ETSI TS 126 510 V18.0.2 (2024-08)

5G;
Media delivery;
interactions and APIs for provisioning
and media session handling
(3GPP TS 26.510 version 18.0.2 Release 18)

Reference
RTS/TSGS-0426510vi02

Keywords
5G

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

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.

Legal Notice

This Technical Specification (TS) has been produced by ETSI 3rd Generation Partnership Project (3GPP).

The present document may refer to technical specifications or reports using their 3GPP identities. These shall be interpreted as being references to the corresponding ETSI deliverables.

The cross reference between 3GPP and ETSI identities can be found under https://webapp.etsi.org/key/queryform.asp.

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.

Contents

Intellectual Property Rights 2
Legal Notice 2
Modal verbs terminology 2
Foreword 10
1 Scope 12
2 References 12
3 Definitions of terms, symbols and abbreviations 14
3.1 Terms 14
3.2 Symbols 14
3.3 Abbreviations 14
4 Functions and roles 15
4.1 Media Application Provider 15
4.2 Media AF 15
4.3 Media Session Handler 15
5 Interactions 16
5.1 Summary 16
5.2 Provisioning (M1) interactions 18
5.2.1 Overview 18
5.2.2 Provisioning Session provisioning 19
5.2.2.1 General 19
5.2.2.2 Enumerate Provisioning Sessions collection operation 19
5.2.2.3 Create Provisioning Session resource operation 20
5.2.2.4 Retrieve Provisioning Session resource operation 20
5.2.2.5 Update Provisioning Session resource operation 20
5.2.2.6 Destroy Provisioning Session resource operation 20
5.2.3 Content protocols discovery 20
5.2.3.1 General 20
5.2.3.2 Create Content Protocols resource operation 21
5.2.3.3 Retrieve Content Protocols resource operation 21
5.2.3.4 Update Content Protocols resource operation 21
5.2.3.5 Destroy Content Protocols resource operation 21
5.2.4 Server Certificate provisioning 21
5.2.4.1 General 21
5.2.4.2 Create Server Certificate resource operation 22
5.2.4.3 Reserve Server Certificate resource operation 22
5.2.4.4 Upload Server Certificate resource operation 23
5.2.4.5 Retrieve Server Certificate resource operation 23
5.2.4.6 Update Server Certificate resource operation 23
5.2.4.7 Destroy Server Certificate resource operation 24
5.2.5 Content Preparation provisioning 24
5.2.5.1 General 24
5.2.5.2 Create Content Preparation Template resource operation 24
5.2.5.3 Retrieve Content Preparation Template resource operation 25
5.2.5.4 Update Content Preparation Template resource operation 25
5.2.5.5 Destroy Content Preparation Template resource operation 25
5.2.6 Edge Resources provisioning 26
5.2.6.1 General 26
5.2.6.2 Create Edge Resources Configuration resource operation 26
5.3.6.3 Retrieve Edge Resources Configuration resource operation 26
5.2.6.4 Update Edge Resources Configuration resource operation 26
5.2.6.5 Destroy Edge Resources Configuration resource operation 27
5.2.7 Dynamic Policy provisioning 27
5.2.7.1 General 27

5.2.7.2 Policy Template life-cycle 28
5.2.7.3 Create Policy Template resource operation 30
5.2.7.4 Retrieve Policy Template resource operation 30
5.2.7.5 Update Policy Template resource operation 30
5.2.7.6 Destroy Policy Template resource operation 31
5.2.8 Content Hosting provisioning 31
5.2.8.1 General 31
5.2.8.2 Create Content Hosting Configuration resource operation 31
5.2.8.3 Retrieve Content Hosting Configuration resource operation 32
5.2.8.4 Update Content Hosting Configuration resource operation 32
5.2.8.5 Destroy Content Hosting Configuration resource operation 33
5.2.8.6 Purge Content Hosting cache operation 33
5.2.9 Content Publishing provisioning 33
5.2.9.1 General 33
5.2.9.2 Create Content Publishing Configuration resource operation 34
5.2.9.3 Retrieve Content Publishing Configuration resource operation 35
5.2.9.4 Update Content Publishing Configuration resource operation 35
5.2.9.5 Destroy Content Publishing Configuration resource operation 35
5.2.9.6 Purge Content Publishing cache operation 36
5.2.10 Real-time Media Communication provisioning 36
5.2.10.1 General 36
5.2.10.2 Create Real-time Media Communication Configuration resource operation 36
5.2.10.3 Retrieve Real-time Media Communication Configuration resource operation 37
5.2.10.4 Update Real-time Media Communication Configuration resource operation 37
5.2.10.5 Destroy Real-time Media Communication Configuration resource operation 37
5.2.11 Metrics Reporting provisioning 37
5.2.11.1 General 37
5.2.11.2 Create Metrics Reporting Configuration resource operation 38
5.2.11.3 Retrieve Metrics Reporting Configuration resource operation 38
5.2.11.4 Update Metrics Reporting Configuration resource operation 38
5.2.11.5 Destroy Metrics Reporting Configuration resource operation 39
5.2.12 Consumption Reporting provisioning 39
5.2.12.1 General 39
5.2.12.2 Create Consumption Reporting Configuration resource operation 39
5.2.12.3 Retrieve Consumption Reporting Configuration resource operation 40
5.2.12.4 Update Consumption Reporting Configuration resource operation 40
5.2.12.5 Destroy Consumption Reporting Configuration resource operation 40
5.2.13 Event Data Processing provisioning 40
5.2.13.1 General 40
5.2.13.2 Create Event Data Processing Configuration resource operation 41
5.2.13.3 Retrieve Event Data Processing Configuration resource operation 41
5.2.13.4 Update Event Data Processing Configuration resource operation 41
5.2.13.5 Destroy Event Data Processing Configuration resource operation 42
5.3 Network media session handling (M3, M5) interactions 42
5.3.1 Overview 42
5.3.2 Service Access Information acquisition 42
5.3.2.1 General 42
5.3.2.2 Create Service Access Information resource operation 43
5.3.2.3 Retrieve Service Access Information resource operation 43
5.3.2.4 Update Service Access Information resource operation 43
5.3.2.5 Destroy Service Access Information resource operation 44
5.3.3 Dynamic Policy invocation 44
5.3.3.1 Procedures 44
5.3.3.2 Create Dynamic Policy Instance resource operation 44
5.3.3.3 Retrieve Dynamic Policy Instance resource operation 47
5.3.3.4 Update Dynamic Policy Instance resource operation 47
5.3.3.5 Destroy Dynamic Policy Instance resource operation 47
5.3.4 Network Assistance invocation 48
5.3.4.1 Procedures 48
5.3.4.2 Create Network Assistance Session resource operation 48
5.3.4.3 Retrieve Network Assistance Session resource operation 50
5.3.4.4 Bit rate recommendation request operation 50

5.3.4.5 Delivery boost request operation....51
5.3.4.6 Update Network Assistance Session resource operation....51
5.3.4.7 Destroy Network Assistance Session resource operation....51
5.3.5 Metrics reporting....52
5.3.5.1 Procedures....52
5.3.5.2 Submit metrics report operation....52
5.3.6 Consumption reporting....53
5.3.6.1 Procedures....53
5.3.6.2 Submit consumption report operation....54
5.4 UE media session handling (M6, M11) interactions....55
5.4.1 Overview....55
5.4.2 Media delivery session life-cycle....55
5.4.2.1 Explicit media session handling initiation/termination....55
5.4.2.2 Implicit media session handling initiation/termination....55
5.4.3 Dynamic Policy invocation....56
5.4.4 Network Assistance invocation....56
5.4.5 Metrics reporting....57
5.4.6 Consumption reporting....57
5.5 5GC policy control (N5/N33) interactions....57
5.5.1 Overview....57
5.5.2 Policy control interactions for Policy Template provisioning....57
5.5.3 Policy control interactions for Dynamic Policies....58
5.5.4 Policy control interactions for AF-based Network Assistance....59
5.6 UE modem interactions....61
5.6.1 Overview....61
5.6.2 ANBR-based Network Assistance....61
5.6.3 RAN-based metrics reporting....61

6 3GPP Service URL....62
6.1 General....62
6.2 3GPP Service URL syntax....62
6.3 Handling of 3GPP Service URLs by Media Client....63

7 General aspects of network APIs....63
7.1 Usage of HTTP....63
7.1.1 HTTP protocol version....63
7.1.2 HTTP endpoint addresses....63
7.1.2.1 Default Media AF endpoint address at reference point M1....63
7.1.2.2 Default Media AF endpoint address at reference point M3....63
7.1.2.3 Default Media AF endpoint address at reference point M5....64
7.1.3 HTTP resource URIs and paths....64
7.1.4 Usage of HTTP headers....64
7.1.4.1 General....64
7.1.4.2 User Agent identification....64
7.1.4.3 Server identification....64
7.1.4.2 Cache control....64
7.1.4.3 Support for conditional HTTP GET requests....64
7.1.4.4 Support for conditional HTTP POST, PUT, PATCH and DELETE requests....65
7.1.5 HTTP message bodies for API resources....65
7.1.6 HTTP response codes....65
7.1.7 HTTP error response message bodies....65
7.2 Explanation of API data model notation....65
7.3 Common OpenAPI data types....67
7.3.1 General....67
7.3.2 Simple data types....67
7.3.3 Structured data types....68
7.3.3.1 IpPacketFilterSet type....68
7.3.3.2 ApplicationFlowDescription type....68
7.3.3.3 M1UnidirectionalQoSSpecification type....69
7.3.3.4 M1QoSSpecification type....69
7.3.3.5 M5BitRateSpecification type....69
7.3.3.6 M5QoSSpecification type....70

7.3.3.7 ChargingSpecification type 70
7.3.3.8 TypedLocation type 70
7.3.3.9 OperationSuccessResponse type 70
7.3.3.10 EdgeProcessingEligibilityCriteria type 71
7.3.3.11 EndpointAddress type 71
7.3.3.12 M1MediaEntryPoint type 72
7.3.3.13 CachingConfiguration type 72
7.3.3.14 BDTWindow type 73
7.3.4 Enumerated data types 73
7.3.4.1 CellIdentifierType enumeration 73
7.3.4.2 SdfMethod enumeration 73
7.3.4.3 ProvisioningSessionType enumeration 73
7.3.4.4 EASRelocationTolerance enumeration 74
7.3.4.5 ContentTransferMode enumeration 74
7.4 Security 74
7.4.1 General 74
7.4.2 Authorising Media Application Provider access to the Media AF at reference point M1 74
7.4.3 Authorising Media Session Handler access to the Media AF at reference point M5 75
8 Maf_Provisioning service 76
8.1 Overview 76
8.2 Provisioning Sessions API 78
8.2.1 Overview 78
8.2.2 Resource structure 78
8.2.3 Data model 78
8.2.3.1 ProvisioningSession resource 78
8.3 Content Protocols Discovery API 81
8.3.1 Overview 81
8.3.2 Resource structure 81
8.3.3 Data model 82
8.3.3.1 ContentProtocols resource 82
8.3.3.2 ContentProtocolDescriptor type 82
8.4 Server Certificates provisioning API 83
8.4.1 Overview 83
8.4.2 Resource structure 83
8.4.3 Data model 84
8.4.3.1 Certificate Signing Request 84
8.4.3.2 Server Certificate resource 84
8.5 Content Preparation Templates provisioning API 85
8.5.1 Overview 85
8.5.2 Resource structure 85
8.5.3 Data model 85
8.6 Edge Resources provisioning API 86
8.6.1 General 86
8.6.2 Resource structure 86
8.6.3 Data model 87
8.6.3.1 EdgeResourcesConfiguration resource type 87
8.6.3.2 EdgeManagementMode enumeration 87
8.6.3.3 EASRequirements type 88
8.6.3.4 M1EASRelocationRequirements type 88
8.7 Policy Templates provisioning API 89
8.7.1 Overview 89
8.7.2 Resource structure 89
8.7.3 Data model 89
8.7.3.1 PolicyTemplate resource 89
8.7.3.2 M1BDTSpecification type 91
8.8 Content Hosting provisioning API 92
8.8.1 Overview 92
8.8.2 Resource structure 92
8.8.3 Data model 93
8.8.3.1 ContentHostingConfiguration resource 93
8.8.3.2 DistributionNetworkType enumeration 96

8.8.3.3 DistributionMode enumeration 96
8.9 Content Publishing provisioning API 97
8.9.1 Overview 97
8.9.2 Resource structure 97
8.9.3 Data model 98
8.9.3.1 ContentPublishingConfiguration resource 98
8.10 Real-time Media Communication provisioning API 101
8.10.1 Overview 101
8.10.2 Resource structure 101
8.10.3 Data model 102
8.10.3.1 RTCConfiguration resource 102
8.10.3.2 M1EndpointAccess 103
8.11 Metrics Reporting provisioning API 104
8.11.1 Overview 104
8.11.2 Resource structure 104
8.11.3 Data model 105
8.11.3.1 MetricsReportingConfiguration resource 105
8.12 Consumption Reporting provisioning API 107
8.12.1 Overview 107
8.12.2 Resource structure 107
8.12.3 Data model 108
8.12.3.1 ConsumptionReportingConfiguration resource 108
8.13 Event Data Processing provisioning API 109
8.13.1 General 109
8.13.2 Resource structure 109
8.13.3 Data model 110
8.13.3.1 EventDataProcessingConfiguration resource type 110

9 Maf_SessionHandling service 111
9.1 Overview 111
9.2 Service Access Information API 112
9.2.1 General 112
9.2.2 Resource structure 112
9.2.3 Data model 113
9.2.3.1 ServiceAccessInformation resource type 113
9.2.3.2 M5EndpointAccess 117
9.2.3.3 EASDiscoveryTemplate type 118
9.2.3.4 M5EASRelocationRequirements type 118
9.3 Dynamic Policy API 119
9.3.1 Overview 119
9.3.2 Resource structure 119
9.3.3 Data model 120
9.3.3.1 DynamicPolicy resource 120
9.3.3.2 ApplicationFlowBinding 121
9.3.3.3 M5BDTSpecification type 121
9.4 Network Assistance API 122
9.4.1 Overview 122
9.4.2 Resource structure 122
9.4.3 Data model 123
9.4.3.1 NetworkAssistanceSession resource 123
9.5 Metrics Reporting API 124
9.5.1 General 124
9.5.2 Reporting procedure 124
9.5.3 Report format 124
9.6 Consumption Reporting API 125
9.6.1 General 125
9.6.2 Reporting procedure 125
9.6.3 Report format 126
9.6.3.1 ConsumptionReport type 126
9.6.3.2 ConsumptionReportingUnit type 127

10 Ancillary network media session handling services 128

10.1 Overview 128
10.2 Resource update notification channel 128
10.2.1 General 128
10.2.2 Topic structure of notification channel 128
10.2.3 Notification message format 128

11 UE media session handling APIs 130
11.1 Introduction 130
11.2 Media Session Handler client API 130
11.2.1 Media Session Handler internal properties 130
11.2.2 General Media Session Handler methods 131
11.2.2.1 Create a media delivery session 131
11.2.2.2 Destroy a media delivery session 132
11.2.3 General Media Session Handler information 132
11.3 Dynamic Policy client API 134
11.3.1 Dynamic Policy methods 134
11.3.1.1 Retrieve Background Data Transfer information 134
11.3.1.2 Activate Dynamic Policy 135
11.3.2 Dynamic Policy information 136
11.4 Network Assistance client API 137
11.4.1 Network Assistance methods 137
11.4.1.1 Bit rate recommendation request 137
11.4.1.2 Delivery boost request 137
11.4.2 Network Assistance information 137
11.5 Consumption Reporting API 139
11.5.1 Consumption Reporting methods 139
11.5.2 Consumption Reporting information 139
11.6 Metrics Reporting client API 140
11.6.1 Consumption Reporting methods 140
11.6.2 Metrics Reporting information 140

Annex A (normative): OpenAPI representation of HTTP REST APIs 142

A.1 General 142

A.2 Data Types applicable to several APIs 142

A.3 OpenAPI representation of Maf_Provisioning APIs 142
A.3.1 Maf_Provisioning_ProvisioningSessions API 142
A.3.2 Maf_Provisioning_ContentProtocols API 142
A.3.3 Maf_Provisioning_ServerCertificates API 142
A.3.4 Maf_Provisioning_ContentPreparationTemplates API 142
A.3.5 Maf_Provisioning_EdgeResources API 142
A.3.6 Maf_Provisioning_PolicyTemplates API 143
A.3.7 Maf_Provisioning_ContentHosting API 143
A.3.8 Maf_Provisioning_ContentPublishing API 143
A.3.9 Maf_Provisioning_RealTimeCommunication API 143
A.3.9 Maf_Provisioning_MetricsReporting API 143
A.3.10 Maf_Provisioning_ConsumptionReporting API 143
A.3.11 Maf_Provisioning_EventDataProcessing API 143

A.4 OpenAPI representation of Maf_SessionHandling APIs 143
A.4.1 Maf_SessionHandling_ServiceAccessInformation API 143
A.4.2 Maf_SessionHandling_DynamicPolicy API 143
A.4.3 Maf_SessionHandling_NetworkAssistance API 144
A.4.4 Maf_SessionHandling_MetricsReporting API 144
A.4.5 Maf_SessionHandling_ConsumptionReporting API 144

Annex B (normative): Controlled vocabularies 145

B.1 Media Delivery locator type 145

Annex C (informative): Usage of TOS/DSCP for traffic identification 146

C.1 General 146

C.2 Differentiated Services/TOS-enabled Collaboration Scenarios 146
C.3 Procedure for using TOS Traffic Class for traffic identification.....147
Annex D (informative): Change history 150
History 152

Foreword

This Technical Specification has been produced by the 3rd Generation Partnership Project (3GPP).

