO calls NFVO via Mv1 reference point (e.g. instantiate indirectly from OSS through MEAO to NFVO).
- Trigger from OSS to NFVO, where MEAO receives a notification from NFVO via Mv1 reference point (e.g. instantiate from OSS to NFVO directly).

5.7.2 Instantiate application in NFV

The message flows of instantiate application in NFV are used to instantiate an application instance as a VNF via NFV APIs defined in the specifications of ETSI GS NFV-SOL 005 [i.7]. Refer to ETSI GS NFV-IFA 013 [15], ETSI GS NFV-IFA 007 [i.3] and ETSI GS NFV-SOL 016 [i.9] for details.

Figure 5.7.2-1 illustrates a message flow of instantiating an application as a VNF through MEAO.

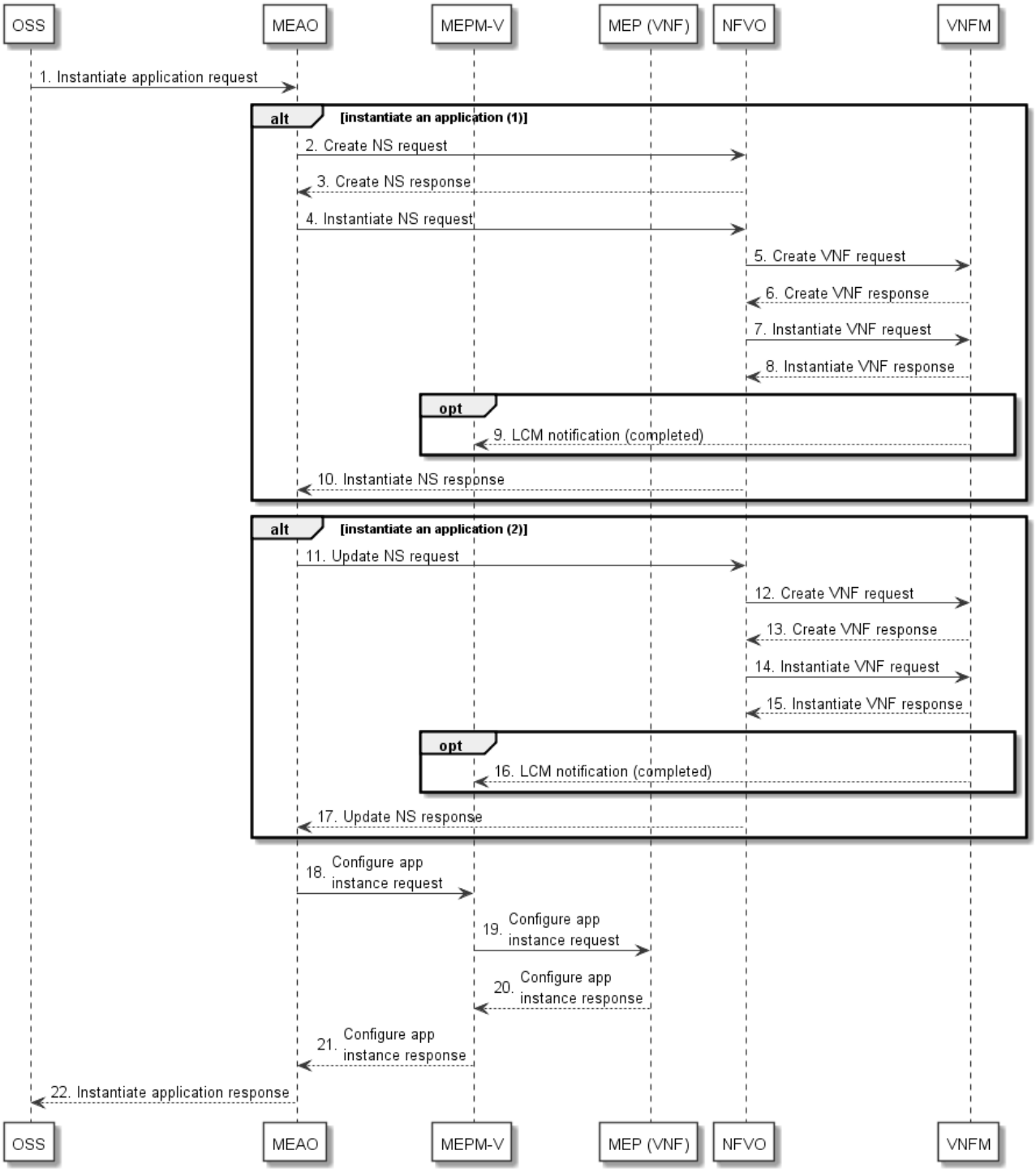

Figure 5.7.2-1: Instantiate application in NFV through MEAO

1) The OSS sends an instantiate application request message to the MEAO for instantiating an application instance on Mm1 after the application package has been onboarded as a VNF package to the NFVO.

The MEAO checks whether the application is instantiated or not. If a NS has not been created, the following steps are executed:

NOTE 1: The details about how to create NSD and onboard NSD to NFVO are not provided in present document.

2) The MEAO sends on Mv1 a create NS request to the NVFO to create a NS.

3) After the NS is created, the NFVO sends a create NS response to the MEAO on Mv1.

4) The MEAO sends to the NFVO the instantiate NS request on Mv1 for creating a VNF for the application and instantiating it.

5) The NFVO sends a create VNF instance request to the VNFM.

6) After a VNF instance is created, the VNFM sends a create VNF response to the NFVO.

7) The NFVO sends an instantiate VNF operation request to the VNFM for instantiating a VNF for the application.

8) After the VNF is instantiated, the VNFM sends a response to the NFVO.

NOTE 2: Refer to ETSI GS NFV-SOL 003 [7] and ETSI GS NFV-IFA 007 [i.3] for details on steps 5 to 8.

9) Optionally, if the MEMP-V subscribes the LCM notification with the VNFM, the VNFM sends a notification (completed) for VNF instantiation to the MEPM-V over Mv2.

10) When receiving the confirmation of the VNF instantiated for the application, the NFVO sends a response to the MEAO on Mv1. The MEAO updates the AppInstanceInfo for the new instantiated application.

Alternatively, if a NS has been instantiated, the following steps are executed:

11) The MEAO sends an update NS operation request message to the NVFO on Mv1 to instantiate a VNF instance for the application and update the existing NS to include the VNF instance.

12) The NFVO sends a create VNF instance request to the VNFM.

13) After a VNF instance is created, the VNFM sends a create VNF response to the NFVO.

14) The NFVO sends an instantiate VNF operation request to VNFM for instantiating the VNF instance of the application.

15) After the VNF instance is instantiated, the VNFM sends a response to the NFVO.

NOTE 3: Refer to ETSI GS NFV-SOL 003 [7] and ETSI GS NFV-IFA 007 [i.3] for details on steps 12 to 15.

16) Optionally, if the MEMP-V subscribes the LCM notification with the VNFM, the VNFM sends a notification (completed) for VNF instantiation to the MEPM-V on Mv2.

17) When receiving the confirmation of the VNF instance instantiated, the NFVO sends a response message to the MEAO. The MEAO updates the AppInstanceInfo for the application instance.

After the application is confirmed being instantiated as a VNF, the following steps are executed for sending configuration information:

18) The MEAO sends the configure application instance request with required configuration parameters to MEPM-V over Mm3*.

19) The MEPM-V forwards the configuration request to the MEP (VNF) to configure the traffic rules, DNS rules, etc. for the application instance.

20) After the configuration is completed, the MEP (VNF) sends a configure application instance response to the MEPM-V.

21) The MEPM-V sends the configure application instance response to the MEAO.

NOTE 4: Refer to the clause 5.3.1 of the present document about the similar application configuration procedure for steps 18 to 21.

Finally, the following step is executed:

22) The MEAO sends the instantiate application response to the OSS.

Figure 5.7.2-2 illustrates a message flow of instantiating an application as a VNF through NFVO directly.

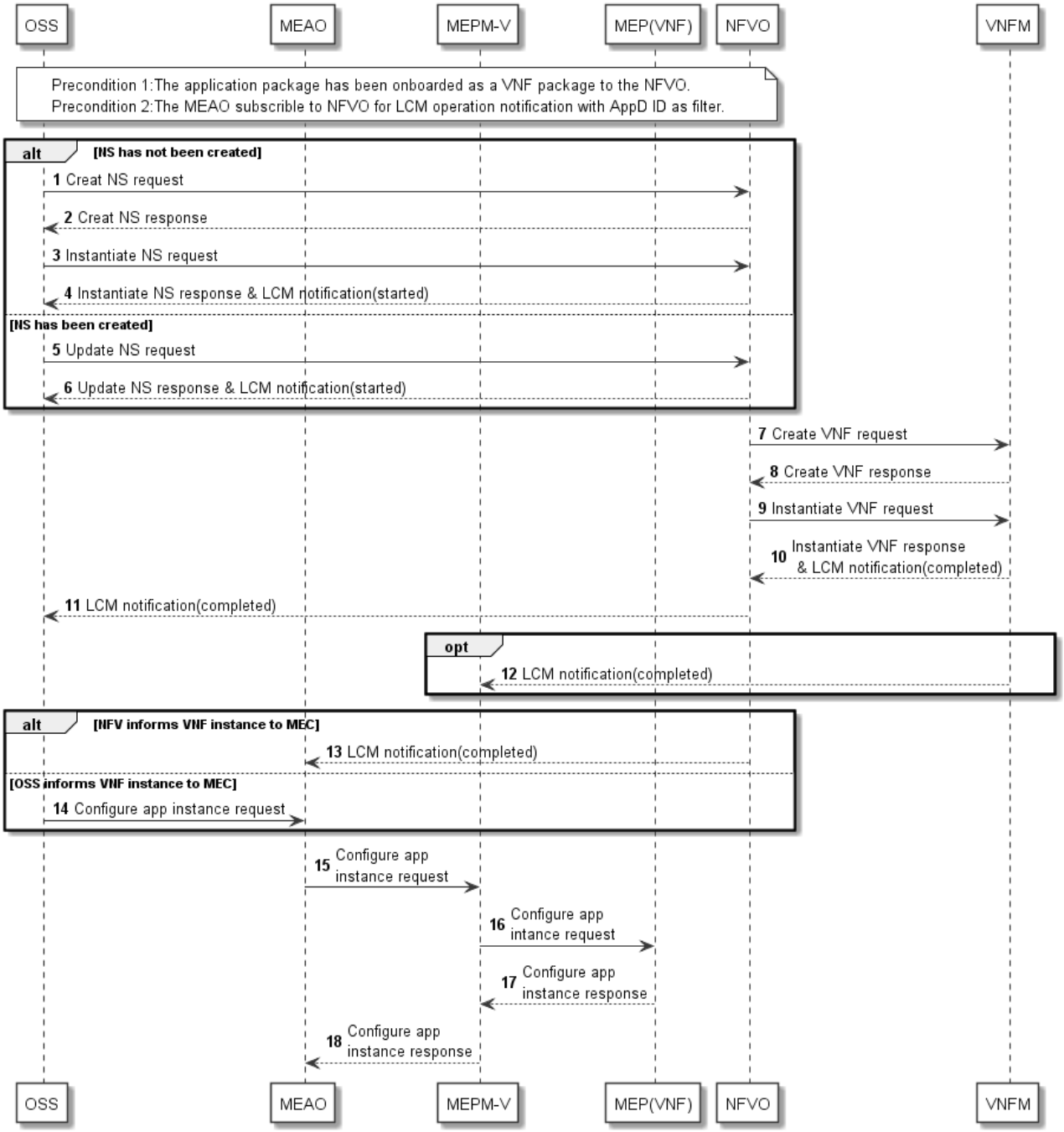

Figure 5.7.2-2: Instantiate application in NFV through NFVO directly

PRECONDITIONS:

- The application has been onboarded as a VNF package to the NFVO.
- The MEAO has subscribed to NFVO for LCM operation notification with AppD ID as filter.

If a NS has not been created, the following steps are executed:

1) The OSS sends a create NS request to the NVFO to create a NS.

2) After the NS is created, the NFVO sends a create NS response to the OSS.

3) The OSS sends to the NFVO the instantiate NS request for creating a VNF for the application and instantiating it.

4) The NFVO sends an instantiate NS response and an LCM notification (started) to the OSS.

Alternatively, if a NS has been instantiated, the following steps are executed:

5) The OSS sends an update NS operation request message to the NFVO to instantiate a VNF instance for the application and update the existing NS to include the VNF instance.

6) The NFVO sends an update NS response and an LCM notification (started) to the OSS.
Refer to ETSI GS NFV-SOL 005 [i.7] and ETSI GS NFV-IFA 013 [15] for detail on steps 1 to 6.

Subsequently, the following steps are executed for instantiation VNF instance:

7) The NFVO sends a create VNF instance request to the VNFM.

8) After a VNF instance is created, the VNFM sends a create VNF response to the NFVO.

9) The NFVO sends an instantiate VNF operation request to the VNFM for instantiating a VNF for the application.

10) The VNFM sends a response to the NFVO. After the VNF is instantiated, the VNFM sends an LCM notification (completed) to the NFVO.
Refer to ETSI GS NFV-SOL 003 [7] and ETSI GS NFV-IFA 007 [i.3] for detail on steps 7 to 10.

11) The NFVO sends an LCM notification (completed) to the OSS.

12) Optionally, if the MEMP-V subscribes the LCM notification with the VNFM, the VNFM sends a notification (completed) for VNF instantiation to the MEPM-V over Mv2.

If the NFVO provides VNF instance information to MEAO, the following step is executed:

13) The NFVO sends a notification (completed) for VNF instantiation to the MEAO over Mv1. In this notification, the NFVO provides the instantiated VNF information including VNF instance identifier and location information to MEAO.

Alternatively, if the OSS provides VNF instance information to MEAO, the following step is executed:

14) The OSS sends a configure application instance request to the MEAO over Mm1. In this request, the OSS provides the instantiated VNF information including VNF instance identifier and location information to MEAO.

Subsequently, the following steps are executed for sending configuration information:

15) The MEAO determines the MEPM-V, which manages the MEP to which the instantiated application as VNF instance intent to associate, according to the VNF instance location information. The MEAO sends the configure application instance request with required configuration parameters to MEPM-V over Mm3*.
The MEAO can send configure application instance request directly if the MEAO has cached the application package information locally. Otherwise, the MEAO shall acquire application package information from NFVO firstly. Refer to figure 5.6.3-1 in clause 5.6.3 of the present document about the similar Query application package information procedure.

16) The MEPM-V forwards the configuration request to the MEP (VNF) to configure the traffic rules, DNS rules, etc. for the application instance.

NOTE 5: This step is only for the completeness of the flow, and will not be specified in the present document.

17) After the configuration is completed, the MEP (VNF) sends a configure application instance response to the MEPM-V.

NOTE 6: This step is only for the completeness of the flow, and will not be specified in the present document.

18) The MEPM-V sends the configure application instance response to the MEAO.

5.7.3 Terminate application in NFV

The message flow of application termination in NFV is to terminate an application instance in MEC system and delete the resources of the VNF instance related to the application via NFV APIs defined in the specifications of ETSI GS NFV-IFA 013 [15], ETSI GS NFV-IFA 007 [i.3], ETSI GS NFV-SOL 003 [7] and ETSI GS NFV-SOL 005 [i.7].

Figure 5.7.3-1 illustrates the message flow of terminate an application instance through MEAO or MEMP-V.

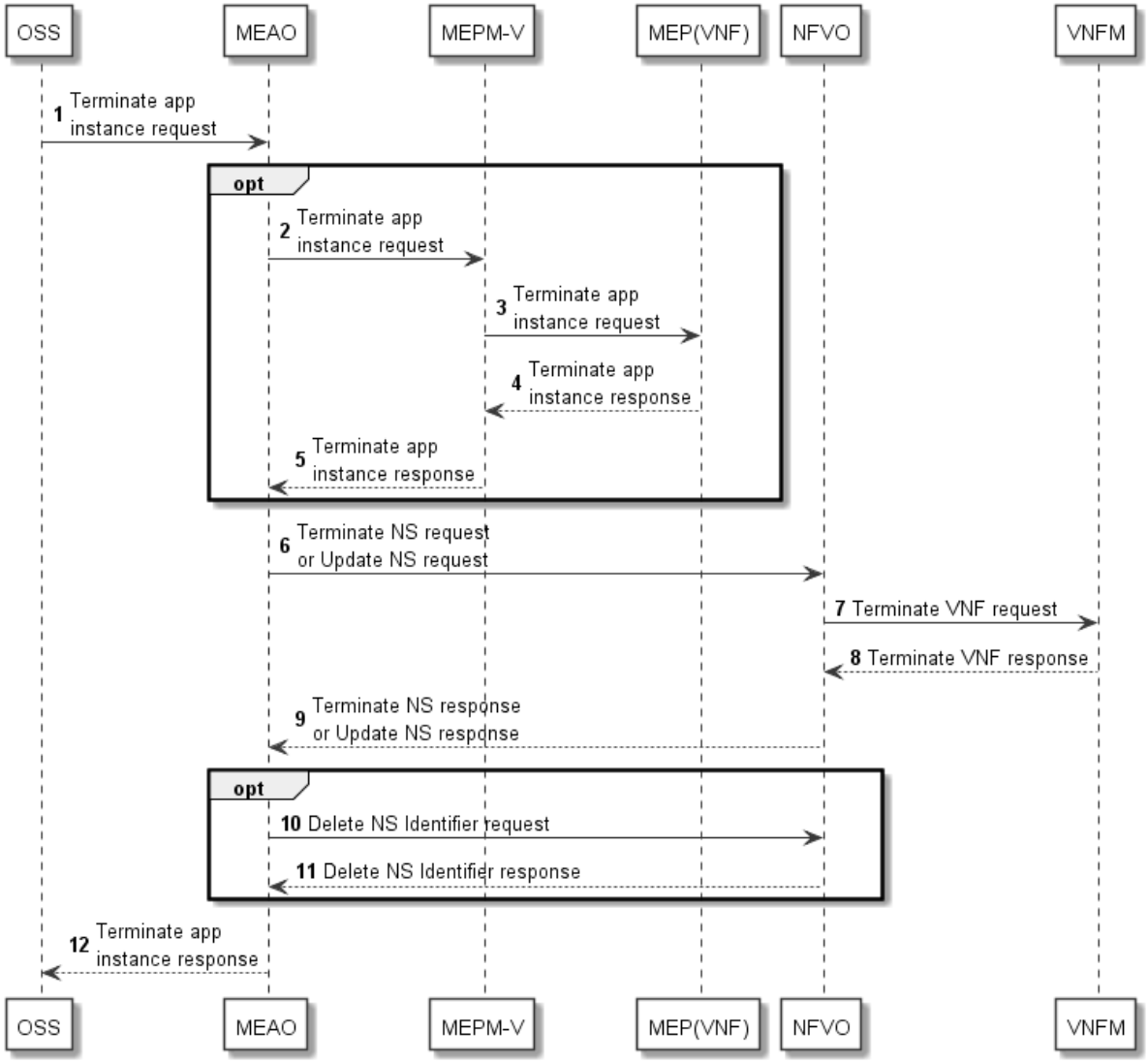

Figure 5.7.3-1: Terminate application instance through MEAO or MEPM-V

1) The OSS sends a terminate application instance request to the MEAO on Mm1 for deleting the application instance.

Optionally, the following steps are executed for taking application instance out of service:

2) The MEAO sends the terminate application instance request to the MEPM-V on Mm3*.

3) The MEPM-V forwards the terminate application instance request to the MEP (VNF) on Mm5. The MEP (VNF) performs terminating the operation of the application instance.
If supported by the MEC application, and graceful termination is requested, the MEP (VNF) notifies the MEC application instance of the termination event. The MEC application instance executes the actions needed before it has been terminated by the MEP (VNF), the actual action(s) the MEC application instance will perform for the application level termination is up to MEC application, and is out of scope of the present document. After the MEC application instance finishes application level termination, it may inform the MEP (VNF) that it is ready to be terminated. The MEP (VNF) may set a timer for the application level termination.

After the timer expires, the MEP (VNF) will shut down the application regardless of the progress of application level termination.

4) The MEP (VNF) sends the response to the MEPM-V on Mm5 after the application instance is terminated.

5) The MEPM-V sends to the MEAO the terminate application instance response.

Subsequently, the following steps are executed for terminating the VNF instance:

6) The MEAO sends to the NFVO a Terminate NS request or an Update NS request with the appropriate removeVnfInstanceId input parameter to trigger the deleting the VNF instance of the application.

7) The NFVO sends a Terminate VNF request to the VNFM to trigger the termination of the application VNF instance.

8) The VNFM performs the operations of Terminate VNF and Delete VNF identifier. It then sends a response to the NFVO.

9) The NFVO sends the Terminate NS response or Update NS response to the MEAO.

NOTE 1: Refer to ETSI GS NFV-SOL 005 [i.7], ETSI GS NFV-IFA 013 [15], ETSI GS NFV-SOL 003 [7] and ETSI GS NFV-IFA 007 [i.3] for details on the steps 6 - 9.

Optionally, if the MEAO has sent a Terminate NS request in step 6, the following steps are executed:

10) The MEAO sends to the NFVO a Delete NS Identifier request. Meanwhile the MEAO deletes the information related to the application instance.

11) The NFVO sends to the MEAO a Delete NS Identifier response.

NOTE 2: Refer to ETSI GS NFV-SOL 005 [i.7] and ETSI GS NFV-IFA 013 [15] for details on the step 10.

Finally, the following step is executed:

12) The MEAO sends to the OSS the terminate application instance response.

Figure 5.7.3-2 illustrates the message flow of terminate VNF instance of application through NFVO directly.

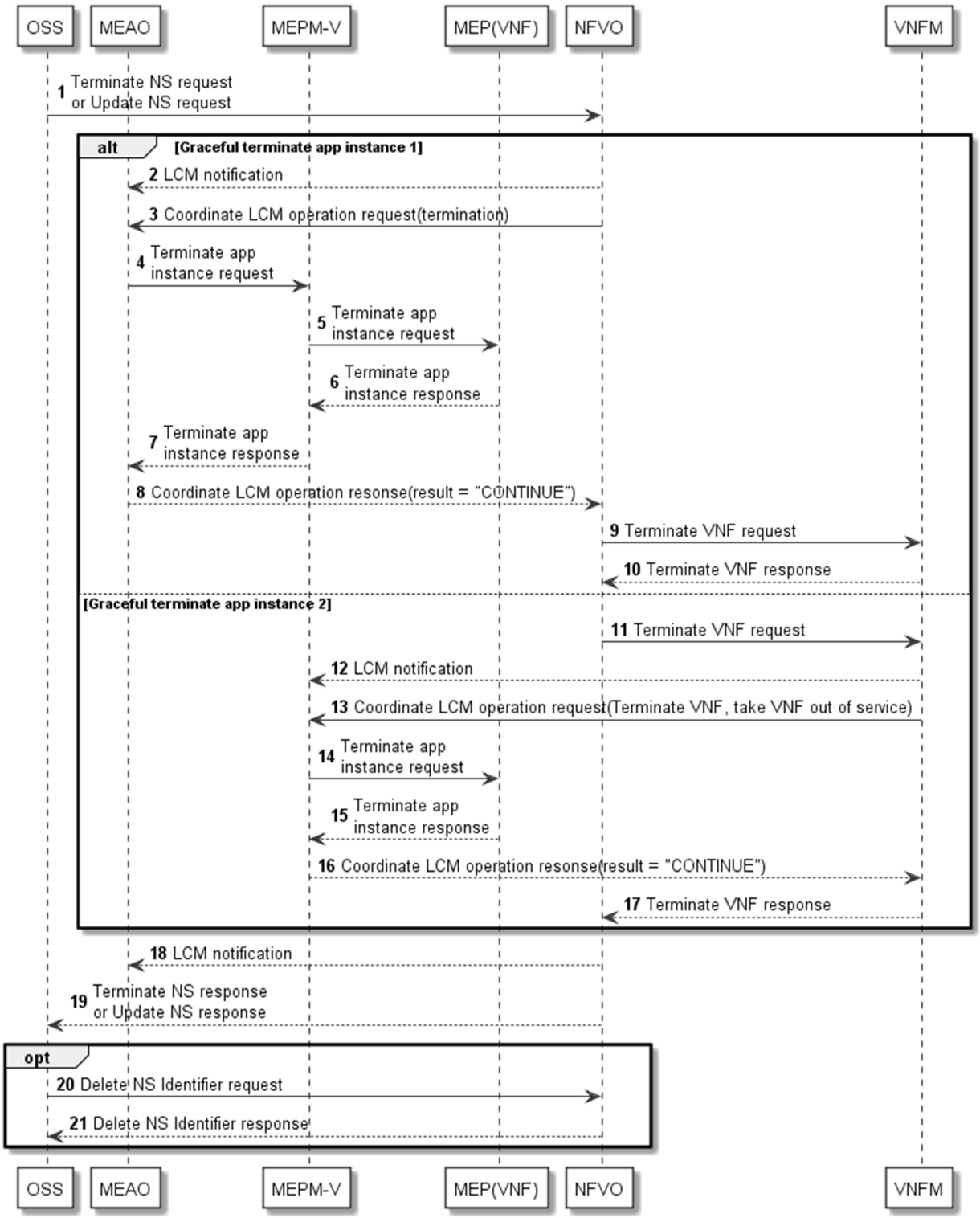

Figure 5.7.3-2: Terminate VNF instance of application through NFVO directly

1) The OSS sends a Terminate NS request or an Update NS request with the appropriate removeVnfInstanceId input parameter to the NFVO directly to trigger terminating and deleting the VNF instance of the application.

For the case supporting graceful termination coordination between NFVO and MEAO, the following steps are executed:

2) If the MEAO has subscribed to LCM notifications with the NFVO, the MEAO receives an LCM notification of Terminate NS or Update NS from the NFVO on Mv1.

3) The NFVO sends to the MEAO a Coordinate LCM Operation request for the termination operation of the VNF instance of the application.

4) The MEAO sends a terminate application instance request to MEPM-V on Mm3* to stop the operation of application instance.

5) The MEPM-V sends the terminate application instance request to MEP (VNF) on Mm5 for stopping the operation of application instance. The MEP (VNF) may set a timer for the application level termination. After the timer expires, the MEP (VNF) will shut down the application regardless of the progress of application level termination.

6) After the application instance has stopped, the MEP (VNF) sends a terminate application instance response to the MEPM-V on Mm5.

7) The MEPM-V sends the terminate application instance response to the MEAO on Mm3*.

NOTE 3: Refer to the clause 5.3.2 of the present document about the similar application termination procedure for the steps 4 to 7.

8) The MEAO sends to the NFVO the Coordinate LCM Operation response with the coordination result "CONTINUE".

9) The NFVO then sends to the VNFM a Terminate VNF request to trigger the termination of the VNF instance.

10) The VNFM terminates the VNF instance and sends to the NFVO the Terminate VNF response.

Alternatively, for the case supporting graceful termination coordination between VNFM and MEPM-V, the following steps are executed:

11) The NFVO sends to the VNFM a Terminate VNF request for the termination of the VNF instance.

12) If the MEPM-V has subscribed to notifications about VNF termination with the VNFM, the MEPM-V receive a notification from the VNFM on Mv2.

13) The VNFM sends to the MEPM-V a Coordinate LCM Operation request to take the VNF instance out of service, as defined in clause 9.11.2.2 of ETSI GS NFV-IFA 008 [16].

14) The MEPM-V sends the terminate application instance request to MEP (VNF) on Mm5 for stopping the operation of application instance. The MEP (VNF) may set a timer for the application level termination. After the timer expires, the MEP (VNF) will shut down the application regardless of the progress of application level termination.

15) After the application instance has stopped, the MEP (VNF) sends a terminate application instance response to the MEPM-V on Mm5.

NOTE 4: Refer to the clause 5.3.2 of the present document about the similar application termination procedure for the steps 14 and 15.

16) The MEPM-V sends to the VNFM the Coordinate LCM Operation response with the coordination result "CONTINUE".

17) The VNFM continues the termination of the VNF instance and sends to the NFVO the terminate VNF response.

Subsequently, the following steps are executed:

18) The NFVO completes the termination or update of the NS instance. If the MEAO has subscribed to NS LCM notifications with the NFVO, the NFVO sends to the MEAO an LCM notification that the NS termination or update has completed. The MEAO deletes the information related to the application instance.

19) The NFVO sends a response to the OSS for the Terminate NS or Update NS request.

Optionally, if the OSS has sent a Terminate NS request in step 1, the following steps are executed:

20) The OSS sends to the NFVO a Delete NS Identifier request.

21) The NFVO sends to the OSS a Delete NS Identifier response.

NOTE 5: Refer to ETSI GS NFV-SOL 005 [i.7] and ETSI GS NFV-IFA 013 [15] for details on the step 20.

5.7.4 Operate application in NFV

The message flows of operate application in NFV are used to operate (i.e. start or stop) an application instance as a VNF via NFV APIs defined in the specifications of ETSI GS NFV-SOL 005 [i.7]. Refer to ETSI GS NFV-IFA 013 [15], ETSI GS NFV-IFA 007 [i.3] and ETSI GS NFV-SOL 003 [7] for details.

Figure 5.7.4-1 illustrates a message flow of operating an application as a VNF through MEAO.

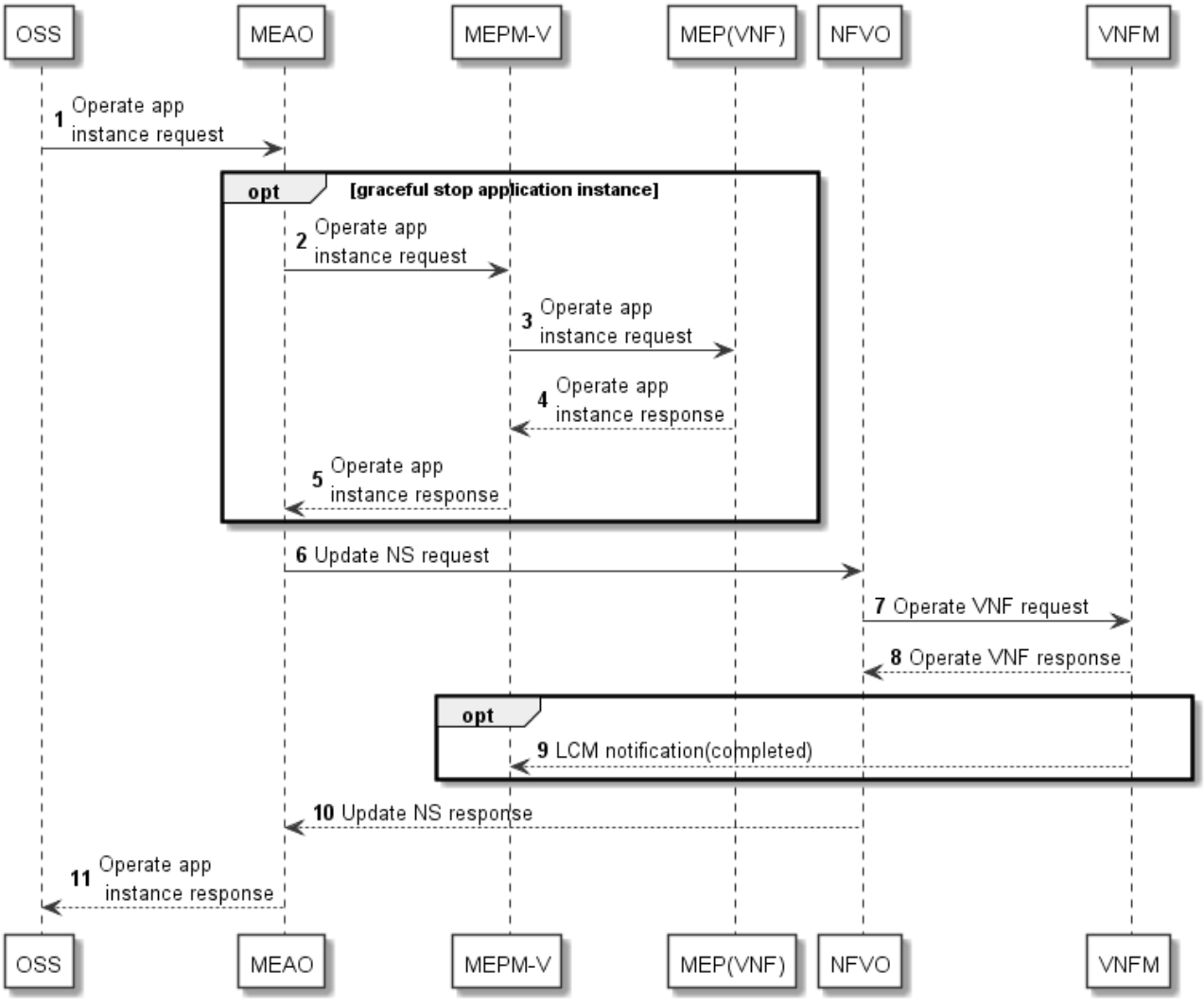

Figure 5.7.4-1: operate application in NFV through MEAO

1) The OSS sends to the MEAO an operate application instance request (i.e. start or stop application instance).

Optionally, if the request is to stop the application instance gracefully, the following steps are executed for stopping application instance:

2) The MEAO sends the operate application instance request to the MEPM-V on Mm3*.

3) The MEPM-V forwards the operate application instance request to the MEP (VNF) on Mm5. The MEP (VNF) performs stopping the operation of the application instance.
If supported by the MEC application, and graceful stop is requested, the MEP (VNF) notifies the MEC application instance of the operate event. The MEC application instance executes the actions needed before it has been stopped by the MEP (VNF), the actual action(s) the MEC application instance will perform for the application level stop is up to MEC application, and is out of scope of the present document. After the MEC application instance finishes application level stop, it may inform the MEP (VNF) that it is ready to be stopped. The MEP (VNF) may set a timer for the application level stop. After the timer expires, the MEP (VNF) continues the stop MEC application instance flow regardless of the progress of application level stop.

4) The MEP(VNF) sends to the MEPM-V an operate application instance response.

5) The MEPM-V sends to the MEAO an operate application instance response.

Subsequently, the following steps are executed to operate the VNF instance of the application:

6) The MEAO sends to the NFVO an Update NS request with the appropriate operate VnfInstanceId input parameter to trigger the changing state of the VNF instance of the application.

7) The NFVO sends to the VNFM an Operate VNF request to trigger the changing state of the application VNF instance.

8) The VNFM performs the operations of the changing state. It then sends a response to the NFVO.

9) Optionally, if the MEPM-V has subscribed to notifications about VNF operation with the VNFM, the MEPM-V receive a notification from the VNFM on Mv2.

10) The NFVO sends to the MEAO an Update NS response.

NOTE 1: Refer to ETSI GS NFV-SOL 005 [i.7], ETSI GS NFV-IFA 013 [15], ETSI GS NFV-SOL 003 [7] and ETSI GS NFV-IFA 007 [i.3] for details on the steps 6 to 10.

Finally, the following step is executed:

11) The MEAO sends to the OSS the operate application instance response.

Figure 5.7.4-2 illustrates a message flow of operating an application as a VNF through NFVO directly.

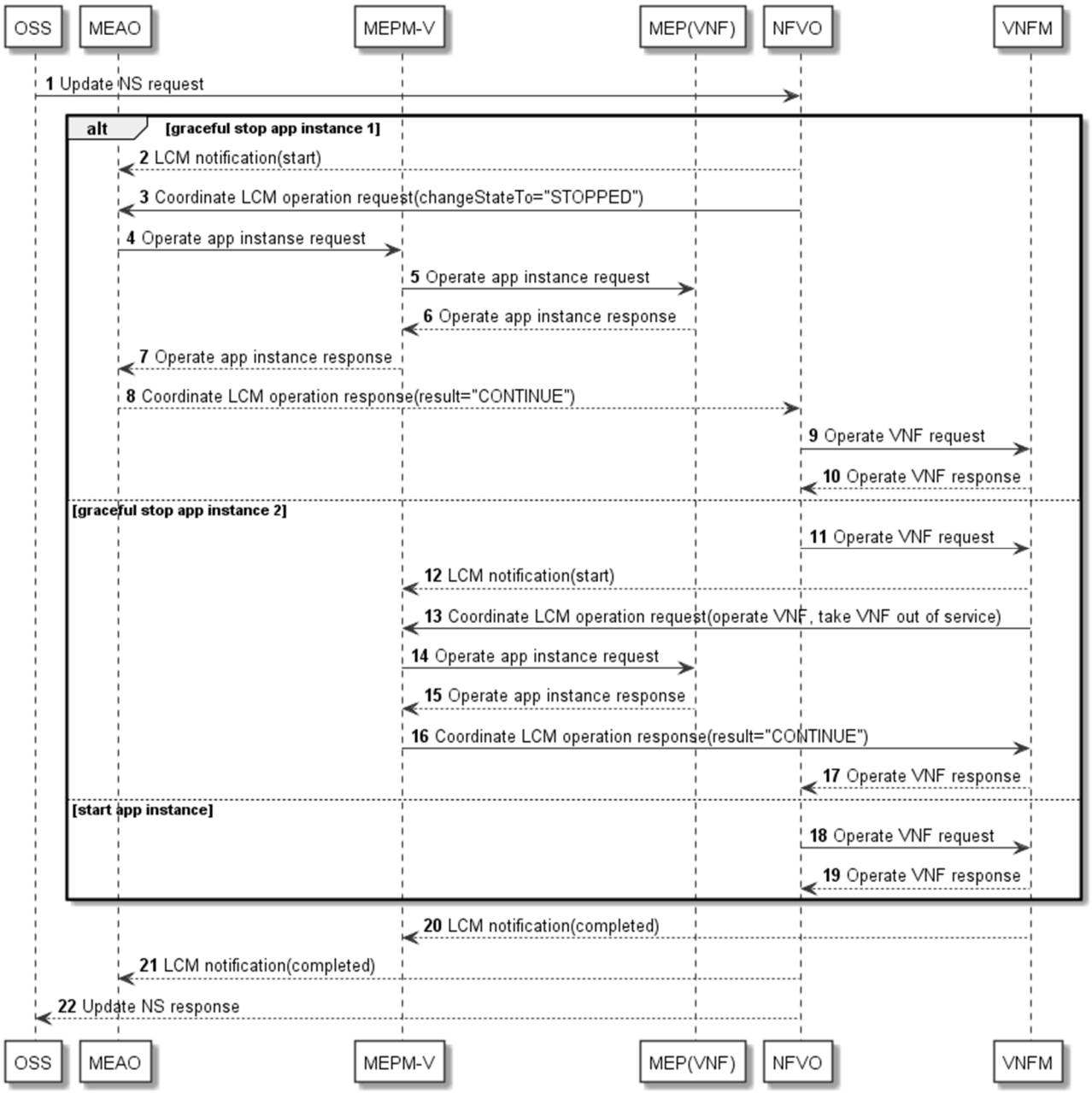

Figure 5.7.4-2: operate application in NFV through NFVO directly

1) The OSS sends to the NFVO directly an Update NS request with the appropriate operated VnfInstanceId input parameter to trigger operate the VNF instance of the application(i.e. start or stop application instance).

For the case of stopping the application instance and supporting graceful stop coordination between NFVO and MEAO, the following steps are executed:

2) If the MEAO has subscribed to LCM notifications with the NFVO, the MEAO receives an LCM notification of Update NS from the NFVO on Mv1.

3) The NFVO sends to the MEAO a Coordinate LCM operation request for the stopping operation of the VNF instance of the application.

4) The MEAO sends an operate application instance request to MEPM-V on Mm3* to stop the operation of application instance.

5) The MEPM-V sends the operate application instance request to MEP (VNF) on Mm5 for stopping the operation of application instance. The MEP (VNF) may set a timer for the application level stop. After the timer expires, the MEP (VNF) continues the stop MEC application instance flow regardless of the progress of application level stop.

6) After the application instance has stopped, the MEP (VNF) sends an operate application instance response to the MEPM-V on Mm5.

7) The MEPM-V sends the operate application instance response to the MEAO on Mm3*.

8) The MEAO sends to the NFVO the Coordinate LCM operation response with the coordination result "CONTINUE".

9) The NFVO then sends to the VNFM an operate VNF request to trigger the stopping of the VNF instance.

10) The VNFM stops the VNF instance and sends to the NFVO the operate VNF response.

Alternatively, for the case of stopping the application instance and supporting graceful stop coordination between VNFM and MEPM-V, the following steps are executed:

11) The NFVO sends to the VNFM an operate VNF request for the stop of the VNF instance.

12) If the MEPM-V has subscribed to notifications about VNF stopping with the VNFM, the MEPM-V receive a notification from the VNFM on Mv2.

13) The VNFM sends to the MEPM-V a Coordinate LCM operation request to take the VNF instance out of service, as defined in clause 9.11.2.2 of ETSI GS NFV-IFA 008 [16].

14) The MEPM-V sends the operate application instance request to MEP (VNF) on Mm5 for stopping the operation of application instance. The MEP (VNF) may set a timer for the application level termination. After the timer expires, the MEP (VNF) continues the stop MEC application instance flow regardless of the progress of application level stop.

15) After the application instance has stopped, the MEP (VNF) sends an operate application instance response to the MEPM-V on Mm5.

16) The MEPM-V sends to the VNFM the Coordinate LCM operation response with the coordination result "CONTINUE".

17) The VNFM continues the stopping of the VNF instance and sends to the NFVO the operate VNF response.

Alternatively, for the case of starting the application instance, the following steps are executed:

18) The NFVO then sends to the VNFM an operate VNF request to trigger the starting of the VNF instance.

19) The VNFM starts the VNF instance and sends to the NFVO the operate VNF response.

Subsequently, the following steps are executed:

20) If the MEP (VNF) has subscribed to VNF LCM notifications with the VNFM, the VNFM sends to the MEP (VNF) an LCM notification that the VNF operation has completed.

21) The NFVO completes update of the NS instance. If the MEAO has subscribed to NS LCM notifications with the NFVO, the NFVO sends to the MEAO an LCM notification that the NS update has completed.

Finally, the following step is executed:

22) The NFVO sends a response to the OSS for the Update NS request.

NOTE 2: Refer to ETSI GS NFV-SOL 005 [i.7] and ETSI GS NFV-IFA 013 [15] for details on the step 21.

6 Information models and interfaces

6.1 Applicable reference points

The following clauses apply to the Mm reference points, for which the relevant sequence diagrams are described in clause 5.

6.2 Information models

6.2.1 Application descriptor information model

6.2.1.1 Introduction

This clause defines data structures to be used by application descriptor information model.

6.2.1.2 Type: AppD

6.2.1.2.1 Description

An application Descriptor (AppD) is a part of application package, and describes application requirements and rules required by application provider.

6.2.1.2.2 Attributes

The attributes of the AppD data type shall follow the indications provided in table 6.2.1.2.2-1.

Table 6.2.1.2.2-1: Attributes of AppD

Attribute name	Cardinality	Data type	Description
appDId	1	String	Identifier of this MEC application descriptor. This attribute shall be globally unique. See note 1.
appName	1	String	Name to identify the MEC application.
appProvider	1	String	Provider of the application and of the AppD.
appSoftVersion	1	String	Identifies the version of software of the MEC application.
appDVersion	1	String	Identifies the version of the application descriptor.
mecVersion	1	String	Identifies version(s) of MEC system compatible with the MEC application described in this version of the AppD. The value shall be formatted as comma-separated list of strings. Each entry shall have the format <x>.<y>.<z> where <x>, <y> and <z> are decimal numbers representing the version of the present document. Whitespace between list entries shall be trimmed before validation.
appInfoName	0..1	String	Human readable name for the MEC application.
appDescription	1	String	Human readable description of the MEC application.
virtualComputeDescriptor	0..1	VirtualComputeDescriptor	Describes CPU and memory requirements, as well as optional additional requirements, such as disk and acceleration related capabilities, of the single VM used to realize this MEC application. See note 5.

Attribute name	Cardinality	Data type	Description
osContainerDescriptor	0..N	OsContainerDescriptor	Describes CPU, memory requirements and limits, and software images of the OS Containers realizing this MEC application corresponding to OS Containers sharing the same host and same network namespace. See notes 5 and 7.
swImageDescriptor	1..N	SwImageDescriptor	Describes the descriptors of the software image to be used by the virtualisation container used to realize this MEC application. See note 5.
virtualStorageDescriptor	0..N	VirtualStorageDescriptor	Defines descriptors of virtual storage resources to be used by the MEC application.
appExtCpd	1..N	AppExternalCpd	Describes external interface(s) exposed by this MEC application. See note 4.
appServiceRequired	0..N	ServiceDependency	Describes services a MEC application requires to run.
appServiceOptional	0..N	ServiceDependency	Describes services a MEC application may use if available.
appServiceProduced	0..N	ServiceDescriptor	Describes services a MEC application is able to produce to the platform or other MEC applications. Only relevant for service-producing apps.
appFeatureRequired	0..N	FeatureDependency	Describes features a MEC application requires to run.
appFeatureOptional	0..N	FeatureDependency	Describes features a MEC application may use if available.
transportDependencies	0..N	TransportDependency	Transports, if any, that this application requires to be provided by the platform. These transports will be used by the application to deliver services provided by this application. Only relevant for service-producing apps. See note 2.
appTrafficRule	0..N	TrafficRuleDescriptor	Describes traffic rules the MEC application requires.
appDNSRule	0..N	DNSRuleDescriptor	Describes DNS rules the MEC application requires.
appLatency	0..1	LatencyDescriptor	Describes the maximum latency tolerated by the MEC application.
terminateAppInstanceOpConfig	0..1	TerminateAppInstanceOpConfig	Configuration parameters for the Terminate application instance operation.
changeAppInstanceStateOpConfig	0..1	ChangeAppInstanceStateOpConfig	Configuration parameters for the change application instance state operation.
userContextTransferCapability	0..1	UserContextTransferCapability	If the application supports the user context transfer capability, this attribute shall be included.
appNetworkPolicy	0..1	AppNetworkPolicy	If present, it represents the application network policy of carrying the application traffic.
mciopId	0..1	String	Identifies the MCIOP in the application package, used in containerized workload management, when the application is realized by a set of OS containers. See note 7.
mcioIdentificationData	0..1	McioIdentificationData	Name and type of the Managed Container Infrastructure Object (MCIO) that realizes this application. It allows the VNFM to identify the MCIO e.g. when querying the Container Infrastructure Service Management (CISM). See note 7. It shall be present when the application is realized by one or a set of OS containers and shall be absent otherwise.

Attribute name	Cardinality	Data type	Description
logicalNode	0..N	LogicalNodeRequirements	The logical node requirements. See notes 6 and 7.
requestAdditionalCapabilities	0..N	RequestedAdditionalCapabilityData	Specifies requirements for additional capabilities. These can be for a range of purposes. One example is acceleration related capabilities. See notes 6 and 7.
mcioConstraintParams	0..N	Enum	The parameter names for constraints expected to be assigned to MCIOs realizing this application. The value specifies the standardized semantical context of the MCIO constraints and the parameter names for the MCIO constraints in the MCIO declarative descriptor. The mcioConstraintParams attribute shall have one of the following values, expressing the associated semantical context. VALUES: • localAffinityCisNode • nodeAdditionalCapabilitySsd • nodeAdditionalCapabilityDpdk • nodeAdditionalCapabilitySriov • nodeAdditionalCapabilityGpu • nodeAdditionalCapabilityFpga • nodeAdditionalCapabilityCpuPin • nodeCapabilityLogicalNuma • nodePool For the associated semantical context of the values, refer to the description under the table 7.1.6.2.2-1 of ETSI GS NFV IFA 011 [1]. See note 7.

NOTE 1: The appDId shall be used as the unique identifier of the application package that contains this AppD.
NOTE 2: This attribute indicates groups of transport bindings which a service-producing MEC application requires to be supported by the platform in order to be able to produce its services. At least one of the indicated groups needs to be supported to fulfil the requirements.
NOTE 3: The support of application descriptor containing descriptions of multiple virtualisation containers and/or application software images is out of scope of the present document.
NOTE 4: External interfaces are used to connect to e.g. other MEC applications, MEC services, UEs and also MEC platform and OSS.
NOTE 5: Only one of virtualComputeDescriptor or osContainerDescriptor shall be present. If virtualComputeDescriptor presents, only a single swImageDescriptor shall be provided.
NOTE 6: If the AppD includes virtualComputeDesc, then logicalNode and requestedAdditionalCapabilites shall not be present.
NOTE 7: This attribute reflects the ETSI NFV interpretation of the cloud native workloads.

6.2.1.3 Type: VirtualComputeDescriptor

6.2.1.3.1 Description

The VirtualComputeDescriptor data type supports the specification of requirements related to virtual compute resources when a MEC application is intended to be deployed in a single VM.

6.2.1.3.2 Attributes

The attributes of VirtualComputeDescriptor shall follow the definition in clause 7.1.9.2.2 of ETSI GS NFV-IFA 011 [1], with the following consideration:

- A VNF corresponds to MEC application, a VNFD corresponds to an AppD in MEC, and the VDU in table 7.1.6.2.2-1 of ETSI GS NFV-IFA 011 [1] corresponds to a MEC application.

6.2.1.4 Type: SwImageDescriptor

6.2.1.4.1 Description

The SwImageDescriptor data type describes the software image of a MEC application.

6.2.1.4.2 Attributes

The attributes of SwImageDescriptor shall follow the definition in clause 7.1.6.5 of ETSI GS NFV-IFA 011 [1], with the following consideration:

- A VNF corresponds to a MEC application, a VNFD corresponds to an AppD in MEC, and the VDU in table 7.1.6.2.2-1 of ETSI GS NFV-IFA 011 [1] corresponds to a MEC application.

6.2.1.5 Type: VirtualStorageDescriptor

6.2.1.5.1 Description

The VirtualStorageDescriptor data type describes the virtual storage required by a MEC application.

6.2.1.5.2 Attributes

The attributes of VirtualStorageDescriptor shall follow the definition in clause 7.1.9.4 of ETSI GS NFV-IFA 011 [1], with the following consideration:

- A VNF corresponds to a MEC application, a VNFD corresponds to an AppD in MEC, and the VDU in table 7.1.6.2.2-1 of ETSI GS NFV-IFA 011 [1] corresponds to a MEC application.

6.2.1.6 Type: AppExternalCpd

6.2.1.6.1 Description

The AppExternalCpd data type supports the specification of MEC application requirements related to external connection point.

6.2.1.6.2 Attributes

The attributes of AppExternalCpd are shown in table 6.2.1.6.2-1.

Table 6.2.1.6.2-1: Attributes of AppExternalCpd

Attribute name	Cardinality	Data type	Description
virtualNetworkInterfaceRequirements	0..N	VirtualNetworkInterfaceRequirements	Specifies requirements on a virtual network interface realizing the CPs instantiated from this CPD. See note 1.
additionalServiceData	0..N	AdditionalServiceData	Additional service identification data of the external CP.
(inherited attributes)			All attributes inherited from Cpd. See note 2.
NOTE 1: An AppD conformant to the present document shall not specify "virtualNetworkInterfaceRequirements" in AppExternalCpd corresponding to primary container cluster network interfaces. NOTE 2: For CPs exposed by MEC Applications realized only by one or set of OS containers and used by the OS containers to connect to the primary container cluster external network, the ability to configure virtualised resources based on cpRole and trunkMode attributes might not be supported by all container technologies.			

For the definition of the VirtualNetworkInterfaceRequirements, refer to clause 7.1.6.6 of ETSI GS NFV-IFA 011 [1]. For the definition of Cpd, refer to clause 7.1.6.3 of ETSI GS NFV-IFA 011 [1].

For the definition of AdditionalServiceData, refer to clause 7.1.18.3 of ETSI GS NFV IFA 011 [1].

6.2.1.7 Type: ServiceDescriptor

6.2.1.7.1 Description

The ServiceDescriptor data type describes a MEC service produced by a service-providing MEC application.

6.2.1.7.2 Attributes

The attributes of a ServiceDescriptor are depicted in table 6.2.1.7.2-1.

Table 6.2.1.7.2-1: Attributes of ServiceDescriptor

Attribute name	Cardinality	Data type	Description
serName	1	String	The name of the service, for example, RNIS, LocationService, etc.
serCategory	0..1	CategoryRef	A Category reference of the service, defined in ETSI GS MEC 011 [17].
version	1	String	The version of the service.
transportsSupported	0..N	Structure (inlined)	Indicates transports and serialization formats supported made available to the service-consuming application. Defaults to REST + JSON if absent.
>transport	1	TransportDescriptor	Information about the transport in this binding.
>serializers	1..N	SerializerType	Information about the serializers in this binding, as defined in the SerializerType type in ETSI GS MEC 011 [17].

6.2.1.8 Type: FeatureDependency

6.2.1.8.1 Description

The FeatureDependency data type supports the specification of requirements of a MEC application related to a feature of MEC platform.

6.2.1.8.2 Attributes

The attributes of a FeatureDependency are depicted in table 6.2.1.8.2-1.

Table 6.2.1.8.2-1: Attributes of FeatureDependency

Attribute name	Cardinality	Data type	Description
featureName	1	String	The name of the feature, for example, UserApps, UEIdentity, etc.
version	1	String	The version of the feature.

6.2.1.9 Type: TrafficRuleDescriptor

6.2.1.9.1 Description

The TrafficRuleDescriptor data type describes traffic rules related to a MEC application.

6.2.1.9.2 Attributes

The attributes of TrafficRuleDescriptor are shown in table 6.2.1.9.2-1.

Table 6.2.1.9.2-1: Attributes of TrafficRuleDescriptor

Attribute name	Cardinality	Data type	Description
trafficRuleId	1	String	Identifies the traffic rule.
filterType	1	Enum (inlined)	Definition of filter type: per FLOW or PACKET If it is per FLOW, the filter matches upstream (e.g. UE->EPC) packets and downstream (e.g. EPC->UE) packets are handled by the same context.
priority	1	Integer	Priority of this traffic rule within the range 0 to 255. If traffic rule conflicts, the one with higher priority take precedence. See note 1.
trafficFilter	1..N	TrafficFilter	The filter used to identify specific flow/packets that need to be handled by the MEC host.
action	1	Enum (inlined)	Identifies the action of the MEC host data plane, when a packet matches the trafficFilter, the example actions include: • DROP, • FORWARD_DECAPSULATED, • FORWARD_ENCAPSULATED, • PASSTHROUGH, • DUPLICATE_DECAPSULATED, • DUPLICATE_ENCAPSULATED
dstInterface	0..2	InterfaceDescriptor	Describes the destination interface information. If the action is FORWARD_DECAPSULATED, FORWARD_ENCAPSULATED or PASSTHROUGH, one value shall be provided. If the action is DUPLICATE_DECAPSULATED or DUPLICATE_ENCAPSULATED, two values shall be provided. See note 2. If the action is DROP, no value shall be provided.
NOTE 1: Value indicates the priority in descending order, i.e. with 0 as the highest priority and 255 as the lowest priority. NOTE 2: Some applications (like inline/tap) require two interfaces. The first interface in the case of inline/tap is on the client (e.g. UE) side and the second on the core network (e.g. EPC) side.			

6.2.1.10 Type: TrafficFilter

6.2.1.10.1 Description

The TrafficFilter data type supports the specification of MEC application requirements related to traffic rules.

NOTE: How the TrafficFilter attributes can be determined prior to instantiation of a MEC application instance is out of scope of the present document.

6.2.1.10.2 Attributes

The attributes of TrafficFilter are shown in table 6.2.1.10.2-1.

Table 6.2.1.10.2-1: Attributes of TrafficFilter

Attribute name	Cardinality	Data type	Description
srcAddress	0..N	String	An IP address or a range of IP addresses. For IPv4, the IP address could be an IP address plus mask, or an individual IP address, or a range of IP addresses. For IPv6, the IP address could be an IP prefix, or a range of IP prefixes.
dstAddress	0..N	String	An IP address or a range of IP addresses. For IPv4, the IP address could be an IP address plus mask, or an individual IP address, or a range of IP addresses. For IPv6, the IP address could be an IP prefix, or a range of IP prefixes.
srcPort	0..N	String	A port or a range of ports.
dstPort	0..N	String	A port or a range of ports.
protocol	0..N	String	Specify the protocol of the traffic filter.
tag	0..N	String	Used for tag based traffic rule.
uri	0..N	String	An URI label, in application layer, i.e. in HTTP message, is used to filter the traffic.
packetLabel	0..N	String	A customized packet label in network layer, as defined by the owner of the MEC platform, is used to filter the traffic.
srcTunnelAddress	0..N	String	Used for GTP tunnel based traffic rule.
tgtTunnelAddress	0..N	String	Used for GTP tunnel based traffic rule.
srcTunnelPort	0..N	String	Used for GTP tunnel based traffic rule.
dstTunnelPort	0..N	String	Used for GTP tunnel based traffic rule.
qCI	0..1	Integer	Used to match all packets that have the same QCI.
dSCP	0..1	Integer	Used to match all IPv4 packets that have the same DSCP.
tC	0..1	Integer	Used to match all IPv6 packets that have the same TC.

6.2.1.11 Type: InterfaceDescriptor

6.2.1.11.1 Description

The InterfaceDescriptor data type describes an interface of a MEC application.

6.2.1.11.2 Attributes

The attributes of InterfaceDescriptor are shown in table 6.2.1.11.2-1.

Table 6.2.1.11.2-1: Attributes of InterfaceDescriptor

Attribute name	Cardinality	Data type	Description
interfaceType	1	Enum (inlined)	Type of interface: TUNNEL, MAC, IP, etc.
tunnelInfo	0..1	TunnelInfo	Included only if the destination address type is tunnel.
srcMACAddress	0..1	String	If the interface type is MAC, the source address identifies the MAC address of the interface.
dstMACAddress	0..1	String	If the interface type is MAC, the destination address identifies the MAC address of the destination. Only used for dstInterface.
dstIPAddress	0..1	String	If the interface type is IP, the destination address identifies the IP address of the destination. Only used for dstInterface.

6.2.1.12 Type: TunnelInfo

6.2.1.12.1 Description

The TunnelInfo data type supports the specification of MEC application requirements related to traffic rules.

6.2.1.12.2 Attributes

The attributes of TunnelInfo are shown in table 6.2.1.12.2-1.

Table 6.2.1.12.2-1: Attributes of TunnelInfo

Attribute name	Cardinality	Data type	Description
tunnelType	1	Enum (inlined)	Type of tunnel: GTP-U, GRE, etc.
tunnelDstAddress	1	String	Destination address of the tunnel.
tunnelSrcAddress	1	String	Source address of the tunnel.
tunnelSpecificData	0..1	Not Specified	Parameters specific to the tunnel.

6.2.1.13 Type: DNSRuleDescriptor

6.2.1.13.1 Description

The DNSRuleDescriptor data type describes DNS rules associated with a MEC application.

NOTE: How the IP addresses within the DNSRuleDescriptor can be determined prior to instantiation of a MEC application instance is out of scope of the present document.

6.2.1.13.2 Attributes

The attributes of DNSRuleDescriptor are shown in table 6.2.1.13.2-1.

Table 6.2.1.13.2-1: Attributes of DNSRuleDescriptor

Attribute name	Cardinality	Data type	Description
dnsRuleId	1	String	Identifies the DNS Rule
domainName	1	String	FQDN of the DNS rule
ipAddressType	1	Enum (inlined)	Specifies the IP address type, value: IP_V6, IP_V4
ipAddress	1	String	IP address given by the DNS rule
ttl	0..1	Integer	Time-to-live value

6.2.1.14 Type: LatencyDescriptor

6.2.1.14.1 Description

The LatencyDescriptor data type describes latency requirements for a MEC application.

6.2.1.14.2 Attributes

The attributes of LatencyDescriptor are shown in table 6.2.1.14.2-1.

Table 6.2.1.14.2-1: Attributes of LatencyDescriptor

Attribute name	Cardinality	Data type	Description
maxLatency	1	Uint32	The value of the maximum latency in nano seconds tolerated by the MEC application. See note.
NOTE: The latency is considered to be the one way end-to-end latency between the client application (e.g. in a device) and the service (i.e. the MEC application instance).			

6.2.1.15 Type: TerminateAppInstanceOpConfig

6.2.1.15.1 Description

The TerminateAppInstanceOpConfig data type supports the specification of MEC application requirements related to terminate application instance operation configuration.

6.2.1.15.2 Attributes

The attributes of TerminateAppInstanceOpConfig shall follow the definition in clause 7.1.5.7 of ETSI GS NFV-IFA 011 [1].

6.2.1.16 Type: ChangeAppInstanceStateOpConfig

6.2.1.16.1 Description

The ChangeAppInstanceStateOpConfig data type supports the specification of MEC application requirements related to change application instance state operation configuration.

6.2.1.16.2 Attributes

The attributes of ChangeAppInstanceStateOpConfig shall follow the definition in clause 7.1.5.8 of ETSI GS NFV-IFA 011 [1].

6.2.1.17 Type: ServiceDependency

6.2.1.17.1 Description

The ServiceDependency data type supports the specification of requirements of a service-consuming MEC application related to a MEC service.

6.2.1.17.2 Attributes

Attributes of a ServiceDependency are depicted in table 6.2.1.17.2-1.

Table 6.2.1.17.2-1: Attributes of ServiceDependency

Attribute name	Cardinality	Data type	Description
serName	1	String	The name of the service, for example, RNIS, LocationService, AMS, etc.
serCategory	0..1	CategoryRef	A Category reference of the service.
version	1	String	The version of the service.
serTransportDependencies	0..N	TransportDependency	Indicates transport and serialization format dependencies of consuming the service. Defaults to REST + JSON if absent. See note.
requestedPermissions	0..N	Not specified	Requested permissions regarding the access of the application to the service. See clause 7.2 of ETSI GS MEC 009 [4]. The format of this attribute is left for the data model design stage.
NOTE: This attribute indicates groups of transport bindings that a service-consuming MEC application supports for the consumption of the MEC service defined by this ServiceDependency structure. If at least one of the indicated groups is supported by the service it may be consumed by the application.			

6.2.1.18 Type: TransportDependency

6.2.1.18.1 Description

The TransportDependency data type supports the specification of requirements of a MEC application related to supported transport bindings (each being a combination of a transport with one or more serializers).

6.2.1.18.2 Attributes

The attributes of a TransportDependency are depicted in table 6.2.1.18.2-1.

Table 6.2.1.18.2-1: Attributes of TransportDependency

Attribute name	Cardinality	Data type	Description
transport	1	TransportDescriptor	Information about the transport in this transport binding.
serializers	1..N	SerializerType	Information about the serializers in this transport binding, as defined in the SerializerType type in ETSI GS MEC 011 [17]. Support for at least one of the entries is required in conjunction with the transport.
labels	1..N	String	Set of labels that allow to define groups of transport bindings. The mechanism of the grouping is defined below this table.

Each "labels" value identifies a group of transport bindings. In a list of TransportDependency structures, all entries that have a "labels" entry with the same value belong to the same group. Each group indicates an alternative set of transport bindings. At least one group of transport bindings needs to be supported to fulfil the requirements.

EXAMPLE 1: An application requires REST_HTTP transport with JSON.

List of TransportDependency structures:

{transport=REST_HTTP, serializers=[JSON], labels=[A]}

EXAMPLE 2: An application can run with JSON or PROTOBUF3 over a topic-based message bus.

List of TransportDependency structures:

{transport=MB_TOPIC_BASED, serializers=[JSON, PROTOBUF3], labels=[A])

EXAMPLE 3: An application requires REST transport with JSON or a topic-based message bus with PROTOBUF3.

List of TransportDependency structures:

{transport=REST_HTTP, serializers=[JSON], labels=[A]},

{MB_TOPIC_BASED, serializers=[PROTOBUF3], labels=[B]}

EXAMPLE 4: An application requires both REST transport with JSON and a topic-based message bus with PROTOBUF3.

List of TransportDependency structures:

{transport=REST_HTTP, serializers=[JSON], labels=[A]},

{transport=MB_TOPIC_BASED, serializers=[PROTOBUF3], labels=[A]}

EXAMPLE 5: An application requires both REST transport with JSON and a topic-based message bus with PROTOBUF3 or Websockets with PROTOBUF3.

List of TransportDependency structures:

{transport=REST_HTTP, serializers=[JSON], labels=[A, B]},

{transport=MB_TOPIC_BASED, serializers=[PROTOBUF3], labels=[A]}

{transport=WEBSOCKETS, serializers=[PROTOBUF3], labels=[B]}

6.2.1.19 Type: TransportDescriptor

6.2.1.19.1 Description

The TransportDescriptor data type describes a transport.

6.2.1.19.2 Attributes

The attributes of a TransportDescriptor are depicted in table 6.2.1.19.2-1.

Table 6.2.1.19.2-1: Attributes of TransportDescriptor

Attribute name	Cardinality	Data type	Description
name	1	String	The name of this transport.
description	0..1	String	Human-readable description of this transport.
type	1	TransportTypes	Type of the transport, as defined in the TransportTypes type in ETSI GS MEC 011 [17].
protocol	1	String	The name of the protocol used. Shall be set to "HTTP" for a REST API.
version	1	String	The version of the protocol used.
security	1	SecurityInfo	Information about the security used by the transport in ETSI GS MEC 011 [17].
implSpecificInfo	0..1	Not specified	Additional implementation specific details of the transport.

6.2.1.20 Type: UserContextTransferCapability

6.2.1.20.1 Description

This data type represents the information of user context transfer capability of application.

6.2.1.20.2 Attributes

The attributes of UserContextTransferCapability are depicted in table 6.2.1.20.2-1.

Table 6.2.1.20.2-1: Attributes of UserContextTransferCapability

Attribute name	Cardinality	Data type	Description
statefulApplication	1	Boolean	If the application is stateful, this attribute shall be set to true. Otherwise, this attribute shall be set to false.
userContextTransferSupport	0..1	Boolean	This attribute shall be present if the application is stateful and shall be absent otherwise. If the application supports the user context transfer, this attribute shall be set to true. Otherwise this attribute shall be set to false.

6.2.1.21 Type: AppNetworkPolicy

6.2.1.21.1 Description

This data type represents the network policy in the application instantiation and operation.

6.2.1.21.2 Attributes

The attributes of AppNetworkPolicy are depicted in table 6.2.1.21.2-1.

Table 6.2.1.21.2-1: Attributes of AppNetworkPolicy

Attribute name	Cardinality	Data type	Description
steeredNetwork	1	Structure (inlined)	This attribute provides an option for the application to specify a type of network to carry the application traffic. See note.
>cellularNetwork	0..1	Boolean	If present, and the application prefers to a cellular network to carry its traffic, this attribute shall be set to true. Otherwise, it shall be set to false.
>wi-fiNetwork	0..1	Boolean	If present, and the application prefers to a Wi-Fi® network to carry its traffic, this attribute shall be set to true. Otherwise, it shall be set to false.
>fixedAccessNetwork	0..1	Boolean	If present, and the application prefers to a fixed access network to carry its traffic, this attribute shall be set to true. Otherwise, it shall be set to false.
NOTE: The network types may depend on the availability of deployed access networks of MEC system.			