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

- x the first digit:
  - 1 presented to TSG for information;
  - 2 presented to TSG for approval;
  - 3 or greater indicates TSG approved document under change control.
- y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.
- z the third digit is incremented when editorial only changes have been incorporated in the document.

In the present document, modal verbs have the following meanings:

shall indicates a mandatory requirement to do something

shall not indicates an interdiction (prohibition) to do something

The constructions "shall" and "shall not" are confined to the context of normative provisions, and do not appear in Technical Reports.

The constructions "must" and "must not" are not used as substitutes for "shall" and "shall not". Their use is avoided insofar as possible, and they are not used in a normative context except in a direct citation from an external, referenced, non-3GPP document, or so as to maintain continuity of style when extending or modifying the provisions of such a referenced document.

should indicates a recommendation to do something

should not indicates a recommendation not to do something

may indicates permission to do something

need not indicates permission not to do something

The construction "may not" is ambiguous and is not used in normative elements. The unambiguous constructions "might not" or "shall not" are used instead, depending upon the meaning intended.

can indicates that something is possible

cannot indicates that something is impossible

The constructions "can" and "cannot" are not substitutes for "may" and "need not".

will indicates that something is certain or expected to happen as a result of action taken by an agency the behaviour of which is outside the scope of the present document

will not indicates that something is certain or expected not to happen as a result of action taken by an agency the behaviour of which is outside the scope of the present document

might indicates a likelihood that something will happen as a result of action taken by some agency the behaviour of which is outside the scope of the present document

might not indicates a likelihood that something will not happen as a result of action taken by some agency the behaviour of which is outside the scope of the present document

In addition:

is (or any other verb in the indicative mood) indicates a statement of fact

is not (or any other negative verb in the indicative mood) indicates a statement of fact

The constructions "is" and "is not" do not indicate requirements.

1 Scope

The present document specifies:

1. The operations and corresponding APIs used by a Media Application Provider to interact with a Media AF component to provision the delivery of media in a Media Delivery System.
2. The operations and corresponding APIs used by a Media Session Handler component to interact with a provisioned Media AF for the purpose of managing media delivery sessions ("media session handling").
3. The operations and corresponding APIs used by a Media-aware Application running in a UE or by a Media Access Client component to interact with a Media Session Handler for the purpose of initiating a media delivery session and for controlling it.

The Media Delivery System may be:

- a 5G Media Streaming (5GMS) System, as specified in TS 26.512 [6] and/or
- a Real-Time media Communication (RTC) System, as specified in TS 26.113 [7].

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[1] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".

[2] 3GPP TS 23.501: "System architecture for the 5G System (5GS)".

[3] 3GPP TS 23.502: "Procedures for the 5G System (5GS); Stage 2".

[4] 3GPP TS 26.501: "5G Media Streaming (5GMS); General description and architecture".

[5] 3GPP TS 26.506: "5G Real-time Media Communication Architecture (Stage 2)".

[6] 3GPP TS 26.512: "5G Media Streaming (5GMS); Protocols".

[7] 3GPP TS 23.113: "Real-Time Media Communication; Protocols and APIs".

[8] 3GPP TS 26.247: "Transparent end-to-end Packet-switched Streaming Service (PSS); Progressive Download and Dynamic Adaptive Streaming over HTTP (3GP-DASH)".

[9] 3GPP TS 26.118: "Virtual Reality (VR) profiles for streaming applications".

[10] ITU-T Recommendation X.509 (2005) | ISO/IEC 9594-8:2005: "Information Technology – Open Systems Interconnection – The Directory: Public-key and attribute certificate frameworks".

[11] IETF RFC 5280: "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", May 2008.

[12] IETF RFC 7468: "Textual Encodings of PKIX, PKCS, and CMS Structures", April 2015.

[13] 3GPP TS 23.558: "Architecture for enabling edge applications".

[14] 3GPP TS 24.558: "Enabling Edge Applications; Protocol specification".

[15] 3GPP TS 29.558: "Enabling Edge Applications; Application Programming Interface (API) specification; Stage 3".

[16] 3GPP TS 23.003: "Numbering, addressing and identification".

[17] 3GPP TS 23.503: "Policy and charging control framework for the 5G System (5GS); Stage 2".

[18] 3GPP TS 29.514: "5G System; Policy Authorization Service; Stage 3".

[19] 3GPP TS 29.522: "5G System. Network Exposure Function Northbound APIs; Stage 3".

[20] 3GPP TS 29.122: "T8 reference point for Northbound APIs".

[21] 3GPP TS 27.007: "AT Command set for User Equipment (UE)".

[22] 3GPP TS 38.321: "NR; Medium Access Control (MAC) protocol specification".

[23] 3GPP TS 36.321: "Evolved Universal Terrestrial Radio Access (E-UTRA); Medium Access Control (MAC) protocol specification".

[24] IETF RFC 9110: "HTTP Semantics", June 2022.

[25] IETF RFC 9111: "HTTP Caching", June 2022.

[26] IETF RFC 9112: "HTTP/1.1", June 2022.

[27] IETF RFC 9113: "HTTP/2", June 2022.

[28] Reserved for future use.

[29] IETF RFC 8446: "The Transport Layer Security (TLS) Protocol Version 1.3", August 2018.

[30] 3GPP TS 29.500: "5G System; Technical Realization of Service Based Architecture; Stage 3".

[31] 3GPP TS 29.501: "5G System; Principles and Guidelines for Services Definition; Stage 3".

[32] OpenAPI: "OpenAPI 3.0.0 Specification", https://github.com/OAI/OpenAPI-Specification/blob/master/versions/3.0.0.md.

[33] 3GPP TS 29.571: "Common Data Types for Service Based Interfaces; Stage 3".

[34] IETF RFC 3339: "Date and Time on the Internet: Timestamps", July 2002.

[35] IETF RFC 3986: "URI Generic Syntax", June 2005.

[36] Standard ECMA-262, 5.1 Edition: "ECMAScript Language Specification", June 2011.

[37] IETF RFC 8259: "The JavaScript Object Notation (JSON) Data Interchange Format", December 2017.

[38] IETF draft-bhutton-json-schema-validation: "JSON Schema Validation: A Vocabulary for Structural Validation of JSON", June 2022.

[39] 3GPP TS 29.517: "5G System; Application Function Event Exposure Service; Stage 3".

[40] 3GPP TS 26.532: "Data Collection and Reporting; Protocols and Formats".

[41] ISO 3166-1: "Codes for the representation of names of countries and their subdivisions — Part 1: Country codes".

[42] ISO 3166-2: "Codes for the representation of names of countries and their subdivisions — Part 2: Country subdivision code".

[43] IETF RFC 2474: "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", December 1998.

[44] IETF RFC 3246: "An Expedited Forwarding PHB (Per-Hop Behavior)", March 2022.

[45] IETF RFC 2597: "Assured Forwarding PHB Group", June 1999.

[46] 3GPP TS 29.554: "5G System; Background Data Transfer Policy Control Service; Stage 3".

[47] IETF RFC 6749: "The OAuth 2.0 Authorization Framework", October 2012.

[48] 3GPP TS 29.222: "Common API Framework for 3GPP Northbound APIs; Stage 3".

[49] IETF RFC 7230: "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", June 2014.

[50] OASIS: "MQTT Version 5.0", https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html

[51] IETC RFC 7519: "JSON Web Token (JWT)", May 2015.

3 Definitions of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the terms given in TR 21.905 [1], TS 26.501 [4], TS 26.506 [5] and the following apply. A term defined in the present document takes precedence over the definition of the same term, if any, in TR 21.905 [1], TS 26.501 [4] or TS 26.506 [5].

Media EAS: Media Application Server deployed as an Edge Application Server.

Media Delivery System: A deployment of a 5GMS System or RTC System.

media delivery session: the time interval during which media is delivered between a Media AS and one or more Media Client participants via reference point M4 at the initiation of an application (which may be a Media-aware Application) associated with each participating Media Client.

media delivery session identifier: a string that uniquely identifies a media delivery session in a Media Delivery System for the purpose of collating information from different system functions.

3.2 Symbols

Void.

3.3 Abbreviations

For the purposes of the present document, the abbreviations given in TR 21.905 [1] and the following apply. An abbreviation defined in the present document takes precedence over the definition of the same abbreviation, if any, in TR 21.905 [1].

5GC	5G Core
AF	Application Function
ANBR	Access Network Bit rate Recommendation
API	Application Programming Interface
AS	Application Server
BDT	Background Data Transfer
CHEM	Coverage and Handoff Enhancements using Multimedia error robustness
DN	Data Network
DS	Differentiated Services
DSCP	DS Code Point
EAS	Edge Application Server
EEC	Edge Enabler Client
EES	Edge Enabler Server
FQDN	Fully Qualified Domain Name

GPSI	Generic Public Subscription Identifier
ICE	Interactive Connectivity Establishment
JSON	JavaScript Object Notation
MFBR	Maximum Flow Bit Rate
NEF	Network Exposure Function
OAM	Operations, Administration and Maintenance
PCC	Policy Control and Charging
PCF	Policy Control Function
PDR	Packet Detection Rule
PHB	Per-Hop Behaviour
QoE	Quality of Experience
QoS	Quality of Service
QFI	QoS Flow Identifier
RTC	Real-Time (media) Communication
STUN	Session Traversal Utilities for NAT, Simple Traversal of User Datagram Protocol (UDP) through Network Address Translators
SWAP	Simple WebRTC Application Protocol
TCP	Transmission Control Protocol
TOS	Type of Service
TURN	Traversal Using Relays around NAT
UE	User Equipment
URI	Uniform Resource Identifier
URL	Uniform Resource Locator

4 Functions and roles

4.1 Media Application Provider

This clause is for future study.

4.2 Media AF

This clause is for future study.

4.3 Media Session Handler

This clause is for future study.

5 Interactions

5.1 Summary

Table 5.1-1 summarises the APIs used to provision and use the various Media Delivery features specified in TS 26.512 [6] (designated "5GMS" in the *Applicability* column) and TS 26.113 [7] (designated "RTC" in the *Applicability* column).

Table 5.1-1: Summary of APIs relevant to Media Delivery features

Media delivery feature	Abstract	Applicability	Reference point	Interactions clause	Relevant APIs	
					API name	API clause
Content hosting	Content is ingested, hosted and distributed by the Media AS according to a Content Hosting Configuration associated with a Provisioning Session. The Media AS may be instantiated in an Edge Data Network. Ingested content may additionally be distributed via eMBMS and/or MBS.	5GMS	M1	5.2.2	Provisioning Sessions API	8.2
				5.2.3	Content protocols discovery API	8.3
				5.2.4	Server Certificates provisioning API	8.4
				5.2.5	Content Preparation Templates provisioning API	8.5
				5.2.6	Edge Resources provisioning API	8.6
				5.2.7	Policy Templates provisioning API	8.7
				5.2.8	Content Hosting provisioning API	8.8
			M5	5.4.2	Service Access Information API	9.2
Content publishing	Content is contributed to and egested from the Media AS according to a Content Hosting Publishing associated with a Provisioning Session. The Media AS may be instantiated in an Edge Data Network.	5GMS	M1	5.2.2	Provisioning Sessions API	8.2
				5.2.3	Content protocols discovery API	8.3
				5.2.4	Server Certificates provisioning API	8.4
				5.2.5	Content Preparation Templates provisioning API	8.5
				5.2.6	Edge Resources provisioning API	8.6
				5.2.7	Policy Templates provisioning API	8.7
				5.2.9	Content Publishing provisioning API	8.9
			M5	5.4.2	Service Access Information API	9.2
Real-Time media Communication (RTC)	Content is exchanged in real time between RTC endpoints in the Media Client and/or the Media AS. The Media AS may be instantiated in an Edge Data Network.	RTC	M1	5.2.2	Provisioning Sessions API	8.2
				5.2.4	Server Certificates provisioning API	8.4
				5.2.6	Edge Resources provisioning API	8.6
				5.2.7	Policy Templates provisioning API	8.7
				5.2.10	Real-time Media Communication provisioning API	8.10
			M5	5.4.2	Service Access Information API	9.2

Dynamic Policy instantiation	The Media Client activates different traffic treatment and charging policies, including Background Data Transfer, selected from a set of Policy Templates provisioned in its Provisioning Session.	5GMS, RTC	M1	5.2.3	Provisioning Sessions API	8.3
				5.2.7	Policy Templates provisioning API	8.7
			M5	5.4.2	Service Access Information API	9.2
				5.4.3	Dynamic Policies API	9.3
Network Assistance	The Media Client requests bit rate recommendations and delivery boosts from the Media AF.	5GMS, RTC	M5	5.4.2	Service Access Information API	9.2
				5.4.4	Network Assistance API	9.4
QoE Metrics reporting	The Media Client submits metrics reports to the Media AF according to a provisioned Metrics Reporting Configuration it obtains from the Service Access Information for its Provisioning Session.	5GMS, RTC	M1	5.2.3	Provisioning Sessions API	8.3
				5.2.11	Metrics Reporting provisioning API	8.10
			M5	5.4.2	Service Access Information API	9.2
				5.4.5	Metrics Reporting API	9.5
Consumption reporting	The Media Client submits consumption reports to the Media AF about content consumed from downlink media delivery sessions according to a provisioned Consumption Reporting Configuration it obtains from the Service Access Information for its Provisioning Session.	5GMS, RTC	M1	5.2.3	Provisioning Sessions API	8.3
				5.2.12	Consumption Reporting provisioning API	8.12
			M5	5.4.2	Service Access Information API	9.2
				5.4.6	Consumption Reporting API	9.6
UE data collection, reporting and exposure	UE data related to media delivery is reported to the Data Collection AF instantiated in the Media AF for exposure to Event consumers.	5GMS	M1	5.2.13	Event Data Processing provisioning API	8.13
			M5	5.4.5	Metrics Reporting API	9.5
				5.4.6	Consumption Reporting API	9.6

5.2 Provisioning (M1) interactions

5.2.1 Overview

A Media Application Provider may use the operations in this clause to provision the different features offered by the Media Delivery System in the Media AF. The Provisioning API exposed by the Media AF to the Media Application Provider at reference point M1 offers the following sets of operations:

1. Provisioning of *Provisioning Sessions* (see clause 5.2.2) to act as an umbrella for the following provisioning information. Each such Provisioning Session is uniquely identified by a system-dependent Provisioning Session identifier as well as by system-independent service identifier that is subsequently used by an application to launch media session handling via a 3GPP Service URL (see clause 6) or used by a Media-aware Application to invoke a method on the Media Session Handler (see clause 5.4.2).