6.2.1.22 Type: OsContainerDescriptor

6.2.1.22.1 Description

The OsContainerDescriptor data type supports the specification of requirements of container compute resources when a MEC application is intended to be realized by one or a set of OS Containers sharing the same host and same networking namespace.

6.2.1.22.2 Attributes

The attributes of OsContainerDescriptor shall follow the definition in clause 7.1.6.13.2 of ETSI GS NFV-IFA 011 [1], with the following consideration:

- A VNF corresponds to MEC application, a VNFD corresponds to an AppD in MEC, and the VDU in table 7.1.6.2.2-1 of ETSI GS NFV-IFA 011 [**1**] corresponds to a MEC application.
- The swImageDesc in table 7.1.6.13.2-1 of ETSI GS NFV-IFA 011 [1] refers to swImageDescriptor in the present document.

6.2.1.23 Type: McioIdentificationData

6.2.1.23.1 Description

The McioIdentificationData data type contains data needed to identify an MCIO when interworking with the CISM.

6.2.1.23.2 Attributes

The attributes of McioIdentificationData shall follow the definition in clause 6.2.75.2 of ETSI GS NFV-SOL 001 [19].

6.2.1.24 Type: LogicalNodeRequirements

6.2.1.24.1 Description

The LogicalNodeRequirements data type describes compute, memory and I/O requirements that are to be associated with the logical node of infrastructure. As an example for illustration purposes, a logical node correlates to the concept of a NUMA cell in libvirt terminology.

6.2.1.24.2 Attributes

The attributes of LogicalNodeRequirements shall follow the definition in clause 7.1.9.6.2 of ETSI GS NFV-IFA 011 [1], with the following consideration:

- A VNF corresponds to MEC application, a VNFD corresponds to an AppD in MEC, and the VDU in table 7.1.6.2.2-1 of ETSI GS NFV-IFA 011 [1] corresponds to a MEC application.

6.2.1.25 Type: RequestedAdditionalCapabilityData

6.2.1.25.1 Description

The RequestedAdditionalCapabilityData data type supports the specification of requested additional capability for a particular application. Such a capability may be for acceleration or specific tasks.

6.2.1.25.2 Attributes

The attributes of RequestedAdditionalCapabilityData shall follow the definition in clause 7.1.9.5.2 of ETSI GS NFV-IFA 011 [1], with the following consideration:

- A VNF corresponds to MEC application, a VNFD corresponds to an AppD in MEC, and the VDU in table 7.1.6.2.2-1 of ETSI GS NFV-IFA 011 [1] corresponds to a MEC application.

6.2.2 Application lifecycle management information model

6.2.2.1 Introduction

This clause defines data structure to be used by application lifecycle management information model.

6.2.2.2 Type: LocationConstraints

6.2.2.2.1 Description

The LocationConstraints data type supports the specification of MEC application requirements related to MEC application deployment location constraints.

The location constraints can be represented as follows:

- as a country code;
- as a civic address combined with a country code;
- as an area, conditionally combined with a country code.

6.2.2.2.2 Attributes

The attributes of LocationConstraints are shown in table 6.2.2.2.2-1.

Table 6.2.2.2.2-1: Attributes of LocationConstraints

Attribute name	Data type	Cardinality	Description
countryCode	String	0..1	The two-letter ISO 3166 [3] country code in capital letters. Shall be present in the case that "area" attribute is absent. May be present if the "area" attribute is present. See note.
civicAddressElement	array(Structure(inlined))	0..N	Zero or more elements comprising the civic address. Shall be absent if the "area" attribute is present.
>caType	Integer	1	Describe the content type of caValue. The value of caType shall comply with section 3.4 of IETF RFC 4776 [2].
>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 [2].
area	Polygon (see IETF RFC 7946 [8])	0..1	Geographic area. Shall be absent if the "civicAddressElement" attribute is present. The content of this attribute shall follow the provisions for the "Polygon" geometry object as defined in IETF RFC 7946 [8], for which the "type" member shall be set to the value "Polygon". See note.
NOTE: If both "countryCode" and "area" are present, no conflicts should exist between the values of these two attributes. In case of conflicts, the API producer (e.g. MEO, MEAO) shall disregard parts of the geographic area signalled by "area" that are outside the boundaries of the country signalled by "countryCode". If "countryCode" is absent, it is solely the "area" attribute that defines the location constraint.			

6.2.2.3 Type: CreateAppInstanceRequest

6.2.2.3.1 Description

The data type of CreateAppInstanceRequest represents the parameters for creating a new application instance resource. It is used by the resource of application instances in clause 7.5.1. It shall comply with attributes in clause 6.2.2.3.2.

6.2.2.3.2 Attributes

The attributes of CreateAppInstanceRequest data type shall follow the specification in table 6.2.2.3.2-1.

Table 6.2.2.3.2-1: Attributes of CreateAppInstanceRequest

Attribute name	Cardinality	Data type	Description
appDId	1	String	The application descriptor identifier. It is managed by the application provider to identify the application descriptor in a globally unique way.
appInstanceName	0..1	String	Human-readable name of the application instance to be created.
appInstanceDescription	0..1	String	Human-readable description of the application instance to be created.
appPlacementInfo	0..1	MepInformation	Describes the information of selected MEC platform for the application instance to associate. See note.
NOTE: This field applies to Mm3* reference point only.			

6.2.2.4 Type: AppInstanceInfo

6.2.2.4.1 Description

The data type of AppInstanceInfo represents the parameters of instantiated application instance resources. It is used by the resource of application instances in clause 7.4.1 and the resource of individual application instance in clause 7.4.2.

6.2.2.4.2 Attributes

The attributes of AppInstanceInfo data type are specified in the table 6.2.2.4.2-1.

Table 6.2.2.4.2-1: Attributes of AppInstanceInfo

Attribute name	Cardinality	Data type	Description
id	1	String	Identifier of the application instance represented by this data type.
appInstanceName	0..1	String	Name of the application instance.
appInstanceDescription	0..1	String	Human-readable description of the application instance to be created.
appDId	1	String	The application descriptor identifier is managed by the application provider to identify the application descriptor in a globally unique way. It is copied from the AppD of the onboarded application package.
appProvider	1	String	The onboarded application package provider name.
appName	1	String	The onboarded application name.
appSoftVersion	1	String	The application software version.
appDVersion	1	String	Version of the application descriptor.
appPkgId	1	String	Identifier of the onboarded application package.
vimConnectionInfo	0..N	map(VimConnectionInfo)	Information about VIM connections to be used for managing the resources for the application instance. The keys of the map, each of which identifies information about a particular VIM connection, are managed by the MEO and referenced from other data structures via the "vimConnectionId" attribute. See notes 1 and 3.
nsInstanceId	0..1	String	Identifier of the NS instance created by NFVO in which the MEC application has been instantiated as a VNF instance. See note 2.
vnfInstanceId	0..1	String	Identifier of the VNF instance created by VNFM that the MEC application has been instantiated as. See note 2.
instantiationState	1	Enum (inlined)	Instantiation state of the application instance: • NOT_INSTANTIATED: the application instance is not instantiated. • INSTANTIATED: the application instance has been instantiated.
instantiatedAppState	0..1	Structure (inlined)	Information specific to an instantiated application. This attribute shall be present if the instantiationState attribute value is INSTANTIATED.
>operationalState	1	Enum (inlined)	Operational state is applicable in the instantiation state INSTANTIATED: • STARTED: the application instance is up and running. • STOPPED: the application instance stops operation.
>appInstLocation	0..1	LocationInformation	Location of the MEC application instance. See note 5 and note 6.
>mcioInfo	0..N	array(McioInfo)	Information on the MCIO(s) representing application instance realized by one or a set of OS containers. See note 7.
communicationInterface	0..1	CommunicationInterface	Interface for communication with other application instances. See clause 7.5.2 of ETSI GS MEC 021 [13] for the data type definition.
_links	1	Structure (inlined)	Links to resources related to this resource.
>self	1	LinkType	Self referring URI.
>instantiate	0..1	LinkType	Link to the "instantiate" task resource, if the related operation is possible based on the current status of this application instance resource (i.e. application instance in NOT_INSTANTIATED state). See note 3.
>terminate	0..1	LinkType	Link to the "terminate" task resource, if the related operation is possible based on the current status of this application instance resource (i.e. application instance is in INSTANTIATED state).

Attribute name	Cardinality	Data type	Description
>operate	0..1	LinkType	Link to the "operate" task resource, if the related operation is supported for this application instance, and is possible based on the current status of this application instance resource (i.e. application instance is in INSTANTIATED state).
>configure_platform_for_app	0..1	LinkType	Link to the "configure_platform_for_app" task resource, if the related operation is supported for this application instance, and is possible based on the current status of this application instance resource (i.e. application instance is in INSTANTIATED state). See note 4.
NOTE 1: This field does not apply if the data structure is used by MEAO. NOTE 2: This field applies if the data structure is used by MEAO. NOTE 3: This field does not apply if the data structure is used on Mm3*. NOTE 4: This field applies if the data structure is used on Mm3*. NOTE 5: This field applies if the data structure is used on Mm1 or Mm3*. NOTE 6: It is not specified in the present document how location information is obtained in the case of MEC in NFV. NOTE 7: This attribute reflects the ETSI NFV interpretation of the cloud native workloads.			

6.2.2.5 Type: AppInstanceSubscriptionFilter

6.2.2.5.1 Description

This data type represents subscription filter criteria to match application instances. The AppInstanceSubscriptionFilter shall comply with provisions in clause 6.2.2.5.2.

6.2.2.5.2 Attributes

The attributes of the data type are specified in table 6.2.2.5.2-1.

Table 6.2.2.5.2-1: Attributes of AppInstanceSubscriptionFilter

Attribute name	Cardinality	Data type	Description
appInstSelectorType	1	Enum (Inlined)	0 = void 1 = APP_IDENTITY 2 = APP_NAME 3 = APP_D_ID 4 = APP_FROM_PROVIDER
appInstances	0..N	array(String)	If appInstIdSelector = APP_IDENTITY match existing application instances with an "application instance identifier" listed in this attribute. If appInstIdSelector = APP_NAME match existing application instances with an "application instance name" listed in this attribute. If appInstIdSelector = APP_D_ID match existing application instances, or those created in the future whilst the subscription is active, based on the application descriptors identified by one of the "application descriptor identities" listed in this attribute. If appInstIdSelector = APP_FROM_PROVIDER this attribute shall not be included.
appsFromProviders	0..N	array(Structure (inlined))	Present only if appInstIdSelector = APP_FROM_PROVIDER. Match existing application instances, or those created in the future whilst the subscription is active, that belong to applications from certain providers.
>appProvider	1	String	Name of the application provider to match.
>appProducts	0..N	array(Structure (inlined))	If present, match application instances that belong to application products with certain product names, from one particular provider.
>>appName	1	String	Name of the application product to match.

Attribute name	Cardinality	Data type	Description
>>versions	0..N	array(Structure (inlined))	If present, match application instances that belong to application products with certain versions and a certain product name, from one particular provider.
>>>appSoftVersion	1	String	Application software version to match.
>>>appDVersion	0..N	array(String)	If present, match application instances that belong to application products with certain appD versions, a certain software version and a certain product name, from one particular provider.

6.2.2.6 Type: AppLcmOpOccSubscriptionFilter

6.2.2.6.1 Description

This data type represents a subscription filter criteria to match an application LCM operation occurrence. The AppLcmOpOccSubscriptionFilter shall comply with provisions in clause 6.2.2.6.2.

6.2.2.6.2 Attributes

The attributes of the data type are specified in table 6.2.2.6.2-1.

Table 6.2.2.6.2-1: Attributes of AppLcmOpOccSubscriptionFilter

Attribute name	Cardinality	Data type	Description
appInstanceSubscriptionFilter	0..1	AppInstanceSubscriptionFilter	If present, this attribute contains filter criteria that selects one or more application instances on which to receive "LCM operation occurrence" notifications.
notificationTypes	0..1	String	Match particular notification types. Permitted values: AppLcmOpOccNotification.
operationTypes	0..N	array(Enum (inlined))	Type of the LCM operation represented by this application instance LCM operation occurrence. Permitted values: • INSTANTIATE. • OPERATE. • TERMINATE. Match particular application lifecycle operation types for the notification of AppLcmOpOccNotification. May be present if the "notificationTypes" attribute contains the value "AppLcmOpOccNotification", and shall be absent otherwise.

Attribute name	Cardinality	Data type	Description
operationStates	0..N	array(Enum (inlined))	Type of the LCM operation state represented by this application instance LCM operation occurrence. Permitted values: • STARTING. • PROCESSING. • COMPLETED. • FAILED. • FAILED_TEMP Match particular LCM operation state values as reported in notifications of AppLcmOpOccNotification. May be present if the "notificationTypes" attribute contains the value "AppLcmOpOccNotification", and shall be absent otherwise.

6.2.2.7 Type: InstantiateAppRequest

6.2.2.7.1 Description

This data type represents request parameters of the "Instantiate Application" operation. It shall comply with the provisions in clause 6.2.2.7.2, which aligns with the clause 6.3.1.3.

6.2.2.7.2 Attributes

The attributes of data type are specified in the table 6.2.2.7.2-1.

Table 6.2.2.7.2-1: Attributes of InstantiateAppRequest

Attribute name	Cardinality	Data type	Description
virtualComputeDescriptor	0..1	VirtualComputeDescriptor	Describes CPU and memory requirements, as well as optional additional requirements, such as disk and acceleration related capabilities, of the single VM to realize the application instance to be created. See notes 1 and 4.
osContainerDescriptor	0..N	array(OsContainerDescriptor)	Describes CPU, memory requirements and limits, and software images of the OS Containers realizing this MEC application corresponding to OS Containers sharing the same host and same network namespace. See notes 1, 4 and 5.
virtualStorageDescriptor	0..N	array(VirtualStorageDescriptor)	Defines descriptors of virtual storage resources to be used by the application instance to be created. See note 1.
selectedMECHostInfo	1..N	array(MECHostInformation)	Describes the information of selected host for the application instance. See note 2.
locationConstraints	0..1	LocationConstraints	Defines the location constraints for the application instance to be created. See note 3.
vimConnectionInfo	0..N	map(VimConnectionInfo)	Information about VIM connections to be used for managing the resources for the application instance, or refer to external/externally-managed virtual links. This attribute shall only be supported and may be present if application-related resource management in direct mode is applicable. See note 2.

Attribute name	Cardinality	Data type	Description
appTermCandsForCoord	0..1	AppTermCandsForCoord	Provides sets of applications as termination candidate alternatives that the MEO/MEAO shall select from when utilizing the coordinate LCM operation exchange in pre-emption situations (see step 3 in clause 5.3.1). If this attribute is omitted, the MEO/MEAO shall make its own selection for the coordinate LCM operation exchange. See note 3.
NOTE 1: This attribute may be provided in the InstantiateAppRequest structure to override the same attribute in the AppD. NOTE 2: This field applies to Mm3 reference point only. NOTE 3: This field applies to Mm1 reference point only. NOTE 4: Only one of virtualComputeDescriptor or osContainerDescriptor shall be present. NOTE 5: This attribute reflects the ETSI NFV interpretation of the cloud native workloads.			

6.2.2.8 Type: OperateAppRequest

6.2.2.8.1 Description

This data type represents request parameters of the "Operate Application" operation. It shall comply with the provisions in clause 6.2.2.8.2, which aligns with the clause 6.3.1.4.

6.2.2.8.2 Attributes

The attributes of data type are specified in the table 6.2.2.8.2-1.

Table 6.2.2.8.2-1: Attributes of OperateAppRequest

Attribute name	Cardinality	Data type	Description
changeStateTo	1	Enum (inlined)	The desired operational state: • STARTED: the application instance is up and running. • STOPPED: the application instance stops operation.
stopType	0..1	Enum (inlined)	The stop type: • FORCEFUL: it will stop the application immediately after accepting the request. • GRACEFUL: it will first arrange to take the application instance out of service after accepting the request. Once that operation is successful or once the timer value specified in the "gracefulStopTimeout" attribute expires, it will stop the application. See notes 1 and 3.
gracefulStopTimeout	0..1	Integer	The time interval (in seconds) to wait for the application instance to be taken out of service during graceful stop, before stopping the application. See note 1 and note 2.
NOTE 1: The "stopType" and "gracefulStopTimeout" attributes shall be absent, when the "changeStateTo" attribute is equal to "STARTED". NOTE 2: The "gracefulStopTimeout" attribute shall be present, when the "changeStateTo" 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". NOTE 3: 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.			

6.2.2.9 Type: TerminateAppRequest

6.2.2.9.1 Description

This data type represents request parameters of the "Terminate Application Request" operation. It shall comply with the provisions in clause 6.2.2.9.2, which aligns with the clause 6.3.1.7.

6.2.2.9.2 Attributes

The attributes of data type are specified in the table 6.2.2.9.2-1.

Table 6.2.2.9.2-1: Attributes of TerminateAppRequest

Attribute name	Cardinality	Data type	Description
terminationType	1	Enum (inlined)	Indicates whether forceful or graceful termination is requested. • FORCEFUL: it will shut down the application instance and release the resources immediately after accepting the request. See note. • GRACEFUL: it will first arrange to take the application instance out of service after accepting the request. Once the operation of taking the application instance out of service finishes or once the timer value specified in the "gracefulTerminationTimeout" attribute expires, it will shut down the application instance and release the resources.
gracefulTerminationTimeout	0..1	Integer	This attribute is only applicable in case of graceful termination. It defines the time to wait for the application instance to be taken out of service before shutting down the application and releasing the resources. The unit is seconds. If not given and the "terminationType" attribute is set to "GRACEFUL", it is expected to wait for the successful taking out of service of the application, no matter how long it takes, before shutting down the application and releasing the resources.
NOTE: If the application instance is still in service, requesting forceful termination can adversely impact service.			

6.2.2.10 Type: AppInstSubscriptionInfo

6.2.2.10.1 Description

The data type represents a subscription to notification of application instance operational state change. It shall comply with provisions in clause 6.2.2.10.2.

6.2.2.10.2 Attributes

The attributes of data type are specified in the table 6.2.2.10.2-1.

Table 6.2.2.10.2-1: Attributes of AppInstSubscriptionInfo

Attribute name	Cardinality	Data type	Description
id	1	String	Identifier of the subscription to application instance operational state change notification.
subscriptionType	1	String	Shall be set to "AppInstanceStateChangeSubscription".
appInstanceState	0..1	Enum (inlined)	Application instance state subscribed to: • NOT_INSTANTIATED: the application instance is not instantiated. • STARTED: the application instance is up and running. • STOPPED: the application instance stops operation.
appInstanceSubscriptionFilter	0..1	AppInstanceSubscriptionFilter	Criteria used to select application instances on which to send notifications related to this subscription.
callbackUri	1	Uri	The URI of the endpoint for the subscription related notification to be sent to.
_links	1	Structure (inlined)	Links to resources related to this resource.
>self	1	LinkType	URI of this resource.

6.2.2.11 Type: AppInstNotification

6.2.2.11.1 Description

This data type represents an application instance notification for informing the subscribers about operational state of application instance resources. It shall comply with provisions in clause 6.2.2.11.2.

6.2.2.11.2 Attributes

The attributes of data type are specified in the table 6.2.2.11.2-1.

Table 6.2.2.11.2-1: Attributes of AppInstNotification

Attribute name	Cardinality	Data type	Description
id	1	String	Identifier of this notification. If a notification is sent multiple times due to multiple subscriptions, the "notificationId" attribute of all these notifications shall have the same value.
notificationType	1	String	Discriminator for the different notification types. Shall be set to "AppInstanceStateChangeSubscription" for this notification type.
appInstanceState	1	Enum (inlined)	Application instance state: • NOT_INSTANTIATED: the application instance is not instantiated. • STARTED: the application instance is up and running. • STOPPED: the application instance stops operation.
subscriptionId	1	String	Identifier of the subscription related to this notification.
timeStamp	1	TimeStamp	Date and time of the notification generation.
appInstanceId	1	String	Identifier of application instance.
appPkgId	1	String	Identifier of the onboarded application package.
appDId	1	String	The application descriptor identifier identifies the application package and the application descriptor in a globally unique way.
appInstLocation	0..1	LocationInformation	Location of the MEC application instance. Shall be present if the application instance is instantiated and shall be absent otherwise.
_links	1	Structure (inlined)	Links to resources related to this notification.
>subscription	1	LinkType	A link to the related subscription.

6.2.2.12 Type: AppInstSubscriptionRequest

6.2.2.12.1 Description

The data type represents the input parameters of "subscription operation" to notification of application lifecycle management for the operational state change of application instance.

6.2.2.12.2 Attributes

The attributes of the data type are specified in the table 6.2.2.12.2-1.

Table 6.2.2.12.2-1: Attributes of AppInstSubscriptionRequest

Name	Data type	Cardinality	Remarks
subscriptionType	String	1	Shall be set to "AppInstanceStateChangeSubscription".
callbackUri	Uri	1	The URI of the endpoint for the subscription related notification to be sent to.
appInstanceState	Enum (inlined)	0..1	Only send notifications for application instances that are in one of the states listed in this attribute. If this attribute is absent, match all states. Application states: • NOT_INSTANTIATED: the application instance is not instantiated. • STARTED: the application instance is up and running. • STOPPED: the application instance stops operation.
appInstanceSubscriptionFilter	AppInstanceSubscriptionFilter	0..1	Criteria used to filter application instances for which to send notifications related to this subscription.

6.2.2.13 Type: AppLcmOpOcc

6.2.2.13.1 Description

This data type represents an application lifecycle management operation occurrence. It shall comply with the provisions in clause 6.2.2.13.2.

6.2.2.13.2 Attributes

The attributes of data type are specified in the table 6.2.2.13.2-1.

Table 6.2.2.13.2-1: Attributes of AppLcmOpOcc

Attribute name	Cardinality	Data type	Description
id	1	String	Identifier of the subscription to application LCM operation occurrence notification.
operationState	1	Enum (inlined)	Operation state: • STARTING: the LCM operation starting. • PROCESSING: the LCM operation is currently in execution. • COMPLETED: the LCM operation has been completed. • FAILED: The LCM operation has failed and it cannot be retried, as it is determined that such action will not succeed. • FAILED_TEMP: The LCM operation has failed and execution has stopped, but the execution of the operation is not considered to be closed.
stateEnteredTime	1	TimeStamp	Date and time when the current state was entered.
startTime	1	TimeStamp	Date and time of the start of the operation.
lcmOperation	1	Enum (inlined)	Type of the actual LCM operation represented by this application instance LCM operation occurrence: • INSTANTIATE. • OPERATE. • TERMINATE. This attribute is associated to the operationParams.

Attribute name	Cardinality	Data type	Description
operationParams	0..1	InstantiateAppRequest OR OperateAppRequest OR TerminateAppRequest	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 LCM operation and the data type of this attribute shall apply: • INSTANTIATE: InstantiateAppRequest. • OPERATE: OperateAppRequest. • TERMINATE: TerminateAppRequest. This attribute shall be present if this data type is returned in a response to reading an individual resource. See note 2.
isCancelPending	0..1	Boolean	If the application LCM operation occurrence operationState is in "PROCESSING" state and the operation is being cancelled, this attribute shall be set to true. Otherwise, it shall be set to false.
cancelMode	0..1	CancelMode	The mode of a cancellation: • GRACEFUL: Indicates ongoing resource management operations in the underlying system are being allowed to complete execution or timing out. • FORCEFUL: Indicates ongoing resource management operations in the underlying system are being forcefully cancelled. Shall be present when isCancelPending equals true and shall be absent otherwise.
_links	1	Structure (inlined)	Links to resources related to this resource.
>self	1	LinkType	URI of this resource.
>appInstance	1	LinkType	Link to the application instance that the operation applies to.
NOTE 1: Void. NOTE 2: This object contains structured data, and shall comply with the provisions of clause 4 of IETF RFC 8259 [5].			

6.2.2.14 Type: AppLcmOpOccSubscriptionRequest

6.2.2.14.1 Description

This data type represents a subscription request to notification of application life cycle management operation occurrence. It shall comply with provisions in clause 6.2.2.14.2.

6.2.2.14.2 Attributes

The attributes of data type are specified in the table 6.2.2.14.2-1.

Table 6.2.2.14.2-1: Attributes of AppLcmOpOccSubscriptionRequest

Attribute name	Cardinality	Data type	Description
subscriptionType	1	String	Shall be set to "AppLcmOpOccStateChangeSubscription".
callbackUri	1	Uri	The URI of the endpoint for the subscription related notification to be sent to.
appLcmOpOccSubscriptionFilter	0..1	AppLcmOpOccSubscriptionFilter	Subscription filter criteria to match specific application LCM operation occurrences.

6.2.2.15 Type: AppLcmOpOccSubscriptionInfo

6.2.2.15.1 Description

This data type represents a subscription to notifications of application life cycle management operation occurrence. It shall comply with provisions in clause 6.2.2.15.2.

6.2.2.15.2 Attributes

The attributes of data type are specified in the table 6.2.2.15.2-1.

Table 6.2.2.15.2-1: Attributes of AppLcmOpOccSubscriptionInfo

Attribute name	Cardinality	Data type	Description
id	1	String	Identifier of this subscription resource.
subscriptionType	1	String	Shall be set to "AppLcmOpOccStateChangeSubscription".
callbackUri	1	Uri	The URI of the endpoint for the subscription related notification to be sent to.
appLcmOpOccSubscriptionFilter	0..1	AppLcmOpOccSubscriptionFilter	Criteria used to select application LCM operation occurrences on which to send notifications related to this subscription.
_links	1	Structure (inlined)	Links to resources related to this resource.
>self	1	LinkType	URI of this resource.

6.2.2.16 Type: AppLcmOpOccNotification

6.2.2.16.1 Description

This data type represents a notification related to state changes of an application LCM operation occurrence which informs the subscribers. It shall comply with provisions in clause 6.2.2.16.2.

6.2.2.16.2 Attributes

The attributes of data type are specified in the table 6.2.2.16.2-1.

Table 6.2.2.16.2-1: Attributes of AppLcmOpOccNotification

Attribute name	Cardinality	Data type	Description
id	1	String	Identifier of this notification. If a notification is sent multiple times due to multiple subscriptions, the "notificationId" attribute of all these notifications shall have the same value.
notificationType	1	String	Discriminator for the different notification types. Shall be set to "AppLcmOpOccStateChangeSubscription" for this notification type.
operationType	1	Enum (inlined)	Type of the LCM operation represented by this application instance LCM operation occurrence. Permitted values: • INSTANTIATE. • OPERATE. • TERMINATE.
operationState	1	Enum (inlined)	Operation state: • STARTING: the LCM operation starting. • PROCESSING: the LCM operation is currently in execution. • COMPLETED: the LCM operation has been completed. • FAILED: The LCM operation has failed and it cannot be retried, as it is determined that such action will not succeed. • FAILED_TEMP: The LCM operation has failed and execution has stopped, but the execution of the operation is not considered to be closed.
subscriptionId	1	String	Identifier of the subscription to this notification.
timeStamp	1	TimeStamp	Date and time of the notification generation.
appLcmOpOccId	1	String	Identifier of application lifecycle management operation occurrence.
appInstanceId	1	String	Identifier of application instance.
_links	1	Structure (inlined)	Links to resources related to this notification.

Attribute name	Cardinality	Data type	Description
>appInstance	1	LinkType	Link to the resource representing the application instance to which the notified change applies.
>subscription	1	LinkType	Link to the related subscription.
>appLcmOpOcc	1	LinkType	Link to the application lifecycle management operation occurrence that this notification is related to.

6.2.2.17 Type: MECHostInformation

6.2.2.17.1 Description

The data type represents the parameters of MEC host information.

6.2.2.17.2 Attributes

The attributes of the data type are specified in table 6.2.2.17.2-1.

Table 6.2.2.17.2-1: Attributes of MECHostInformation

Attribute name	Cardinality	Data type	Description
hostName	0..1	String	Human-readable name of MEC host.
hostId	1	KeyValuePairs	Deployment-specific information to identify a MEC host. See note.
NOTE: This information can be structured to cater for host identification schemes that are more complex than a simple identifier, e.g. when referring to the structure of an NFVI.			

6.2.2.18 Type: VimConnectionInfo

6.2.2.18.1 Description

The VimConnectionInfo data type specifies the connection information of VIM for managing the resources of the application instance.

6.2.2.18.2 Attributes

The attributes of VimConnectionInfo data type are specified in the table 6.2.2.18.2-1.

Table 6.2.2.18.2-1: Attributes of VimConnectionInfo

Attribute name	Cardinality	Data type	Description
id	1	String	The identifier of the VIM Connection. This identifier is managed by the MEO.
vimId	0..1	String	The identifier of the VIM instance. This identifier is managed by the MEO. Shall be present to address additional information about the VIM if such information has been configured into the MEPM by means outside the scope of the present document, and should be absent otherwise.
vimType	1	String	Discriminator for the different types of the VIM information. The value of this attribute determines the structure of the "interfaceInfo" and "accessInfo" attributes, based on the type of the VIM. The set of permitted values is expected to change over time as new types or versions of VIMs become available.

Attribute name	Cardinality	Data type	Description
interfaceInfo	0..1	KeyValuePairs	Information about the interface or interfaces to the VIM, if applicable, such as the URI of an interface endpoint to communicate with the VIM. The applicable keys are dependent on the content of vimType. Alternatively, such information may have been configured into the VNFM and bound to the vimId.
accessInfo	0..1	KeyValuePairs	Authentication credentials for accessing the VIM, and other access-related information such as tenants or infrastructure resource groups. The applicable keys are dependent on the content of vimType. If the VimConnectionInfo structure is part of an HTTP response message content, sensitive attributes that are children of this attributes (such as passwords) shall not be included. If the VimConnectionInfo structure is part of an HTTP request message content, sensitive attributes that are children of this attribute (such as passwords) shall be present if they have not been provisioned out of band.
extra	0..1	KeyValuePairs	VIM type specific additional information. The applicable structure, and whether or not this attribute is available, is dependent on the content of vimType.

6.2.2.19 Type: AppInstanceSubscriptionLinkList

6.2.2.19.1 Description

The data type represents a subscription link list of notification on application lifecycle management. It shall comply with provisions in clause 6.2.2.19.2.

6.2.2.19.2 Attributes

The attributes of data type are specified in the table 6.2.2.19.2-1.

Table 6.2.2.19.2-1: Attributes of SubscriptionLinkList

Attribute name	Cardinality	Data type	Description
_links	1	Structure (inlined)	List of hyperlinks related to the resource.
>self	1	LinkType	URI of this resource.
>subscriptions	0..N	array (Structure (inlined))	A link list to the subscriptions.
>>href	1	Uri	The URI referring to the subscription.
>>subscriptionType	1	AppInstanceSubscriptionType	Type of the subscription.

6.2.2.20 Referenced simple data types and enumerations

6.2.2.20.1 Introduction

This clause defines simple data types and enumerations.

6.2.2.20.2 Simple data types

The simple data type defined for this API are provided in table 6.2.2.20.2-1.

Table 6.2.2.20.2-1: Simple data types

Type name	Description
AppInstanceSubscriptionType	String representing the type of a subscription. Permitted values: • "AppInstanceStateChangeSubscription": subscription to notifications relating application operational state change. • "AppLcmOpOccStateChangeSubscription": subscription to notification relating application lifecycle management operation occurrence state change. • "AppIdentifierCreationSubscription": subscription to notification relating application instance identifier creation. • "AppIdentifierDeletionSubscription": subscription to notification relating application instance identifier deletion.

6.2.2.20.3 Enumeration

None defined.

6.2.2.21 Type: ConfigPlatformForAppRequest

6.2.2.21.1 Description

This data type represents the parameters for configuring the MEP to run an application instance. It shall comply with the provisions in clause 6.2.2.21.2.

6.2.2.21.2 Attributes

The attributes of data type are specified in the table 6.2.2.21.2-1.

Table 6.2.2.21.2-1: Attributes of ConfigPlatformForAppRequest

Attribute name	Cardinality	Data type	Description
appServiceRequired	0..N	array(ServiceDependency)	Describes services a MEC application requires to run.
appServiceOptional	0..N	array(ServiceDependency)	Describes services a MEC application may use if available.
appServiceProduced	0..N	array(ServiceDescriptor)	Describes services a MEC application is able to produce to the platform or other MEC applications. Only relevant for service-producing apps.
appFeatureRequired	0..N	array(FeatureDependency)	Describes features a MEC application requires to run.
appFeatureOptional	0..N	array(FeatureDependency)	Describes features a MEC application may use if available.
transportDependencies	0..N	array(TransportDependency)	Transports, if any, that this application requires to be provided by the platform. These transports will be used by the application to deliver services provided by this application. Only relevant for service-producing apps.
appTrafficRule	0..N	array(TrafficRuleDescriptor)	Describes traffic rules the MEC application requires.
appDNSRule	0..N	array(DNSRuleDescriptor)	Describes DNS rules the MEC application requires.
appLatency	0..1	LatencyDescriptor	Describes the maximum latency tolerated by the MEC application.
userContextTransferCapability	0..1	UserContextTransferCapability	If the application supports the user context transfer capability, this attribute shall be included.
appNetworkPolicy	0..1	AppNetworkPolicy	If present, it represents the application network policy of carrying the application traffic.