2. Discovery of the set of content ingest and/or egest protocols supported by the Media AS for a particular Provisioning Session (see clause 5.2.3):

   - For downlink media streaming according to TS 26.512 [6], discovery of the content ingest protocols available at reference point M2 and the content distribution protocols available at reference point M4.

   - For uplink media streaming according to TS 26.512 [6], discovery of the content contribution protocols available at reference point M4 and the content egest protocols available at reference point M2.

3. Provisioning of *Server Certificates* within the scope of a Provisioning Session (see clause 5.2.4) to be used by the Media AS to assert its identity to the Media Access Function in Media Clients during media delivery sessions at reference point M4.

4. Provisioning of *Content Preparation Templates* within the scope of a Provisioning Session (see clause 5.2.5) that can be used by the Media AS to manipulate media content ingested at reference point M2 or contributed at reference point M4.

5. Provisioning of *Edge Resources* within the scope of a Provisioning Session (see clause 5.2.6) to be used to instantiate the Media AS as a set of Edge Application Servers (EAS) in an Edge Data Network (EDN) using the APIs specified in TS 29.558 [15].

5. Provisioning of *Policy Templates* within the scope of a Provisioning Session (see clause 5.2.7) that can be applied to M4 downlink/uplink media delivery sessions in order to realise different Service Operation Points as part of the Dynamic Policies feature (see clause 5.4.3).

7. Provisioning of media delivery by the Media AS within the scope of a Provisioning Session using the abovementioned building blocks:

   - For downlink media streaming according to TS 26.512 [6], provisioning of the *Content Hosting* feature of the Media AS (see clause 5.2.8), which offers functionality equivalent to that of a public Content Delivery Network (CDN): content ingest at reference point M2 for onward distribution by the Media AS to Media Clients via reference point M4 or via other distribution systems such as eMBMS or MBS.

     After discovering the set of ingest and distribution content protocols supported by the Media AS (see clause 5.2.2), the Media Application Provider may provision a Server Certificate (see clause 5.2.4), Content Preparation Template (see clause 5.2.5) and/or Edge Resources Configuration (see clause 5.2.6) for each Content Hosting distribution configuration to reference. The Media Application Provider may also provision one or more Policy Templates (see clause 5.2.7) to realise Service Operation Points pertaining to downlink media delivery.

   - For uplink media streaming according to TS 26.512 [6], provisioning of the *Content Publishing* feature of the Media AS (see clause 5.2.9), including content contribution by Media Clients at reference point M4 and subsequent content egest of content at reference point M2 after optional manipulation by a Content Preparation Template.

     After discovering the set of contribution and egest content protocols supported by the Media AS (see clause 5.2.2), the Media Application Provider may provision a Server Certificate (see clause 5.2.4), Content Preparation Template (see clause 5.2.5) and/or Edge Resources Configuration (see clause 5.2.6) for each Content Publishing contribution configuration to reference. The Media Application Provider may also

provision one or more Policy Templates (see clause 5.2.7) to realise Service Operation Points relevant to the parent Provisioning Session.

- For real-time media communication according to TS 26.113 [7], provisioning of the RTC functionality of the Media AS (see clause 5.2.10).

  The Media Application Provider may provision the WebRTC Signalling Function and ICE Function (including TURN and STUN services) of the Media AS to facilitate communication between two RTC endpoints. Additionally, the Media Application Provider may provision Server Certificates (see clause 5.2.4) for presentation by these subfunctions to Media Clients. Alternatively, the Media Application Provider may provide these subfunctions itself and inform the Media AF of their endpoint addresses at the time of provisioning.

  The Media Application Provider may additionally provision an Edge Resources Configuration (see clause 5.2.6) for the RTC Configuration to reference. The Media Application Provider may also provision one or more Policy Templates (see clause 5.2.7) for the RTC Configuration to reference that the Media Session Handler is then able to instantiate for RTC-based media delivery sessions.

8. Provisioning of *QoE metrics reporting* within the scope of a Provisioning Session (see clause 5.2.11) to configure how and how often the Media Client should report Quality of Experience metrics to the Media AF during the course of media delivery sessions at reference point M4.

9. Provisioning of *consumption reporting* within the scope of a Provisioning Session (see clause 5.2.12) to configure how often the Media Client should report downlink media consumption to the Media AF during the course of media delivery sessions at reference point M4.

10. Provisioning of rules for processing of UE data (see clause 5.2.13) related to media delivery sessions by the Data Collection AF instantiated in the Media AF (as defined in clause 4.7 of TS 26.501 [4]), and for restricting its exposure over reference points R5 and R6 by means of Event Data Processing Configurations and Data Access Profiles for a particular Event ID.

NOTE: The *Network Assistance* feature is not provisioned by the Media Application Provider at reference point M1. Instead, it is provisioned at the discretion of the Media Delivery System operator using means beyond the scope of the present document.

5.2.2 Provisioning Session provisioning

5.2.2.1 General

Prior to configuring media delivery features specified in subsequent clauses of the present document, the Media Application Provider shall create a new Provisioning Session resource in the Media AF at reference point M1. The Media Application Provider shall nominate a globally unique *external service identifier* that will be used by the Media Session Handler to launch media delivery sessions and this identifier shall be associated with exactly one Provisioning Session in each Media Delivery System.

The operations specified in the following clauses are used to manage a Provisioning Session resource, as specified in clause 8.2.3.1.

HTTP responses for successful and operation-specific failure cases are specified in the following clauses. For all other failure cases, an HTTP response indicating a response code in accordance with clause 7.1.6 shall be returned to the API client. In all failure cases a message body in accordance with clause 7.1.7 shall be included in the response message.

5.2.2.2 Enumerate Provisioning Sessions collection operation

This procedure is used by the Media Application Provider to enumerate the current set of Provisioning Sessions in the Media AF. The HTTP `GET` method shall be used for this purpose. The request URL shall be that of the Provisioning Sessions collection.

If the operation is successful, the Media AF shall return an HTTP *200 (OK)* response. The resource body shall be a JSON array of Provisioning Session resource identifiers. The array shall contain only resource identifiers for Provisioning Sessions that the Media Application Provider is entitled to manipulate, according to the credentials supplied with the request. This may, for example, be the subset of Provisioning Sessions tagged with the Application

Server Provider identifier of the Media Application Provider. The array shall be empty if no Provisioning Sessions visible to the invoker currently exist in the collection.

5.2.2.3 Create Provisioning Session resource operation

This operation is used by the Media Application Provider to create a new Provisioning Session. The Media Application Provider shall use the HTTP `POST` method to create a new Provisioning Session.

Upon successful creation, the Media AF shall return a *201 (Created)* response message that includes the resource identifier of the newly created Provisioning Session resource (the *Provisioning Session identifier*) in the body of the HTTP response message and the URL of the resource, including its resource identifier, shall be provided as the value of the `Location` HTTP header field. The response message body shall be a representation of the current state of the Provisioning Session resource (see clause 8.2.3.1), including any property values set by the Media AF.

If the request is acceptable but the Media AF is unable to create the Provisioning Session resource, the create operation shall fail with an HTTP response status code of *500 (Internal Server Error)* and an error message body per clause 7.1.7. In this case, the Provisioning Session resource shall remain in an uncreated state in the Media AF.

This operation may be performed multiple times by a Media Application Provider to create different Provisioning Session resources. Each such resource is assigned a different Provisioning Session identifier by the Media AF.

5.2.2.4 Retrieve Provisioning Session resource operation

This operation is used by the Media Application Provider to retrieve the current state of an existing Provisioning Session from the Media AF. The HTTP `GET` method shall be used for this purpose, citing the resource identifier of the target Provisioning Session in the request URL.

If the target Provisioning Session exists, the Media AF shall return an HTTP *200 (OK)* response with a representation of the Provisioning Session resource in the response message body.

5.2.2.5 Update Provisioning Session resource operation

The Update operation is not permitted for the Provisioning Sessions resource. Any attempt to do so using the HTTP `PUT` or `PATCH` methods shall result in the HTTP response *405 (Method Not Allowed)* that includes an error message body per clause 7.1.7.

To achieve an equivalent outcome, the Media Application Provider should instead destroy the existing Provisioning Session resource using the operation specified in clause 5.2.2.5 and create a new one using the operation specified in clause 5.2.2.2.

5.2.2.6 Destroy Provisioning Session resource operation

This operation is used by the Media Application Provider to destroy an existing Provisioning Session. The Media AF shall use the HTTP `DELETE` method for this purpose, citing the resource identifier of the target Provisioning Session in the request URL.

If the target Provisioning Session resource exists, the Media AF shall return a *204 (No Content)* HTTP response message with an empty message body. The Media AF shall release any associated resources in the Media AF and Media AS, purge any cached data, and destroy any sub-resources associated with the target Provisioning Session.

Any subsequent operations citing the resource identifier of a destroyed Provisioning Session should result in a *410 (Gone)* or else a *404 (Not Found)* HTTP response message that includes an error message body per clause 7.1.7.

5.2.3 Content protocols discovery

5.2.3.1 General

The set of downlink content ingest and/or uplink content egest protocols supported by the Media AS at reference point M2 and the set of downlink content distribution and/or uplink content contribution protocols supported by the Media AS at reference point M4 are described by the Content Protocols resource exposed by the Media AF at reference point M1, as specified in clause 8.3.3.1. This resource shall exist in the Media AF as a sub-resource of each created

Provisioning Session and may therefore be different for each one, for example to offer different content protocols depending on properties of the parent Provisioning Session or 5GMS Application Provider.

NOTE: The information contained in the Content Protocols resource is useful to the Media Application Provider when it provides Service Access Information to the Media-aware Application at reference point M8.

HTTP responses for successful and operation-specific failure cases are specified in the following clauses. For all other failure cases, an HTTP response indicating a response code in accordance with clause 7.1.6 shall be returned to the API client. In all failure cases a message body in accordance with clause 7.1.7 shall be included in the response message.

5.2.3.2 Create Content Protocols resource operation

The Create operation is not permitted for the Content Protocols resource. Any usage of the HTTP `POST` method in relation to its well-known resource URL shall result in the HTTP response *405 (Method Not Allowed)* that includes an error message body per clause 7.1.7.

5.2.3.3 Retrieve Content Protocols resource operation

This operation is used by the Media Application Provider to retrieve from the Media AF a list of downlink content ingest protocols and/or uplink content egest protocols supported by the Media AS at reference point M2 and a list of downlink content distribution and/or uplink content contribution protocols supported by the Media AS at reference point M4. The HTTP `GET` method shall be used for this purpose, citing the well-known URL of the Content Protocols resource.

If the operation is successful, the Media AF shall return a *200 (OK)* response that includes a Content Protocols resource in the response message body, as specified in clause 8.3.3.1.

5.2.3.4 Update Content Protocols resource operation

The Update operation is not permitted for the Content Protocols resource. Any usage of the HTTP `PUT` or `PATCH` methods in relation to its well-known resource URL shall result in the HTTP response *405 (Method Not Allowed)* that includes an error message body per clause 7.1.7.

5.2.3.5 Destroy Content Protocols resource operation

The Destroy operation is not permitted for the Content Protocols resource. Any usage of the HTTP `DELETE` method in relation to its well-known resource URL shall result in the HTTP response *405 (Method Not Allowed)* that includes an error message body per clause 7.1.7.

5.2.4 Server Certificate provisioning

5.2.4.1 General

Each X.509 server certificate [10] presented by the Media AS to the Media Client at reference point M4 is represented by a Server Certificate resource at reference point M1. The Server Certificates Provisioning API specified in clause 8.4 enables a Server Certificate resource to be created within the scope of a Provisioning Session, and subsequently referenced by a Content Hosting Configuration or Content Publishing Configuration created in the scope of the same Provisioning Session. That API supports two alternative provisioning methods for Server Certificate resources: one in which a certificate is generated by the Media Delivery System operator on behalf of the Media Application Provider; the other in which a certificate is generated by the Media Application Provider from a Certificate Signing Request solicited from the Media AF. Both methods shall be supported by implementations of the Media AF.

Under no circumstances shall the Media AF reveal the private key associated with a Certificate Signing Request to the Media Application Provider.

HTTP responses for successful and operation-specific failure cases are specified in the following clauses. For all other failure cases, an HTTP response indicating a response code in accordance with clause 7.1.6 shall be returned to the API client. In all failure cases a message body in accordance with clause 7.1.7 shall be included in the response message.

5.2.4.2 Create Server Certificate resource operation

This operation is used by the Media Application Provider to request that the Media Delivery System generates a new X.509 certificate [10] on its behalf within the scope of a Provisioning Session. In this case, the certificate's Common Name (*CN*) and a single Subject Alternative Name (*subjectAltName*, see section 4.2.1.6 of RFC 5280 [11]) is assigned in a domain under the control of the Media Delivery System operator and the use of supplementary domain name aliases is not supported. The first Subject Alternative Name (*subjectAltName*) extension field of the certificate should be identical to its Common Name. Both fields may include a single wildcard ("*") character at the start to indicate applicability to several different subdomains of the same domain.

NOTE 1: Modern TLS client implementations ignore the obsolete Common Name (*CN*) field of the X.509 certificate in favour of the first Subject Alternative Name (*subjectAltName*) extension field.

The Media Application Provider shall use the HTTP `POST` method to create a new Server Certificate resource in the Media AF. The request URL shall be a well-known sub-resource of the Provisioning Session resource representing its Server Certificates resource collection, as specified in clause 8.4.2. The HTTP request message body shall be omitted.

Upon successful creation, the Media AF shall return a *201 (Created)* HTTP response message and the URL of the newly created Server Certificate resource, including its resource identifier, shall be provided in the HTTP `Location` header field. The response message body may optionally convey a copy of the X.509 certificate corresponding to the newly created Server Certificate resource, as specified in clause 8.4.3.2.

NOTE 2: The X.509 certificate corresponding to the newly created Server Certificate resource may not be available immediately for interrogation and use. See clause 5.2.4.5 below for more details.

If the request is acceptable but the Media AF is unable to provision the X.509 certificate, the create operation shall fail with an HTTP response status code of *500 (Internal Server Error)* and an error message body per clause 7.1.7. In this case, the Server Certificate resource shall remain in an uncreated state in the Media AF.

This operation may be performed multiple times by a Media Application Provider to provision different Server Certificate resources within the scope of a Provisioning Session. Each such resource is assigned a different Server Certificate resource identifier by the Media AF.

5.2.4.3 Reserve Server Certificate resource operation

This operation is used by the Media Application Provider to solicit a Certificate Signing Request (CSR) from the Media AF for the purpose of generating an X.509 certificate [10] independently of the Media Delivery System. In this case, the certificate's Common Name (*CN*) is assigned in a domain under the control of the Media Application Provider itself, or that of a third party acting on its behalf. The first Subject Alternative Name (*subjectAltName*) extension field of the certificate should be identical to its Common Name. The *CN* and *subjectAltName* fields may include a single wildcard ("*") character at the start to indicate applicability to several different subdomains of the same domain.

NOTE: Modern TLS client implementations ignore the obsolete Common Name (*CN*) field of the X.509 certificate in favour of the first Subject Alternative Name (*subjectAltName*) extension field.

The Media Application Provider may specify additional domains in its certificate reservation request to the Media AF. If provided, these domain name aliases shall be included in the returned Certificate Signing Request using the Subject Alternative Name (*subjectAltName*) extension (see section 4.2.1.6 of RFC 5280 [11]). In this case, the Media Application Provider is responsible for ensuring that any FQDN aliases it subsequently provisions in Content Hosting Configurations or Content Publishing Configurations matching these additional domains resolve to the canonical domain name of the Media AS in the target Media Application System.