6.2.2.22 Type: MepInformation

6.2.2.22.1 Description

The data type represents the parameters of MEC platform information. It shall comply with the provisions in clause 6.2.2.22.2.

6.2.2.22.2 Attributes

The attributes of the data type are specified in table 6.2.2.22.2-1.

Table 6.2.2.22.2-1: Attributes of MepInformation

Attribute name	Cardinality	Data type	Description
mepName	0..1	String	Human-readable name of MEC platform.
mepId	1	String	Deployment-specific identifier of MEC platform.

6.2.2.23 Type: AppTermCandsForCoord

6.2.2.23.1 Description

The AppTermCandsForCoord data type represents the parameters to provide candidates of applications to terminate in pre-emption situations for LCM coordination exchanges.

6.2.2.23.2 Attributes

The attributes of the data type are specified in table 6.2.2.23.2-1.

Table 6.2.2.23.2-1: Attributes of AppTermCandsForCoord

Attribute name	Cardinality	Data type	Description
terminationOptions	1..N	array(Structure(inline))	Sets of application options for the MEO/MEAO to select from as candidates for termination. The MEO/MEAO shall select one or more of these alternate options to pass to the OSS when utilizing the LCM coordination exchange in pre-emption situations. For each option, the MEO/MEAO may select all, or a subset, of the candidate set's members.
>appInstIdTerminationCands	1..N	array(String)	List of application instance identifiers, constituting a candidate set for termination.

6.2.2.24 Void

6.2.2.25 Type: AppInstIdCreationSubscriptionRequest

6.2.2.25.1 Description

The data type represents the input parameters of "subscription operation" to notification of application instance identifier creation.

6.2.2.25.2 Attributes

The attributes of the data type are specified in the table 6.2.2.25.2-1.

Table 6.2.2.25.2-1: Attributes of AppInstIdCreationSubscriptionRequest

Name	Data type	Cardinality	Remarks
subscriptionType	String	1	Shall be set to "AppIdentifierCreationSubscription".
callbackUri	Uri	1	The URI of the endpoint for the subscription related notification to be sent to.
appInstanceSubscriptionFilter	AppInstanceSubscriptionFilter	0..1	Criteria used to filter application instances for which to send notifications related to this subscription. See note.
NOTE: If present, the value of attribute "appInstSelectorType" in appInstanceSubscriptionFilter can only be set as "APP_D_ID" or "APP_FROM_PROVIDER".			

6.2.2.26 Type: AppInstIdCreationSubscriptionInfo

6.2.2.26.1 Description

The data type represents a subscription to notification of application instance identifier creation. It shall comply with provisions in clause 6.2.2.26.2.

6.2.2.26.2 Attributes

The attributes of data type are specified in the table 6.2.2.26.2-1.

Table 6.2.2.26.2-1: Attributes of AppInstIdCreationSubscriptionInfo

Attribute name	Cardinality	Data type	Description
id	1	String	Identifier of the subscription to application instance operational state change notification.
subscriptionType	1	String	Shall be set to "AppIdentifierCreationSubscription".
callbackUri	1	Uri	The URI of the endpoint for the subscription related notification to be sent to.
appInstanceSubscriptionFilter	0..1	AppInstanceSubscriptionFilter	Criteria used to select application instances on which to send notifications related to this subscription.
_links	1	Structure (inlined)	Links to resources related to this resource.
>self	1	LinkType	URI of this resource.

6.2.2.27 Type: AppInstanceIdentifierCreationNotification

6.2.2.27.1 Description

This data type represents a notification for informing the subscribers about the creation of a new "individual application instance" resource and the associated application instance identifier. It shall comply with provisions in clause 6.2.2.27.2.

6.2.2.27.2 Attributes

The attributes of data type are specified in the table 6.2.2.27.2-1.

Table 6.2.2.27.2-1: Attributes of AppInstanceIdentifierCreationNotification

Attribute name	Cardinality	Data type	Description
id	1	String	Identifier of this notification. If a notification is sent multiple times due to multiple subscriptions, the "notificationId" attribute of all these notifications shall have the same value.
notificationType	1	String	Discriminator for the different notification types. Shall be set to "AppIdentifierCreationSubscription" for this notification type.
subscriptionId	1	String	Identifier of the subscription related to this notification.
timeStamp	1	TimeStamp	Date and time of the notification generation.
appInstanceId	1	String	The created application instance Identifier.
_links	1	Structure (inlined)	Links to resources related to this notification.
>subscription	1	LinkType	A link to the related subscription.
>appInstance	1	LinkType	Link to the resource representing the created application instance.

6.2.2.28 Type: AppInstIdDeletionSubscriptionRequest

6.2.2.28.1 Description

The data type represents the input parameters of "subscription operation" to notification of application instance identifier deletion.

6.2.2.28.2 Attributes

The attributes of the data type are specified in the table 6.2.2.28.2-1.

Table 6.2.2.28.2-1: Attributes of AppInstIdDeletionSubscriptionRequest

Name	Data type	Cardinality	Remarks
subscriptionType	String	1	Shall be set to "AppIdentifierDeletionSubscription".
callbackUri	Uri	1	The URI of the endpoint for the subscription related notification to be sent to.
appInstanceSubscriptionFilter	AppInstanceSubscriptionFilter	0..1	Criteria used to filter application instances for which to send notifications related to this subscription.

6.2.2.29 Type: AppInstIdDeletionSubscriptionInfo

6.2.2.29.1 Description

The data type represents a subscription to notification of application instance identifier deletion. It shall comply with provisions in clause 6.2.2.29.2.

6.2.2.29.2 Attributes

The attributes of data type are specified in the table 6.2.2.29.2-1.

Table 6.2.2.29.2-1: Attributes of AppInstIdDeletionSubscriptionInfo

Attribute name	Cardinality	Data type	Description
id	1	String	Identifier of the subscription to application instance operational state change notification.
subscriptionType	1	String	Shall be set to "AppIdentifierDeletionSubscription".
callbackUri	1	Uri	The URI of the endpoint for the subscription related notification to be sent to.
appInstanceSubscriptionFilter	0..1	AppInstanceSubscriptionFilter	Criteria used to select application instances on which to send notifications related to this subscription.
_links	1	Structure (inlined)	Links to resources related to this resource.
>self	1	LinkType	URI of this resource.

6.2.2.30 Type: AppInstanceIdentifierDeletionNotification

6.2.2.30.1 Description

This data type represents a notification for informing the subscribers about the deletion of an "individual application instance" resource and the associated application instance identifier. It shall comply with provisions in clause 6.2.2.30.2.

6.2.2.30.2 Attributes

The attributes of data type are specified in the table 6.2.2.30.2-1.

Table 6.2.2.30.2-1: Attributes of AppInstanceIdentifierDeletionNotification

Attribute name	Cardinality	Data type	Description
id	1	String	Identifier of this notification. If a notification is sent multiple times due to multiple subscriptions, the "notificationId" attribute of all these notifications shall have the same value.
notificationType	1	String	Discriminator for the different notification types. Shall be set to "AppIdentifierDeletionSubscription" for this notification type.
subscriptionId	1	String	Identifier of the subscription related to this notification.
timeStamp	1	TimeStamp	Date and time of the notification generation.
appInstanceId	1	String	The deleted application instance Identifier.
_links	1	Structure (inlined)	Links to resources related to this notification.
>subscription	1	LinkType	A link to the related subscription.
>appInstance	1	LinkType	Link to the resource representing the deleted application instance.

6.2.2.31 Type: LocationInformation

6.2.2.31.1 Description

The LocationInformation data type represents the location information of the site hosting the MEC application instance. The location information can be represented as a country code, plus a civic address and/or geographical position.

6.2.2.31.2 Attributes

The attributes of LocationInformation are shown in table 6.2.2.31.2-1.

Table 6.2.2.31.2-1: Attributes of LocationInformation

Attribute name	Cardinality	Data type	Description
countryCode	1	String	The two-letter ISO 3166 [3] country code in capital letters where an instance is deployed.
civicAddress	0..1	Structure (inlined)	Provides the civic address of the site hosting the MEC application instance.
>civicAddressElement	1..N	array(Structure (inlined))	Provides elements comprising a single civic address as described in section 3.4, with accompanying example in section 5 of IETF RFC 4776 [2].
>>caType	1	Integer	Describe the content type of caValue. The value of caType shall comply with section 3.4 of IETF RFC 4776 [2].
>>caValue	1	String	Content of civic address element corresponding to the caType. The format caValue shall comply with section 3.4 of IETF RFC 4776 [2].
geographicalPosition	0..1	String	Geographical position (i.e. latitude and longitude) where an instance is deployed. The content of this attribute shall follow the provisions for the "Point" geometry object as defined in IETF RFC 7946 [8].
NOTE: At least one of civicAddress or geographicalPosition shall be present. If both are present they shall specify the same location, bound by the precision of the provided coordinates.			

6.2.2.32 Type: CancelMode

6.2.2.32.1 Description

This data type represents the valid modes of cancelling an application LCM operation. It shall comply with the provisions in clause 6.2.2.32.2. In either mode, the server shall not start any new application LCM and resource management operations. If graceful cancellation is selected, the server shall wait for existing operations to complete or timeout. If forcefully cancellation is selected, the server shall cancel the ongoing application LCM and resource management operations and then wait for those cancellations to complete or timeout. In either mode, the server shall place the operation occurrence into the FAILED_TEMP state once the cancellation is completed.

6.2.2.32.2 Attributes

The attributes of data type are specified in the table 6.2.2.32.2-1.

Table 6.2.2.32.2-1: Attributes of CancelMode

Attribute name	Cardinality	Data type	Description
CancelMode	1	Enum (inlined)	Indicates the intervention action to be taken. • GRACEFUL: Indicates ongoing resource management operations in the underlying system are allowed to complete execution or time out. • FORCED: Indicates ongoing resource management operations in the underlying system are to be cancelled without allowing them to complete execution or time out.

6.2.2.33 Type: McioInfo

6.2.2.33.1 Description

This data type represents the information about an MCIO representing the application instance realized by one or a set of OS containers.

6.2.2.33.2 Attributes

The attributes of McioInfo shall follow the definition in clause 8.3.3.33.2 of ETSI GS NFV-IFA 013 [15], with the following consideration:

- A VNF corresponds to MEC application, a VNFC instance corresponds to a MEC application instance.

6.2.3 Application package information model

6.2.3.1 Introduction

This clause defines data structures to be used by the APIs of application package management.

6.2.3.2 Type: CreateAppPkg

6.2.3.2.1 Description

The data type CreateAppPkg represents the parameters for creating a new application package resource. It shall comply with attributes in clause 6.2.3.2.2.

6.2.3.2.2 Attributes

The attributes of CreateAppPkg data type shall follow the specification in table 6.2.3.2.2-1.

Table 6.2.3.2.2-1: Attributes of CreateAppPkg

Attribute name	Cardinality	Data type	Description
appPkgName	1	String	Name of the application package to be onboarded.
appPkgVersion	1	String	Version of the application package to be onboarded. The appPkgName with appPkgVersion can be used to uniquely identify the application package.
appProvider	0..1	String	The provider's name of the application package to be onboarded.
checksum	1	Checksum	Checksum of the onboarded application package.
userDefinedData	0..1	KeyValuePairs	User defined data for the application package.
appPkgPath	1	Uri	Address information of the application package. See note.
NOTE: It is for further study how to convey appPkgPath, and align with ETSI GS NFV-SOL 005 [i.7].			

6.2.3.3 Type: AppPkgInfo

6.2.3.3.1 Description

The data type AppPkgInfo represents the parameters for an application package resource. It shall comply with provisions in clause 6.2.3.3.2.

6.2.3.3.2 Attributes

The attributes of AppPkgInfo data type are specified in the table 6.2.3.3.2-1.

Table 6.2.3.3.2-1: Attributes of AppPkgInfo

Attribute name	Cardinality	Data type	Description
id	1	String	Identifier of the application package resource. This identifier is allocated by the MEO.
appDId	1	String	The application descriptor identifier. It is managed by the application provider to identify the application package and the application descriptor in a globally unique way. It is copied from the AppD of the onboarded application package.
appProvider	0..1	String	The provider's name of the onboarded application package.
appName	1	String	Name of the onboarded application.
appSoftwareVersion	1	String	Software version of the application. This is updated when there is any change to the software in the onboarded application package.
appDVersion	1	String	Version of the application descriptor.
checksum	1	Checksum	Checksum of the onboarded application package.
signingCertificate	0..1	String	The singleton signing certificate if it is included as a file in the AppD archive.
softwareImages	1..N	array(AppPkgSWImageInfo)	Information of application software image in application package. See note 1.
additionalArtifacts	0..N	array(AppPkgArtifactInfo)	Additional information of application package artifacts that are not application software images. See note 2.
onboardingState	1	Enum (inlined)	Onboarding state of application package: • CREATED: The application package resource has been created. • UPLOADING: The associated application package content is being uploaded. • PROCESSING: The associated application package content is being processed, e.g. validation. • ONBOARDED: The associated application package content is successfully onboarded.

Attribute name	Cardinality	Data type	Description
operationalState	1	Enum (inlined)	Operational state of the onboarded application package: • ENABLED: the application package can be used for instantiation of new application instances. • DISABLED: the application package cannot be used for further application instantiation requests.
usageState	1	Enum (inlined)	Usage state of the onboarded instance of the application package: • IN_USE: application instances instantiated from this package exist. • NOT_IN_USE: No application instance instantiated from this package exist.
mecInfo	1..N	array(String)	The MEC version that compatible with this application. This information is copied from the AppD.
userDefinedData	0..1	KeyValuePairs	User defined data for the application package.
onboardingFailureDetails	0..1	ProblemDetails	Failure details of current onboarding procedure. See ETSI GS MEC 009 [**4**].
_links	1	Structure (inlined)	Links to resources related to this resource.
>self	1	LinkType	Self referring URI.
>appD	1	LinkType	Link to the appD resource.
>appPkgContent	1	LinkType	Link to the "Onboarded application package content" resource.
>vnfPkgInfo	0..1	LinkType	Link to the corresponding VNF package resource at NFVO. See note 3.
NOTE 1: The data type of application software image information data model is related to virtualisation method and needs for further study. NOTE 2: The data type of additional information of application package artifacts is not specified in the present document. NOTE 3: This attribute applies only for the MEAO.			

Table 6.2.3.3.2-2 provides the attribute mapping information between AppPkgInfo and VnfPkgInfo after on-boarding application package to VNF as a VNF package.

Table 6.2.3.3.2-2: Attributes mapping between AppPkgInfo and VnfPkgInfo

AppPkgInfo Attribute	Map to VnfPkgInfo Attribute	Description
id	id	See note 1
appDId	vnfdId	See note 1
appProvider	vnfProvider	See note 2
appName	vnfProductName	See note 1
appSoftwareVersion	vnfSoftwareVersion	See note 1
appDVersion	vnfdVersion	See note 1
checksum	checksum	See note 4
signingCertificate	signingCertificate	See note 2
softwareImages	softwareImages	See note 1
additionalArtifacts	additionalArtifacts	See note 2
onboardingState	onboardingState	See note 3
operationalState	operationalState	See note 3
usageState	usageState	See note 3
mecInfo	vnfmInfo	See note 4
userDefinedData	userDefinedData	See note 2
onboardingFailureDetails	onboardingFailureDetails	See note 2
_links	_links	See note 4
>self	>self	See note 4
>appD	>vnfd	See note 5
>appPkgContent	>packageContent	See note 5
NOTE 1: MEAO maps those attributes between AppPkgInfo and VnfPkgInfo. NOTE 2: MEAO maps those attributes between AppPkgInfo and VnfPkgInfo if it is included in AppPkgInfo. NOTE 3: The state attributes of the MEC package info correspond to the states of the VNF package. NOTE 4: MEAO does not map those attributes between AppPkgInfo and VnfPkgInfo. NOTE 5: MEAO queries information from those URI in VnfPkgInfo and cache locally to those URI in AppPkgInfo.		

6.2.3.4 Type: AppPkgSubscriptionInfo

6.2.3.4.1 Description

The data type represents a subscription to notification of application package management for the onboarding, or operational state change of application package. It shall comply with provisions in clause 6.2.3.4.2.

6.2.3.4.2 Attributes

The attributes of data type are specified in the table 6.2.3.4.2-1.

Table 6.2.3.4.2-1: Attributes of AppPkgSubscriptionInfo

Attribute name	Cardinality	Data type	Description
id	1	String	Identifier of the subscription to application package notification.
subscriptionType	1	AppPkgSubscriptionType	Type of subscription.
callbackUri	1	Uri	The URI of the endpoint for the subscription related notification to be sent to.
_links	1	Structure (inlined)	Links to resources related to this resource.
>self	1	LinkType	URI of this resource.

6.2.3.5 Type: AppPkgSubscriptionLinkList

6.2.3.5.1 Description

The data type represents a subscription link list of notification on application package management. It shall comply with provisions in clause 6.2.3.5.2.

6.2.3.5.2 Attributes

The attributes of data type are specified in the table 6.2.3.5.2-1.

Table 6.2.3.5.2-1: Attributes of AppPkgSubscriptionLinkList

Attribute name	Cardinality	Data type	Description
_links	1	Structure (inlined)	Links to resources related to this resource.
>self	1	LinkType	URI of this resource.
>subscriptions	0..N	array (Structure (inlined))	A link list to the subscriptions to an application package.
>>href	1	Uri	The URI referring to the subscription.
>>subscriptionType	1	AppPkgSubscriptionType	Type of the subscription.

6.2.3.6 Type: AppPkgNotification

6.2.3.6.1 Description

This data type represents an application package management notification for informing the subscribers about onboarding application package resources. The notification is triggered when a new application package is onboarded.

It shall comply with provisions in clause 6.2.3.6.2.

6.2.3.6.2 Attributes

The attributes of data type are specified in the table 6.2.3.6.2-1.

Table 6.2.3.6.2-1: Attributes of AppPkgNotification

Attribute name	Cardinality	Data type	Description
id	1	String	Identifier of this notification. If a notification is sent multiple times due to multiple subscriptions, the "notificationId" attribute of all these notifications shall have the same value.
notificationType	1	String	Discriminator for the different notification types: • "AppPackageOnBoarded": notification of the new onboarded application package. • "AppPackageEnabled": notification of the operational state change of onboarded application package. • "AppPackageDisabled": notification of the onboarded application package disabled. • "AppPackageDeleted": notification of the application package deleted.
subscriptionId	1	String	Identifier of the subscription to this notification.
timeStamp	1	TimeStamp	Date and time of the notification generation.
appPkgId	1	String	Identifier of the onboarded application package.
appDId	1	String	The application descriptor identifier identifies the application package and the application descriptor in a globally unique way.
operationalState	1	Enum (inlined)	Operational state of the application package: • ENABLED: the application package can be used for instantiation of new application instances. • DISABLED: the application package cannot be used for further application instantiation requests.
_links	1	Structure (inlined)	Links to resources related to this notification.
>subscription	1	LinkType	A link to the related subscription.

6.2.3.7 Type: AppPkgSubscription

6.2.3.7.1 Description

The data type represents the input parameters of "subscription operation" to notification of application package management for the onboarding, or operational state change of application package.

6.2.3.7.2 Attributes

The attributes of data type are specified in the table 6.2.3.7.2-1.

Table 6.2.3.7.2-1: Attributes of AppPkgSubscription

Name	Data type	Cardinality	Remarks
callbackUri	Uri	1	The URI of the endpoint for the subscription related notification to be sent to.
subscriptionType	AppPkgSubscriptionType	1	Type of the subscription.
appPkgFilter	array(AppPkgFilter)	0..N	The attribute-based filter is to filter application packages on which the query applies.

6.2.3.8 Type: AppPkgInfoModifications

6.2.3.8.1 Description

The data type represents modifications of the "AppPkgInfo" data type that can be requested to perform "application package operation".

6.2.3.8.2 Attributes

The attributes of data type are specified in the table 6.2.3.8.2-1.

Table 6.2.3.8.2-1: Attributes of AppPkgInfoModifications

Name	Data type	Cardinality	Remarks
operationalState	Enum (inlined)	1	New value of the "operationalState" attribute of the "OnboardedAppPkgInfo" structure. Permitted values • DISABLED: to disable the individual application package. • ENABLED: to enable the individual application package.

6.2.3.9 Referenced simple data types and enumerations

6.2.3.9.1 Introduction

This clause defines simple data types and enumerations.

6.2.3.9.2 Simple data types

The simple data type defined for this API are provided in table 6.2.3.9.2-1.

Table 6.2.3.9.2-1: Simple data types

Type name	Description
AppPkgSubscriptionType	String representing the type of a subscription. Permitted values: • "AppPackageOnBoardingSubscription": subscription to notifications relating newly onboarded application packages. • "AppPackageChangeSubscription": subscription to notifications relating onboarded application package operational state change or usage state change. • "AppPackageDeletionSubscription": subscription to notification relating application package deletion.

6.2.3.9.3 Enumeration

None defined.

6.2.3.10 Type: AppPkgFilter

6.2.3.10.1 Description

This data type represents subscription filter criteria to match application package. The AppPkgFilter shall comply with provisions in clause 6.2.3.10.2.

6.2.3.10.2 Attributes

The attributes of the data type are specified in table 6.2.3.10.2-1.

Table 6.2.3.10.2-1: Attributes of AppPkgFilter

Attribute name	Cardinality	Data type	Description
appPkgInfoId	0..1	String	Match the application package identifier which is allocated by the MEO. See note.
appDId	0..1	String	Match the application descriptor identifier which is allocated by the application provider. See note.
appProvider	0..1	String	Match the provider's name of the onboarded application.
appName	0..1	String	Match the name of the onboarded application.
appSoftwareVersion	0..1	String	Match the software version of the application package.
appDVersion	0..1	String	Match the version of the application descriptor.

Attribute name	Cardinality	Data type	Description
operationalState	0..1	Enum (inlined)	Match particular operational state of the application package: • ENABLED: the application package can be used for instantiation of new application instances. • DISABLED: the application package cannot be used for further application instantiation requests. May be present if the "subscriptionType" attribute contains the value "AppPackageChangeSubscription", and shall be absent otherwise.
usageState	0..1	Enum (inlined)	Match particular usage state of the application package: • IN_USE: application instances instantiated from this package exist. • NOT_IN_USE: No application instance instantiated from this package exist. May be present if the "subscriptionType" attribute contains the value "AppPackageChangeSubscription", and shall be absent otherwise.
NOTE: The attributes "appPkgInfoId ", and "appDId" are alternatives to reference particular application package in a filter. They should not be used both in the same filter instance, but one alternative should be chosen.			

6.2.4 Granting information model

6.2.4.1 Introduction

This clause defines data types used in the granting resource.

6.2.4.2 Type: GrantRequest

6.2.4.2.1 Description

This type represents a grant request. Refer to clause 9.5.2.2 of ETSI GS NFV-SOL 003 [7].

6.2.4.2.2 Attributes

The attributes of the data type are specified in table 6.2.4.2.2-1.

Table 6.2.4.2.2-1: Attributes of GrantRequest

Attribute name	Cardinality	Data type	Description
appInstanceId	1	String	Identifier of the application instance which this grant request is related to. Shall also be provided for application instances that not yet exist but are planned to exist in the future, i.e. if the grant is requested for Instantiate.
appLcmOpOccId	1	String	The identifier of the application lifecycle management operation occurrence associated to the GrantRequest.
appDId	1	String	Identifier of the AppD that defines the application for which the LCM operation is to be granted.
operation	1	Enum (inlined)	The lifecycle management operation for which granting is requested: • INSTANTIATE. • OPERATE. • TERMINATE. See notes 1 and 2.
addResources	0..N	array(ResourceDefinition)	List of resource definitions in the AppD for resources to be added by the LCM operation which is related to this grant request, with one entry per resource.

Attribute name	Cardinality	Data type	Description
tempResources	0..N	array(ResourceDefinition)	List of resource definitions in the AppD for resources to be temporarily instantiated during the runtime of the LCM operation which is related to this grant request. See note 3.
removeResources	0..N	array(ResourceDefinition)	Removed by the LCM operation which is related to this grant request, with one entry per resource.
updateResources	0..N	array(ResourceDefinition)	Provides the definitions of resources to be modified by the LCM operation which is related to this grant request, with one entry per resource.
additionalParams	0..1	KeyValuePairs	MEPM, specific to the application and the LCM operation.
_links	1	Structure (inlined)	Links to resources related to this request.
>appLcmOpOcc	1	LinkType	Related lifecycle management operation occurrence.
>appInstance	1	LinkType	Related application instance.
NOTE 1: Other application LCM operations can be executed by the MEPM without requesting granting. NOTE 2: If the granting request is for Instantiate, addResources shall be present. NOTE 3: The MEO will assume that the MEPM will be responsible to both allocate and release the temporary resource during the runtime of the LCM operation. This means, the resource can be allocated and consumed after the "start" notification for the LCM operation is sent by the MEPM, and the resource will be released before the "result" notification of the application LCM operation is sent by the MEPM.			

6.2.4.3 Type: ResourceDefinition

6.2.4.3.1 Description

This type provides information of an existing or proposed resource used by the application. Refer to clause 9.5.3.2 of ETSI GS NFV-SOL 003 [7].

6.2.4.3.2 Attributes

The attributes of the data type are specified in table 6.2.4.3.2-1.

Table 6.2.4.3.2-1: Attributes of ResourceDefinition

Attribute name	Cardinality	Data type	Description
id	1	String	Identifier of this "ResourceDefinition" structure, unique at least within the scope of the "GrantRequest" structure.
type	1	Enum (inlined)	Type of the resource definition referenced. Permitted values: • COMPUTE. • STORAGE. • LINKPORT • OSCONTAINER, see note.
vduId	0..1	String	Reference to the related VDU in the AppD applicable to this resource. Shall only be present if a VDU is applicable to this resource.
resourceTemplateId	1..N	array(String)	Reference to a resource template in the AppD as follows: • If type="COMPUTE": VirtualComputeDescriptor, • If type="LINKPORT": AppExtCpd, • If type="STORAGE": VirtualStorageDescriptor, • If type="OSCONTAINER": osContainerDescriptor. Cardinality may be greater than "1" when Type ="OSCONTAINER" and multiple references to OsContainerDescriptor are present in the AppD. Cardinality shall be "1" otherwise.
resource	1	Structure (inlined)	Resource information for an existing resource. Shall be present for resources that are planned to be deleted or modified. Shall be absent otherwise.
>vimConnectionInfo	1	VimConnectionInfo	Specifies the connection information of VIM for the resources of the application instance.
>resourceId	1	String	Identifier of the resource in the scope of the VIM.
NOTE: This permitted value reflects the ETSI NFV interpretation of the cloud native workloads.			

6.2.4.4 Type: Grant

6.2.4.4.1 Description

This type represents a grant. Refer to clause 9.5.2.3 of ETSI GS NFV-SOL 003 [7].

6.2.4.4.2 Attributes

The attributes of the data type are specified in table 6.2.4.4.2-1.

Table 6.2.4.4.2-1: Attributes of Grant

Attribute name	Cardinality	Data type	Description
id	1	String	Identifier of the Grant.
appInstanceId	1	String	Identifier of the application instance which this Grant is related to.
appLcmOpOccId	1	String	The identifier of the application lifecycle management operation occurrence associated to the Grant.
vimConnections	0..N	map(VimConnectionInfo)	Provides information regarding VIM connections that are approved to be used by the MEPM to allocate resources, and provides parameters of these VIM connections. See note 1.
zones	0..N	array(ZoneInfo)	Identifies resource zones where the resources are approved to be allocated by the MEPM.
zoneGroups	0..N	array(ZoneGroupInfo)	Information about groups of resource zones that are related and that the MEO has chosen to fulfil a zoneGroup constraint in the Grant request.
addResources	0..N	array(GrantInfo)	List of resources that are approved to be added, with one entry per resource.
tempResources	0..N	array(GrantInfo)	List of resources that are approved to be temporarily instantiated during the runtime of the lifecycle operation, with one entry per resource.
removeResources	0..N	array(GrantInfo)	List of resources that are approved to be removed, with one entry per resource.
updateResources	0..N	array(GrantInfo)	List of resources that are approved to be modified, with one entry per resource.
vimAssets	0..1	Structure (inlined)	Information about assets for the application that are managed by the MEO in the VIM, such as software images. See note 2.
>softwareImages	0..N	array(VimSoftwareImage)	Mappings between software images defined in the AppD and software images managed in the VIM.
extVirtualLinks	0..N	array(ExtVirtualLinkData)	Information about external VLs to connect the application instance to. See note 3.
additionalParams	0..1	KeyValuePairs	MEPM, specific to the application and the LCM operation.
_links	1	Structure (inlined)	Links to resources related to this request.
>appLcmOpOcc	1	LinkType	Related lifecycle management operation occurrence.
>appInstance	1	LinkType	Related application instance.

NOTE 1: This interface allows to signal the use of multiple VIMs per application. However, due to the partial support of this feature in the present release, it is recommended in the present document that the number of entries in the "vims" attribute in the Grant is not greater than 1.
NOTE 2: The further condition will be defined by ETSI GS NFV-SOL 003 [7].
NOTE 3: External VLs can be passed in the application lifecycle management operation requests such as Instantiate, and/or in the grant response. The MEO may choose to override in the grant response external VL instances that have been passed previously in the associated application lifecycle management request, if the lifecycle management request has originated from the MEO itself.

6.2.4.5 Type: GrantInfo

6.2.4.5.1 Description

This type contains information about a Compute, storage or network resource whose addition/update/deletion was granted. It shall comply with the provisions defined in table 6.2.4.5.2-1. Refer to clause 9.5.3.3 of ETSI GS NFV-SOL 003 [7].

6.2.4.5.2 Attributes

The attributes of the data type are specified in table 6.2.4.5.2-1.

Table 6.2.4.5.2-1: Definition of the GrantInfo data type

Attribute name	Data type	Cardinality	Description
resourceDefinitionId	String	1	Identifier of the related "ResourceDefinition" structure from the related "GrantRequest" structure.
vimConnectionId	String	0..1	Identifier of the VIM connection to be used to manage this resource. Shall be present for new resources, and shall be absent for resources that have already been allocated.
zoneId	String	0..1	Reference to the identifier of the "ZoneInfo" structure in the "Grant" structure defining the resource zone into which this resource is to be placed. Shall be present for new resources if the zones concept is applicable to them (typically, Compute resources), and shall be absent for resources that have already been allocated.
resourceGroupId	String	0..1	Identifier of the "infrastructure resource group", logical grouping of virtual resources assigned to a tenant within an Infrastructure Domain, to be provided when allocating the resource. If the VIM connection referenced by "vimConnectionId" applies to multiple infrastructure resource groups, this attribute shall be present for new resources. If the VIM connection referenced by "vimConnectionId" applies to a single infrastructure resource group, this attribute may be present for new resources. This attribute shall be absent for resources that have already been allocated.
containerNamespace	String	0..1	The value of the namespace in which the MCIOs of an application with containerized components shall be deployed. This attribute shall be present if the granted resources are managed by a CISM. The attribute shall be absent if the granted resources are not managed by a CISM. See note.
mcioConstraints	KeyValuePair	0..N	The constraint values to be assigned to MCIOs of an application with containerized components. The key in the key-value pair indicates the parameter name of the MCIO constraint in the MCIO declarative descriptor and shall be one of the possible enumeration values of the "mcioConstraintsParams" attribute as specified in clause 7.1.6.2.2 of ETSI GS NFV-IFA 011 [1]. The value in the key-value pair indicates the value to be assigned to the MCIO constraint. This attribute shall be present if the granted resources are managed by a CISM. The attribute shall be absent if the granted resources are not managed by a CISM. See note.
NOTE: This attribute reflects the ETSI NFV interpretation of the cloud native workloads.			

6.2.4.6 Type: ZoneInfo

6.2.4.6.1 Description

This type provides information regarding a resource zone. Refer to clause 9.5.3.4 of ETSI GS NFV-SOL 003 [7].

6.2.4.6.2 Attributes

The attributes of the data type are specified in table 6.2.4.6.2-1.

Table 6.2.4.6.2-1: Definition of the ZoneInfo data type

Attribute name	Data type	Cardinality	Description
id	String	1	The identifier of this ZoneInfo instance, for the purpose of referencing it from other structures in the "Grant" structure.
zoneId	String	1	The identifier of the resource zone, as managed by the resource management layer (typically, the VIM).
vimConnectionId	String	0..1	Identifier of the connection to the VIM that manages the resource zone. The applicable "VimConnectionInfo" structure, which is referenced by vimConnectionId, can be obtained from the " vimConnectionInfo" attribute of the "AppInstanceInfo" structure.

6.2.4.7 Type: ZoneGroupInfo

6.2.4.7.1 Description

This type provides information regarding a resource zone group. A resource zone group is a group of one or more related resource zones which can be used in resource placement constraints. To fulfil such constraint, the MEO may decide to place a resource into any zone that belongs to a particular group. Refer to clause 9.5.3.5 of ETSI GS NFV-SOL 003 [7].

NOTE: A resource zone group can be used to support overflow from one resource zone into another, in case a particular deployment supports only non-elastic resource zones.

6.2.4.7.2 Attributes

The attributes of the data type are specified in table 6.2.4.7.2-1.