The Media Application Provider shall separately arrange for the FQDN carried in the Common Name of the certificate generated, or those of all Subject Alternative Name (*subjectAltName*) extensions in the same certificate (see section 4.2.1.6 of RFC 5280 [11]), to resolve to the address of a Media AS in the target Media Delivery System after provisioning the Content Hosting feature per clause 5.2.8.2 or the Content Publishing feature per clause 5.2.9.2.

The Media Application Provider shall use the HTTP `POST` method to create a new Server Certificate. The request URL shall be a well-known sub-resource of the Provisioning Session resource representing its Server Certificates resource collection, as specified in clause 8.4.2, including the query parameter specified there. Domain name aliases (if any) shall be conveyed in the HTTP request message body, encoded as a JSON [37] array of strings; otherwise the request message body shall be omitted. Upon successful creation of the resource, the Media AF shall return a *201 (Created)* response message and the URL of the resource, including the resource identifier of the reserved Server Certificate

resource, shall be returned in the HTTP `Location` header. The HTTP response message shall provide a Certificate Signing Request as specified in clause 8.4.3.1.

If the list of additional domains in the HTTP request message is malformed, the Media AF shall return a *400 (Bad Request)* response message.

If the request is acceptable but the Media AF is unable to generate a Certificate Signing Request, the creation operation shall fail with an HTTP response status code of *500 (Internal Server Error)* and an error message body per clause 7.1.7. In this case, the Server Certificate resource shall remain in an uncreated state in the Media AF.

This operation may be performed multiple times by a Media Application Provider to provision different Server Certificate resources within the scope of a Provisioning Session. Each such resource is assigned a different Server Certificate resource identifier by the Media AF.

5.2.4.4 Upload Server Certificate resource operation

This operation is used by a Media Application Provider to upload an X.509 certificate [10] to the Media AF that it has generated in response to a Certificate Signing Request solicited using the reservation operation specified in clause 5.2.4.3 above. The Media Application Provider shall use the HTTP `PUT` method for this purpose. The `Content-Type` request header and the body of the HTTP request message shall be as specified in clause 8.4.3.2.

The Media AF shall verify that the party originating the upload is the same party that reserved the Server Certificate resource using the operations specified in clause 5.2.4.3 above before accepting the supplied X.509 certificate. The Media AF shall also verify that the X.509 certificate uploaded corresponds to the Certificate Signing Request it issued for the Server Certificate resource in question. If there is a mismatch, the HTTP response *403 (Forbidden)* shall be returned.

Attempting to upload an X.509 certificate to a Server Certificate resource URL that has not been reserved using the operation specified in clause 5.2.4.3 above shall elicit a *404 (Not Found)* HTTP response.

On success, the HTTP response *204 (No Content)* shall be returned with an empty response body.

5.2.4.5 Retrieve Server Certificate resource operation

This operation is used by the Media Application Provider to download a Server Certificate resource from the Media AF for inspection. The HTTP `GET` method shall be used for this purpose, citing the resource identifier of the target Server Certificate in the request URL.

If the requested resource exists and is populated with an X.509 certificate [10], the Media AF shall return *200 (OK)* HTTP response message with a representation of the target Server Certificate in the response message body in accordance with clause 8.4.3.2.

In the case where the X.509 certificate was provisioned by the Media Delivery System on behalf of the Media Application Provider according to clause 5.2.4.2 above, the HTTP response *204 (No Content)* shall be returned until such time as the X.509 certificate is generated and available for download. The optional HTTP response header `Retry-After` should be included in such a response, indicating when the certificate is expected to become available for inspection and use.

In cases where the X.509 certificate is to be generated by the Media Application Provider from a Certificate Signing Request obtained according to clause 5.2.4.3 above, the HTTP response *204 (No Content)* shall be returned with an empty message response body until such time as the X.509 certificate has been uploaded using the operation specified in clause 5.2.4.4 above.

5.2.4.6 Update Server Certificate resource operation

The Update operation is not permitted for the Server Certificate resource. Any attempt to do so using the HTTP `PUT` method shall result in the HTTP response *405 (Method Not Allowed)* that includes an error message body per clause 7.1.7.

Updating a previously uploaded Server Certificate in the Media AF is not permitted for security reasons. To supply a replacement X.509 certificate, for example when a previously supplied certificate is shortly due to expire, the Media Application Provider should instead use one of the operations specified in clause 5.2.4.2 or 5.2.4.3 above to create or

reserve a new Server Certificate resource and, once the certificate is available for use, update the Content Hosting Configuration to reference it.

5.2.4.7 Destroy Server Certificate resource operation

This operation is used by the Media Application Provider to remove a Server Certificate resource from a Provisioning Session in the Media AF. The HTTP `DELETE` method shall be used for this purpose, citing the resource identifier of the target Server Certificate in the request URL.

On success, the HTTP response *204 (No Content)* shall be returned with an empty response body and afterwards the identifier of the Service Certificate resource is no longer valid. The party that originally created (see clause 5.2.4.2) or reserved (see clause 5.2.4.3) the Server Certificate resource is responsible for ensuring that the serial number of the destroyed certificate is appropriately revoked. Only the party that created (see clause 5.2.4.2) or reserved (see clause 5.2.4.3) the Server Certificate resource is permitted to destroy it. Any attempt by another party to destroy a Server Certificate resource shall elicit the HTTP response *405 (Method Not Allowed)*.

The HTTP response *409 (Conflict)* shall be returned with an error message body per clause 7.1.7 if an attempt is made to destroy a Server Certificate resource that is currently referenced by a Content Hosting Configuration or Content Publishing Configuration resource.

Attempting to destroy a Server Certificate resource that has been reserved but never uploaded shall elicit a *200 (OK)* HTTP response with an empty message body. In this case, the Media AF should release any resources associated with the reservation.

Any subsequent operations citing the resource identifier of a destroyed Server Certificate should result in a *410 (Gone)* or else a *404 (Not Found)* HTTP response message that includes an error message body per clause 7.1.7.

5.2.5 Content Preparation provisioning

5.2.5.1 General

For downlink media delivery, the Media AS may be required to process content ingested at reference point M2 before distributing it at reference point M4. For uplink media delivery, the Media AS may be required to process content contributed by Media Clients before publishing it to the Media Application Provider at reference point M2. These content processing operations are described by a Content Preparation Template resource provisioned in the Media AF by the Media Application Provider at reference point M1, as specified in clause 8.5, and subsequently configured in the Media AS by the Media AF at reference point M3 using an API outside the scope of the present document.

HTTP responses for successful and operation-specific failure cases are specified in the following clauses. For all other failure cases, an HTTP response indicating a response code in accordance with clause 7.1.6 shall be returned to the API client. In all failure cases a message body in accordance with clause 7.1.7 shall be included in the response message.

5.2.5.2 Create Content Preparation Template resource operation

This operation is used by the Media Application Provider to register a new Content Preparation Template with a Provisioning Session in the Media AF. The Media Application Provider shall use the HTTP `POST` method to upload a new Content Preparation Template resource to the Media AF. The request URL shall be a well-known sub-resource of the Provisioning Session resource representing its Content Preparation Templates resource collection, as specified in clause 8.5.2. The HTTP request message body shall be a Content Preparation Template as specified in clause 8.5.3. The MIME content type of the Content Preparation Template shall be supplied in the `Content-Type` HTTP request header.

Upon successful creation, the Media AF shall respond with a *201 (Created)* response message and the URL of the newly created resource, including its resource identifier, shall be provided as the value of the `Location` HTTP header field. The response message body shall be a representation of the current state of the Content Preparation Template resource (see clause 8.5.2.1), including any properties assigned by the Media AF.

This operation may be performed multiple times by a Media Application Provider to provision different Content Preparation Template resources within the scope of a Provisioning Session. Each such resource is assigned a different Content Preparation Template identifier by the Media AF.

If the MIME content type indicated in `Content-Type` is not understood by the Media AF, the creation of the Content Preparation Template resource shall fail with HTTP error response status code *415 (Unsupported Media Type)* that includes an error message body per clause 7.1.7.

If the request is acceptable but the Media AF is unable to provision the resources required by the supplied Content Preparation Template, the create operation shall fail with an HTTP response status code of *500 (Internal Server Error)* and an error message body per clause 7.1.7. In this case, the Content Preparation Template resource shall remain in an uncreated state in the Media AF.

5.2.5.3 Retrieve Content Preparation Template resource operation

This operation is used by the Media Application Provider to retrieve the current state of a Content Preparation Template resource from the Media AF. The HTTP `GET` method shall be used for this purpose, citing the resource identifier of the target Server Certificate in the request URL.

If the operation is successful, the Media AF shall respond with *200 (OK)* and shall provide a representation of the requested resource in the HTTP message response body. The `Content-Type` response header shall have the same value as that supplied when the Content Preparation Template resource was created using the operation specified in clause 5.2.5.2.

5.2.5.4 Update Content Preparation Template resource operation

This operation is used by the Media Application Provider to modify or replace an existing Content Preparation Template resource. The HTTP `PATCH` or HTTP `PUT` methods shall be used for this purpose. The replacement Content Preparation Template resource representation shall be provided in the body of the HTTP request message.

If the HTTP request is acceptable but the operation results in no change to the resource representation, a *204 (No Content)* HTTP response message with an empty body should be returned.

If the operation is otherwise successful, the Media AF shall return a *200 (OK)* HTTP response message and shall provide a representation of the current state of the resource in the message body to confirm successful update.

If the Media AF does not support modification of the Content Preparation Template, the update operation shall fail with a *403 (Forbidden)* HTTP response that includes an error message body per clause 7.1.7.

If the MIME content type indicated in `Content-Type` is not acceptable to the Media AF, the creation of the Content Preparation Template resource shall fail with HTTP error response status code *415 (Unsupported Media Type)*.

If the request is acceptable but the Media AF is unable to provision the resources required by the supplied Content Preparation Template, the update operation shall fail with an HTTP response status code of *500 (Internal Server Error)* and an error message body per clause 7.1.7. In this case, the Content Preparation Template resource in the Media AF shall remain in the state immediately prior to the update operation.

5.2.5.5 Destroy Content Preparation Template resource operation

This operation is used by the Media Application Provider to destroy a Content Preparation Template resource. The HTTP `DELETE` method shall be used for this purpose, citing the resource identifier of the target Content Preparation Template in the request URL.

If the operation is successful, the Media AF shall return a *204 (No Content)* response message with an empty message body.

If the Content Preparation Template is still referenced by a Content Hosting Configuration or Content Publishing Configuration, the operation shall fail with HTTP error response status code *409 (Conflict)* that includes an error message body per clause 7.1.7.

5.2.6 Edge Resources provisioning

5.2.6.1 General

These operations are used by the Media Application Provider at reference point M1 to provision edge computing resources in the Media AF for the purpose of instantiating Edge Application Server (EAS) instances of the Media AS in an Edge Data Network (EDN), as defined in TS 23.558 [13].

HTTP responses for successful and operation-specific failure cases are specified in the following clauses. For all other failure cases, an HTTP response indicating a response code in accordance with clause 7.1.6 shall be returned to the API client. In all failure cases a message body in accordance with clause 7.1.7 shall be included in the response message.

5.2.6.2 Create Edge Resources Configuration resource operation

This operation is used by the Media Application Provider to create a new Edge Resources Configuration resource in the Media AF. The HTTP `POST` method shall be used for this purpose. The request URL shall be a well-known sub-resource of the Provisioning Session resource representing its Edge Resources Configurations resource collection, as specified in clause 8.6.2. The request message body shall be an Edge Resources Configuration resource representation, as specified in clause 8.6.3.1.

- If the *edgeManagmentMode* is set to *EM_AF_DRIVEN* (indicating AF-driven edge resource management), the Media AF is responsible for requesting and managing the required edge resources and for handling EAS relocation in relation to media delivery sessions that fall within the scope of the parent Provisioning Session.

- If the *edgeManagementMode* is set to *EM_CLIENT_DRIVEN* (indicating client-driven edge resource management), the Media AF shall only request edge resources based on requests from the Edge Enabler Client (EEC) instantiated in the Media Session Handler at reference point EDGE-1 (as defined in clause 6 of TS 23.558 [13]).

If the operation is successful, the Media AF shall generate a resource identifier representing the new Edge Resources Provisioning Configuration. In this case, the Media AF shall respond with a *201 (Created)* HTTP response message and the URL of the newly created resource, including its resource identifier, shall be provided as the value of the `Location` HTTP header field. The response message body shall be a representation of the current state of the Edge Resources Configuration resource (see clause 8.6.3.1), including any properties assigned by the Media AF.

If the request is acceptable but the Media AF is unable to provision the resources required by the supplied Edge Resources Configuration, the create operation shall fail with an HTTP response status code of *500 (Internal Server Error)* and an error message body per clause 7.1.7. In this case, the Edge Resources Configuration resource shall remain in an uncreated state in the Media AF.

This operation may be performed multiple times by a Media Application Provider to provision different Edge Resources Configuration resources within the scope of a Provisioning Session. Each such resource is assigned a different Edge Resources Configuration identifier by the Media AF.

5.3.6.3 Retrieve Edge Resources Configuration resource operation

This operation is used by the Media Application Provider to retrieve the current state of an existing Edge Resources Provisioning Configuration resource from the Media AF. The HTTP `GET` method shall be used for this purpose.

If the procedure is successful, the Media AF shall return a *200 (OK)* response message that includes a representation of the target Edge Resources Configuration resource (see clause 8.6.3.1) in the response message body.

5.2.6.4 Update Edge Resources Configuration resource operation

This operation is invoked by the Media Application Provider to modify the properties of an existing Edge Resources Configuration resource. All writeable properties except *edgeManagementMode* may be updated. The HTTP `PATCH` or HTTP `PUT` methods shall be used for this purpose. The replacement Edge Resources Configuration resource representation shall be provided in the body of the HTTP request message.

If the HTTP request is acceptable but the operation results in no change to the resource representation, a *204 (No Content)* HTTP response message with an empty body should be returned.

If the operation is otherwise successful, the Media AF shall return a *200 (OK)* HTTP response message and shall provide a representation of the resulting state of the resource in the message body to confirm successful update.

Attempts to modify read-only properties of the target Edge Resources Configuration resource, such as the edge management mode, shall be rejected by the Media AF with a *403 (Forbidden)* HTTP response that includes an error message body per clause 7.1.7.

If the request is acceptable but the Media AF is unable to provision the resources required by the supplied Edge Resources Configuration, the update operation shall fail with an HTTP response status code of *500 (Internal Server Error)* and an error message body per clause 7.1.7. In this case, the Edge Resources Configuration resource in the Media AF shall remain in the state immediately prior to the update operation.

5.2.6.5 Destroy Edge Resources Configuration resource operation

This operation is used by the Media Application Provider to destroy an Edge Resources Configuration resource in the Media AF. The HTTP `DELETE` method shall be used for this purpose, citing the resource identifier of the target Edge Resources Configuration in the request URL. This operation makes the configuration unusable for future media delivery sessions, but it does not affect any ongoing media delivery sessions.

If the operation is successful, the Media AF shall return a *204 (No Content)* HTTP response message with an empty message body.

The HTTP response *409 (Conflict)* shall be returned with an error message body per clause 7.1.7 if an attempt is made to destroy an Edge Resources Configuration resource that is currently referenced by a Content Hosting Configuration or Content Publishing Configuration resource.

Any subsequent operations citing the resource identifier of a destroyed Edge Resources Configuration should result in a *410 (Gone)* or else a *404 (Not Found)* HTTP response message that includes an error message body per clause 7.1.7.

5.2.7 Dynamic Policy provisioning

5.2.7.1 General

These operations are used by the Media Application Provider to configure Policy Templates for the media delivery sessions of a particular Provisioning Session.

A Policy Template, identified by its *policyTemplateId*, represents a set of PCF/NEF API parameters which defines the service quality and/or associated charging for the corresponding media delivery session(s). The Policy Template is configured as part of the provisioning procedures with the Media AF using the API specified in clause 8.7 and is subsequently instantiated by a Media Session Handler using the interactions specified in clause 5.3.3.