Table 6.2.4.7.2-1: Definition of the ZoneGroupInfo data type

Attribute name	Data type	Cardinality	Description
zoneId	array(String)	1..N	References of identifiers of "ZoneInfo" structures, each of which provides information about a resource zone that belongs to this group.

6.2.4.8 Type: ExtVirtualLinkData

6.2.4.8.1 Description

This type represents an external VL. Refer to clause 4.4.1.11 of ETSI GS NFV-SOL 003 [7].

6.2.4.8.2 Attributes

It shall comply with the provisions defined in table 6.2.4.8.2-1.

Table 6.2.4.8.2-1: Definition of the ExtVirtualLinkData data type

Attribute name	Data type	Cardinality	Description
id	String	1	The identifier of the external VL instance. The identifier is assigned by the MEC entity that manages this VL instance.
vimConnectionId	String	0..1	Identifier of the VIM connection to manage this resource.
resourceId	String	1	The identifier of the resource in the scope of the VIM.
extCps	array(AppExtCpData)	1..N	External CPs of the application instance to be connected to this external VL.
extLinkPorts	array(ExtLinkPortData)	0..N	Externally provided link ports to be used to connect external connection points to this external VL. If this attribute is not present, the MEPM shall create the link ports on the external VL.

6.2.4.9 Type: ExtLinkPortData

6.2.4.9.1 Description

This type represents an externally provided link port to be used to connect an external connection point to an external VL. Refer to clause 5.5.3.9a of ETSI GS NFV-SOL 003 [7].

6.2.4.9.2 Attributes

It shall comply with the provisions defined in table 6.2.4.9.2-1.

Table 6.2.4.9.2-1: Definition of the ExtLinkPortData data type

Attribute name	Data type	Cardinality	Description
id	String	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.

6.2.4.10 Type: ResourceHandle

6.2.4.10.1 Description

This type represents the information that allows addressing a virtualised resource that is used by an application instance. Information about the resource is available from the VIM. Refer to clause 4.4.1.7 of ETSI GS NFV-SOL 003 [7].

6.2.4.10.2 Attributes

It shall comply with the provisions defined in table 6.2.4.10.2-1.

Table 6.2.4.10.2-1: Definition of the ResourceHandle data type

Attribute name	Data type	Cardinality	Description
vimConnectionId	String	0..1	Identifier of the VIM connection to manage the resource. The applicable "VimConnectionInfo" structure, which is referenced by vimConnectionId, can be obtained from the "vimConnectionInfo" attribute of the "AppInstance" structure.
resourceId	String	1	Identifier of the resource in the scope of the VIM or the CISM or the resource provider. See note 2.
vimLevelResourceType	String	0..1	Type of the resource in the scope of the VIM. See note 1.

NOTE 1: The value set of the "vimLevelResourceType" attribute is within the scope of the VIM and can be used as information that complements the ResourceHandle. This value set is different from the value set of the "type" attribute in the ResourceDefinition.

NOTE 2: When the container infrastructure service is a Kubernetes® instance the resourceId shall be populated in the following way:

- For a compute MCIO, it is the instance identifier that Kubernetes® assigns, which is unique cluster wide per resource type.
- For a storage MCIO modelled as a persistent volume claim, it is the name of the persistent volume claim, i.e. the value of the 'claimName' field in the Kubernetes® manifest, or a compound name built by Kubernetes® if the persistent volume claim is defined inline in another template instead of in its own manifest.
- For a network MCIO representing a NetworkAttachmentDefinition, a Service or an Ingress, it is the value of the 'metadata.name' field in Kubernetes® manifest.

6.2.4.11 Type: VimSoftwareImage

6.2.4.11.1 Description

This type contains a mapping between a software image definition in the AppD and the corresponding software image managed by the MEO in the VIM which is needed during compute resource instantiation. Refer to clause 9.5.3.10 of ETSI GS NFV-SOL 003 [7].

6.2.4.11.2 Attributes

It shall comply with the provisions defined in table 6.2.4.11.2-1.

Table 6.2.4.11.2-1: Definition of the VimSoftwareImage data type

Attribute name	Data type	Cardinality	Description
vimConnectionId	String	0..1	Identifier of the VIM connection to access the software image referenced in this structure.
appDSoftwareImageId	String	1	Identifier which references the software image descriptor in the AppD.
vimSoftwareImageId	String	1	Identifier of the software image in the resource management layer (i.e. VIM). See note.
NOTE: For an OS container image, the value of this attribute is a string concatenating the name and tag of the image in the CIR separated by a colon ':' with no spaces, e.g. "dbImage:001".			

6.2.4.12 Type: AppExtCpData

6.2.4.12.1 Description

This type represents configuration information for external CPs created from a CPD. Refer to clause 4.4.1.10 of ETSI GS NFV-SOL 003 [7].

6.2.4.12.2 Attributes

It shall comply with the provisions defined in table 6.2.4.12.2-1.

Table 6.2.4.12.2-1: Definition of the AppExtCpData data type

Attribute name	Data type	Cardinality	Description
cpdId	String	1	The identifier of the CPD in the AppD.
cpConfig	map(AppExtCpConfig)	1..N	List of instance data that need to be configured on the CP instances created from the respective CPD.

6.2.4.13 Type: AppExtCpConfig

6.2.4.13.1 Description

This type represents an externally provided link port or network address information per instance of an external connection point. In case a link port is provided, the MEPM shall use that link port when connecting the external CP to the external VL. In case a link port is not provided, the MEPM shall create a link port on the external VL, and use that link port to connect the external CP to the external VL. Refer to clause 4.4.1.10a of ETSI GS NFV-SOL 003 [7].

6.2.4.13.2 Attributes

This type shall comply with the provisions defined in table 6.2.4.13.2-1.

Table 6.2.4.13.2-1: Definition of the AppExtCpConfig data type

Attribute name	Data type	Cardinality	Description
cpInstanceId	String	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	String	0..1	Identifier of a pre-configured link port to which the external CP will be associated. See note.
cpProtocolData	array(CpProtocol Data)	0..N	Parameters for configuring the network protocols on the link port that connects the CP to a VL. See note.
NOTE: The following conditions apply to the attributes "linkPortId" and " cpProtocolData": 1) The "linkPortId" and "cpProtocolData" attributes shall both be absent for the deletion of an existing external CP instance addressed by cpInstanceId. 2) At least one of these attributes shall be present for a to-be-created external CP instance or an existing external CP instance. 3) If the "linkPortId" attribute is absent, the MEPM shall create a link port. 4) If the "cpProtocolData" attribute is absent, the "linkPortId" attribute shall be provided referencing a pre-created link port, and the MEPM can use means outside the scope of the present document to obtain the pre-configured address information for the connection point from the resource representing the link port. 5) 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".			

6.2.4.14 Type: CpProtocolData

6.2.4.14.1 Description

This type represents network protocol data. Refer to clause 4.4.1.10b of ETSI GS NFV-SOL 003 [7].

6.2.4.14.2 Attributes

This type shall comply with the provisions defined in table 6.2.4.14.2-1.

Table 6.2.4.14.2-1: 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.			

6.2.4.15 Type: IpOverEthernetAddressData

6.2.4.15.1 Description

This type represents network address data for IP over Ethernet. Refer to clause 4.4.1.10c of ETSI GS NFV-SOL 003 [7].

6.2.4.15.2 Attributes

It shall comply with the provisions defined in table 6.2.4.15.2-1.

Table 6.2.4.15.2-1: Definition of the IpOverEthernetAddressData data type

Attribute name	Data type	Cardinality	Description
macAddress	String	0..1	MAC address. If this attribute is not present, it shall be chosen by the VIM. See note 1.
ipAddresses	array(Structure (inlined))	0..N	List of IP addresses to assign to the CP 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	array(String)	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	String	1	Lowest IP address belonging to the range.
>>maxAddress	String	1	Highest IP address belonging to the range.
>subnetId	String	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.			

6.2.5 Common information model

6.2.5.1 Introduction

This clause defines common data structures used by other information models.

6.2.5.2 Type: LinkType

6.2.5.2.1 Description

This data type represents a type of link.

6.2.5.2.2 Attributes

The attributes of LinkType are specified in the table 6.2.5.2.2-1.

Table 6.2.5.2.2-1: Attributes of LinkType

Attribute name	Cardinality	Data type	Description
href	1	Uri	URI referring to a resource.

6.2.5.3 Type: KeyValuePairs

6.2.5.3.1 Description

This data type represents a list of key-value pairs. The order of the pairs in the list is not significant. In JSON, a set of key-value pairs is represented as an object. It shall comply with the provisions defined in clause 4 of IETF RFC 8259 [5]. In the following example, a list of key-value pairs with four keys ("aString", "aNumber", "anArray" and "anObject") is provided to illustrate that the values associated with different keys can be of different type.

EXAMPLE:

```
{
    "aString" : "ETSI ISG MEC",
    "aNumber" : 0.01,
    "anArray" : [1,2,3],
    "anObject" : {"organization" : "ETSI", "ISG" : "MEC"}
}
```

6.2.5.4 Type: TimeStamp

6.2.5.4.1 Description

This data type represents the time stamp as Unix-time since January 1, 1970, 00:00:00 UTC.

The TimeStamp shall comply with provisions in clause 6.2.5.4.2.

6.2.5.4.2 Attributes

The attributes of data type are specified in the table 6.2.5.4.2-1.

Table 6.2.5.4.2-1: Attributes of TimeStamp

Attribute name	Cardinality	Data type	Description
seconds	1	Uint32	The seconds part of the Time. Time is defined as Unix-time since January 1, 1970, 00:00:00 UTC.
nanoSeconds	1	Uint32	The nanoseconds part of the Time. Time is defined as Unix-time since January 1, 1970, 00:00:00 UTC.

6.2.5.5 Void

6.2.5.6 Type: Checksum

6.2.5.6.1 Description

This type represents the checksum of an application package.

6.2.5.6.2 Attributes

It shall comply with the provisions defined in table 6.2.5.6.2-1.

Table 6.2.5.6.2-1: 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 [18]. For example, SHA-256, SHA-512.
hash	String	1	The hexadecimal value of the checksum.

6.3 Interfaces

6.3.1 Application lifecycle management interface

6.3.1.1 Description

This interface allows the OSS to invoke lifecycle management operations towards the MEO/MEAO or allows the MEO to invoke lifecycle management operations towards the MEPM or allows the MEAO to invoke lifecycle management operations towards the NFVO.

The following operations are defined:

- Create application instance identifier.
- Application instantiation.
- Application instance terminate.
- Delete application instance identifier.
- Query application instance information.
- Change application instance state.
- Query application lifecycle operation Status.
- Subscribe to notifications relating to application lifecycle management.

An identifier (i.e. lifecycleOperationOccurrenceId) is generated for each application lifecycle operation occurrence, except for query application instance information, create application instance identifier, delete application instance identifier, query application lifecycle operation status and subscribe to notifications relating to application lifecycle management.

6.3.1.2 Create application instance identifier operation

6.3.1.2.1 Description

This operation creates an application instance identifier, and an associated instance of an AppInstanceInfo, identified by that identifier, in the NOT_INSTANTIATED state without instantiating the application or doing any additional lifecycle operation(s). It allows returning right away an application instance identifier that may be used in subsequent lifecycle operations, like the application instantiation operation.

Table 6.3.1.2.1-1 lists the information flow exchanged between the MEPM and the MEO, between the MEO and the OSS, between the MEAO and the OSS, or between the MEAO and the MEPM-V.

Table 6.3.1.2.1-1: Create application instance identifier operation

Message	Requirement	Direction
CreateAppInstanceIdentifierRequest	Mandatory	MEO → MEPM, OSS → MEO, OSS → MEAO, MEAO → MEPM-V
CreateAppInstanceIdentifierResponse	Mandatory	MEPM → MEO, MEO → OSS, MEAO → OSS, MEPM-V → MEAO

6.3.1.2.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 6.3.1.2.2-1.

Table 6.3.1.2.2-1: Create application instance identifier operation input parameters

Attribute name	Cardinality	Data type	Description
appDId	1	String	Identifier that identifies the application package that will be instantiated. The application package is identified by the appDId of the application descriptor included in the package.
appInstanceName	0..1	String	Human-readable name of the application instance to be created.
appInstanceDescription	0..1	String	Human-readable description of the application instance to be created.
appPlacementInfo	0..1	MepInformation	Describes the information of selected MEC platform for the application instance to associate. See note.
NOTE: This field applies to Mm3* reference point only.			

6.3.1.2.3 Output parameters

The output parameters returned by the operation shall follow the indications provided in table 6.3.1.2.3-1.

Table 6.3.1.2.3-1: Create application instance identifier operation output parameters

Attribute name	Cardinality	Data type	Description
appInstanceId	1	String	The application instance identifier just created.

6.3.1.2.4 Operation results

In case of success, an instance of an AppInstanceInfo, in the NOT_INSTANTIATED state, has been created and may be used in subsequent lifecycle operations and the corresponding AppInstanceIdentifierCreationNotification has been sent. In case of failure, appropriate error information is returned.

6.3.1.3 Application instantiation operation

6.3.1.3.1 Definition

This operation instantiates a MEC application instance.

Table 6.3.1.3.1-1 lists the information flow for application instantiation.

Table 6.3.1.3.1-1: Instantiate application instance operation

Message	Requirement	Direction
InstantiateAppRequest	Mandatory	OSS → MEO, MEO → MEPM, OSS → MEAO.
InstantiateAppResponse	Mandatory	MEO → OSS, MEPM → MEO, MEAO → OSS.

6.3.1.3.2 Input parameters

The input parameters for this operation are shown in table 6.3.1.3.2-1.

Table 6.3.1.3.2-1: Instantiate application instance operation input parameters

Attribute name	Cardinality	Data type	Description
appInstanceId	1	String	Identifier of the application instance created by "Create application instance identifier" operation.
virtualComputeDescriptor	0..1	VirtualComputeDescriptor	Describes CPU and memory requirements, as well as optional additional requirements, such as disk and acceleration related capabilities, of the single VM used to realize the MEC application instance to be created. See note 1 and note 5.

Attribute name	Cardinality	Data type	Description
osContainerDescriptor	0..N	OsContainerDescriptor	Describes CPU, memory requirements and limits, and software images of the OS Containers realizing this MEC application corresponding to OS Containers sharing the same host and same network namespace. See notes 1, 5 and 6.
virtualStorageDescriptor	0..N	VirtualStorageDescriptor	Defines descriptors of virtual storage resources to be used by the MEC application instance to be created. See note 1.
selectedMECHostInfo	1..N	MECHostInformation	Describes the information of selected MEC host for the MEC application instance. See note 2.
locationConstraints	0..1	LocationConstraints	Defines the location constraints for the MEC application instance to be created. See note 4.
vimConnectionInfo	0..N	VimConnectionInfo	Information about VIM connections to be used for managing the resources for the application instance, or refer to external/externally-managed virtual links. This attribute shall only be supported and may be present if application-related resource management in direct mode is applicable. See note 2.
appTermCandsForCoord	0..1	AppTermCandsForCoord	Provides sets of applications as termination candidate alternatives that the MEO/MEAO shall select from when utilizing the coordinate LCM operation exchange in pre-emption situations (see step 3 in clause 5.3.1). If this attribute is omitted, the MEO/MEAO shall make its own selection for the coordinate LCM operation exchange. See note 4.
NOTE 1: The same field of AppD will be override by the value of the field in this table. NOTE 2: This field applies to Mm3 reference point only. NOTE 3: Void. NOTE 4: This field applies to Mm1 reference point only. NOTE 5: Only one of virtualComputeDescriptor or osContainerDescriptor shall be present. NOTE 6: This attribute reflects the ETSI NFV interpretation of the cloud native workloads.			

6.3.1.3.3 Output parameters

The output parameters returned by the operation shall follow the indications provided in table 6.3.1.3.3-1.

Table 6.3.1.3.3-1: Instantiate application instance operation output parameters

Attribute name	Cardinality	Data type	Description
lifecycleOperationOccurrenceId	1	String	The identifier of the application lifecycle operation occurrence.

6.3.1.3.4 Operation results

In case of success, the MEC application instance has been instantiated, initially configured, and Lifecycle Change Notifications have been sent accordingly. In case of failure, appropriate error information is returned in lifecycle change notification.

The responder shall first return the lifecycleOperationOccurrenceId and second send the "start" Lifecycle Change Notification before additional notifications or messages as part of this operation are issued, or operations towards the VIM are invoked.

On successful as well as unsuccessful completion of the operation, the responder shall send the "result" Lifecycle Change Notification.

6.3.1.4 Change application instance operational state operation

6.3.1.4.1 Description

This operation enables requesting to change the state of a MEC application instance, including starting and stopping the application instance.

NOTE 1: These operations are complementary to instantiating and terminating a MEC application instance.

NOTE 2: In the present document, only starting and stopping the MEC application instance(s) are supported. Extension of this operation to support other MEC application state changes is left for future specification.

A MEC application instance may be in the following operational states:

- STARTED: the MEC application instance is up and running.
- STOPPED: the MEC application instance has been shut down.

In the operational state STOPPED, the virtualisation container, where the MEC application instance run, are shut down but not terminated. In addition, if the workflow requires a graceful stop, as part of this process, the MEC platform will interact with the MEC application instance to gracefully stop the MEC application. Once a MEC application is instantiated, i.e. all instantiation steps have been completed, the MEC application instance is in the operational state STARTED.

Figure 6.3.1.4.1-1 illustrates the application instance operational state diagram. The desired change of operational state is indicated as an input in the OperateAppInstanceRequest message.

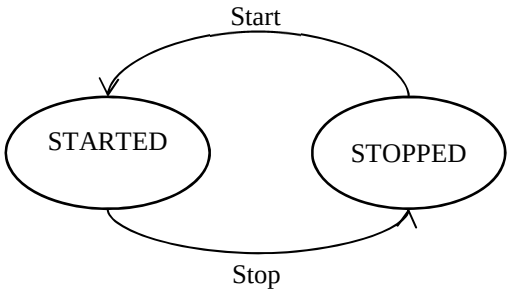

Figure 6.3.1.4.1-1: Change application instance operational state diagram

It depends on the MEC application capabilities, and is declared in the AppD, whether this operation is supported for a particular MEC application.

Table 6.3.1.4.1-1 lists the information flow exchanged between the initiator and the responder.

Table 6.3.1.4.1-1: Change application instance state operation

Message	Requirement	Direction
OperateAppInstanceRequest	Mandatory	OSS → MEO, MEO → MEPM, OSS → MEAO, MEAO → MEPM-V.
OperateAppInstanceResponse	Mandatory	MEO → OSS, MEPM → MEO, MEAO → OSS, MEPM-V → MEAO.

6.3.1.4.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 6.3.1.4.2-1.

Table 6.3.1.4.2-1: Change application instance state operation input parameters

Attribute name	Cardinality	Data type	Description
appInstanceId	1	String	Identifier of the MEC application instance.
changeStateTo	1	Enum (inlined)	The desired operational state to change the MEC application instance to. Permitted values are: STARTED, STOPPED.
stopType	0..1	Enum (inlined)	Signals whether forceful or graceful stop is requested. Allowed values are: FORCEFUL and GRACEFUL. In case of FORCEFUL stop, the MEC application is stopped immediately. Note that if the MEC application is still in service, this may adversely impact network service, and therefore, operator policies apply to determine if FORCEFUL stop is allowed in the particular situation. In case of GRACEFUL stop, the MEC system gives time to the MEC application for application level stop (e.g. via Mp1 interaction). Once this was successful, or after a timeout, the MEC system stops the MEC application. If the MEC application does not support Mp1, the stopType shall be set to FORCEFUL.
gracefulStopTimeout	0..1	Integer	The time interval to wait for the application instance to stop service during graceful stop, before stopping the application instance. If not given, it is expected that the MEC system waits for the successful application level stop, no matter how long it takes, before stopping the MEC application (see note). Minimum timeout or timeout range are specified by the application vendor defined in the AppD. Not relevant in case of forceful stop.
NOTE: This implies that no application instance stop will be attempted if application level stopping fails or hangs.			

6.3.1.4.3 Output parameters

The output parameters returned by the operation shall follow the indications provided in table 6.3.1.4.3-1.

Table 6.3.1.4.3-1: Change application instance state operation output parameters

Attribute name	Cardinality	Data type	Description
lifecycleOperationOccurrenceId	1	String	The identifier of the MEC application lifecycle operation occurrence.

6.3.1.4.4 Operation results

In case of success, the state of the MEC application instance has been changed. In case of failure, appropriate error information is provided in the "result" Lifecycle Change Notification.

The producer shall first return the lifecycleOperationOccurrenceId and second send the "start" Lifecycle Change Notification before additional notifications or messages as part of this operation are issued, or operations towards the VIM are invoked.

On successful as well as unsuccessful completion of the operation, the MEPM shall send the "result" Lifecycle Change Notification.

6.3.1.5 Query application instance information operation

6.3.1.5.1 Description

This operation provides information about application instances. The applicable application instances may be chosen based on filtering criteria, and the information may be restricted to selected attributes.

Table 6.3.1.5.1-1 lists the information flow exchanged between: the OSS and the MEO; the MEO and MEPM; the OSS and the MEAO; the MEAO and MEPM-V.

Table 6.3.1.5.1-1: Query application instance information operation

Message	Requirement	Direction
QueryAppInstanceInfoRequest	Mandatory	OSS → MEO, MEO → MEPM, OSS → MEAO, MEAO → MEPM-V.
QueryAppInstanceInfoResponse	Mandatory	MEO → OSS, MEPM → MEO, MEAO → OSS, MEPM-V → MEAO.

6.3.1.5.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 6.3.1.5.2-1.

Table 6.3.1.5.2-1: Query application instance information operation input parameters

Attribute name	Cardinality	Data type	Description
filter	1	Filter	Filter to select the application instance(s) about which information is queried. See note.
attributeSelector	0..N	String	Provides a list of attribute names. If present, only these attributes will be returned for the application instance(s) matching the filter. If absent, the complete information will be returned for the application instance(s) matching the filter.
NOTE: See table 7.4.1.3.2-1 for the attribute-based filter and selector.			

6.3.1.5.3 Output parameters

The output parameters returned by the operation shall follow the indications provided in table 6.3.1.5.3-1.

Table 6.3.1.5.3-1: Query application instance information operation output parameters

Attribute name	Cardinality	Data type	Description
appInstanceInfo	0..N	AppInstanceInfo	The information about the selected application instance(s) that are returned. If attributeSelector is present, only the attributes listed in attributeSelector will be returned for the selected application instance(s). See note.
NOTE: The lower cardinality is 0 since there may be no matches to the provided filter.			

6.3.1.5.4 Operation results

In case of success, information related to the application instances that match the filter is returned. In case of failure, appropriate error information is returned.

6.3.1.6 Query application lifecycle operation status

6.3.1.6.1 Description

This operation provides the status of an application lifecycle management operation.

Table 6.3.1.6.1-1 lists the information flow exchanged between: the OSS and the MEO; the MEO and MEPM; the OSS and the MEAO; the MEAO and MEPM-V.

Table 6.3.1.6.1-1: Query application lifecycle operation status operation

Message	Requirement	Direction
QueryAppLcmOperationStatusRequest	Mandatory	OSS → MEO, MEO → MEPM, OSS → MEAO, MEAO → MEPM-V.
QueryAppLcmOperationStatusResponse	Mandatory	MEO → OSS, MEPM → MEO, MEAO → OSS, MEPM-V → MEAO.

6.3.1.6.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 6.3.1.6.2-1.

Table 6.3.1.6.2-1: Query application lifecycle operation status operation input parameters

Attribute name	Cardinality	Data type	Description
lifecycleOperationOccurrenceId	1	String	Identifier of the application lifecycle operation occurrence.

6.3.1.6.3 Output parameters

The output parameters returned by the operation shall follow the indications provided in table 6.3.1.6.3-1.

Table 6.3.1.6.3-1: Query application lifecycle operation status operation output parameters

Attribute name	Cardinality	Data type	Description
operationStatus	1	Enum (inlined)	Indicates the operation status (which includes, for example, PROCESSING, SUCCESSFULLY_DONE, FAILED and operation-specific states).

6.3.1.6.4 Operation results

In success of the operation, the status of the queried operation will be returned. In case of failure, appropriate error code will be returned.

6.3.1.7 Application instance terminate operation

6.3.1.7.1 Description

This operation terminates a MEC application instance.

A MEC application instance may be terminated gracefully or forcefully. Graceful termination means that the MEC Platform Manager gives time to the MEC application for application level termination, and after the MEC application has terminated in application level, the MEC system releases the resources used by the MEC application. Forceful termination means that the MEC Platform Manager immediately shuts down the MEC application and releases the resources. A time interval is specified for graceful termination, after the timer specified by the time interval expires, the MEC Platform Manager will shut down the MEC application and release the resources. The graceful termination requires that the MEC application supports Mp1 reference point.

Table 6.3.1.7.1-1 lists the information flow exchanged between the initiator and the responder.

Table 6.3.1.7.1-1: Terminate application instance operation

Message	Requirement	Direction
TerminateAppInsRequest	Mandatory	OSS → MEO, MEO → MEPM, OSS → MEAO, MEAO → MEPM-V.
TerminateAppInsResponse	Mandatory	MEO → OSS, MEPM → MEO, MEAO → OSS, MEPM-V → MEAO.

6.3.1.7.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 6.3.1.7.2-1.

Table 6.3.1.7.2-1: Terminate application instance operation input parameters

Attribute name	Cardinality	Data type	Description
appInstanceId	1	String	Identifier of the MEC application instance to be terminated.
terminationType	1	Enum (inlined)	Signals whether FORCEFUL or GRACEFUL termination is requested. In case of FORCEFUL termination, the MEC application is shut down immediately, and resources are released. Note that if the MEC application is still in service, this may adversely impact user experience. In case of GRACEFUL termination, the MEC system gives time to the MEC application for application level termination (e.g. via Mp1 interaction). Once this was successful, or after a timeout, the MEC system shuts down the MEC application and releases the resources. If the MEC application does not support Mp1, the terminationType shall be set to FORCEFUL termination.

Attribute name	Cardinality	Data type	Description
gracefulTerminationTimeout	0..1	Integer	The time interval given to MEC application for application level termination during graceful termination, before shutting down the MEC application and releasing the resources. If not given, it is expected that the MEC system waits for the successful application level termination, no matter how long it takes, before shutting down the MEC application and releasing the resources (see note). Minimum timeout or timeout range are specified by the application vendor defined in the AppD. Not relevant in case of forceful termination.
NOTE: This implies that no MEC application shutdown and resource release will be attempted if taking the MEC application out of service fails or hangs.			

6.3.1.7.3 Output parameters

The output parameters returned by the operation shall follow the indications provided in table 6.3.1.7.3-1.

Table 6.3.1.7.3-1: Terminate application instance operation output parameters

Attribute name	Cardinality	Data type	Description
lifecycleOperationOccurrenceId	1	String	The identifier of the MEC application lifecycle operation occurrence.

6.3.1.7.4 Operation results

In case of success, the MEC application instance has been terminated, resources used by the MEC application instance have been released. In case of failure, appropriate error information is returned.

6.3.1.8 Delete application instance identifier operation

6.3.1.8.1 Description

This operation deletes an application instance identifier and the associated instance of an AppInstanceInfo in the NOT_INSTANTIATED state.

Table 6.3.1.8.1-1 lists the information flow exchanged between: the OSS and the MEO; the MEO and MEPM; the OSS and the MEAO; the MEAO and MEPM-V.

Table 6.3.1.8.1-1: Delete application instance Identifier operation

Message	Requirement	Direction
DeleteAppInstanceIdentifierRequest	Mandatory	MEO → MEPM, OSS → MEO, OSS → MEAO, MEAO → MEPM-V.
DeleteAppInstanceIdentifierResponse	Mandatory	MEPM → MEO, MEO→ OSS, MEAO → OSS, MEPM-V → MEAO.

6.3.1.8.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 6.3.1.8.2-1.

Table 6.3.1.8.2-1: Delete application instance Identifier operation input parameters

Attribute name	Cardinality	Data type	Description
appInstanceId	1	String	Application instance identifier to be deleted.

6.3.1.8.3 Output parameters

No output parameter.

6.3.1.8.4 Operation results

In case of success, the application instance identifier and the associated instance of the AppInstanceInfo has been deleted and is no longer used; and the corresponding AppInstanceIdentifierDeletionNotification has been sent. If the application instance was not terminated (i.e. the application instance is in INSTANTIATED state), the operation shall be rejected.

In case of failure, appropriate error information is returned.

6.3.1.9 Subscribe to application lifecycle management notifications

6.3.1.9.1 Description

This operation allows a subscriber to subscribe to notifications relating to MEC application lifecycle management, including notification of operational state changes, application LCM operation occurrence state change, as well as notification on the creation/deletion of an application instance identifier, with its associated application instance.

6.3.1.9.2 Subscribe

6.3.1.9.2.1 Description

The subscriber subscribes with a filter to the notifications related to MEC application lifecycle management operational state changes, application LCM operation occurrence state change, as well as creation/deletion of MEC application instance identifiers and the associated application instances.

Table 6.3.1.9.2.1-1 lists the information flow exchanged between: the OSS and the MEO; the MEO and MEPM; the OSS and the MEAO; the MEAO and MEPM-V.

Table 6.3.1.9.2.1-1: Subscribe operation

Message	Requirement	Direction
SubscribeRequest	Mandatory	OSS →MEO, MEO → MEPM, OSS → MEAO, MEAO → MEPM-V.
SubscribeResponse	Mandatory	MEO →OSS, MEPM → MEO, MEAO → OSS, MEPM-V → MEAO.

6.3.1.9.2.2 Input parameters

The input parameters of a subscribe request shall follow the information in table 6.3.1.9.2.2-1.

Table 6.3.1.9.2.2-1: Subscribe operation input parameters

Attribute name	Cardinality	Data type	Description
subscriptionType	1	AppInstanceSubscriptionType	Match particular notification types. See clause 6.2.2.20.2.
appInstanceSubscriptionFilter	0..1	AppInstanceSubscriptionFilter	If present, this attribute contains filter criteria that selects one or more application instances on which to receive notifications. See note.
appInstanceState	0..1	Enum (inlined)	Match particular application instance operational states for the notifications of AppInstNotification. Only send notifications for application instances that are in one of the states listed in this attribute. If this attribute is absent, match all states. Application states: • NOT_INSTANTIATED: the application instance is not instantiated. • STARTED: the application instance is up and running. • STOPPED: the application instance stops operation. May be present if the "subscriptionType" attribute contains the value "AppInstanceStateChangeSubscription", and shall be absent otherwise.
operationTypes	0..N	Enum (inlined)	Match particular application lifecycle operation types for the notifications of AppLcmOpOccNotification. Type of the LCM operation represented by this application instance LCM operation occurrence. Permitted values: • INSTANTIATE. • OPERATE. • TERMINATE. May be present if the "subscriptionType" attribute contains the value "AppLcmOpOccStateChangeSubscription", and shall be absent otherwise.
operationStates	0..N	Enum (inlined)	Match particular LCM operation state values as reported in notifications of AppLcmOpOccNotification. Type of the LCM operation state represented by this application instance LCM operation occurrence. Permitted values: • STARTING. • PROCESSING. • COMPLETED. • FAILED. • FAILED_TEMP. May be present if the "subscriptionType" attribute contains the value "AppLcmOpOccStateChangeSubscription", and shall be absent otherwise.
NOTE: When subscribing for notifications regarding the creation of application instance identifiers and the associated application instance information object instances, selecting the application instances in the filter is not possible.			

6.3.1.9.2.3 Output parameters

The output parameters returned in the response to a subscribe request shall follow the indications in table 6.3.1.9.2.3-1.

Table 6.3.1.9.2.3-1: Subscribe operation output parameters

Attribute name	Cardinality	Data type	Description
subscriptionId	1	String	Identifier of the subscription realized.

6.3.1.9.2.4 Operation results

After a successful subscription, the subscriber (such as OSS or MEO) will be registered to receive notifications related to MEC application lifecycle management operational state changes, application LCM operation occurrence state change, as well as creation/deletion of application instance identifiers, with their associated application instances.

The result of the subscribe request shall indicate if the subscription has been successful or not with a standard success/error result. For a particular subscription, only notifications matching the filter will be delivered to the subscriber.

6.3.1.9.3 Notify

6.3.1.9.3.1 Description

The notify operation notifies a subscriber about events related to application lifecycle management operational state changes, application LCM operation occurrence state change, as well as events related to the creation/deletion of application instance identifiers, with their associated application instances.

This operation distributes notifications to subscribers, and is a one-way operation issued only by the producer. In order to receive notifications, a consumer (such as OSS or MEO) has to perform an explicit subscribe operation beforehand.

Table 6.3.1.9.3.1-1 lists the information flow exchanged between: the OSS and the MEO; the MEO and MEPM; the OSS and the MEAO; the MEAO and MEPM-V.

Table 6.3.1.9.3.1-1: Notify operation

Message	Requirement	Direction
Notify	Mandatory	MEPM → MEO, MEO → OSS, MEAO → OSS, MEPM-V → MEAO.

The following notifications may be sent by this operation:

- AppInstNotification. See clause 6.2.2.11
- AppLcmOpOccNotification. See clause 6.2.2.16
- AppInstanceIdentifierCreationNotification. See clause 6.2.2.27
- AppInstanceIdentifierDeletionNotification. See clause 6.2.2.30

6.3.1.9.4 Query subscription

6.3.1.9.4.1 Definition

The query operation enables a subscriber (such as OSS or MEO) to query subscription(s) for application instance operational state change notification.

Table 6.3.1.9.4.1-1 lists the information flow exchanged between: the OSS and the MEO; the MEO and MEPM; the OSS and the MEAO; the MEAO and MEPM-V.

Table 6.3.1.9.4.1-1: Query subscription operation

Message	Requirement	Direction
QuerySubscriptionRequest	Mandatory	OSS → MEO, MEO → MEPM, OSS → MEAO, MEAO → MEPM-V.
QuerySubscriptionResponse	Mandatory	MEO → OSS, MEPM → MEO, MEAO → OSS, MEPM-V → MEAO.

6.3.1.9.4.2 Input parameters

The input parameters shall follow the information in table 6.3.1.9.4.2-1.

Table 6.3.1.9.4.2-1: Query subscription operation input parameters

Attribute name	Cardinality	Data type	Description
n/a			

6.3.1.9.4.3 Output parameters

The output parameters returned by the operation shall follow the information in table 6.3.1.9.4.3-1.

Table 6.3.1.9.4.3-1: Query subscription operation output parameters

Attribute name	Cardinality	Data type	Description
appInstSubscriptionLinkList	1	AppInstanceSubscription LinkList	A list of subscriptions to application instances.

6.3.1.9.4.4 Operation results

The result of this operation shall indicate whether the query request is success or not. If successful, the information of subscription(s) shall be delivered to the requester (such as OSS or MEO).

6.3.1.9.5 Delete subscription operation

6.3.1.9.5.1 Definition

The delete operation deletes subscription(s) to application lifecycle management notification.

Table 6.3.1.9.5.1-1 lists the information flow exchanged between: the OSS and the MEO; the MEO and MEPM; the OSS and the MEAO; the MEAO and MEPM-V.

Table 6.3.1.9.5.1-1: Delete subscription operation

Message	Requirement	Direction
DeleteSubscriptionRequest	Mandatory	OSS → MEO, MEO → MEPM, OSS → MEAO, MEAO → MEPM-V.
DeleteSubscriptionResponse	Mandatory	MEO → OSS, MEPM → MEO, MEAO → OSS, MEPM-V → MEAO.

6.3.1.9.5.2 Input parameters

The input parameters of this operation shall follow the information in table 6.3.1.9.5.2-1.

Table 6.3.1.9.5.2-1: Delete subscription operation input parameters

Attribute name	Cardinality	Data type	Description
subscriptionId	1..N	String	The identifier(s) of application instance subscription(s).

6.3.1.9.5.3 Output parameters

The output parameters returned by the operation shall follow the information in table 6.3.1.9.5.3-1.

Table 6.3.1.9.5.3-1: Delete subscription operation output parameters

Attribute name	Cardinality	Data type	Description
n/a			

6.3.1.9.5.4 Operation results

The result of this operation shall indicate whether the delete request is success or not.

6.3.1.10 Configure platform for application instance operation

6.3.1.10.1 Description

This operation allows MEAO to send configuration information to MEPM-V. The configuration information includes the traffic rules, DNS rules, the required and optional services, and services produced by the application instance, etc. This operation may trigger MEPM-V to send the configuration information to the MEP to prepare it for running an application instance.

Table 6.3.1.10.1-1 lists the information flow exchanged between the MEAO and MEPM-V.

Table 6.3.1.10.1-1: Configure platform for application instance operation

Message	Requirement	Direction
ConfigPlatformForAppInstanceRequest	Mandatory	MEAO → MEPM-V
ConfigPlatformForAppInstanceResponse	Mandatory	MEPM-V → MEAO

6.3.1.10.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 6.3.1.10.2-1.

Table 6.3.1.10.2-1: Configure platform for application instance operation input parameters

Attribute name	Cardinality	Data type	Description
appInstanceId	1	String	The identifier of application instance (i.e. VNF instance identifier returned from NFVO after instantiation).
configurationParameters	1	ConfigPlatformForAppRequest	The platform configuration information associated to the application instance.

6.3.1.10.3 Output parameters

The output parameters returned by the operation shall follow the indications provided in table 6.3.1.10.3-1.

Table 6.3.1.10.3-1: Configure platform for application instance operation output parameters

Attribute name	Cardinality	Data type	Description
lifecycleOperationOccurrenceId	1	String	The identifier of the application lifecycle operation occurrence.

6.3.1.10.4 Operation results

In case of success, the configuration information for the MEC application in the MEP(VNF) has been updated according to the input parameters specified in the operation. In case of failure, appropriate error information is returned.

The responder shall first return the lifecycleOperationOccurrenceId and second send the "start" Lifecycle Change Notification before additional notifications or messages as part of this operation are issued, or operations towards MEP(VNF) are invoked.

On successful as well as unsuccessful completion of the operation, the responder shall send the "result" Lifecycle Change Notification.

6.3.2 Void

6.3.3 Application package management interface

6.3.3.1 Fetch onboarded application package operation

6.3.3.1.1 Definition

This operation enables the OSS or the MEPM to fetch onboarded application package. Table 6.3.3.1.1-1 lists the information flow exchanged between the MEO and the MEPM, the OSS and the MEO, or the OSS and the MEAO.

Table 6.3.3.1.1-1: Fetch onboarded package operation

Message	Requirement	Direction
FetchAppPackageRequest	Mandatory	OSS → MEO, MEPM → MEO, OSS → MEAO.
FetchAppPackageResponse	Mandatory	MEO → OSS, MEO → MEPM, MEAO → OSS.

6.3.3.1.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 6.3.3.1.2-1.

Table 6.3.3.1.2-1: Fetch onboarded package operation input parameters

Attribute name	Cardinality	Data type	Description
appPkgId	1	String	Identifier of the onboarded application package to be fetched.

6.3.3.1.3 Output parameters

The output parameters returned by the operation shall follow the indications provided in table 6.3.3.1.3-1.

Table 6.3.3.1.3-1: Fetch onboarded package operation output parameters

Attribute name	Cardinality	Data type	Description
appPackage	0..1	Binary	The application package.

6.3.3.1.4 Operation results

After success operation, the MEO/MEAO has provided to the requester a copy of the requested application package.

6.3.3.2 Query application package information operation

6.3.3.2.1 Definition

This interface allows the OSS or the MEPM to query information about the Application Package. Table 6.3.3.2.1-1 lists the information flow exchanged between the MEO and the MEPM, between the OSS and the MEO, or between the OSS and the MEAO.

Table 6.3.3.2.1-1: Query application package operation

Message	Requirement	Direction
QueryAppPkgInfoRequest	Mandatory	OSS → MEO, MEPM → MEO, OSS → MEAO.
QueryAppPkgInfoResponse	Mandatory	MEO → OSS, MEO → MEPM, MEAO → OSS.

6.3.3.2.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 6.3.3.2.2-1.

Table 6.3.3.2.2-1: Query application package operation input parameters

Attribute name	Cardinality	Data type	Description
filter	1	Filter	Filter defining the application packages on which the query applies, based on attributes of the application package. It may also be used to specify one or more application packages to be queried by providing their identifiers. See note.
attributeSelector	0..N	String	It provides a list of attribute names of the application package. If present, only these attributes will be returned for the application package matching the filter. If absent, the complete application package will be returned.
NOTE: See table 7.3.1.3.2-1 for the attribute-based filter and selector.			

6.3.3.2.3 Output parameters

The output parameters returned by the operation shall follow the indications provided in table 6.3.3.2.3-1.

Table 6.3.3.2.3-1: Query application package operation output parameters

Attribute name	Cardinality	Data type	Description
queryResult	0..N	AppPkgInfo	Details of the application packages matching the attribute filter. If an attribute selector is present, only the attributes listed in attribute elector will be returned for the selected entities.

6.3.3.2.4 Operation results

After successful operation, the MEO/MEAO has queried the internal application package information objects. The result of the operation indicates if it has been successful or not with a standard success/error result. For a particular query, information about the application package that the consumer has access to and that are matching the filter shall be returned.

6.3.3.3 Subscribe operation

6.3.3.3.1 Definition

This operation enables the OSS or MEPM to subscribe with a filter for the notifications related to events of application packages sent by the MEO. Table 6.3.3.3.1-1 lists the information flow exchanged between the OSS and MEO, the MEO and MEPM, or the OSS and MEAO.

Table 6.3.3.3.1-1: Subscribe operation

Message	Requirement	Direction
SubscribeRequest	Mandatory	OSS → MEO, MEPM → MEO, OSS → MEAO.
SubscribeResponse	Mandatory	MEO → OSS, MEO → MEPM, MEAO → OSS.

6.3.3.3.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 6.3.3.3.2-1.

Table 6.3.3.3.2-1: Subscribe operation input parameters

Attribute name	Cardinality	Data type	Description
inputFilter	1	Filter	Input filter for selecting the application package(s) and the related events notifications to subscribe to. This filter may contain information about specific types of events to subscribe to, or attributes of the application package.

6.3.3.3.3 Output parameters

The output parameters returned by the operation shall follow the indications provided in table 6.3.3.3.3-1.

Table 6.3.3.3.3-1: Subscribe operation output parameters

Attribute name	Cardinality	Data type	Description
subscriptionId	1	String	Identifier of the subscription realized.

6.3.3.3.4 Operation results

After successful subscription, the OSS or MEPM is registered to receive notifications related to events of application packages sent by the MEO/MEAO. The result of the operation shall indicate if the subscription has been successful or not with a standard success/error result. For a particular subscription, only notifications matching the filter will be delivered to the OSS or MEPM.

6.3.3.4 Notify application package operation

6.3.3.4.1 Definition

This operation distributes notifications to subscribers and can only be invoked as an operation by the MEO/MEAO.

In order to receive notifications, the OSS or MEPM shall have a subscription.

The following notifications shall be notified/sent to subscribers by this operation:

- AppPackageOnBoardingNotification.
- AppPackageStateChangeNotification.

The format of both notification is the AppPkgNotification type specified in clause 6.2.3.6.2.

Table 6.3.3.4.1-1: Notify operation

Message	Requirement	Direction
Notify	Mandatory	MEO →OSS, MEO → MEPM, MEAO → OSS.

6.3.3.5 Onboarding operation

6.3.3.5.1 Definition

This operation will onboard an application package in the MEO or through MEAO to NFVO, for the latter case, if necessary, the MEAO converts the application package to the VNF package according to the VNF package format as described in clause 5.6.2.

Table 6.3.3.5.1-1 lists the information flow exchanged between the OSS and the MEO, or between the OSS and MEAO.

Table 6.3.3.5.1-1: Onboard application package operation

Message	Requirement	Direction
OnboardAppPkgRequest	Mandatory	OSS → MEO, OSS → MEAO.
OnboardAppPkgResponse	Mandatory	MEO → OSS, MEAO → OSS.

6.3.3.5.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 6.3.3.5.2-1.

Table 6.3.3.5.2-1: Onboard application package operation input parameters

Attribute name	Cardinality	Data type	Description
name	1	String	Name of the application package to be onboarded.
version	1	String	Version of the application package to be onboarded.
provider	1	String	Provider of the application package to be onboarded.
checksum	1	Checksum	Checksum of the on-boarded application package.
userDefinedData	0..N	KeyValuePairs	User defined data for the application package. See note 1.
appPackagePath	1	URI	Address information based on which the application package may be obtained. See note 2.
NOTE 1: This data type is not specified in the present document. NOTE 2: This Structure may be the address information related to an FTP server when the application package is stored, or be a URI where the MEO or NFVO may download the application package.			

6.3.3.5.3 Output parameters

The output parameters returned by the operation shall follow the indications provided in table 6.3.3.5.3-1.

Table 6.3.3.5.3-1: Onboard application package operation output parameters

Attribute name	Cardinality	Data type	Description
appPkgId	1	String	Identifier of the on-boarded the Application package.
appDId	1	String	Identifier that identifies the application package in a globally unique way. See note.
NOTE: This identifier, which is managed by the application provider, identifies the application package and the AppD in a globally unique way.			

6.3.3.5.4 Operation results

The result of the operation indicates whether the on-boarding of the application package has been successful or not with a standard success/error result.

The appPkgId of onboarded application package will only be returned when the operations has been successful.

Once on-boarded, the application package will be known to and validated by the MEO or NFVO. It will be in "Enabled, Not in use" state, allowing its use for application lifecycle management. For details of state model of application, refer to clause A.2.

6.3.3.6 Enable operation

6.3.3.6.1 Definition

This operation will enable a previously disabled application package, allowing again its use for instantiation of new application instances. The "In use/Not in use" sub-state shall not change as a result of the operation.

Table 6.3.3.6.1-1 lists the information flow exchanged between the OSS and the MEO, or between the OSS and the MEAO.

Table 6.3.3.6.1-1: Enable application package operation

Message	Requirement	Direction
EnableAppPkgRequest	Mandatory	OSS → MEO, OSS → MEAO.
EnableAppPkgResponse	Mandatory	MEO → OSS, MEAO → OSS.

6.3.3.6.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 6.3.3.6.2-1.

Table 6.3.3.6.2-1: Enable application package operation input parameters

Attribute name	Cardinality	Data type	Description
appPkgId	1	String	Identifier of the on-boarded application package.

6.3.3.6.3 Output parameters

No output parameter.

6.3.3.6.4 Operation results

The result of the operation indicates if it has been successful or not with a standard success/error result.

If the application was already enabled, this operation will return an error.

6.3.3.7 Disable operation

6.3.3.7.1 Definition

This operation will disable a previously enabled application package, preventing any further use for instantiation of new network application instance with this application package. The "In use/Not in use" sub-state shall not change as a result of the operation. After an application package is disabled successfully, the state of this application package is "disabled, not in use" or "disabled, in use", see clause A.2 for the state of the application package.

Table 6.3.3.7.1-1 lists the information flow exchanged between the OSS and the MEO, or between the OSS and the MEAO.

Table 6.3.3.7.1-1: Disable application package operation

Message	Requirement	Direction
DisableAppPkgRequest	Mandatory	OSS → MEO, OSS → MEAO.
DisableAppPkgResponse	Mandatory	MEO → OSS, MEAO → OSS.

6.3.3.7.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 6.3.3.7.2-1.

Table 6.3.3.7.2-1: Disable application package operation input parameters

Attribute name	Cardinality	Data type	Description
onboardedAppPkgId	1	String	Identifier of the on-boarded application package.

6.3.3.7.3 Output parameters

No output parameter.

6.3.3.7.4 Operation results

The result of the operation indicates if it has been successful or not with a standard success/error result.

If the application package was already disabled, this operation will return an error.

6.3.3.8 Void

6.3.3.9 Delete operation

6.3.3.9.1 Definition

This operation will delete one application package.

An application package shall only be deleted when it is disabled and there is no instantiated application instance using it.

Table 6.3.3.9.1-1 lists the information flow exchanged between the OSS and the MEO, or between the OSS and the MEAO.

Table 6.3.3.9.1-1: Delete application package operation

Message	Requirement	Direction
DeleteAppPkgRequest	Mandatory	OSS → MEO, OSS → MEAO.
DeleteAppPkgResponse	Mandatory	MEO → OSS, MEAO → OSS.

6.3.3.9.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 6.3.3.9.2-1.

Table 6.3.3.9.2-1: Delete application package operation input parameters

Attribute name	Cardinality	Data type	Description
appPkgId	1	String	Identifier of information held by the MEO/MEAO about the specific on-boarded application package, which is to be deleted. This identifier was allocated by the MEO/MEAO.

6.3.3.9.3 Output parameters

No output parameter.

6.3.3.9.4 Operation results

The result of the operation indicates if it has been successful or not with a standard success/error result.

6.3.3.10 Abort application package deletion operation

This operation is no longer supported.

6.3.3.11 Query subscription operation

6.3.3.11.1 Definition

This operation enables the OSS or MEPM to query subscription(s) for events of application packages sent by the OSS or MEO. Table 6.3.3.11.1-1 lists the information flow exchanged between the OSS and the MEO, the MEO and the MEPM, or the OSS and the MEAO.

Table 6.3.3.11.1-1: Query subscription operation

Message	Requirement	Direction
QuerySubscriptionRequest	Mandatory	OSS → MEO, MEPM → MEO, OSS → MEAO.
QuerySubscriptionResponse	Mandatory	MEO → OSS, MEO → MEPM, MEAO → OSS.

6.3.3.11.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 6.3.3.11.2-1.

Table 6.3.3.11.2-1: Query subscription operation input parameters

Attribute name	Cardinality	Data type	Description
n/a			

6.3.3.11.3 Output parameters

The output parameters returned by the operation shall follow the indications provided in table 6.3.3.11.3-1.

Table 6.3.3.11.3-1: Query subscription operation output parameters

Attribute name	Cardinality	Data type	Description
appPkgSubscriptionInfo	0 .. N	AppPkgSubscriptionInfo	A list of application package subscriptions

6.3.3.11.4 Operation results

The result of this operation shall indicate whether the query request is success or not. If successful query, the information of subscription(s) shall be delivered to the requester OSS or MEPM.

6.3.4 Granting interface

6.3.4.1 Introduction

This interface allows the MEPM to obtain from the MEO permission and configuration parameters for an application lifecycle operation. Further, this interface allows to retrieve the granting result.

6.3.4.2 Granting request

6.3.4.2.1 Definition

The MEPM sends the granting request for permission on an operation of application instance. Table 6.3.4.2.1-1 lists the information flow for the granting request.

Table 6.3.4.2.1-1: Granting request

Message	Requirement	Direction
Granting request	Mandatory	MEPM → MEO
Granting response	Mandatory	MEO → MEPM

6.3.4.2.2 Input parameters

The input parameters for the granting request is shown in table 6.3.4.2.2-1.

Table 6.3.4.2.2-1: Input parameters of granting request

Attribute name	Cardinality	Data type	Description
grantRequest	1	GrantRequest	The parameters for the granting request.

6.3.4.2.3 Output parameters

The output parameters shall follow the indications in table 6.3.4.2.3-1.

Table 6.3.4.2.3-1: Output parameters of granting request

Attribute name	Cardinality	Data type	Description
grant	1	Grant	Result of the granting request

6.3.4.2.4 Operation results

In case of success, the MEO returns the granting result in the grant message.

6.3.5 LCM Coordination interface

6.3.5.1 Description

LCM Coordination interfaces enable the MEO/MEAO to request the OSS, or NFV management entities to request MEC management entities, to coordinate MEC application LCM operations.

The following operations are defined:

- CoordinateAppLcmOperation produced by OSS on Mm1;
- CoordinateLcmOperation produced by MEAO on Mv1;
- CoordinateLcmOperation produced by MEPM-V on Mv2.

Further, standardized coordination actions are defined.

6.3.5.2 Coordinate application lifecycle operation produced by OSS

6.3.5.2.1 Description

This operation enables a MEO/MEAO to request the coordination of an LCM operation with management operations executed in the OSS on an application instance. The operation shall follow the definition in clause 6.1 of ETSI GS NFV-IFA 013 [15], where the coordination action used within that is defined in clause 6.3.5.5.2 of the present document.

6.3.5.3 Coordinate application lifecycle operation produced by MEAO

6.3.5.3.1 Description

This operation enables a NFVO to request the coordination of an LCM operation with a MEAO on an NS instance in which the target MEC application has been instantiated as a VNF instance.

The operation shall follow the definition in clause 6.1 of ETSI GS NFV-IFA 013 [15], with the following consideration:

- The OSS in ETSI GS NFV-IFA 013 [15] corresponds to the MEAO in the present document.

6.3.5.4 Coordinate application lifecycle operation produced by MEPM-V

6.3.5.4.1 Description

This operation enables a VNFM to request the coordination of an LCM operation with a MEPM-V on MEC application that has been instantiated as a VNF instance.

The operation shall follow the definition in clause 8.3 of ETSI GS NFV-IFA 008 [16], with the following consideration:

- The EM in ETSI GS NFV-IFA 008 [16] corresponds to the MEPM-V in the present document.

6.3.5.5 Standardized coordination actions

6.3.5.5.1 Introduction

The clauses below define standardized LCM coordination actions.

6.3.5.5.2 Select application instances to terminate

This action allows to obtain information from the OSS to select a number of applications that can be terminated in order to allow the instantiation of a new, typically higher-priority application during a resource shortage.

The coordination action shall follow the provisions defined in tables 6.3.5.5.2-1, 6.3.5.5.2-2 and 6.3.5.5.2-3.

Table 6.3.5.5.2-1: Definitions

Attribute	Definition
coordinationActionName	"urn:etsi:mec:coord:select-apps-to-terminate"
operationStage	This attribute shall not be present.

Table 6.3.5.5.2-2: Input parameters

Attribute name	Cardinality	Data type	Description
appInstanceId	1	String	The identifier of an application to be instantiated in a resource shortage situation.
terminationOptions	1..N	array(Structure(inline))	Suggested application instance termination options for the OSS to select between.
>appInstIdTerminationCands	1..N	array(String)	List of application instance identifiers all of which are suggested for termination as part of the actual termination option.

Table 6.3.5.5.2-3: Output parameters

Attribute name	Cardinality	Data type	Description
terminateApps	0..N	array(Structure(inline))	List of application instances for termination.
>appInstanceId	1	String	The application instance identifier.
>terminationType	0..1	Enum (inlined)	Indicates whether forceful or graceful termination is to be performed (see note 1). If absent, graceful termination shall be assumed by the MEO/MEAO using its default timeout period. • FORCEFUL • GRACEFUL
>gracefulTerminationTimeout	0..1	Integer	Shall be absent if termination type indicates forceful termination and may be present otherwise. It defines the time to wait for the application instance to be taken out of service before shutting down the application and releasing the resources (see note 2). The unit is seconds.
NOTE 1: See clause 6.2.2.9 for further description relating to this attribute. Requesting forceful termination can adversely impact service and is therefore not recommended. NOTE 2: If specified, the timeout overrides the default timeout period at the MEO/MEAO. Setting the default timeout is out of scope of the present document.			

7 API definitions

7.1 Introduction

This clause defines the RESTFul resources and operations over reference point Mm1 and Mm3 APIs for:

- application package management; and
- application life cycle management.

7.2 Global definitions and resource structure

All resource URIs of APIs shall have the following root:

{apiRoot}/{apiName}/{apiVersion}/

Where:

- The "apiRoot" consists of the scheme ("https"), host and optional port, and an optional prefix string. It can be discovered using the service registry.

- The "apiName" shall be set to "app_pkgm" for application package management interface, or "app_lcm" for application life cycle management interface.
- The "apiVersion" shall be set to "v1" for the present document. All resource URIs in the sub-clauses below is defined relative to the above root URI.

The API shall support HTTP over TLS (also known as HTTPS) using TLS version 1.2 as defined by IETF RFC 5246 [10]. TLS 1.3 (including the new specific requirements for TLS 1.2 implementations) defined by IETF RFC 8446 [14] should be supported. HTTP without TLS shall not be used. Versions of TLS earlier than 1.2 shall neither be supported nor used.

The API shall require using the OAuth 2.0 client credentials grant type according to IETF RFC 6749 [11] with bearer tokens according to IETF RFC 6750 [12]. See clause 6.16 of ETSI GS MEC 009 [4] for more information. How the token endpoint and client credentials are provisioned into the MEC applications is out of scope of the present document.

Due to the specific structure how application packages are identified, there are two resource sub-trees with identical structure provided which only differ in the identifier per individual application package resource. Application packages can be identified by a MEO-managed/MEAO-managed identifier known as appPkgId which is assigned during the application package onboarding process, or by an identifier known as appDId defined by the application vendor during application package creation. The set of packages identified by the appDId is a subset of the application packages identified by the appPkgId, containing all those packages that have completed their onboarding process and are available for use by the MEPM.

For any given appDId value, there shall be at most one associated appPkgId value in the whole resource tree visible to the MEPM.

Figure 7.2-1 illustrates the resource URI structure of application package management on the reference point of Mm1or Mm3.

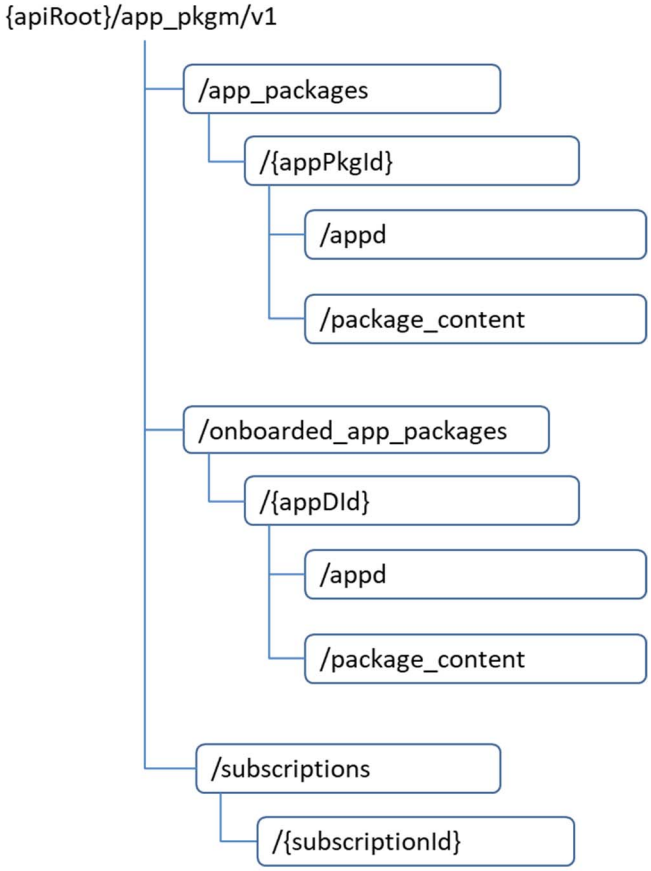

Figure 7.2-1: The resource URI structure of application package management

Figure 7.2-2 illustrates the resource URI structure of application life cycle management interface on the reference point of Mm1 or Mm3.

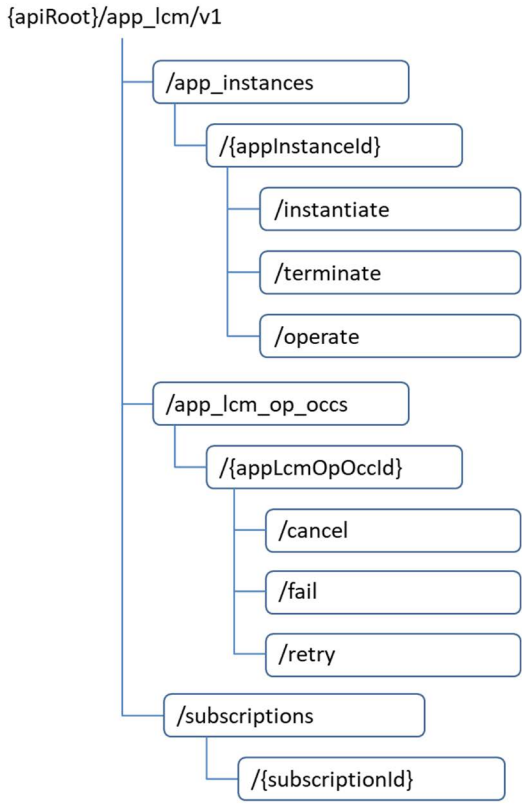

Figure 7.2-2: The resource URI structure of application life cycle management

Figure 7.2-3 illustrates the resource URI structure of granting on Mm3.

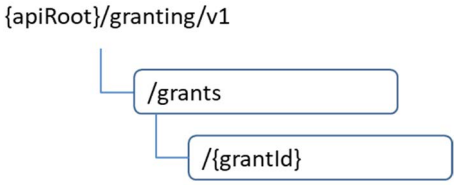

Figure 7.2-3: The resource URI structure of granting

Figure 7.2-4 illustrates the resource URI structure of application life cycle management interface on the reference point of Mm3*.

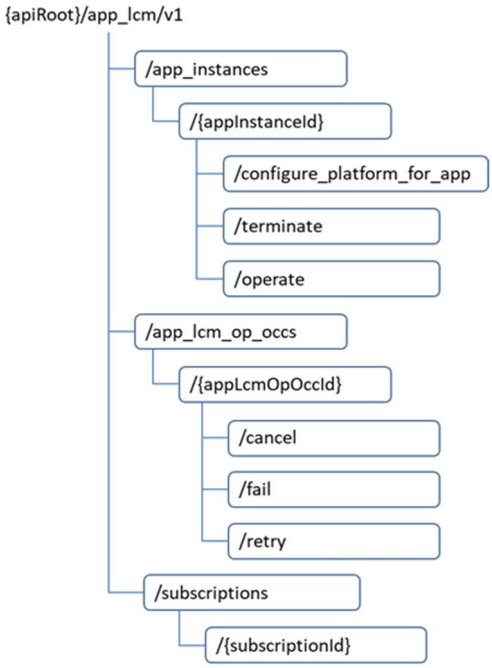

Figure 7.2-4: The resource URI structure of application life cycle management on Mm3*

Table 7.2-1 summarizes the resources and associated HTTP methods for application package management API over the Mm1 reference point.

Table 7.2-1: Overview of resources and methods of MEO's/MEAO's application package management on Mm1

Resource name	Resource URI	HTTP Method	Description
Application packages	/app_packages /onboarded_app_packages	POST	Create a new resource for on-boarded application package.
		GET	Query on-boarded application package information.
Individual application package	/app_packages/{appPkgId} /onboarded_app_packages/{appDId}	GET	Read information of an individual on-boarded application package.
		PATCH	Enable or disable an individual on-boarded application package.
		DELETE	Delete an individual on-boarded application package.
Application descriptor	/app_packages/{appPkgId}/appd /onboarded_app_packages/{appDId}/appd	GET	Read application descriptor of an onboarded application package.
Application package content	/app_packages/{appPkgId}/package_content /onboarded_app_packages/{appDId}/package_content	GET	Fetch an on-boarded application package.
		PUT	Upload an application package by providing the content of application package.
Subscriptions	/subscriptions	POST	Subscribe to notification related to on-boarding and/or changes of application packages.
		GET	Query multiple subscriptions
Individual subscription	/subscriptions/{subscriptionId}	GET	Read resource of an individual subscription.
		DELETE	Terminate an individual subscription.
Notification endpoint	(client provided)	POST	Notify application package on-boarding or change.

Table 7.2-2 summarizes the resources and associated HTTP methods for application life cycle management APIs over the Mm1 reference point.

Table 7.2-2: Overview of resources and methods of MEO's/MEAO's application life cycle management on Mm1

Resource name	Resource URI	HTTP Method	Description
Application instances	/app_instances	POST	Create an application instance resource.
		GET	Query multiple application instance resources.
Individual application instance	/app_instances/{appInstanceId}	GET	Read an application instance resource.
		DELETE	Delete individual application instance resource.
Instantiate application instance task	/app_instances/{appInstanceId}/instantiate	POST	Instantiate the application instance.
Terminate application instance task	/app_instances/{appInstanceId}/terminate	POST	Terminate the application instance.
Operate application instance task	/app_instances/{appInstanceId}/operate	POST	Start or stop the application instance.
Application LCM operation occurrences	/app_lcm_op_occs	GET	Query multiple application lifecycle operation occurrences.
Individual application LCM operation occurrence	/app_lcm_op_occs/{appLcmOpOccId}	GET	Read the operation state of the individual application lifecycle operation occurrence.
Application LCM operation cancel	/app_lcm_op_occs/{appLcmOpOccId}/cancel	POST	Cancel an ongoing individual application LCM operation.
Application LCM operation fail	/app_lcm_op_occs/{appLcmOpOccId}/fail	POST	Finally fail an ongoing individual application LCM operation.
Application LCM operation retry	/app_lcm_op_occs/{appLcmOpOccId}/retry	POST	Retry an ongoing individual application LCM operation.
Subscriptions	/subscriptions	POST	Subscribe to notifications related to application instances' lifecycle change.
		GET	Query multiple subscriptions.
Individual subscription	/subscriptions/{subscriptionId}	GET	Read an individual subscription resource.
		DELETE	Terminate an individual subscription.
Notification endpoint	(client provided)	POST	Notify about application instance's lifecycle change.

Table 7.2-3 summarizes the resources and associated HTTP methods for MEO's application package management APIs over Mm3 reference points.

Table 7.2-3: Overview of resources and methods of MEO's application package management on Mm3

Resource name	Resource URI	HTTP Method	Description
Application packages	/app_packages /onboarded_app_packages	GET	Query information about multiple on-boarded application packages.
Individual application package	/app_packages/{appPkgId} /onboarded_app_packages/{appDId}	GET	Read information about individual on-boarded application package.
Application package content	/app_packages/{appPkgId}/package_content /onboarded_app_packages/{appDId}/package_content	GET	Fetch an on-boarded application package.
Application descriptor	/app_packages/{appPkgId}/appd /onboarded_app_packages/{appDId}/appd	GET	Read the application descriptor of the on-boarded application package.
Subscriptions	/subscriptions	POST	Subscribe to notification related to on-boarding and/or changes of application packages.
		GET	Query multiple subscriptions.
Individual subscription	/subscriptions/{subscriptionId}	GET	Read an individual subscription.
		DELETE	Terminate an individual subscription.
Notification endpoint	(client provided)	POST	Notify application package on-boarding or change.

Table 7.2-4 summarizes the resources and associated HTTP methods for MEPM's application life cycle management API over Mm3 reference points.