When a Policy Template requires media to be delivered in a specific Data Network and/or network slice at reference point M4, the *applicationSessionContext* array shall be present with at least one of the following properties populated:

- The *dnn* property contains the name of the Data Network in which the Media AS is hosted.
- When Network Slicing is used, the *sliceInfo* property contains information about the network slice which is serving the UE.

When a Policy Template is intended to influence the network QoS of Service Data Flows used for media delivery, the *qoSSpecifications* array shall be populated with objects of type *M1QoSSpecification* (see clause 7.3.3.4). Each member of the array describes the QoS limits of an application service component that a Media Client is permitted request when instantiating the Policy Template:

- The *componentReference* property is a string used by the Media Session Handler to reference this *M1QoSSpecification* when instantiating the Policy Template. It shall be unique for all members of the same *qoSSpecifications* array.
- The *qosReference* value, as specified in clause 5.6.2.7 of TS 29.514 [18], is obtained with the Service Level Agreement. See TS 23.502 [3] for detailed usage.
- The *maximumBitRate* properties of the *downlinkQosSpecification* and *uplinkQosSpecification* objects define the maximal bit rates which are permitted to be requested by a Media Session Handler on (respectively) downlink

and uplink Service Data Flows. These values are defined by configuration of the 5G System and are therefore populated by the Media AF rather than by the Media Application Provider.

- The *maximumAuthorisedBitRate* properties of the *downlinkQosSpecification* and *uplinkQosSpecification* objects define the maximal bit rates which a Media Session Handler is authorised to request on (respectively) downlink and uplink Service Data Flows. Higher bit rates are not authorised by the Media Application Provider when the Policy Template is instantiated.

- The *minimumPacketLossRate* properties of the *downlinkQosSpecification* and *uplinkQosSpecification* objects define the minimal packet loss rates which are permitted to be requested by a Media Session Handler on (respectively) downlink and uplink Service Data Flows. Lower packet loss rates are not permitted by the Media Application Provider when the Policy Template is instantiated.

- The *pduSetQosLimits* properties of the *downlinkQosSpecification* and *uplinkQosSpecification* objects define the minimal delay budget and minimal error rates for PDU Sets which are permitted to be requested by a Media Session Handler on (respectively) downlink and uplink Service Data Flows. Lower delay and error rates are not permitted by the Media Application Provider when the Policy Template is instantiated.

- The *pduSetMarking* flag is used to specify whether Media Clients instantiating this Policy Template for uplink media delivery, or Media AS instances for downlink media delivery, are required to apply PDU Set marking to media transport protocol PDUs falling within the scope of a Dynamic Policy Instance based on this Policy Template.

NOTE: PDU Set marking is used by the 5G System to satisfy the QoS requirements of application flows.

When a Policy Template is intended to be used for differential charging, the *chargingSpecification* property shall be present.

When a Policy Template is intended to be used for Background Data Transfer, the properties of a new Background Data Transfer policy are specified by the Media Application Provider in the *bdtSpecification* property (of type *M1BDTSpecification*).

- The *startDate* and *endDate* indicate the time period for which the Background Data Transfer specification is valid. A Background Data Transfer specification may be removed from its parent Policy Template by the Media AF when it expires.

- The *windows* property indicates the time windows over which the Background Data Transfer may occur.

  - Each such time window is characterised by a start time (*startTime*), a duration (*duration*) and the days of the week on which the time window is scheduled (*daysOfWeek*).

  - The *numberOfUes* property indicates the maximum number of UEs permitted to instantiate the Policy Template and make use of Background Data Transfers during a single time window instance.

  - The *averageDataVolumePerUe* that reflects the average data volume that each UE is expected to transfer during a single time window instance.

NOTE: The product of the *numberOfUes* and *averageDataVolumePerUe* properties represents an estimate of the maximum data volume that may be transferred during any given time window instance.

  - The *aggregateUplinkBitRateLimit* and *aggregateDownlinkBitRateLimit* properties specify limits on the total aggregate bit rate of all currently instantiated Policy Templates to be enforced by the Media AF's admission control function. If omitted, the Media AF may instantiate a Policy Template with a Background Data Transfer specification regardless of additional costs that may be incurred by the Media Application Provider as a result.

HTTP responses for successful and operation-specific failure cases are specified in the following clauses. For all other failure cases, an HTTP response indicating a response code in accordance with clause 7.1.6 shall be returned to the API client. In all failure cases a message body in accordance with clause 7.1.7 shall be included in the response message.

5.2.7.2 Policy Template life-cycle

The state of a Policy Template is exposed by the Media AF in the *state* property of the Policy Template resource and has one of the values specified in table 5.2.7.2-1.

Table 5.2.7.2-1: Policy Template states

Policy Template state	Meaning
PENDING	The Policy Template is awaiting validation by the Media Delivery System, potentially because not all required parameters have yet been provided. This is the default state after Policy Template creation.
INVALID	One or more of the Policy Template's properties failed validation by the Media Delivery System.
READY	After successful validation by the Media Delivery System the Policy Template moves into this state.
SUSPENDED	The Media Delivery System may move a Policy Template into this state under certain conditions defined within the Service Level Agreement.

Figure 5.2.7.2-1 below is a state diagram showing the life-cycle of a Policy Template resource in the Media AF.

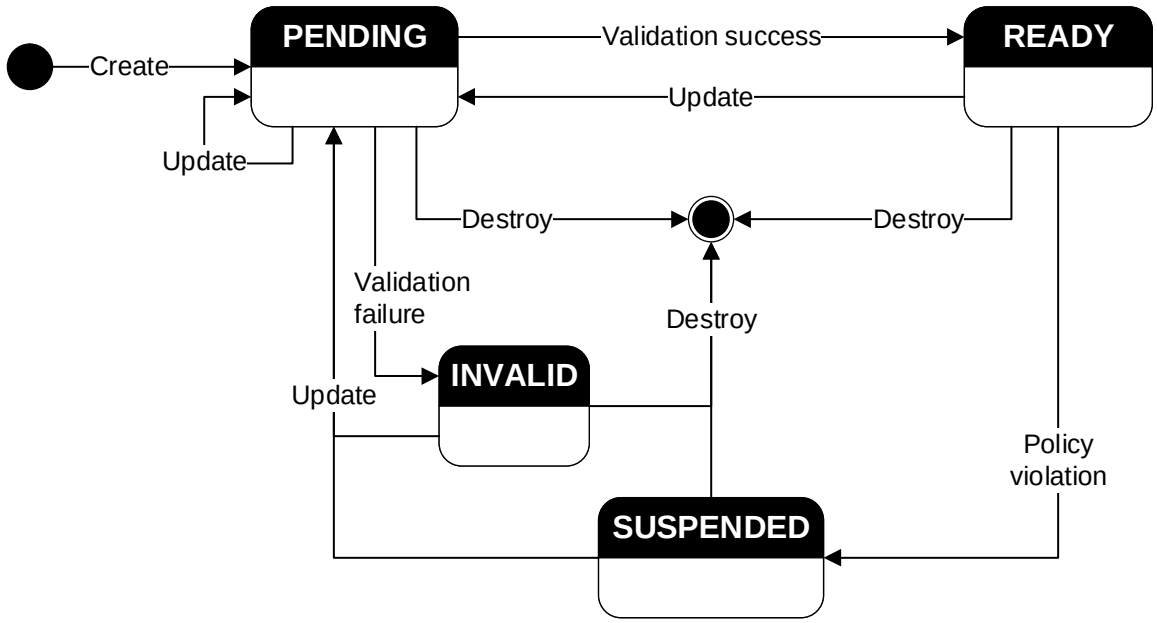

Figure 5.2.7.2-1: Policy Template Resource State Diagram

Policy Templates require Media Delivery System operator verification, and a Policy Template resource that is newly created cannot be used immediately.

1. Upon creation, a Policy Template resource shall be in the *PENDING* state. Once all mandatory properties are provided, the Media AF triggers validation.
2. If the Policy Template is not deemed to be valid by the operator of the Media Delivery System, it shall move to the *INVALID* state, from where it can be updated to remedy the defect.
3. Once it has been successfully validated by the Media Delivery System operator, a Policy Template resource shall take the *READY* state, indicating that it may be applied to media delivery sessions.
4. If it is subsequently updated by the Media Application Provider, a Policy Template resource shall return to the *PENDING* state, awaiting revalidation by the operator of the Media Delivery System.
5. Finally, a Policy Template resource may be *SUSPENDED* by the Media Delivery System operator, e.g., in case of a violation of the usage terms or for some other reasons, which renders it unusable. The update of any property moves the state from *SUSPENDED* into *PENDING* and triggers revalidation.

A Policy Template resource may be destroyed when it is in any of the abovementioned states.

The Media AF shall verify the status of a Policy Template resource prior to allowing a Dynamic Policy Instance to instantiate it. Only a Policy Template resource in the *READY* state is eligible to be instantiated in this way.

The Media AF shall indicate the current state of a Policy Template in the Policy Template resource in machine-readable form as well as indicating a human-readable reason for this state.

5.2.7.3 Create Policy Template resource operation

This operation is used by the Media Application Provider to create a new Policy Template resource. The HTTP `POST` method shall be used for this purpose. The request URL shall be a well-known sub-resource of the Provisioning Session resource representing its Policy Templates resource collection, as specified in clause 8.7.2. The HTTP request message body shall be a Policy Template resource representation, as specified in clause 8.7.3.1.

If the Policy Template includes the *bdtSpecification* property, the Media AF shall attempt to create a series of new Background Data Transfer policies in the PCF using the procedure specified in clause 5.5.1A. The individual Background Data Transfer policy associated with each time window shall be created prior to the start of that time window.

If the procedure is successful, the Media AF shall generate a resource identifier to uniquely identify the newly created Policy Template resource. In that case, it shall return a *201 (Created)* HTTP response message and the URL of the newly created resource, including its resource identifier, shall be provided as the value of the `Location` HTTP header field. The response message body shall be a representation of the current state of the Policy Template resource (see clause 8.7.3.1), including any property values set by the Media AF.

If the request is acceptable but the Media AF is unable to provision the resources required by the supplied Policy Template (for example if it fails to create a new Background Data Transfer policy), the create operation shall fail with an HTTP response status code of *500 (Internal Server Error)* and an error message body per clause 7.1.7. In this case, the Policy Template resource shall remain in an uncreated state in the Media AF.

This operation may be performed multiple times by a Media Application Provider to provision different Policy Template resources within the scope of a Provisioning Session. Each such resource is assigned a different Policy Template identifier by the Media AF.

The default state of a newly created Policy Template resource is *PENDING*. If all mandatory property values have been provided, the Policy Template resource is eligible for validation, as specified in clause 5.2.7.2.

5.2.7.4 Retrieve Policy Template resource operation

This operation is used by the Media Application Provider to retrieve the current state of an existing Policy Template resource in the Media AF. The HTTP `GET` method shall be used for this purpose, citing the resource identifier of the target Policy Template in the request URL.

If the operation is successful, the Media AF shall return a *200 (OK)* response that includes a representation of the target Policy Template resource (see clause 8.7.3.1) in the response message body.

5.2.7.5 Update Policy Template resource operation

This operation is invoked by the Media Application Provider to modify the properties of an existing Policy Template resource in the Media AF. All available properties except *state* may be updated. The HTTP `PATCH` or HTTP `PUT` methods shall be used for this purpose. The replacement Policy Template resource representation shall be provided in the body of the HTTP request message.

Any update to the Policy Template resource shall automatically change its state back to *PENDING*, which makes it temporarily unusable by ongoing media delivery sessions. Accordingly, any attempt to instantiate the Policy Template (see clause 5.4.3) shall fail. If all mandatory property values have been provided by the Media Application Provider, the updated Policy Template is eligible for revalidation.

If the HTTP request is acceptable but the operation results in no change to the resource representation, a *204 (No Content)* HTTP response message with an empty body should be returned.

If the operation is otherwise successful, the Media AF shall return a *200 (OK)* response message that includes a representation of the target Policy Template resource in the response message body, confirming successful update.

Attempts to modify read-only properties of the target Policy Template resource, such as its state, shall be rejected by the Media AF with a *403 (Forbidden)* HTTP response that includes an error message body per clause 7.1.7.

If the request is acceptable but the Media AF is unable to provision the resources required by the supplied Policy Template, the update operation shall fail with an HTTP response status code of *500 (Internal Server Error)* and an error

message body per clause 7.1.7. In this case, the Policy Template resource in the Media AF shall remain in the state immediately prior to the update operation.

5.2.7.6 Destroy Policy Template resource operation

This operation is used by the Media Application Provider to destroy a Policy Template resource in the Media AF. The HTTP `DELETE` method shall be used for this purpose, citing the resource identifier of the target Policy Template in the request URL.

If the operation is successful, the Media AF shall return a *204 (No Content)* response message with an empty message body.

All currently active media delivery sessions using the destroyed Policy Template shall revert to a default network QoS as a result of destroying the Policy Template resource.

Any subsequent operations citing the resource identifier of a destroyed Policy Template should result in a *410 (Gone)* or else a *404 (Not Found)* HTTP response message that includes an error message body per clause 7.1.7.

5.2.8 Content Hosting provisioning

5.2.8.1 General

These operations are used by the Media Application Provider at reference point M1 to provision the Content Hosting feature for downlink media delivery.

HTTP responses for successful and operation-specific failure cases are specified in the following clauses. For all other failure cases, an HTTP response indicating a response code in accordance with clause 7.1.6 shall be returned to the API client. In all failure cases a message body in accordance with clause 7.1.7 shall be included in the response message.

5.2.8.2 Create Content Hosting Configuration resource operation

This operation is used by the Media Application Provider at reference point M1 to activate the Content Hosting feature for a particular Provisioning Session. The Media Application Provider shall use the HTTP `POST` method for this purpose. The request URL shall be a well-known sub-resource of the Provisioning Session resource, as specified in clause 8.8.2. The HTTP request message body shall be a Content Hosting Configuration resource representation, as specified in clause 8.8.3.1. There is at most one Content Hosting Configuration at a time for a given Provisioning Session.

Regarding the configuration of content ingest by the Media AS from the Media Application Provider at reference point M2:

- If the Content Hosting Configuration uses the pull-based content ingest method, i.e., the *ingestConfiguration.mode* attribute is set to *PULL*, then the *ingestConfiguration.baseURL* property shall be nominated by the Media Application Provider in the request message body. The Media AF shall return the *IngestConfiguration.baseURL* property value unchanged in its response message body.

- If the Content Hosting Configuration uses the push-based content ingest method, i.e., the *ingestConfiguration.mode* attribute is set to *PUSH*, then the *ingestConfiguration.baseURL* property shall be nominated by the Media AF and returned in the response message body. It shall not be set by the Media Application Provider in the request message body.

Regarding the configuration(s) of content distribution by the Media AS to the Media Client at reference point M4:

- In all cases, the *distributionConfiguration.canonicalDomainName* and *distributionConfiguration.baseURL* properties are read-only: they shall always be omitted from the creation request and shall be assigned by the Media AF, allowing their values to be inspected by the Media Application Provider in the returned Content Hosting Configuration resource representation, or by using the operation specified in clause 5.2.8.3 below.

- If the *distributionConfiguration.certificateId* property is present and valid, the Media AF shall assign a canonical domain name for the Media AS to expose at reference point M4 that matches the Common Name and the first Subject Alternative Name in the referenced Server Certificate resource (taking into account wildcard matching)

regardless of whether the corresponding X.509 certificate was created using the operation specified in clause 5.2.4.2 or those specified in clauses 5.2.4.3 and 5.2.4.4.

- The Media Application Provider may nominate an alternative domain name to be advertised to the Media Client in the Service Access Information by setting the *distributionConfiguration.domainNameAlias* property when (and only when) creating the Content Hosting Configuration resource. If valid, the value of this property shall then appear in the *distributionConfiguration.baseURL* assigned by the Media AF instead of *distributionConfiguration.canonicalDomainName*. The Media Application Provider shall ensure that this domain name alias resolves to the canonical domain name of the Media AS notified by the Media AF in its response by means of suitable DNS configuration.