Table 7.2-4: Overview of resources and methods of MEPM's application life cycle management on Mm3

Resource name	Resource URI	HTTP Method	Description
Application instances	/app_instances	POST	Create an application instance resource
		GET	Query multiple application instances
Individual application instance	/app_instances/{appInstanceId}	GET	Read application instance
		DELETE	Delete individual application instance
Instantiate application instance task	/app_instances/{appInstanceId}/instantiate	POST	Instantiate an application instance
Terminate application instance task	/app_instances/{appInstanceId}/terminate	POST	Terminate an application instance
Operate application instance task	/app_instances/{appInstanceId}/operate	POST	Start or stop an application instance
Application LCM operation occurrences	/app_lcm_op_occs	GET	Query multiple application lifecycle operation occurrences
Individual application LCM operation occurrence	/app_lcm_op_occs/{appLcmOpOccId}	GET	Read an individual application lifecycle management operation occurrence
Application LCM operation cancel	/app_lcm_op_occs/{appLcmOpOccId}/cancel	POST	Cancel an ongoing individual application LCM operation
Application LCM operation fail	/app_lcm_op_occs/{appLcmOpOccId}/fail	POST	Finally fail an ongoing individual application LCM operation
Application LCM operation retry	/app_lcm_op_occs/{appLcmOpOccId}/retry	POST	Retry an ongoing individual application LCM operation
Subscriptions	/subscriptions	POST	Subscribe to notifications related to application instance's lifecycle change
		GET	Query multiple subscriptions
Individual subscription	/subscriptions/{subscriptionId}	GET	Query an individual subscription
		DELETE	Terminate an individual subscription

Resource name	Resource URI	HTTP Method	Description
Notification endpoint	(client provided)	POST	Notify about application instance's lifecycle change

Table 7.2-5 summarizes the resources and associated HTTP methods for MEO's application life cycle management API over Mm3 reference points.

Table 7.2-5: Overview of resources and methods of MEO's application life cycle management on Mm3

Resource name	Resource URI	HTTP Method	Description
Grants	/grants	POST	Request a grant for a particular application LCM operation
Individual grant	/grants/{grantId}	GET	Read the status of grant for the application LCM operation

Table 7.2-6 summarizes the resources and associated HTTP methods for MEPM-V's application life cycle management API over Mm3* reference points.

Table 7.2-6: Overview of resources and methods of MEPM-V's application life cycle management on Mm3*

Resource name	Resource URI	HTTP Method	Description
Application instances	/app_instances	POST	Create an application instance resource
		GET	Query multiple application instances
Individual application instance	/app_instances/{appInstanceId}	GET	Read application instance
		DELETE	Delete individual application instance
Configure application instance task	/app_instances/{appInstanceId}/configure_platform_for_app	POST	Providing configuration information in AppD to the MEPM-V, intended to configure the MEP to run an application instance which is instantiated from the AppD
Terminate application instance task	/app_instances/{appInstanceId}/terminate	POST	Terminate an application instance at application level
Operate application instance task	/app_instances/{appInstanceId}/operate	POST	Start or stop an application instance at application level
Application LCM operation occurrences	/app_lcm_op_occs	GET	Query multiple application lifecycle operation occurrences
Individual application LCM operation occurrence	/app_lcm_op_occs/{appLcmOpOccId}	GET	Read an individual application lifecycle management operation occurrence
Application LCM operation cancel	/app_lcm_op_occs/{appLcmOpOccId}/cancel	POST	Cancel an ongoing individual application LCM operation
Application LCM operation fail	/app_lcm_op_occs/{appLcmOpOccId}/fail	POST	Finally fail an ongoing individual application LCM operation
Application LCM operation retry	/app_lcm_op_occs/{appLcmOpOccId}/retry	POST	Retry an ongoing individual application LCM operation
Subscriptions	/subscriptions	POST	Subscribe to notifications related to application instance's lifecycle change
		GET	Query multiple subscriptions
Individual subscription	/subscriptions/{subscriptionId}	GET	Query an individual subscription
		DELETE	Terminate an individual subscription
Notification endpoint	(client provided)	POST	Notify about application instance's lifecycle change

7.3 Resources of application package management on Mm1 and Mm3

7.3.1 Resource: application packages

7.3.1.1 Description

This resource is used to represent application packages of data type "AppPkgInfo" specified in clause 6.2.

7.3.1.2 Resource definition

The possible resource URIs are:

- Resource URI: {apiRoot}/app_pkgm/v1/app_packages/
- Resource URI: {apiRoot}/app_pkgm/v1/onboarded_app_packages/

Resource URI variables for this resource are defined in table 7.3.1.2-1.

Table 7.3.1.2-1: Resource URI variables for the resource

Name	Definition
apiRoot	See clause 7.2

7.3.1.3 Resource methods

7.3.1.3.1 POST

The POST method is used to create a resource for on-boarding an application package to a MEO/MEAO, which refers to the procedure of "onboarding operation" as described in clause 6.3.3.5. The POST method is also used to create a resource for on-boarding an application package from OSS through MEAO to NFVO as described in clause 5.6.2.

The POST method is supported on Mm1 only.

This method shall comply with the URI request and response data structures, and response codes, as specified in tables 7.3.1.3.1-1 and 7.3.1.3.1-2.

Table 7.3.1.3.1-1: URI query parameters of POST method on the resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.3.1.3.1-2: Data structures supported by POST request/response on this resource

Request message content	**Data type**	**Cardinality**	**Remarks**	
	CreateAppPkg	1	The POST method is used to create a new resource for onboarding an application package onto a MEO/MEAO.	
Response message content	**Data type**	**Cardinality**	**Response codes**	**Remarks**
	AppPkgInfo	1	201 Created	Indicates a successful request. The response message content shall contain a representation of the application package resource defined in clause 6.2. The HTTP response includes a "Location" HTTP header that contains the URI of the created resource.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.3.1.3.2 GET

This GET method queries information relating to on-boarded application packages in the MEO/MEAO matching the filtering criteria. It refers to the procedure of "query operation" of application package as described in clause 6.3.3.2. This GET method also queries information relating to on-boarded application packages in the NFVO matching the filtering criteria by the OSS through MEAO to NFVO.

The GET method is supported on Mm1 and Mm3.

This method shall comply with the URI request and response data structures, and response codes, as specified in tables 7.3.1.3.2-1 and 7.3.1.3.2-2.

Table 7.3.1.3.2-1: URI query parameters of GET method on the resource

Attribute name	Cardinality	Description
filter	0..1	Attribute-based filtering parameters according to ETSI GS MEC 009 [4]. The API producer shall support receiving filtering parameters as part of the URI query string. All attribute names that appear in the AppPkgInfo and in data types referenced from it shall be supported in attribute-based filtering parameters. See clause 6.19 in ETSI GS MEC 009 [4] for details.
all_fields	0..1	Include all complex attributes in the response. See clause 6.18 in ETSI GS MEC 009 [4] for details. The API producer shall support this parameter.
fields	0..1	Complex attributes of AppPkgInfo to be included into the response. See clause 6.18 in ETSI GS MEC 009 [4] for details. The API producer should support this parameter.
exclude_fields	0..1	Complex attributes of AppPkgInfo to be excluded from the response. See clause 6.18 in ETSI GS MEC 009 [4] for details. The API producer should support this parameter.
exclude_default	0..1	Indicates to exclude the following complex attributes of AppPkgInfo from the response. The following attributes shall be excluded from the AppPkgInfo structure in the response message content if this parameter is provided, or none of the parameters "all_fields", "fields", "exclude_fields", "exclude_default" are provided: • checksum; • softwareImages; • additionalArtifacts.

Table 7.3.1.3.2-2: Data structures supported by GET request/response on this resource

Request message content	Data type	Cardinality	Remarks	
	n/a			
Response message content	**Data type**	**Cardinality**	**Response codes**	**Remarks**
	AppPkgInfo	0..N	200 OK	Indicate the success of request. The response message content shall contain a list of representations of the "individual application package" resources that match the attribute filter.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	400 Bad Request	Error: Invalid attribute-based filtering expression. In the returned ProblemDetails structure, the "detail" attribute shall convey more information about the error.
	ProblemDetails	1	400 Bad Request	Error: Invalid attribute selector. In the returned ProblemDetails structure, the "detail" attribute shall convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.

	Data type	Cardinality	Response codes	Remarks
Response message content	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.3.1.3.3 PUT

Not supported.

7.3.1.3.4 DELETE

Not supported.

7.3.1.3.5 PATCH

Not supported.

7.3.2 Resource: individual application package

7.3.2.1 Description

This resource is used to represent an individual application package of data type "AppPkgInfo" specified in clause 6.2.

7.3.2.2 Resource definition

The possible resource URIs are:

- Resource URI: {apiRoot}/app_pkgm/v1/app_packages/{appPkgId}
- Resource URI: {apiRoot}/app_pkgm/v1/onboarded_app_packages/{appDId}

Resource URI variables for this resource are defined in table 7.3.2.2-1.

Table 7.3.2.2-1: Resource URI variables for the resource

Name	Definition
apiRoot	See clause 7.2
appPkgId	Identifier of an individual application package resource

7.3.2.3 Resource methods

7.3.2.3.1 POST

Not supported.

7.3.2.3.2 GET

This GET method is used to query the information related to individual application package resources.

The GET method is supported on Mm1 and Mm3.

This method shall comply with the URI request and response data structures, and response codes, as specified in tables 7.3.2.3.2-1 and 7.3.2.3.2-2.

Table 7.3.2.3.2-1: URI query parameters of GET method on the resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.3.2.3.2-2: Data structures supported by GET request/response on this resource

Request message content	**Data type**	**Cardinality**	**Remarks**	
	n/a			
Response message content	**Data type**	**Cardinality**	**Response codes**	**Remarks**
	AppPkgInfo	1	200 OK	Indicates the success of request. The response message content shall contain a representation of the resource.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.3.2.3.3 PUT

Not supported.

7.3.2.3.4 DELETE

This DELETE method realizes the procedure of "delete operation" of application package resource in MEO/MEAO or the procedure of "delete operation" of application package resource in NFVO by OSS through MEAO as described in clause 6.3.3.9.

The DELETE method is supported on Mm1 only.

This method shall comply with the URI request and response data structures, and response codes, as specified in tables 7.3.2.3.4-1 and 7.3.2.3.4-2, which refer to table 6.3.3.9.2-1.

Table 7.3.2.3.4-1: URI query parameters supported by the DELETE method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.3.2.3.4-2: Data structures supported by the DELETE request/response on this resource

Request message content	Data type	Cardinality	Remarks	
	n/a			
Response message content	**Data type**	**Cardinality**	**Response codes**	**Remarks**
	n/a		204 No Content	Upon successful deletion of application package resource, the response message content shall be empty.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.3.2.3.5 PATCH

This PATCH method updates the operational state of an individual application package resource used by the procedure of "enable operation" as described in clause 6.3.3.6, "disable operation" as described in clause 6.3.3.7.

The PATCH method is supported on Mm1 only.

This method shall comply with the URI query parameters, request and response data structures, and response codes, as specified in tables 7.3.2.3.5-1 and 7.3.2.3.5-2.

Table 7.3.2.3.5-1: URI query parameters of PATCH method on the resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.3.2.3.5-2: Data structures supported by PATCH request/response on this resource

Request message content	Data type	Cardinality	Remarks	
	AppPkgInfoModifications	1	Parameters for application package information modification.	
Response message content	**Data type**	**Cardinality**	**Response Codes**	**Remarks**
	AppPkgInfoModifications	1	200 OK	Shall be returned when the operation has been completed successfully.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

	Data type	Cardinality	Response Codes	Remarks
Response message content	ProblemDetails	1	409 Conflict	Shall be returned upon the following errors: - 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: • Disable an application package resource of which the operational state is not ENABLED. • Enable an application package resource of which the operational state is not DISABLED. - The operation of onboarding the application package has not completed. The response message content shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.3.3 Resource: subscriptions

7.3.3.1 Description

This resource is used to represent subscriptions to notifications about application package changes. The subscriber can use this resource to subscribe to notifications related to the application package management.

7.3.3.2 Resource definition

The possible resource URIs are:

- Resource URI: {apiRoot}/app_pkgm/v1/subscriptions.

Resource URI variables for this resource are defined in table 7.3.3.2-1.

Table 7.3.3.2-1: Resource URI variables for the resource

Name	Definition
apiRoot	See clause 7.2

7.3.3.3 Resource methods

7.3.3.3.1 POST

The POST method is used to subscribe to notifications about on-boarding an application package, or about operational state changes of on-boarded application package, which is mapped to the procedure of "subscription operation" as described in clause 6.3.3.3.

The POST method for subscriptions is supported on Mm1 and Mm3.

This method shall comply with the URI request and response data structures, and response codes, as specified in the table 7.3.3.3.1-1.

Table 7.3.3.3.1-1: Data structures supported by POST request/response on this resource

Request message content	**Data type**	**Cardinality**	**Remarks**	
	AppPkgSubscription	1	The input parameters of "subscribe operation" to notifications about changes related to application package management for the on-boarding, or operational state change of application package.	
Response message content	**Data type**	**Cardinality**	**Response codes**	**Remarks**
	AppPkgSubscriptionInfo	1	201 Created	Upon success, a response message content representing the created subscription shall be returned.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.3.3.3.2 GET

This GET method is used to retrieve the information of subscriptions to individual application package resource in MEO or MEAO. Upon success, the response contains the list of links to the subscriptions that are present for the requestor.

The GET method is supported on Mm1 and Mm3.

This method shall comply with the URI query parameters, request and response data structures, and response codes, as specified in tables 7.3.3.3.2-1 and 7.3.3.3.2-2.

Table 7.3.3.3.2-1: URI query parameters supported by the GET method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.3.3.3.2-2: Data structures supported by the GET request/response on this resource

Request message content	Data type	Cardinality	Remarks	
	n/a			
Response message content	**Data type**	**Cardinality**	**Response codes**	**Remarks**
	AppPkgSubscriptionLinkList	1	200 OK	Upon success, a response message content containing a list of zero or more subscriptions shall be returned.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.3.3.3.3 PUT

Not supported.

7.3.3.3.4 DELETE

Not Supported.

7.3.3.3.5 PATCH

Not Supported.

7.3.4 Resource: individual subscription

7.3.4.1 Description

This resource is used to represent an individual subscription to notifications about application package changes, which is mapped to the procedure of "subscription operation" as described in clause 6.3.3.3.

7.3.4.2 Resource definition

The possible resource URIs are:

- Resource URI: {apiRoot}/app_pkgm/v1/subscriptions/{subscriptionId}.

Resource URI variables for this resource are defined in table 7.3.4.2-1.

Table 7.3.4.2-1: Resource URI variables for the resource

Name	Definition
apiRoot	See clause 7.2.
subscriptionId	Identifier of an individual subscription to notifications about application package changes.

7.3.4.3 Resource methods

7.3.4.3.1 POST

Not supported.

7.3.4.3.2 GET

This GET method is used to retrieve the individual subscription information to the application package resource in MEO or MEAO.

The GET method is supported on Mm1 and Mm3.

This method shall comply with the URI query parameters, request and response data structures, and response codes, as specified in tables 7.3.4.3.2-1 and 7.3.4.3.2-2.

Table 7.3.4.3.2-1: URI query parameters supported by the GET method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.3.4.3.2-2: Data structures supported by GET request/response on this resource

Request message content	Data type	Cardinality	Remarks	
	n/a			
Response message content	**Data type**	**Cardinality**	**Response codes**	**Remarks**
	AppPkgSubscriptionInfo	1	200 OK	Upon success, a response message content containing a representation of the resource shall be returned.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.3.4.3.3 PUT

Not supported.

7.3.4.3.4 DELETE

This DELETE method is used to delete the individual subscription to notifications about application package changes in MEO or MEAO.

The DELETE method is supported on Mm1 and Mm3.

This method shall comply with the URI request and response data structures, and response codes, as specified in tables 7.3.4.3.4-1 and 7.3.4.3.4-2.

Table 7.3.4.3.4-1: URI query parameters supported by DELETE method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.3.4.3.4-2: Data structures supported by DELETE request/response on this resource

Request message content	Data type	Cardinality	Remarks	
	n/a			
Response message content	**Data type**	**Cardinality**	**Response codes**	**Remarks**
	n/a		204 No Content	The subscription resource was deleted successfully. The response message content shall be empty.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.3.4.3.5 PATCH

Not supported.

7.3.5 Resource: notification endpoint

7.3.5.1 Description

This resource is used to represent a notification endpoint, which is mapped to the procedure of "notify application package operation" as described in clause 6.3.3.3. The API producer can use this resource to send notifications related to application package management events to a subscribed API consumer.

7.3.5.2 Resource definition

The resource of callback URI is provided by the subscriber when subscribing to the notification.

Resource URI variables for this resource are defined in table 7.3.5.2-1.

Table 7.3.5.2-1: Resource URI variables for the resource

Name	Definition
n/a	

7.3.5.3 Resource methods

7.3.5.3.1 POST

The POST method delivers a notification from the application package management resource in MEO or MEAO to the subscriber.

The POST method is supported on Mm1 and Mm3.

This method shall follow the provisions specified in tables 7.3.5.3.1-1 and 7.3.5.3.1-2 for URI parameters, request and response data structures, and response codes.

Table 7.3.5.3.1-1: URI query parameters supported by POST method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.3.5.3.1-2: Data structures supported by POST request/response on this resource

Request message content	Data type	Cardinality	Remarks	
	AppPkgNotification	1	A notification of an application package for on-boarding or operational state change.	
Response message content	**Data type**	**Cardinality**	**Response codes**	**Remarks**
	n/a		204 No Content	The notification was delivered successfully. The response message content shall be empty.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.3.5.3.2 GET

Not supported.

7.3.5.3.3 PUT

Not supported.

7.3.5.3.4 DELETE

Not supported.

7.3.5.3.5 PATCH

Not supported.

7.3.6 Resource: application descriptor

7.3.6.1 Description

This resource represents an application descriptor "AppD" contained in an on-boarded application package. The client can use this resource to obtain the content of the AppD specified in clause 6.2 over either the Mm1 or Mm3 reference point.

7.3.6.2 Resource definition

The possible resource URIs are:

- Resource URI: {apiRoot}/app_pkgm/v1/app_packages/{appPkgId}/appd
- Resource URI: {apiRoot}/app_pkgm/v1/onboarded_app_packages/{appDId}/appd

Resource URI variables for this resource are defined in table 7.3.6.2-1.

Table 7.3.6.2-1: Resource URI variables for the resource

Name	Definition
apiRoot	See clause 7.2.
appPkgId	Identifier of an on-boarded individual application package.

7.3.6.3 Resource methods

7.3.6.3.1 POST

This method is not supported. When this method is requested on this resource, the "405 Method Not Allowed" response is returned.

7.3.6.3.2 GET

This GET method reads the content of the AppD of on-boarded individual application package resources. The format of the AppD is left for future specification.

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" (with appropriate indication of character set) and the AppD 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 AppD shall be returned embedded in a ZIP file.
- If the "Accept" header contains both "text/plain" (with appropriate indication of character set) and "application/zip", it is up to the MEO or MEAO to choose the format to return for a single-file AppD; for a multi-file AppD, a ZIP file shall be returned.

The default format of the ZIP file can be referred to ETSI GS NFV-SOL 004 [18] where only the YAML files representing the AppD, and information needed to navigate the ZIP file and to identify the file that is the entry point for parsing the AppD (such as TOSCA-meta or manifest files or naming conventions) are included.

The GET method is supported on Mm1 and Mm3.

This method shall comply with the URI request and response data structures, and response codes, as specified in tables 7.3.6.3.2-1 and 7.3.6.3.2-2.

Table 7.3.6.3.2-1: URI query parameters of GET method on the resource

Attribute name	Cardinality	Description
filter	0..1	Attribute-based filtering parameters according to ETSI GS MEC 009 [4]. The API producer shall support receiving filtering parameters as part of the URI query string. All attribute names that appear in the AppD and in data types referenced from it shall be supported in attribute-based filtering parameters. See clause 6.19 in ETSI GS MEC 009 [4] for detail.
all_fields	0..1	Include all complex attributes in the response. See clause 6.18 in ETSI GS MEC 009 [4] for details. The API producer shall support this parameter.
fields	0..1	Complex attributes of AppD to be included into the response. See clause 6.18 in ETSI GS MEC 009 [4] for details. The API producer should support this parameter.
exclude_fields	0..1	Complex attributes of AppD to be excluded from the response. See clause 6.18 in ETSI GS MEC 009 [4] for details. The API producer should support this parameter.
exclude_default	0..1	Indicates to exclude the following complex attributes of AppD from the response. The following attributes shall be excluded from the AppPkgInfo structure in the response message content if this parameter is provided, or none of the parameters "all_fields", "fields", "exclude_fields", "exclude_default" are provided: - virtualComputeDescriptor; - swImageDescriptor; - virtualStorageDescriptor; - appExtCpd; - terminateAppInstanceOpConfig; - changeAppInstanceStateOpConfig.

Table 7.3.6.3.2-2: Data structures supported by GET request/response on this resource

Request message content	Data type	Cardinality	Remarks	
	n/a			
Response message content	**Data type**	**Cardinality**	**Response codes**	**Remarks**
	AppD	1	200 OK	Indicates the success of request, and the content of the AppD is returned. The response message content shall contain a copy of the file representing the AppD or a ZIP file that contains the file or multiple files representing the AppD. The "Content-Type" HTTP header shall be set according to the format of the returned file, which is selected according to "Accept" HTTP header options passed in the request.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that the request contained either: • incorrect parameters; or • an invalid attribute-based filtering expression; or • an invalid attribute selector. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the specific error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

	Data type	Cardinality	Response codes	Remarks
Response message content	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.3.6.3.3 PUT

This method is not supported. When this method is requested on this resource, the "403 Forbidden" response is returned.

7.3.6.3.4 DELETE

This method is not supported. When this method is requested on this resource, the "405 Method Not Allowed" response is returned.

7.3.6.3.5 PATCH

Not supported.

7.3.7 Resource: application package content

7.3.7.1 Description

This resource represents an individual application package identified by the application package identifier, or application descriptor identifier allocated by the MEO or MEAO. The client can use this resource to upload or fetch the content of the application package.

7.3.7.2 Resource definition

The possible resource URIs are:

- Resource URI: {apiRoot}/app_pkgm/v1/app_packages/{appPkgId}/package_content
- Resource URI: {apiRoot}/app_pkgm/v1/onboarded_app_packages/{appDId}/package_content

Resource URI variables for this resource are defined in table 7.3.7.2-1.

Table 7.3.7.2-1: Resource URI variables for the resource

Name	Definition
apiRoot	See clause 7.2
appPkgId	Identifier of an individual application package
appDId	Identifier of an application descriptor

7.3.7.3 Resource methods

7.3.7.3.1 POST

Not supported.

7.3.7.3.2 GET

The GET method is used to fetch the onboarded application package content identified by appPkgId or appDId. The client can use this resource to fetch the content of the application package.

The GET method is supported on Mm1 and Mm3.

This method shall comply with the URI query parameters, request and response data structures, and response codes, as specified in tables 7.3.7.3.2-1 and 7.3.7.3.2-2.

Table 7.3.7.3.2-1: URI query parameters of GET method on the resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.3.7.3.2-2: Data structures supported by GET request/response on this resource

	Data type	Cardinality	Remarks	
Request message content	n/a		The request may contain a "Range" HTTP header to obtain single range of bytes from the application package file. This can be used to continue an aborted transmission. If the MEO or MEAO does not support range requests, it should return the whole file with a 200 OK response instead.	
Response message content	**Data type**	**Cardinality**	**Response codes**	**Remarks**
	n/a	1	200 OK	On success, a copy of the on-boarded application package is returned in the response message content. The "Content-Type" HTTP header shall be set according to the type of the file, i.e. to "application/zip" for an application package.
	n/a	1	206 Partial Content	On success, if the MEO or MEAO supports range requests, a single consecutive byte range from the content of the application package file shall be returned. The response message content shall contain the requested part of the application package file. The "Content-Range" HTTP header shall be provided according to IETF RFC 9110 [6]. The "Content-Type" HTTP header shall be set as defined above for the "200 OK" response.

	Data type	Cardinality	Response codes	Remarks
Response message content	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should 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 application package file (e.g. "access after end of file"). The response message content may contain a ProblemDetails structure.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.3.7.3.3 PUT

This PUT method uploads the content of application package.

The POST method is supported on Mm1 only.

This method shall comply with the URI query parameters, request and response data structures, and response codes, as specified in tables 7.3.7.3.3-1 and 7.3.7.3.3-2.

Table 7.3.7.3.3-1: URI query parameters of PUT method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.3.7.3.3-2: Data structures supported by PUT request/response on this resource

Request message content	**Data type**	**Cardinality**	**Remarks**	
	n/a		The message content shall contain a ZIP file that represents the application package. The "Content-Type" HTTP header shall be set to "application/zip".	
Response message content	**Data type**	**Cardinality**	**Response Codes**	**Remarks**
	n/a		202 Accepted	The application package has been accepted for uploading, but the processing has not been completed. It is expected to take some time for processing. The response message content shall be empty.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	409 Conflict	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 onboarding state of the application package resource is not CREATED. The response message content shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.3.7.3.4 DELETE

Not supported.

7.3.7.3.5 PATCH

Not supported.

7.4 Resources of application lifecycle management on Mm1

7.4.1 Resource: application instances

7.4.1.1 Description

This resource represents application instances of data type "AppInstanceInfo" specified in clause 6.2. The consumer can use this resource to request resource allocation for application instance and query the information of resource.

7.4.1.2 Resource definition

The possible resource URIs are:

- Resource URI: {apiRoot}/app_lcm/v1/app_instances.

Resource URI variables for this resource are defined in table 7.4.1.2-1.

Table 7.4.1.2-1: Resource URI Variables for the resource

Name	Definition
apiRoot	See clause 7.2

7.4.1.3 Resource methods

7.4.1.3.1 POST

The POST method is used to create an application instance resource, which refers to the procedure of "creating application instance resource operation" described in clause 6.3.

This method shall comply with the URI request and response data structures, and response codes, as specified in tables 7.4.1.3.1-1 and 7.4.1.3.1-2.

Table 7.4.1.3.1-1: URI query parameters of POST method on the resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.4.1.3.1-2: Data structures supported by POST request/response on this resource

Request message content	**Data type**	**Cardinality**	**Remarks**	
	CreateAppInstance Request	1	The POST method is used to create an application instance resource.	
Response message content	**Data type**	**Cardinality**	**Response codes**	**Remarks**
	AppInstanceInfo	1	201 Created	An application instance identifier and the related resource has been created successfully. The response message content shall contain a representation of the created resource. The HTTP response shall include a "Location" HTTP header that contains the resource URI of the created application instance.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
Response message content	**Data type**	**Cardinality**	**Response codes**	**Remarks**
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.4.1.3.2 GET

The GET method retrieves information about the application instances resources. This refers to the procedure of "query application instance information operation" as described in clause 6.3.1.5.

This method shall comply with the URI request and response data structures, and response codes, as specified in tables 7.4.1.3.2-1 and 7.4.1.3.2-2.

Table 7.4.1.3.2-1: URI query parameters of GET method on the resource

Name	Cardinality	Description
filter	0..1	Attribute-based filtering parameters according to ETSI GS MEC 009 [4]. The API producer shall support receiving filtering parameters as part of the URI query string. All attribute names that appear in the AppInstanceInfo and in data types referenced from it shall be supported in attribute-based filtering parameters. See clause 6.19 in ETSI GS MEC 009 [4] for details.
all_fields	0..1	Include all complex attributes in the response. See clause 6.18 in ETSI GS MEC 009 [4] for details. The API producer shall support this parameter.
fields	0..1	Complex attributes to be included into the response. See clause 6.18 in ETSI GS MEC 009 [4] for details. The API producer should support this parameter.
exclude_fields	0..1	Complex attributes to be excluded from the response. See clause 6.18 in [4] for details. The API producer should support this parameter.
exclude_default	0..1	Indicates to exclude the following complex attributes from the response. See clause 6.18 in ETSI GS MEC 009 [4] for details. The API producer shall support this parameter. The following attributes shall be excluded from the AppInstanceInfo structure in the response message content if this parameter is provided, or none of the parameters "all_fields", "fields", "exclude_fields", "exclude_default" are provided: - vimConnectionInfo; - instantiate; - terminate; - operate.

Table 7.4.1.3.2-2: Data structures supported by GET request/response on this resource

Request message content	Data type	Cardinality	Remarks	
	n/a	0	The method is to query the information of application instances.	
Response message content	**Data type**	**Cardinality**	**Response codes**	**Remarks**
	AppInstanceInfo	0..N	200 OK	Information about zero or more application instances was queried successfully. The response message content shall contain in an array the representations of zero or more application instances.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.4.1.3.3 PUT

Not supported.

7.4.1.3.4 DELETE

Not supported.

7.4.1.3.5 PATCH

Not supported.

7.4.2 Resource: individual application instance

7.4.2.1 Description

This method is to retrieve information about an individual application instance. The client can use this resource to read the information of, or delete the individual application instance.

7.4.2.2 Resource definition

The possible resource URIs are:

- Resource URI: {apiRoot}/app_lcm/v1/app_instances/{appInstanceId}.

Resource URI Variables for this resource are defined in table 7.4.2.2-1.

Table 7.4.2.2-1: Resource URI Variables for the resource

Name	Definition
apiRoot	See clause 7.2
appInstanceId	Identifier of an individual application instance

7.4.2.3 Resource methods

7.4.2.3.1 POST

Not supported.

7.4.2.3.2 GET

The GET method retrieves the information of an individual application instance via reading an individual application instance resource, which is used by the procedure of "query application instance information operation" as described in clause 6.3.1.5.

This method shall comply with the URI request and response data structures, and response codes, as specified in tables 7.4.2.3.2-1 and 7.4.2.3.2-2.

Table 7.4.2.3.2-1: URI query parameters of GET method on the resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.4.2.3.2-2: Data structures supported by GET request/response on this resource

Request message content	**Data type**	**Cardinality**	**Remarks**	
	n/a	0		
Response message content	**Data type**	**Cardinality**	**Response codes**	**Remarks**
	AppInstanceInfo	1	200 OK	Information about an individual application instance was read successfully. The response message content shall contain a representation of the read resource.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

	Data type	Cardinality	Response codes	Remarks
Response message content	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.4.2.3.3 PUT

Not supported.

7.4.2.3.4 DELETE

The DELETE method deletes an individual application instance resource, which refers to the procedure of "delete application instance identifier operation" as described in clause 6.3.1.8.

The method shall comply with the URI request and response data structures, and response codes, as specified in tables 7.4.2.3.4-1 and 7.4.2.3.4-2.

Table 7.4.2.3.4-1: URI query parameters supported by the DELETE method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.4.2.3.4-2: Data structures supported by the DELETE request/response on this resource

Request message content	Data type	Cardinality	Remarks	
	n/a			
Response message content	**Data type**	**Cardinality**	**Response codes**	**Remarks**
	n/a		204 No content	The application instance resource and the associated application instance identifier were deleted successfully. The response message content shall be empty.
	ProblemDetails	1	409 Conflict	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 application instance resource is in INSTANTIATED state. The response message content shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.4.2.3.5 PATCH

Not supported.

7.4.3 Resource: subscriptions

7.4.3.1 Description

This resource represents subscriptions to notifications of related to an application instance. The subscriber can use this resource to subscribe to notifications related to the application instance related changes, such as application instance operational state change or application LCM operation occurrence state change, as well as creation/deletion of MEC application instance identifiers and the associated application instances.

When this resource represents a subscription to notifications regarding application instance operational state changes, it shall follow the data type of "AppInstSubscriptionRequest" as specified in clause 6.2.2.12. The notifications related to this subscription shall follow the data type of "AppInstNotification" as specified in clause 6.2.2.11.

When this resource represents a subscription to the notifications regarding to application instance LCM operation occurrence state change, it shall follow the data type of "AppLcmOpOccSubscriptionRequest" as specified in clause 6.2.2.14. The notifications related to this subscription shall follow the data type of "AppLcmOpOccNotification" as specified in clause 6.2.2.16.

When this resource represents a subscription to the notifications regarding to creation of application instance identifier, it shall follow the data type of "AppInstIdCreationSubscriptionRequest" as specified in clause 6.2.2.25. The notifications related to this subscription shall follow the data type of "AppInstanceIdentifierCreationNotification" as specified in clause 6.2.2.27.

When this resource represents a subscription to the notifications regarding to deletion of application instance identifier, it shall follow the data type of "AppInstIdDeletionSubscriptionRequest" as specified in clause 6.2.2.28. The notifications related to this subscription shall follow the data type of "AppInstanceIdentifierDeletionNotification" as specified in clause 6.2.2.30.

7.4.3.2 Resource definition

The possible resource URIs are:

- Resource URI: {apiRoot}/app_lcm/v1/subscriptions.

Resource URI Variables for this resource are defined in table 7.4.3.2-1.

Table 7.4.3.2-1: Resource URI variables for the resource

Name	Definition
apiRoot	See clause 7.2

7.4.3.3 Resource methods

7.4.3.3.1 POST

The POST method is to subscribe to the notification of application instance related change, which is mapped to the procedure of "subscription operation".

This method shall comply with the URI query parameters, request and response data structures, and response codes, as specified in tables 7.4.3.3.1-1 and 7.4.3.3.1-2.