If the operation is successful, the Media AF shall return a *201 (Created)* HTTP response message and the request URL shall be returned as the value of the `Location` HTTP header field. The response message body shall be a representation of the current state of the Content Hosting Configuration resource (see clause 8.8.3.1), including any properties assigned by the Media AF.

If any resources referenced by the supplied Content Hosting Configuration resource representation are invalid, the create operation shall fail with an HTTP response status code of *400 (Bad Request)* and an error message body per clause 7.1.7. In this case, the Content Hosting Configuration resource shall remain in an uncreated state in the Media AF.

If *distributionConfiguration.domainNameAlias* is set in the supplied Content Hosting Configuration resource representation but its value is not a syntactically valid Fully-Qualified Domain Name or if the *distributionConfiguration.certificateId* property is absent or if the supplied domain name alias does match any of one of the Subject Alternative Names listed in the Server Certificate referenced by the *distributionConfiguration.certificateId* property, the create operation shall fail with an HTTP response status code of *400 (Bad Request)* and an error message body per clause 7.1.7. In this case, the Content Hosting Configuration resource shall remain in an uncreated state in the Media AF.

NOTE: Even if multiple distribution configurations in the same Content Hosting Configuration reference the same Server Certificate resource, they may each nominate a different domain name alias from among its Subject Alternative Names.

Attempting to create a Content Hosting Configuration in the scope of a Provisioning Session of any type other than *MS_DOWNLINK* shall fail with an HTTP response status code of *403 (Forbidden)* and an error message body per clause 7.1.7. In this case, the Content Hosting Configuration resource shall remain in an uncreated state in the Media AF.

If the request is acceptable but the Media AF is unable to provision the resources required by the supplied Content Hosting Configuration, the create operation shall fail with an HTTP response status code of *500 (Internal Server Error)* and an error message body per clause 7.1.7. In this case, the Content Hosting Configuration resource shall remain in an uncreated state in the Media AF.

5.2.8.3 Retrieve Content Hosting Configuration resource operation

This operation is used by the Media Application Provider to retrieve the current state of an existing Content Hosting Configuration resource from the Media AF. The HTTP `GET` method shall be used for this purpose.

If the operation is successful, the Media AF shall return a *200 (OK)* response message that includes a representation of the target Content Hosting Configuration resource (see clause 8.8.3.1) in the response message body.

5.2.8.4 Update Content Hosting Configuration resource operation

This operation is invoked by the Media Application Provider to modify the properties of an existing Content Hosting Configuration resource. All writeable properties except *domainNameAlias* may be updated. The HTTP `PATCH` or HTTP `PUT` methods shall be used for this purpose. The replacement Content Hosting Configuration resource representation shall be provided in the body of the HTTP request message.

If the HTTP request is acceptable but the operation results in no change to the resource representation, a *204 (No Content)* HTTP response message with an empty body should be returned.

If the operation is otherwise successful, the Media AF shall return a *200 (OK)* HTTP response message and shall provide a representation of the current state of the target resource in the message body to confirm successful update.

If any resources referenced by the supplied Content Hosting Configuration resource representation are invalid, the create operation shall fail with an HTTP response status code of *400 (Bad Request)* and an error message body per clause 7.1.7. In this case, the Content Hosting Configuration resource shall remain in the state immediately prior to the update operation.

Attempts to modify read-only properties of the target Content Hosting Configuration resource, such as the canonical domain name of a distribution configuration, shall be rejected by the Media AF with a *403 (Forbidden)* HTTP response that includes an error message body per clause 7.1.7.

If the request is acceptable but the Media AF is unable to provision the resources required by the supplied Content Hosting Configuration, the update operation shall fail with an HTTP response status code of *500 (Internal Server Error)* and an error message body per clause 7.1.7. In this case, the Content Hosting Configuration resource in the Media AF shall remain in the state immediately prior to the update operation.

5.2.8.5 Destroy Content Hosting Configuration resource operation

This operation is used by the Media Application Provider to destroy a Content Hosting Configuration resource in the Media AF and to terminate the related content distribution. The HTTP `DELETE` method shall be used for this purpose, citing the well-known sub-resource of the target Provisioning Session in the request URL. As a result, the Media AF shall release any associated network resources, purge any cached content in the Media AS, and remove any corresponding configurations.

If the procedure is successful, the Media AF shall return a *200 (OK)* HTTP response message with an empty message body.

5.2.8.6 Purge Content Hosting cache operation

This operation is used by the Media Application Provider to purge content from the Media AS Content Hosting cache. The HTTP `POST` method shall be used for this purpose with a regular expression describing the media resource URLs to be purged provided in the body of the request. The message request body shall be encoded using the *application/x-www-form-urlencoded* MIME content type as a key–value pair, with the key being the string *pattern* and the value being the regular expression.

On receiving a purge request, the Media AF shall immediately invalidate all media resources in the Media AS cache matching the regular expression by declaring them as stale. A subsequent Media Client request at reference point M4 for a purged media resource will trigger the fetching (and possible caching) of the current version from the Media Application Provider's content origin via reference point M2 in case of a Pull-based ingest. For Push-based ingest, M4 requests for purged content shall be responded to with a *404 (Not Found)* HTTP response until such time as a new version of the object is published by the Media Application Provider to the Media AS via at reference point M2.

If the procedure is successful, the Media AF shall return one of the following response messages:

- *204 (No Content)* if no cache entries were purged, for example because no current cache entries matched the regular expression supplied in the original request. The response message body shall be empty in this case.
- *200 (OK)* if some cache entries were purged. The body of the response message shall indicate the total number of cache entries purged in all Media AS instances distributing the content.

The HTTP response *400 (Bad Request)* shall be returned in the case where the request message body – or the regular expression contained in it – are found by the Media AF to be syntactically malformed.

5.2.9 Content Publishing provisioning

5.2.9.1 General

These operations are used by the Media Application Provider at reference point M1 to provision the Content Publishing feature for uplink media delivery.

HTTP responses for successful and operation-specific failure cases are specified in the following clauses. For all other failure cases, an HTTP response indicating a response code in accordance with clause 7.1.6 shall be returned to the API client. In all failure cases a message body in accordance with clause 7.1.7 shall be included in the response message.

5.2.9.2 Create Content Publishing Configuration resource operation

This operation is used by the Media Application Provider at reference point M1 to activate the Content Publishing feature for a particular Provisioning Session. The Media Application Provider shall use the HTTP `POST` method for this purpose. The request URL shall be a well-known sub-resource of the Provisioning Session resource, as specified in clause 8.9.2. The HTTP request message body shall be a Content Publishing Configuration resource representation, as specified in clause 8.9.3.1. There is at most one Content Publishing Configuration at a time for a given Provisioning Session.

Regarding the configuration of content egest from the Media AS to the Media Application Provider at reference point M2:

- If the Content Publishing Configuration uses the push-based content egest method, i.e., the *egestConfiguration.mode* attribute is set to *PUSH*, then the *egestConfiguration.baseURL* property shall be nominated by the Media Application Provider in the request message body. The Media AF shall return the *egestConfiguration.baseURL* property value unchanged in its response message body.

- If the Content Publishing Configuration uses the pull-based content egest method, i.e., the *egestConfiguration.mode* attribute is set to *PULL*, then the *egestConfiguration.baseURL* property shall be nominated by the Media AF and returned in the response message body. It shall not be set by the Media Application Provider in the request message body.

Regarding the configuration(s) of content contribution by the Media Client to the Media AS at reference point M4:

- When more than one content contribution configuration is provided in the HTTP request message body, the operation to create the Content Publishing Configuration resource shall be successful if and only if all such contribution configurations are acceptable to the Media AF.

- In all cases, the *contributionConfiguration.canonicalDomainName* and *contributionConfiguration.baseURL* properties are read-only: they shall always be omitted from the creation request and shall be assigned by the Media AF, allowing their values to be inspected by the Media Application Provider in the returned Content Publishing Configuration resource representation, or by using the operation specified in clause 5.2.9.3 below.

- If the *contributionConfiguration.certificateId* property is present and valid, the Media AF shall assign a canonical domain name for the Media AS to expose at reference point M4 that matches the Common Name and the first Subject Alternative Name in the referenced Server Certificate resource (taking into account wildcard matching) regardless of whether the corresponding X.509 certificate was created using the operation specified in clause 5.2.4.2 or those specified in clauses 5.2.4.3 and 5.2.4.4.

- The Media Application Provider may nominate an alternative domain name to be advertised to the Media Client in the Service Access Information by setting the *contributionConfiguration.domainNameAlias* property when (and only when) creating the Content Publishing Configuration resource. If valid, the value of this property shall then appear in the *contributionConfiguration.baseURL* assigned by the Media AF instead of *contributionConfiguration.canonicalDomainName*. The Media Application Provider shall ensure that this domain name alias resolves to the canonical domain name of the Media AS notified by the Media AF in its response by means of suitable DNS configuration.

If the operation is successful, the Media AF shall return a *201 (Created)* HTTP response message and the request URL shall be returned as the value of the `Location` HTTP header field. The response message body shall be a representation of the current state of the Content Publishing Configuration resource (see clause 8.9.3.1), including any properties assigned by the Media AF.

If any resources referenced by the supplied Content Publishing Configuration resource representation are invalid, the create operation shall fail with an HTTP response status code of *400 (Bad Request)* and an error message body per clause 7.1.7. In this case, the Content Publishing Configuration resource shall remain in an uncreated state in the Media AF.

If *contributionConfiguration.domainNameAlias* is set in the supplied Content Hosting Configuration resource representation but its value is not a syntactically valid Fully-Qualified Domain Name or if the *contributionConfiguration.certificateId* property is absent or if the supplied domain name alias does match any of one of the Subject Alternative Names listed in the Server Certificate referenced by the *contributionConfiguration.certificateId* property, the create operation shall fail with an HTTP response status code of *400 (Bad Request)* and an error message body per

clause 7.1.7. In this case, the Content Publishing Configuration resource shall remain in an uncreated state in the Media AF.

NOTE: Even if multiple contribution configurations in the same Content Publishing Configuration reference the same Server Certificate resource, they may each nominate a different domain name alias from among its Subject Alternative Names.

Attempting to create a Content Publishing Configuration in the scope of a Provisioning Session of any type other than *MS_UPLINK* shall fail with an HTTP response status code of *403 (Forbidden)* and an error message body per clause 7.1.7. In this case, the Content Publishing Configuration resource shall remain in an uncreated state in the Media AF.

If the request is acceptable but the Media AF is unable to provision the resources required by the supplied Content Publishing Configuration, the create operation shall fail with an HTTP response status code of *500 (Internal Server Error)* and an error message body per clause 7.1.7. In this case, the Content Publishing Configuration resource shall remain in an uncreated state in the Media AF.

5.2.9.3 Retrieve Content Publishing Configuration resource operation

This operation is used by the Media Application Provider to retrieve the current state of an existing Content Publishing Configuration resource from the Media AF. The HTTP `GET` method shall be used for this purpose.

If the operation is successful, the Media AF shall return a *200 (OK)* response message that includes a representation of the target Content Publishing Configuration resource (see clause 8.9.3.1) in the response message body.

5.2.9.4 Update Content Publishing Configuration resource operation

This operation is invoked by the Media Application Provider to modify the properties of an existing Content Publishing Configuration resource. All writeable properties may be updated. The HTTP `PATCH` or HTTP `PUT` methods shall be used for this purpose.

When more than one content contribution configuration is provided in the HTTP request message body, the operation to create the Content Publishing Configuration resource shall be successful if and only if all such contribution configurations are acceptable to the Media AF.

If the HTTP request is acceptable but the operation results in no change to the resource representation, a *204 (No Content)* HTTP response message with an empty body should be returned.

If the operation is otherwise successful, the Media AF shall return a *200 (OK)* HTTP response message and shall provide a representation of the current state of the target resource in the message body to confirm successful update.

Attempts to modify read-only properties of the target Content Publishing Configuration resource, such as the canonical domain name of a contribution configuration, shall be rejected by the Media AF with a *403 (Forbidden)* HTTP response that includes an error message body per clause 7.1.7.

If the request is acceptable but the Media AF is unable to provision the resources required by the supplied Content Publishing Configuration, the update operation shall fail with an HTTP response status code of *500 (Internal Server Error)* and an error message body per clause 7.1.7. In this case, the Content Publishing Configuration resource in the Media AF shall remain in the state immediately prior to the update operation.

5.2.9.5 Destroy Content Publishing Configuration resource operation

This operation is used by the Media Application Provider to destroy a Content Publishing Configuration resource and to terminate the related egest of content. The HTTP `DELETE` method shall be used for this purpose. As a result, the Media AF shall release any associated network resources, purge any cached content, and delete any corresponding configurations.

If the procedure is successful, the Media AF shall return a *204 (No Content)* HTTP response message with an empty message body.

5.2.9.6 Purge Content Publishing cache operation

When pull-based content egest is provisioned in the Content Publishing Configuration, this operation is used by the Media Application Provider to purge content from the Media AS Content Publishing cache. The HTTP `POST` method shall be used for this purpose with a regular expression describing the media resource URLs to be purged provided in the body of the request. The message request body shall be encoded using the *application/x-www-form-urlencoded* MIME content type as a key–value pair, with the key being the string *pattern* and the value being the regular expression.

On receiving a purge request, the Media AF shall immediately invalidate all media resources in the Media AS cache matching the regular expression by declaring them as stale. Requests at reference point M2 for purged media resources should be responded to with a *410 (Gone)* HTTP response or else a *404 (Not Found)* response.

If the procedure is successful, the Media AF shall return one of the following response messages:

- *204 (No Content)* if no cache entries were purged, for example because no current cache entries matched the regular expression supplied in the original request. The response message body shall be empty in this case.
- *200 (OK)* if some cache entries were purged. The body of the response message shall indicate the total number of cache entries purged in all Media AS instances egesting the content.

The HTTP response *400 (Bad Request)* shall be returned in the case where the request message body – or the regular expression contained in it – are found by the Media AF to be syntactically malformed.

5.2.10 Real-time Media Communication provisioning

5.2.10.1 General

These operations are used by the Media Application Provider at reference point M1 to provision the configuration information for RTC-based media delivery sessions.

HTTP responses for successful and operation-specific failure cases are specified in the following clauses. For all other failure cases, an HTTP response indicating a response code in accordance with clause 7.1.6 shall be returned to the API client. In all failure cases, a message body in accordance with clause 7.1.7 shall be included in the response message.

5.2.10.2 Create Real-time Media Communication Configuration resource operation

This operation is used by the Media Application Provider at reference point M1 to activate the RTC feature for a particular Provisioning Session. The Media Application Provider shall use the HTTP `POST` method for this purpose. The request URL shall be a well-known sub-resource of the Provisioning Session resource, as specified in clause 8.10.2. The HTTP request message body shall be an RTC Configuration resource representation, as specified in clause 8.10.3.1. There is at most one RTC Configuration resource at a time for a given Provisioning Session.

The Media Application Provider may request that the Media Delivery System provides additional support services to facilitate communication between Media Clients wishing to engage in an RTC-based media delivery session:

- If the *enableStunService* flag is set to *true*, the Media AF shall configure the Media AS to provide a STUN service to Media Clients and the Media AF shall populate information about the endpoint(s) of this service in *stunServerEndpoints*. Otherwise, the Media AS is not required to provide a STUN service. Otherwise, the Media Application Provider may populate *stunServerEndpoints* with information about a STUN service it provides.
- If the *enableTurnService* flag is set to *true*, the Media AF shall configure the Media AS to provide a TURN service to Media Clients and the Media AF shall populate information about the endpoint(s) of this service in *turnServerEndpoints*. Otherwise, the Media AS is not required to provide a TURN service. Otherwise, the Media Application Provider may populate *turnServerEndpoints* with information about a TURN service it provides.
- If the *enableSwapService* flag is set to *true*, the Media AF shall configure the Media AS to provide a SWAP service to Media Clients and the Media AF shall populate information about the endpoint(s) of this service in *swapServerEndpoints*. Otherwise, the Media AS is not required to provide a SWAP service. Otherwise, the Media Application Provider may populate *swapServerEndpoints* with information about a SWAP service it provides.

If the operation is successful, the Media AF shall return a *201 (Created)* HTTP response message and the request URL shall be returned as the value of the `Location` HTTP header field. The response message body shall be a representation

of the current state of the RTC Configuration resource (see clause 8.10.3.1), including any properties assigned by the Media AF.

If the request is acceptable but the Media AF is unable to provision the resources required by the supplied RTC Configuration, the create operation shall fail with an HTTP response status code of *500 (Internal Server Error)* and an error message body per clause 7.1.7. In this case, the RTC Configuration resource shall remain in an uncreated state in the Media AF.

5.2.10.3 Retrieve Real-time Media Communication Configuration resource operation

This operation is used by the Media Application Provider to retrieve the current state of an existing RTC Configuration resource from the Media AF. The HTTP `GET` method shall be used for this purpose.

If the operation is successful, the Media AF shall return a *200 (OK)* response message that includes a representation of the target RTC Configuration resource (see clause 8.10.3.1) in the response message body.

5.2.10.4 Update Real-time Media Communication Configuration resource operation

This operation is invoked by the Media Application Provider to modify the properties of an existing RTC Configuration resource. All writeable properties may be updated. The HTTP `PATCH` or HTTP `PUT` methods shall be used for this purpose.

If the HTTP request is acceptable but the operation results in no change to the resource representation, a *204 (No Content)* HTTP response message with an empty body should be returned.

If the operation is otherwise successful, the Media AF shall return a *200 (OK)* HTTP response message and shall provide a representation of the current state of the target resource in the message body to confirm successful update.

Attempts to modify read-only properties of the target RTC Configuration resource, such as the STUN service endpoint information, shall be rejected by the Media AF with a *403 (Forbidden)* HTTP response that includes an error message body per clause 7.1.7.

If the request is acceptable but the Media AF is unable to provision the resources required by the supplied RTC Configuration, the update operation shall fail with an HTTP response status code of *500 (Internal Server Error)* and an error message body per clause 7.1.7. In this case the RTC Configuration resource in the Media AF shall remain in the state immediately prior to the update operation.

5.2.10.5 Destroy Real-time Media Communication Configuration resource operation

This operation is used by the Media Application Provider to destroy an RTC Configuration resource. The HTTP `DELETE` method shall be used for this purpose. As a result, the Media AF shall release any associated network resources and invalidate the configuration.

If the procedure is successful, the Media AF shall return a *204 (No Content)* HTTP response message with an empty message body.

5.2.11 Metrics Reporting provisioning

5.2.11.1 General

These operations are used by the Media Application Provider at reference point M1 to provision QoE metrics reporting functionality associated with downlink or uplink media delivery. The Media Application Provider may provision several Metrics Reporting Configurations within the scope of a Provisioning Session with different properties which determine whether and how often the Media Session Handler submits QoE metrics reports to the Media AF as well as the format and contents of these reports. To this end, each Metrics Reporting Configuration shall specify a *metrics scheme*, which may be specified by 3GPP or by another party. The chosen metrics scheme URI is indicated in the *scheme* property of the Metrics Reporting Configuration. This clause defines the basic operations; more details are provided in clause 8.11.

A given Metrics Reporting Configuration is uniquely identified within the scope of its parent Provisioning Session by the *metricsReportingConfigurationId* property of the corresponding Metrics Reporting Configuration resource, as specified in clause 8.11.3.1.

Where metrics reporting is not required for the entire duration of a media delivery session, *reportingStartOffset* and/or *reportingDuration* parameters may additionally be specified for a Metrics Reporting Configuration indicating the portion of each media delivery session for which metrics reports are to be submitted by the Media Session Handler.

HTTP responses for successful and operation-specific failure cases are specified in the following clauses. For all other failure cases, an HTTP response indicating a response code in accordance with clause 7.1.6 shall be returned to the API client. In all failure cases a message body in accordance with clause 7.1.7 shall be included in the response message.

5.2.11.2 Create Metrics Reporting Configuration resource operation

This operation is used by the Media Application Provider to create a Metrics Reporting Configuration resource within the scope of an existing Provisioning Session. The Media Application Provider shall use the HTTP `POST` method for this purpose. The request URL shall be a well-known sub-resource of the Provisioning Session resource representing its Metrics Reporting Configurations resource collection, as specified in clause 8.11.2. The HTTP request message body shall be a Metrics Reporting Configuration resource representation, as specified in clause 8.11.3.1.

Upon successful creation of the resource, the Media AF shall return a *201 (Created)* response message and the resource URL for the newly-created Metrics Reporting Configuration resource, including its resource identifier, shall be returned as the value of the `Location` HTTP header field. The response message body shall be a representation of the current state of the Metrics Reporting Configuration resource (see clause 8.11.3.1), including any properties assigned by the Media AF.

If the metrics scheme or any of the metrics cited in the supplied Metrics Reporting Configuration is not supported by the Media AF, the create operation shall fail with an HTTP response status code of *400 (Bad Request)* and an error message body per clause 7.1.7. In this case, the Metrics Reporting Configuration resource shall remain in an uncreated state in the Media AF.

If the request is acceptable but the Media AF is unable to provision the resources required by the supplied Metrics Reporting Configuration, the create operation shall fail with an HTTP response status code of *500 (Internal Server Error)* and an error message body per clause 7.1.7. In this case, the Metrics Reporting Configuration resource shall remain in an uncreated state in the Media AF.

This operation may be performed multiple times by a Media Application Provider to provision different Metrics Reporting Configuration resources within the scope of a Provisioning Session. Each such resource is assigned a different Metrics Reporting Configuration identifier by the Media AF.

5.2.11.3 Retrieve Metrics Reporting Configuration resource operation

This operation is used by the Media Application Provider to retrieve the current state of an existing Metrics Reporting Configuration resource in the Media AF. The HTTP `GET` method shall be used for this purpose, citing the resource identifier of the target Metrics Reporting Configuration in the request URL.

If successful, the Media AF shall return a *200 (OK)* HTTP response message that includes a representation of the target Metrics Reporting Configuration resource (see clause 8.11.3.1) in the response message body.

5.2.11.4 Update Metrics Reporting Configuration resource operation

This operation is invoked by the Media Application Provider to entirely replace or modify certain properties of an existing Metrics Reporting Configuration resource. All available properties may be updated. The HTTP `PATCH` or HTTP `PUT` methods shall be used for this purpose. The replacement Metric Reporting Configuration resource representation shall be provided in the body of the HTTP request message.

If the HTTP request is acceptable but the operation results in no change to the resource representation, a *204 (No Content)* HTTP response message with an empty body should be returned.

If the operation is otherwise successful, the Media AF shall return a *200 (OK)* HTTP response message that includes a representation of the current state of the target resource in the message body to confirm successful update.

If the metrics scheme or any of the metrics cited in the supplied Metrics Reporting Configuration is not supported by the Media AF, the update operation shall fail with an HTTP response status code of *400 (Bad Request)* and an error message body per clause 7.1.7. In this case, the Metrics Reporting Configuration resource in the Media AF shall remain in the state immediately prior to the update operation.

Attempts to modify read-only properties of the target Metrics Reporting Configuration resource shall be rejected by the Media AF with a *403 (Forbidden)* HTTP response that includes an error message body per clause 7.1.7.

If the request is acceptable but the Media AF is unable to provision the resources required by the supplied Metrics Reporting Configuration, the update operation shall fail with an HTTP response status code of *500 (Internal Server Error)* and an error message body per clause 7.1.7. In this case, the Metrics Reporting Configuration resource in the Media AF shall remain in the state immediately prior to the update operation.

5.2.11.5 Destroy Metrics Reporting Configuration resource operation

This operation is used by the Media Application Provider to destroy a Metrics Reporting Configuration resource and to terminate the related metrics reporting procedure by Media Clients. The HTTP `DELETE` method shall be used for this purpose, citing the resource identifier of the target Metrics Reporting Configuration in the request URL. As a result, the Media AF shall release any associated resources, discard any pending metrics reports, and remove any corresponding configurations.

If the operation is successful, the Media AF shall return a *204 (No Content)* HTTP response message with an empty message body.

Any subsequent operations citing the resource identifier of a destroyed Metrics Reporting Configuration should result in a *410 (Gone)* or else a *404 (Not Found)* HTTP response message that includes an error message body per clause 7.1.7.

5.2.12 Consumption Reporting provisioning

5.2.12.1 General

These operations are used by the Media Application Provider at reference point M1 to activate and to configure consumption reporting functionality associated with downlink media delivery. The Media Application Provider may provision a single Consumption Reporting Configuration within the scope of a Provisioning Session which determines whether and how often the Media Session Handler submits consumption reports to the Media AF. This clause defines the basic operations. More details are provided in clause 8.12.

HTTP responses for successful and operation-specific failure cases are specified in the following clauses. For all other failure cases, an HTTP response indicating a response code in accordance with clause 7.1.6 shall be returned to the API client. In all failure cases a message body in accordance with clause 7.1.7 shall be included in the response message.

5.2.12.2 Create Consumption Reporting Configuration resource operation

This operation is used by the Media Application Provider to activate the Consumption Reporting feature for a particular Provisioning Session. The Media Application Provider shall use the HTTP `POST` method for this purpose. The request URL shall be a well-known sub-resource of the Provisioning Session resource, as specified in clause 8.12.2. The HTTP request message body shall be a Consumption Reporting Configuration resource representation, as specified in clause 8.12.3.1. There is at most one Consumption Reporting Configuration at a time for a given Provisioning Session.

Upon successful creation of the resource, the Media AF shall return a *201 (Created)* response message and the request URL shall be returned as the value of the `Location` HTTP header field. The response message body shall be a representation of the current state of the Consumption Reporting Configuration resource (see clause 8.12.3.1), including any properties assigned by the Media AF.

If the supplied Consumption Reporting Configuration is not supported by the Media AF, the create operation shall fail with an HTTP response status code of *400 (Bad Request)* and an error message body per clause 7.1.7. In this case, the Consumption Reporting Configuration resource shall remain in an uncreated state in the Media AF.

If the request is acceptable but the Media AF is unable to provision the resources required by the supplied Consumption Reporting Configuration, the create operation shall fail with an HTTP response status code of *500 (Internal Server Error)* and an error message body per clause 7.1.7. In this case, the Consumption Reporting Configuration resource shall remain in an uncreated state in the Media AF.

5.2.12.3 Retrieve Consumption Reporting Configuration resource operation

This operation is used by the Media Application Provider to obtain the current Consumption Reporting Configuration from the Media AF. The HTTP `GET` method shall be used for this purpose.

If successful, the Media AF shall return a *200 (OK)* HTTP response message that includes a representation of the target Consumption Reporting Configuration resource (see clause 8.12.3.1) in the response message body.

5.2.12.4 Update Consumption Reporting Configuration resource operation

This operation is invoked by the Media Application Provider to modify the current Consumption Reporting Configuration. All available parameters may be updated. The HTTP `PATCH` or HTTP `PUT` methods shall be used for this purpose. The replacement Consumption Reporting Configuration resource representation shall be provided in the body of the HTTP request message.

If the HTTP request is acceptable but the operation results in no change to the resource representation, a *204 (No Content)* HTTP response message with an empty body should be returned.

If the operation is otherwise successful, the Media AF shall return a *200 (OK)* HTTP response message that includes a representation of the current state of the Consumption Reporting Configuration resources to confirm successful update.

If the supplied Consumption Reporting Configuration is not acceptable to the Media AF, the update operation shall fail with an HTTP response status code of *400 (Bad Request)* and an error message body per clause 7.1.7. In this case, the Consumption Reporting Configuration resource shall remain in the state immediately prior to the update operation.

Attempts to modify read-only properties of the target Consumption Reporting Configuration resource shall be rejected by the Media AF with a *403 (Forbidden)* HTTP response that includes an error message body per clause 7.1.7.

If the request is acceptable but the Media AF is unable to provision the resources required by the supplied Consumption Reporting Configuration, the update operation shall fail with an HTTP response status code of *500 (Internal Server Error)* and an error message body per clause 7.1.7. In this case, the Consumption Reporting Configuration resource in the Media AF shall remain in the state immediately prior to the update operation.

5.2.12.5 Destroy Consumption Reporting Configuration resource operation

This operation is used by the Media Application Provider to destroy the Consumption Reporting Configuration resource and to terminate the related consumption reporting procedure by Media Clients. The HTTP `DELETE` method shall be used for this purpose, citing the well-known sub-resource of the target Provisioning Session in the request URL. As a result, the Media AF shall release any associated resources, purge any cached data, and remove any corresponding configurations.

If the operation is successful, the Media AF shall return a *204 (No Content)* HTTP response message with an empty message body.

5.2.13 Event Data Processing provisioning

5.2.13.1 General

These operations are used by the Media Application Provider at reference point M1 to configure the collection and processing of UE data related to media delivery (as defined in TS 26.531 [46]) and to restrict its exposure over reference points R5 and R6 by configuring the Data Collection AF instantiated in the Media AF (such as that defined in clause 4.7 of TS 26.501 [4]) with one or more Event Data Processing Configurations and Data Access Profiles for a particular Event ID. The Media Application Provider may provision several Event Data Processing Configurations within the scope of a Provisioning Session with different properties.

Each instance of a Data Access Profile specifies a set of data processing operations to be performed by the Data Collection AF on its collected UE data in order to synthesize the event data to be exposed to a specific Event service consumer entity. In this release, eligible Event service consumer entities of Media Delivery event data are the NWDAF, the Event Consumer AF and the NEF.

The Event Data Processing Provisioning API is specified in clause 8.13.

HTTP responses for successful and operation-specific failure cases are specified in the following clauses. For all other failure cases, an HTTP response indicating a response code in accordance with clause 7.1.6 shall be returned to the API client. In all failure cases a message body in accordance with clause 7.1.7 shall be included in the response message.

5.2.13.2 Create Event Data Processing Configuration resource operation

This operation is used by the Media Application Provider to create a new Event Data Processing Configuration within the scope of an existing Provisioning Session in the form of one or more Data Access Profiles. The HTTP `POST` method shall be used for this purpose. The request URL shall be a well-known sub-resource of the Provisioning Session resource representing its Event Data Processing Configurations resource collection, as specified in clause 8.13.2. The request message body shall be an Event Data Processing Configuration resource representation, as specified in clause 8.13.3.1.

If the operation is successful, the Media AF shall generate a resource identifier representing the new Event Data Processing Configuration. In this case, the Media AF shall return a *201 (Created)* response message and the URL of the newly created resource, including its resource identifier, shall be provided as the value of the `Location` HTTP header field. The response message body shall be a representation of the current state of the Event Data Processing Configuration resource (see clause 8.13.3.1), including any properties assigned by the Media AF.

If the event identified in the supplied Event Data Processing Configuration is not supported by the Media AF, or if any of the data access profiles are unacceptable, the create operation shall fail with an HTTP response status code of *400 (Bad Request)* and an error message body per clause 7.1.7. In this case, the Event Data Processing Configuration resource shall remain in an uncreated state in the Media AF.

If the Data Collection AF is not instantiated in the Media AF, the create operation shall fail with an HTTP response status code of *404 (Not Found)* and an error message body per clause 7.1.7. In this case, the Event Data Processing Configuration resource shall remain in an uncreated state in the Media AF.

If the request is acceptable but the Media AF is unable to provision the resources required by the supplied Event Data Processing Configuration, the create operation shall fail with an HTTP response status code of *500 (Internal Server Error)* and an error message body per clause 7.1.7. In this case, the Event Data Processing Configuration resource shall remain in an uncreated state in the Media AF.

This operation may be performed multiple times by a Media Application Provider to provision different Event Data Processing Configuration resources within the scope of a Provisioning Session. Each such resource is assigned a different Event Data Processing Configuration identifier by the Media AF.