Table 7.4.3.3.1-1: URI query parameters supported by the POST method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.4.3.3.1-2: Data structures supported by the POST request/response on this resource

	Data type	Cardinality	Remarks	
Request message content	{SubscriptionRequest}	1	The subscription request message content contains the data type of the specific subscription to application instance state change event that is to be created: • AppInstSubscriptionRequest • AppLcmOpOccSubscriptionRequest • AppInstIdCreationSubscriptionRequest • AppInstIdDeletionSubscriptionRequest	
Response message content	**Data type**	**Cardinality**	**Response codes**	**Remarks**
	{SubscriptionInfo}	1	201 Created	Upon success, a response message content representing the created subscription is returned. The response message content shall contain a representation of the created SubscriptionInfo with the appropriate data type: • AppInstSubscriptionInfo • AppLcmOpOccSubscriptionInfo • AppInstIdCreationSubscriptionInfo • AppInstIdDeletionSubscriptionInfo The HTTP response shall include a "Location" HTTP header that contains the resource URI of the created subscription resource.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.4.3.3.2 GET

The GET method retrieves the information of multiple subscriptions to notifications related to an application instance.

Upon success, the response contains the list of links to the subscriptions that are present for the requestor.

Table 7.4.3.3.2-1: URI query parameters supported by the GET method on this resource

Name	Data type	Cardinality	Remarks
subscriptionType	AppInstanceSubscriptionType	0..1	Query parameter to filter on a specific subscription type.

Table 7.4.3.3.2-2: Data structures supported by the GET request/response on this resource

Request message content	Data type	Cardinality	Remarks	
	n/a			
Response message content	**Data type**	**Cardinality**	**Response codes**	**Remarks**
	AppInstanceSubscriptionLinkList	1	200 OK	Upon success, a response message content containing a list of all subscriptions is returned.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.4.3.3.3 PUT

Not supported.

7.4.3.3.4 DELETE

Not Supported.

7.4.3.3.5 PATCH

Not supported.

7.4.4 Resource: individual subscription

7.4.4.1 Description

This resource represents an individual subscription to notifications related to an application instance.

7.4.4.2 Resource definition

The possible resource URIs are:

- Resource URI: {apiRoot}/app_lcm/v1/subscriptions/{subscriptionId}.

Resource URI variables for this resource are defined in table 7.4.4.2-1.

Table 7.4.4.2-1: Resource URI variables for the resource

Name	Definition
apiRoot	See clause 7.2.
subscriptionId	Represents an individual subscription to notification related to an application instance.

7.4.4.3 Resource methods

7.4.4.3.1 POST

Not supported.

7.4.4.3.2 GET

The GET method retrieves the individual subscription information by reading an individual subscription resource. This method shall comply with the URI query parameters, request and response data structures, and response codes, as specified in tables 7.4.4.3.2-1 and 7.4.4.3.2-2.

Table 7.4.4.3.2-1: URI query parameters supported by the GET method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.4.4.3.2-2: Data structures supported by the GET request/response on this resource

Request message content	Data type	Cardinality	Remarks	
	n/a			
Response message content	**Data type**	**Cardinality**	**Response codes**	**Remarks**
	{SubscriptionInfo}	1	200 OK	Upon success, a response message content containing a representation of the individual subscription resource shall be returned. The allowed data types for the SubscriptionInfo are: • AppInstSubscriptionInfo • AppLcmOpOccSubscriptionInfo • AppInstIdCreationSubscriptionInfo • AppInstIdDeletionSubscriptionInfo
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the ProblemDetails structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.4.4.3.3 PUT

Not supported.

7.4.4.3.4 DELETE

The DELETE method is used to delete an individual subscription to notifications related to application instances.

This method shall comply with the URI query parameters, request and response data structures, and response codes, as specified in tables 7.4.4.3.4-1 and 7.4.4.3.4-2.

Table 7.4.4.3.4-1: URI query parameters supported by the DELETE method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.4.4.3.4-2: Data structures supported by the DELETE request/response on this resource

Request message content	Data type	Cardinality	Remarks	
	n/a			
Response message content	Data type	Cardinality	Response codes	Remarks
	n/a		204 No Content	The subscription resource was deleted successfully. The response message content shall be empty.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the ProblemDetails structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.4.4.3.5 PATCH

Not supported.

7.4.5 Resource: notification endpoint

7.4.5.1 Description

This resource represents a notification endpoint, which is mapped to the procedure of "notify application instance operational state change", "notify application LCM operation occurrence state change", or "creation/deletion of MEC application instance identifiers and the associated application instances". The producer can use this resource to send notifications related to application instance management events to a subscribed consumer.

7.4.5.2 Resource definition

The resource of callback URI is provided by the subscriber when subscribing to the notification.

Resource URI variables for this resource are defined in table 7.4.5.2-1.

Table 7.4.5.2-1: Resource URI variables for the resource

Name	Definition
n/a	

7.4.5.3 Resource methods

7.4.5.3.1 POST

The POST method delivers a notification from the application lifecycle management resource to the subscriber.

This method shall follow the provisions specified in tables 7.4.5.3.1-1 and 7.4.5.3.1-2 for URI parameters, request and response data structures, and response codes.

Table 7.4.5.3.1-1: URI query parameters supported by POST method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.4.5.3.1-2: Data structures supported by POST request/response on this resource

	Data type	Cardinality	Response codes	Remarks
Request message content	{Notification}	1		A notification of an event related to an application instance. The allowed data types of application state change notification are: • AppInstNotification • AppLcmOpOccNotification • AppInstanceIdentifierCreationNotification • AppInstanceIdentifierDeletionNotification
Response message content	**Data type**	**Cardinality**	**Response codes**	**Remarks**
	n/a		204 No Content	The notification was delivered successfully. The response message content shall be empty.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.4.5.3.2 GET

Not supported.

7.4.5.3.3 PUT

Not supported.

7.4.5.3.4 DELETE

Not supported.

7.4.5.3.5 PATCH

Not supported.

7.4.6 Resource: instantiate application instance task

7.4.6.1 Description

This resource represents the task of instantiating an application instance. The client can use this resource to instantiate an application instance.

7.4.6.2 Resource definition

The possible resource URIs are:

- Resource URI: {apiRoot}/app_lcm/v1/app_instances/{appInstanceId}/instantiate

Resource URI variables for this resource are defined in table 7.4.6.2-1.

Table 7.4.6.2-1: Resource URI Variables for the resource

Name	Definition
apiRoot	See clause 7.2.
appInstanceId	The identifier of the application instance. 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 the new application instance resource. It can also be retrieved from the "id" attribute in the response message content.	

7.4.6.3 Resource methods

7.4.6.3.1 POST

The POST method is to instantiate the application instance.

This method shall comply with the URI request and response data structures, and response codes, as specified in tables 7.4.6.3.1-1 and 7.4.6.3.1-2.

Table 7.4.6.3.1-1: URI query parameters supported by the POST method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.4.6.3.1-2: Data structures supported by a POST request/response on this resource

Request message content	Data type	Cardinality	Remarks	
	InstantiateAppRequest	1	The request message content contains the information necessary to instantiate an application instance.	
Response message content	**Data type**	**Cardinality**	**Response codes**	**Remarks**
	n/a		202 Accepted	The request was accepted for processing, but the processing has not yet been completed. The response message content shall be empty. The HTTP response shall include a "Location" HTTP header that contains the URI of the newly-created "application LCM operation occurrence" resource that corresponds to this application instance instantiation operation.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit the appropriate credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	409 Conflict	The operation cannot currently be executed due to a conflict with the state of the resource. Typically, this is because the application instance resource is not in NOT_INSTANTIATED state. The response message content shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.4.6.3.2 GET

Not supported.

7.4.6.3.3 PUT

Not supported.

7.4.6.3.4 DELETE

Not supported.

7.4.6.3.5 PATCH

Not supported.

7.4.7 Resource: terminate application instance task

7.4.7.1 Description

This resource represents the task of terminating an application instance. The client can use this resource to terminate an application instance.

7.4.7.2 Resource definition

The possible resource URIs are:

- Resource URI: {apiRoot}/app_lcm/v1/app_instances/{appInstanceId}/terminate

Resource URI variables for this resource are defined in table 7.4.7.2-1.

Table 7.4.7.2-1: Resource URI Variables for the resource

Name	Definition
apiRoot	See clause 7.2.
appInstanceId	The identifier of the application instance. 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 the new application instance resource. It can also be retrieved from the "id" attribute in the response message content.	

7.4.7.3 Resource methods

7.4.7.3.1 POST

The POST method is used to terminate an application instance.

This method shall comply with the URI request and response data structures, and response codes, as specified in tables 7.4.7.3.1-1 and 7.4.7.3.1-2.

Once the MEPM has successfully completed the underlying application instance LCM operation occurrence, it shall set the "instantiationState" attribute in the representation of the "individual application instance" resource to the value "NOT_INSTANTIATED".

Table 7.4.7.3.1-1: URI query parameters supported by the POST method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.4.7.3.1-2: Data structures supported by a POST request/response on this resource

Request message content	Data type	Cardinality	Remarks	
	TerminateAppRequest	1	Parameters for the termination, as defined in clause 6.2.2.9.	
Response message content	**Data type**	**Cardinality**	**Response codes**	**Remarks**
	n/a		202 Accepted	The request was accepted for processing, but the processing has not yet been completed. The response message content shall be empty. The HTTP response shall include a "Location" HTTP header that contains the URI of the newly-created "application LCM operation occurrence" resource that corresponds to this operation.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit the appropriate credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	409 Conflict	The operation cannot be executed currently, due to a conflict with the state of the resource. Typically, this is because the application instance resource is in NOT_INSTANTIATED state. The response message content shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.4.7.3.2 GET

Not supported.

7.4.7.3.3 PUT

Not supported.

7.4.7.3.4 DELETE

Not supported.

7.4.7.3.5 PATCH

Not supported.

7.4.8 Resource: operate application instance task

7.4.8.1 Description

This resource represents the task of changing the operational state of the application instance. The client can use this resource to start or stop an application instance.

7.4.8.2 Resource definition

The possible resource URIs are:

- Resource URI: {apiRoot}/app_lcm/v1/app_instances/{appInstanceId}/operate

Resource URI variables for this resource are defined in table 7.4.8.2-1.

Table 7.4.8.2-1: Resource URI Variables for the resource

Name	Definition
apiRoot	See clause 7.2.
appInstanceId	The identifier of the application instance. 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 the new application instance resource. It can also be retrieved from the "id" attribute in the response message content.	

7.4.8.3 Resource methods

7.4.8.3.1 POST

The POST method is used to change the operational state, i.e. start or stop, of the application instance.

This method shall comply with the URI request and response data structures, and response codes, as specified in tables 7.4.8.3.1-1 and 7.4.8.3.1-2.

Table 7.4.8.3.1-1: URI query parameters supported by the POST method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.4.8.3.1-2: Data structures supported by POST request/response on this resource

Request message content	**Data type**	**Cardinality**	**Remarks**	
	OperateAppRequest	1	The request message content contains the information necessary to change the operational state of application instance, as described in clause 6.2.2.8.	
Response message content	**Data type**	**Cardinality**	**Response codes**	**Remarks**
	n/a		202 Accepted	The request was accepted for processing, but it is possible that the processing is not yet completed. The response message content shall be empty. The HTTP response shall include a "Location" HTTP header that contains the URI of the newly-created "application LCM operation occurrence" resource that corresponds to this application instance instantiation operation.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit the appropriate credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	409 Conflict	The operation cannot be executed currently, due to a conflict with the state of the resource. Typically, this is because the application instance resource is not in INSTANTIATED state. The response message content shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.4.8.3.2 GET

Not supported.

7.4.8.3.3 PUT

Not supported.

7.4.8.3.4 DELETE

Not supported.

7.4.8.3.5 PATCH

Not supported.

7.4.9 Resource: application LCM operation occurrences

7.4.9.1 Description

This resource represents application instance LCM operation occurrences. The client can use this resource to query status information about multiple application instance lifecycle management operation occurrences.

7.4.9.2 Resource definition

The possible resource URIs are:

- Resource URI: {apiRoot}/app_lcm/v1/app_lcm_op_occs.

Resource URI variables for this resource are defined in table 7.4.9.2-1.

Table 7.4.9.2-1: Resource URI Variables for the resource

Name	Definition
apiRoot	See clause 7.2

7.4.9.3 Resource methods

7.4.9.3.1 POST

Not supported.

7.4.9.3.2 GET

The GET method retrieves information of operation status about multiple application instance lifecycle management operation occurrences.

This method shall comply with the URI query parameters, request and response data structures, and response codes, as specified in tables 7.4.9.3.2-1 and 7.4.9.3.2-2.

Table 7.4.9.3.2-1: URI query parameters of GET method on the resource

Name	Cardinality	Remarks
filter	0..1	Attribute-based filtering parameters according to ETSI GS MEC 009 [4]. The API producer shall support receiving filtering parameters as part of the URI query string. All attribute names that appear in the AppLcmOpOcc and in data types referenced from it shall be supported in attribute-based filtering parameters. See clause 6.19 in ETSI GS MEC 009 [4] for details.
all_fields	0..1	Include all complex attributes in the response. See clause 6.18 in ETSI GS MEC 009 [4] for details. The API producer shall support this parameter.
fields	0..1	Complex attributes of AppLcmOpOcc to be included into the response. See clause 6.18 in ETSI GS MEC 009 [4] for details. The API producer should support this parameter.
exclude_fields	0..1	Complex attributes of AppLcmOpOcc to be excluded from the response. See clause 6.18 in ETSI GS MEC 009 [4] for details. The API producer should support this parameter.
exclude_default	0..1	Indicates to exclude the following complex attributes of AppLcmOpOcc from the response. The following attributes shall be excluded from the AppLcmOpOcc structure in the response message content if this parameter is provided, or none of the parameters "all_fields", "fields", "exclude_fields", "exclude_default" are provided: • operationParams; • _links.

Table 7.4.9.3.2-2: Data structures supported by GET request/response on this resource

Request message content	Data type	Cardinality	Remarks	
	n/a			
Response message content	**Data type**	**Cardinality**	**Response codes**	**Remarks**
	AppLcmOpOcc	0..N	200 OK	Status information for zero or more application instance lifecycle management operation occurrences was queried successfully. The response message content shall contain in an array the status information about zero or more application lifecycle operation occurrences.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

	Data type	Cardinality	Response codes	Remarks
Response message content	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.4.9.3.3 PUT

Not supported.

7.4.9.3.4 DELETE

Not supported.

7.4.9.3.5 PATCH

Not supported.

7.4.10 Resource: individual application LCM operation occurrence

7.4.10.1 Description

This method represents an individual application LCM operation occurrence. The client can use this resource to read status information about an individual application lifecycle management operation occurrence.

7.4.10.2 Resource definition

The possible resource URIs are:

- Resource URI: {apiRoot}/app_lcm/v1/app_lcm_op_occs/{appLcmOpOccId}.

Resource URI Variables for this resource are defined in table 7.4.10.2-1.

Table 7.4.10.2-1: Resource URI Variables for the resource

Name	Definition
apiRoot	See clause 7.2.
appLcmOpOccId	Identifies an individual application LCM operation occurrence.
NOTE: This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request triggering an application instance LCM operation. It can also be retrieved from the "appLcmOpOccId" attribute in the AppLcmOpOccNotification.	

7.4.10.3 Resource methods

7.4.10.3.1 POST

Not supported.

7.4.10.3.2 GET

The GET method reads the status information of an individual application LCM operation occurrence, which is used by the procedure of "query application lifecycle operation status" as described in clause 6.3.1.6.

The method shall comply with the URI query parameters, request and response data structures, and response codes, as specified in the tables 7.4.10.3.2-1 and 7.4.10.3.2-2.

Table 7.4.10.3.2-1: URI query parameters of GET method on the resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.4.10.3.2-2: Data structures supported by GET request/response on this resource

Request message content	Data type	Cardinality	Remarks	
	n/a	0		
Response message content	**Data type**	**Cardinality**	**Response codes**	**Remarks**
	AppLcmOpOcc	1	200 OK	Information about an application LCM operation occurrence was read successfully. The response message content shall contain status information about an application lifecycle management operation occurrence.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.4.10.3.3 PUT

Not supported.

7.4.10.3.4 DELETE

Not supported.

7.4.10.3.5 PATCH

Not supported.

7.4.11 Resource: application LCM operation occurrence cancel task

7.4.11.1 Description

This resource represents the task of cancelling an ongoing application LCM operation. The client can use this resource to intervene in an ongoing application LCM operation.

7.4.11.2 Resource definition

The possible resource URIs are:

Resource URI: {apiRoot}/app_lcm/v1/app_lcm_op_occs/{appLcmOpOccId}/cancel.

Resource URI Variables for this resource are defined in table 7.4.11.2-1.

Table 7.4.11.2-1: Resource URI Variables for the resource

Name	Definition
apiRoot	See clause 7.2.
appLcmOpOccId	Identifies an individual application LCM operation occurrence.
NOTE: This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request triggering an application instance LCM operation. It can also be retrieved from the "appLcmOpOccId" attribute in the AppLcmOpOccNotification.	

7.4.11.3 Resource methods

7.4.11.3.1 POST

The POST method is used to cancel an ongoing application lifecycle operation whose related "Individual application LCM operation occurrence" resource is in "PROCESSING" state. This method shall comply with the URI request and response data structures, and response codes, as specified in tables 7.4.11.3.1-1 and 7.4.11.3.1-2.

Before returning the "202 Accepted" response, the "isCancelPending" and "cancelMode" attributes in the representation of the parent resource shall be updated according to the provisions in clause 5.4.5.

In case of success of processing the asynchronous request, the "operationState" attribute in the representation of the parent resource shall be changed to "FAILED_TEMP". In addition, the "isCancelPending" and "cancelMode" attributes in the representation of the parent resource according to the provisions in clause 5.4.5 to reflect the new status, and the applicable "result" notification shall be emitted to indicate that the execution of the underlying application 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 7.4.11.3.1-1: URI query parameters supported by the POST method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.4.11.3.1-2: Data structures supported by POST request/response on this resource

	Data type	Cardinality	Response codes	Remarks
Request message content	CancelMode	1		The request message content contains the information on the cancellation mode to apply to the ongoing application LCM operation, as described in clause 5.4.5.
Response message content	n/a		202 Accepted	The request was accepted for processing, but it is possible that the processing is not yet completed. The response message content shall be empty.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit the appropriate credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	409 Conflict	The operation cannot be executed currently, due to a conflict with the state of the resource. The response message content shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.4.11.3.2 GET

Not supported.

7.4.11.3.3 PUT

Not supported.

7.4.11.3.4 DELETE

Not supported.

7.4.11.3.5 PATCH

Not supported.

7.4.12 Resource: application LCM operation occurrence fail task

7.4.12.1 Description

This resource represents the task of failing an application LCM operation. The client can use this resource to mark an application LCM operation occurrence as "finally failed", i.e. change the state of the related application LCM operation occurrence resource to "FAILED", if it is not assumed that a subsequent retry will succeed. Once the operation is marked as "finally failed", it cannot be retried anymore.

7.4.12.2 Resource definition

The possible resource URIs are:

Resource URI: {apiRoot}/app_lcm/v1/app_lcm_op_occs/{appLcmOpOccId}/fail.

Resource URI Variables for this resource are defined in table 7.4.12.2-1.

Table 7.4.12.2-1: Resource URI Variables for the resource

Name	Definition
apiRoot	See clause 7.2.
appLcmOpOccId	Identifies an individual application LCM operation occurrence.
NOTE: This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request triggering an application instance LCM operation. It can also be retrieved from the "appLcmOpOccId" attribute in the AppLcmOpOccNotification.	

7.4.12.3 Resource methods

7.4.12.3.1 POST

The POST method marks an application lifecycle management operation occurrence as "finally failed" if that operation occurrence is in "FAILED_TEMP" state. This method shall comply with the URI request and response data structures, and response codes, as specified in tables 7.4.12.3.1-1 and 7.4.12.3.1-2.

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 5.4.5 shall be emitted to indicate that the execution of the underlying application LCM operation occurrence has finally and un-recoverably failed.

Table 7.4.12.3.1-1: URI query parameters supported by the POST method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.4.12.3.1-2: Data structures supported by POST request/response on this resource

Request message content	Data type	Cardinality	Remarks	
	n/a		The POST request message content to this resource shall be empty.	
Response message content	**Data type**	**Cardinality**	**Response codes**	**Remarks**
	AppLcmOpOcc		200 OK	Shall be returned when the state of the application lifecycle management operation occurrence has been changed successfully. The response shall include a representation of the "Individual application lifecycle management operation occurrence" resource.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit the appropriate credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	409 Conflict	The operation cannot be executed currently, due to a conflict with the state of the resource. The response message content shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.4.12.3.2 GET

Not supported.

7.4.12.3.3 PUT

Not supported.

7.4.12.3.4 DELETE

Not supported.

7.4.12.3.5 PATCH

Not supported.

7.4.13 Resource: application LCM operation occurrence retry task

7.4.13.1 Description

This resource represents the task of retrying an application LCM operation. The client can use this resource to retry in an application LCM operation that is in a transient failure state.

7.4.13.2 Resource definition

The possible resource URIs are:

Resource URI: {apiRoot}/app_lcm/v1/app_lcm_op_occs/{appLcmOpOccId}/retry.

Resource URI Variables for this resource are defined in table 7.4.13.2-1.

Table 7.4.13.2-1: Resource URI Variables for the resource

Name	Definition
apiRoot	See clause 7.2.
appLcmOpOccId	Identifies an individual application LCM operation occurrence.
NOTE: This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request triggering an application instance LCM operation. It can also be retrieved from the "appLcmOpOccId" attribute in the AppLcmOpOccNotification.	

7.4.13.3 Resource methods

7.4.13.3.1 POST

The POST method is used to initiate retrying an application lifecycle operation that has experience a temporary failure, i.e. the related "application LCM operation occurrence" is in "FAILED_TEMP" state. This method shall comply with the URI request and response data structures, and response codes, as specified in tables 7.4.13.3.1-1 and 7.4.13.3.1-2.

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 5.4.5 shall be emitted to indicate that the underlying application LCM operation occurrence proceeds.

Table 7.4.13.3.1-1: URI query parameters supported by the POST method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.4.13.3.1-2: Data structures supported by POST request/response on this resource

Request message content	**Data type**	**Cardinality**	**Remarks**	
	n/a		The POST request message content to this resource shall be empty.	
Response message content	**Data type**	**Cardinality**	**Response codes**	**Remarks**
	n/a		202 Accepted	The request was accepted for processing, but it is possible that the processing is not yet completed. The response message content shall be empty.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit the appropriate credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	409 Conflict	The operation cannot be executed currently, due to a conflict with the state of the resource. The response message content shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.4.13.3.2 GET

Not supported.

7.4.13.3.3 PUT

Not supported.

7.4.13.3.4 DELETE

Not supported.

7.4.13.3.5 PATCH

Not supported.

7.5 Resources of granting on Mm3

7.5.1 Resource: grants

7.5.1.1 Description

This resource represents grants. The client can use this resource to obtain permission from the MEO to perform a particular application lifecycle operation.

7.5.1.2 Resource definition

The possible resource URIs are:

- Resource URI: {apiRoot}/granting/v1/grants.

Resource URI variables for this resource are defined in table 7.5.1.2-1.

Table 7.5.1.2-1: Resource URI Variables for the resource

Name	Definition
apiRoot	See clause 7.2.

7.5.1.3 Resource methods

7.5.1.3.1 POST

The POST method requests a grant for a particular application lifecycle operation.

This method shall comply with the URI query parameters, request and response data structures, and response codes, as specified in tables 7.5.1.3.1-1 and 7.5.1.3.1-2.

As the result of successfully processing this request, a new "individual grant" resource shall be created. In the synchronous case which is indicated by responding with "201 Created", that resource shall be created before the 200 OK response is returned. In the asynchronous case which is indicated by responding with "202 Accepted", this resource may be created after the response is returned.

Table 7.5.1.3.1-1: URI query parameters of POST method on the resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.5.1.3.1-2: Data structures supported by POST request/response on this resource

Request message content	Data type	Cardinality	Remarks	
	GrantRequest	1	The POST method is to request a grand for an application lifecycle operation.	
Response message content	**Data type**	**Cardinality**	**Response codes**	**Remarks**
	Grant	1	201 Created	Shall be returned when the grant was created successfully (synchronous mode). A representation of the created "Individual grant" resource shall be returned in the response message content. The HTTP response shall include a "Location" HTTP header that indicates the URI of the "Individual grant" resource just created.
	n/a		202 Accepted	Shall be returned when the request was accepted for processing, and it is expected to take some time to create the grant (asynchronous mode). The HTTP response shall include a "Location" HTTP header that indicates the URI of the "Individual grant" resource just created.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.5.1.3.2 GET

This method is not supported.

7.5.1.3.3 PUT

Not supported.

7.5.1.3.4 DELETE

Not supported.

7.5.1.3.5 PATCH

Not supported.

7.5.2 Resource: individual grant

7.5.2.1 Description

This resource represents an individual grant. The client can use this resource to read the grant.

It is determined by means outside the scope of the present document, such as configuration or policy, how long an individual grant is available.

7.5.2.2 Resource definition

The possible resource URIs are:

- Resource URI: {apiRoot}/granting/v1/grants/{grantId}.

Resource URI Variables for this resource are defined in table 7.5.2.2-1.

Table 7.5.2.2-1: Resource URI Variables for the resource

Name	Definition
apiRoot	See clause 7.2.
grantId	Identifier of the individual grant.
NOTE: This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request granting a new application lifecycle operation. It can also be retrieved from the "id" attribute in the response message content.	

7.5.2.3 Resource methods

7.5.2.3.1 POST

Not supported.

7.5.2.3.2 GET

The GET method reads a grant.

This method shall comply with the URI query parameters, request and response data structures, and response codes, as specified in the tables 7.5.2.3.2-1 and 7.5.2.3.2-2.

Table 7.5.2.3.2-1: URI query parameters of GET method on the resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.5.2.3.2-2: Data structures supported by GET request/response on this resource

Request message content	Data type	Cardinality	Remarks	
	n/a	0		
Response message content	**Data type**	**Cardinality**	**Response codes**	**Remarks**
	Grant	1	200 OK	Shall be returned when the grant was read successfully. A representation of the "individual grant" resource shall be returned in the response message content.
	n/a		202 Accepted	Shall be returned when the process of creating the grant is ongoing, no grant is available yet. The response message content shall be empty. The HTTP response shall include a "Location" HTTP header that contains the URI of the newly-created resource.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.5.2.3.3 PUT

Not supported.

7.5.2.3.4 DELETE

Not supported.

7.5.2.3.5 PATCH

Not supported.

7.6 Resources of MEPM's application lifecycle management on Mm3

Refer to clause 7.4.

7.7 Resources of MEPM-V's application lifecycle management on Mm3*

7.7.1 Resource: application instances

Refer to clause 7.4.1.

7.7.2 Resource: individual application instance

Refer to clause 7.4.2.

7.7.3 Resource: subscriptions

Refer to clause 7.4.3.

7.7.4 Resource: individual subscription

Refer to clause 7.4.4.

7.7.5 Resource: notification endpoint

Refer to clause 7.4.5.

7.7.6 Resource: configure_platform_for_app task

7.7.6.1 Description

This resource represents the task of providing configuration information in AppD to the MEPM-V, intended to configure the MEP to run an application instance which is instantiated from the AppD. The configuration information includes the traffic rules, DNS rules, the required and optional services, and services produced by the application instance, etc. The client can use this resource to provide to the MEPM-V configuration information for the MEP to run an application instance after the corresponding VNF instance has been instantiated by NFV-MANO.

7.7.6.2 Resource definition

The possible resource URIs are:

Resource URI: {apiRoot}/app_lcm/v1/app_instances/{appInstanceId}/configure_platform_for_app

Resource URI variables for this resource are defined in table 7.7.6.2-1.

Table 7.7.6.2-1: Resource URI Variables for the resource

Name	Definition
apiRoot	See clause 7.2.
appInstanceId	The identifier of the application instance. 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 the new application instance resource. It can also be retrieved from the "id" attribute in the response message content.	

7.7.6.3 Resource methods

7.7.6.3.1 POST

The POST method is to provide configuration information in AppD to the MEPM-V, intended to configure the MEP to run the application instance.

This method shall comply with the URI request and response data structures, and response codes, as specified in tables 7.7.6.3.1-1 and 7.7.6.3.1-2.

Table 7.7.6.3.1-1: URI query parameters supported by the POST method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.7.6.3.1-2: Data structures supported by a POST request/response on this resource

Request message content	**Data type**	**Cardinality**	**Remarks**	
	ConfigPlatformForAppRequest	1	The request message content contains the information necessary to provide configuration information in AppD as described in clause 6.2.2.21.	
Response message content	**Data type**	**Cardinality**	**Response codes**	**Remarks**
	n/a		202 Accepted	The request has been accepted for processing, but the processing has not yet been completed. The response message content shall be empty. The HTTP response shall include a "Location" HTTP header that contains the URI of the newly-created "application LCM operation occurrence" resource that corresponds to this application instance configure operation.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit the appropriate credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. Typically, this is because the application instance resource is in NOT_INSTANTIATED state. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.7.6.3.2 GET

Not supported.

7.7.6.3.3 PUT

Not supported.

7.7.6.3.4 DELETE

Not supported.

7.7.6.3.5 PATCH

Not supported.

7.7.7 Resource: terminate application instance task

Refer to clause 7.4.7.

7.7.8 Resource: operate application instance task

Refer to clause 7.4.8.

7.7.9 Resource: application LCM operation occurrences

Refer to clause 7.4.9.

7.7.10 Resource: individual application LCM operation occurrence

Refer to clause 7.4.10.

7.7.11 Resource: application LCM operation occurrence cancel task

Refer to clause 7.4.11.

7.7.12 Resource: application LCM operation occurrence fail task

Refer to clause 7.4.12.

7.7.13 Resource: application LCM operation occurrence retry task

Refer to clause 7.4.13.

Annex A (informative): Application package state model

A.1 Introduction

This annex presents the state model of application package. The steps before onboarding an application package is out of scope of this model.

A.2 State model

This clause describes the state model of application package in the MEO. It consists of the onboarding phase and onboarded phase.

Figure A.2-1 shows the state model of application package.

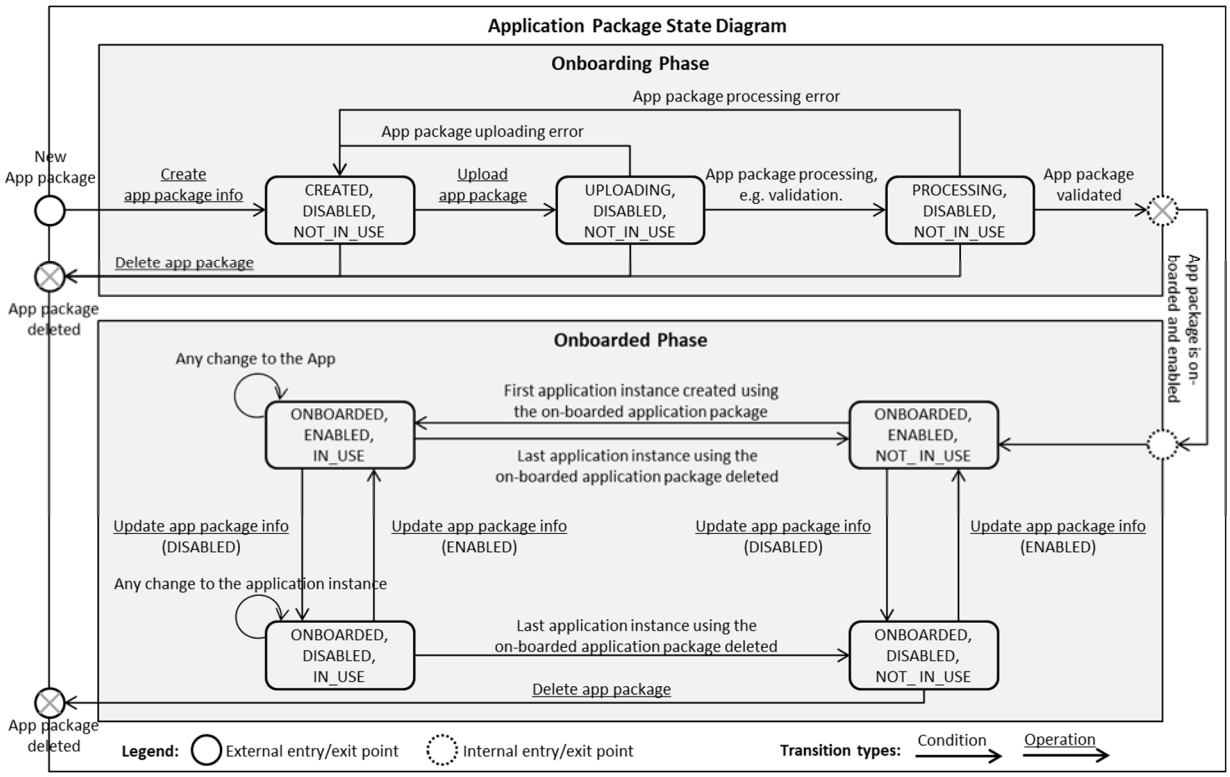

Figure A.2-1: Application package state model

The onboarding state, operational state and usage state are represented by the "onboardingState", "operationalState" and "usageState" attributes respectively in the "AppPkgInfo" information element specified in table 6.2.3.3.2-1.

Annex B (informative): Bibliography

- ETSI GS MEC 003: "Multi-access Edge Computing (MEC); Framework and Reference Architecture".

History

Document history		
V1.1.1	July 2017	Publication
V2.1.1	November 2019	Publication
V2.2.1	February 2022	Publication
V3.1.1	June 2023	Publication