5.2.13.3 Retrieve Event Data Processing Configuration resource operation

This operation is used by the Media Application Provider to retrieve the current state of an existing Event Data Processing Configuration resource in the Media AF. The HTTP `GET` method shall be used for this purpose, citing the resource identifier of the target Event Data Processing Configuration in the request URL.

If successful, the Media AF shall return a *200 (OK)* HTTP response message that includes a representation of the target Event Data Processing Configuration resource (see clause 8.13.3.1) in the response message body.

5.2.13.4 Update Event Data Processing Configuration resource operation

This operation is invoked by the Media Application Provider to entirely replace or modify certain properties of an existing Event Data Processing Configuration resource. All available properties may be updated. The HTTP `PATCH` or HTTP `PUT` methods shall be used for this purpose. The replacement Event Data Processing Configuration resource representation shall be provided in the body of the HTTP request message.

If the HTTP request is acceptable but the operation results in no change to the resource representation, a *204 (No Content)* HTTP response message with an empty body should be returned.

If the operation is otherwise successful, the Media AF shall return a *200 (OK)* HTTP response message that includes a representation of the current state of the target resource in the message body to confirm successful update.

If the event identified in the supplied Event Data Processing Configuration is not supported by the Media AF, or if any of the data access profiles are unacceptable, the update operation shall fail with an HTTP response status code of *400*

(Bad Request) and an error message body per clause 7.1.7. In this case, the Event Data Processing Configuration resource in the Media AF shall remain in the state immediately prior to the update operation.

Attempts to modify read-only properties of the target Event Data Processing Configuration resource shall be rejected by the Media AF with a *403 (Forbidden)* HTTP response that includes an error message body per clause 7.1.7.

If the request is acceptable but the Media AF is unable to provision the resources required by the supplied Event Data Processing Configuration, the update operation shall fail with an HTTP response status code of *500 (Internal Server Error)* and an error message body per clause 7.1.7. In this case, the Event Data Processing Configuration resource in the Media AF shall remain in the state immediately prior to the update operation.

5.2.13.5 Destroy Event Data Processing Configuration resource operation

This operation is used by the Media Application Provider to destroy an existing Event Data Processing Configuration resource and to terminate the related collection of UE data and exposure of events to event consumer subscribers. The HTTP `DELETE` method shall be used for this purpose, citing the resource identifier of the target Event Data Processing Configuration in the request URL. As a result, the Data Collection AF shall process any reported UE data still outstanding and shall delete any corresponding data collection and reporting client configurations as well as any event subscriptions.

If the operation is successful, the Media AF shall return a *204 (No Content)* HTTP response message with an empty message body.

Any subsequent operations citing the resource identifier of a destroyed Event Data Processing Configuration should result in a *410 (Gone)* or else a *404 (Not Found)* HTTP response message that includes an error message body per clause 7.1.7.

5.3 Network media session handling (M3, M5) interactions

5.3.1 Overview

This clause specifies the set of operations used by the Media Session Handler within a Media Client to invoke services on the Media AF at reference point M5 relating to downlink or uplink media delivery. A subset of these operations is also exposed by the Media AF to the Media AS at reference point M3.

5.3.2 Service Access Information acquisition

5.3.2.1 General

Service Access Information is the set of parameters and addresses needed by the Media Client to activate reception of a downlink media delivery session, to activate an uplink media delivery session for content contribution or to obtain configuration parameters to initiate real-time media communication (RTC).

The Media Session Handler may obtain Service Access Information in one of two ways:

1. From the Media-aware Application via reference point M6. In this case, the Service Access Information is initially acquired by the Media-aware Application from the Media Application Provider via reference point M8 and the Media-aware Application shall pass the parameters to the Media Session Handler using one of the session launch mechanisms specified in clause 10.2.

2. From the Media AF via reference point M5. In this case, the Service Access Information is derived by the Media AF from a Provisioning Session established at reference point M1 and the Media AF exposes this to the Media Session Handler using the operations specified in this clause. At the start of a media delivery session, a minimal set of baseline Service Access Information parameters is passed to the Media Session Handling using one of the session launch mechanisms specified in clause 10.2 and this causes it to fetch the full Service Access Information from the Media AF using the procedure specified in clause 5.3.2.3.

The data model of the Service Access Information resource acquired by the Media Session Handler of the Media Client is specified in clause 9.2.3. The Service Access Information typically includes:

- For downlink media streaming according to TS 26.512 [6], a set of Media Entry Points that can be consumed by the Media Access Function. One of these is selected by the Media Session Handler or by the Media-aware Application and is handed to the Media Access Function via reference point M11 or M7 respectively.
- For uplink media according to TS 26.512 [6], a description of an entry point for the publishing of the uplink streaming content.
- For RTC according to TS 26.113 [7] specifies a configuration for the Media Client to assist in establishing interactive connectivity with other RTC session participants.

Service Access Information additionally includes configuration information to allow the Media Session Handler to invoke procedures for dynamic policy (see clause 5.3.3), network assistance (clause 5.3.4), QoE metrics reporting (clause 5.3.5) and consumption reporting (clause 5.3.6).

If an Edge Resources Configuration with client-driven management (*EM_CLIENT_DRIVEN*) is provisioned in the applicable Provisioning Session (see clause 5.2.6), the Media AF shall convey a Client Edge Resources Configuration to the Media Session Handler as part of the Service Access Information it provides at reference point M5.

HTTP responses for successful and operation-specific failure cases are specified in the following clauses. For all other failure cases, an HTTP response indicating a response code in accordance with clause 7.1.6 shall be returned to the API client. In all failure cases a message body in accordance with clause 7.1.7 shall be included in the response message.

5.3.2.2 Create Service Access Information resource operation

The Create operation is not permitted for the Service Access Information resource collection. Any usage of the HTTP `POST` method shall result in the HTTP response *405 (Method Not Allowed)*.

5.3.2.3 Retrieve Service Access Information resource operation

This operation shall be used by the Media Session Handler to acquire Service Access Information from the Media AF. The Media Session Handler shall use the `GET` method for this purpose, citing the external service identifier associated with the target Provisioning Session (see clause 5.2.3) in the request URL. The request message body shall be empty.

If successful, the Media AF shall reply with a *200 (OK)* HTTP response message that includes a representation of the Service Access Information associated with the target media streaming session resource in the response message body, along with HTTP response headers in line with clause 7.1.3.2.

Once it has obtained an initial set of Service Access Information, the Media Session Handler shall periodically check for updated Service Access Information by issuing a conditional HTTP `GET` request in line with clause 7.1.3.2. The periodicity of polling for updated Service Access Information shall be guided by the value of the `Expires` and/or `Cache-control: max-age` headers that shall be included along with every response message for this operation.

The Media AF instance may nominate an MQTT [50] endpoint URL in the *notificationURL* property of the Service Access Information resource representation it returns to the Media Session Handler. If this property is present, the Media Session Handler shall connect to the MQTT channel at the indicated endpoint and subscribe to the root topic as described in clause 10.2.2. The usage and message formats for MQTT are described in clause 10.2.3. The Media AF shall notify the Media Session Handler about updates to the Service Access Information resource corresponding to the root topic of the MQTT channel.

5.3.2.4 Update Service Access Information resource operation

The Update operation is not permitted for members of the Service Access Information resource collection. Any usage of the HTTP `PUT` or `PATCH` methods in relation to the URLs of its members shall result in the HTTP response *405 (Method Not Allowed)*.

5.3.2.5 Destroy Service Access Information resource operation

The Destroy operation is not permitted for members of the Service Access Information resource collection. Any usage of the HTTP `DELETE` method in relation to the URLs of its members shall result in the *405 (Method Not Allowed)* HTTP response.

5.3.3 Dynamic Policy invocation

5.3.3.1 Procedures

To take advantage of the Dynamic Policy feature of the Media Delivery System, a Media Session Handler instantiates a Policy Template that was previously provisioned within the scope of a Provisioning Session using the operations specified in clause 5.2.7. The parameters in the Policy Template are used by the Media AF in combination with a dynamic QoS specification supplied by the Media Session Handler to request specific QoS and/or charging policies from the PCF (either directly or via the NEF, as specified in clause 5.5.3) for that media delivery session.

The following procedures are followed by a Media Session Handler to manage Dynamic Policy Instance resources in the Media AF via reference point M5. Instantiating a Policy Template as a dynamic policy requires a Policy Template identifier (provided in Service Access Information that is either retrieved from the Media AF using the operation specified in clause 5.3.2.3 or else supplied via reference point M6), a set of Service Data Flow description(s), an optional dynamic QoS specification and potentially other parameters defined in clause 5.7 of TS 26.501 [4].

- The Policy Template identifier identifies the desired Policy Template (as previously provisioned per clause 5.2.7.3) to be applied to the specified application flow(s). A Policy Template includes properties such as specific QoS (e.g. background data) or different charging treatments.

- The Media AF combines the information from the Policy Template with dynamic QoS specification supplied by the Media Session Handler and uses this complete set of parameters to invoke the PCF according to clause 5.5.3.

- The set of Service Data Flow description(s) allow the identification and classification by the 5G System of the application traffic involved in a media delivery session. These take the form of an IP packet filter set (as defined in clause 5.7.6 of [2]) or the Fully-Qualified Domain Name (FQDN) of a Media AS at reference point M4.

- The Dynamic Policy Instance may specify a target network slice and Data Network Name.

NOTE: It is not defined in this release how a Media AF in an external Data Network selects a specific DNN or S-NSSAI.

The application flow specifications for Dynamic Policy Instances relating to concurrent media delivery sessions at the same Media Client shall be non-overlapping. The Media AF is responsible for enforcing these constraints.

HTTP responses for successful and operation-specific failure cases are specified in the following clauses. For all other failure cases, an HTTP response indicating a response code in accordance with clause 7.1.6 shall be returned to the API client. In all failure cases a message body in accordance with clause 7.1.7 shall be included in the response message.

5.3.3.2 Create Dynamic Policy Instance resource operation

In order to instantiate a new dynamic policy, the Media Session Handler shall first create a resource for the Dynamic Policy Instance in the Media AF. The Media Session Handler shall use the HTTP `POST` message for this purpose. The body of the HTTP `POST` message shall be a Dynamic Policy Instance resource representation that includes a Provisioning Session identifier, the resource identifier of the target Policy Template and a set of Service Data Flow descriptions identifying the application flow(s) to be policed.

1. The *provisioningSessionId* property associates the Dynamic Policy Instance resource with a Provisioning Session.

2. The *policyTemplateId* property uniquely identifies the Policy Template on which the Dynamic Policy Instance is based.

3. For each application flow to be managed by the Dynamic Policy Instance resource, an instance of the *Application FlowBinding* object shall be present in the *appplicationFlowBindings* array. The *applicationFlowDescription* property of of this object shall be populated by the Media Session Handler and shall declare a Service Data Flow template according to TS 23.503 [17] that describes application flow in question. Exactly one of the following filtering specifications shall be populated in the *ApplicationFlowDescription* object to identify traffic belonging to a media delivery application flow:

   - a *packetFilter* object (including 5-tuples, Type of Service, Security Parameter Index, etc.). A Media Client shall not attempt to instantiate more than one Dynamic Policy Instance at the same time that cites the same set of packet filters.

   - a *domainName* populated with the fully-qualified Internet domain name of a Media AS at reference point M4. A Media Client shall not attempt to instantiate more than one Dynamic Policy Instance at the same time that cites the same *domainName.*

   In addition, the top-level media type of the application flow may be declared in the *mediaType* property.

   When the policy binding for the chosen Policy Template indicates that PDU Set marking is enabled (i.e., the *pduSetMarking* flag is set to *true* in Service Access Information), the Media Session Handler shall also populate the *mediaTransportParameters* property with the media transport protocol parameters to be used by the Media Access Function on the application flow in question to label uplink PDUs belonging to the same PDU Set and/or to indicate the last PDU in each PDU Set and/or to indicate the end of a data burst comprising one or more PDU Sets.

   When the policy binding for the chosen Policy Template indicates that PDU Set marking is enabled (i.e., the *pduSetMarking* flag is set to *true* in Service Access Information), the Media Session Handler shall also populate the *mediaTransportParameters* property with the media transport protocol parameters to be used by the Media AS on the application flow in question to label downlink PDUs belonging to the same PDU Set and/or to indicate the last PDU in each PDU Set and/or to indicate the end of a data burst comprising one or more PDU Sets.

4. When the Media Session Handler attempts to activate a QoS-related Policy Template, the *qosSpecification* property shall also be present in the *ApplicationFlowBinding* object containing the following properties specified in clause 7.3.3.6 to describe the QoS requirements of the media application flows described by the bound *applicationFlowDescription* property:

   - *downlinkBitRates* shall indicate the maximum requested bit rate, minimum desired bit rate and minimum requested bit rate in the downlink direction.

   - *uplinkBitRates* shall indicate the maximum requested bit rate, minimum desired bit rate and minimum requested bit rate in the uplink direction.

   - *desiredPacketLatency* may indicate the desired packet latency in both the downlink and uplink directions.

   - *desiredPacketLossRate* may indicate the desired packet loss rate in both the downlink and uplink directions.

   - *desiredDownlinkPduSetQosParameters* may be populated to indicate the desired delay budget and error rate for PDU Sets in the downlink direction, as well as indicating whether the loss of a single PDU in a PDU Set is significant for the receiving application.

   - *desiredUplinkPduSetQosParameters* may be populated to indicate the desired delay budget and error rate for PDU Sets in the uplink direction, as well as indicating whether the loss of a single PDU in a PDU Set is significant for the receiving application.

5. When the Media Session Handler instantiates a Policy Template that is provisioned with a Background Data Transfer (BDT) specification per clause 5.2.7.1, the *bdtSpecification* property shall be present and it shall contain the following properties:

   - *estimatedDataTransferVolume*, indicating the data volume that the Media Client estimates it will use during the current Background Data Transfer time window.

   - The *windows* property indicates time windows over which Background Data Transfers are offered to the Media Session Handler.

- The *maximimumDownlinkBitRate* and *maximimumUplinkBitRate* properties indicate the maximum bit rate for Background Data Transfers in the downlink and uplink directions respectively that the Media Session Handler is bidding for. In response, the Media AF populates these properties with the maximum permitted bit rate for Background Data Transfers in the downlink and uplink directions respectively when the dynamic policy is in force.

6. When the 5G System employs a traffic enforcement function to ensure that traffic complies with the policy described by the *qosSpecification* property, the Media AF shall explicitly indicate this in the Dynamic Policy resource representation by setting the *qosEnforcement* property to *true*.

If the operation is successful, the Media AF shall create a new Dynamic Policy Instance resource. In this case, the Media AF shall return a *201 (Created)* HTTP response message to the Media Session Handler, and the URL of the newly created Dynamic Policy Instance resource, including its resource identifier, shall be provided as the value of the `Location` HTTP header field. The response message body shall be a representation of the current state of the Dynamic Policy Instance resource (see clause 9.3.3.1), including any properties assigned by the Media AF.

Upon successful creation of the Dynamic Policy Instance resource, notifications of updates to the resource may be notified asynchronously to the Media Session Handler:

- If the *notificationURL* property is present in the Service Access Information, the Media Session Handler shall subscribe to the MQTT sub-topic corresponding to the *resourceId* of the Dynamic Policy Instance and shall expect to receive asynchronous notifications published by the Media AF on the MQTT notification channel of type NOTIFICATION_DYNAMIC_POLICY_INSTANCE concerning changes to the Dynamic Policy Instance, including details about new Background Data Transfer opportunities.

- The Media AF shall use the MQTT notification channel signalled in the Service Access Information (if any, see clause 5.3.2.3) to notify the Media Session Handler subscriber about updates to the Dynamic Policy Instance resource. A notification message of type *NOTIFICATION_DYNAMIC_POLICY_INSTANCE* shall be published to the MQTT sub-topic corresponding to the *resourceId* of the Dynamic Policy Instance.

The usage and message formats for the MQTT notification channel are specified in clause 10.2.

When the Dynamic Policy Instance is successfully instantiated, the Media AF triggers the creation of a corresponding PCC rule in the 5G System according to clause 5.5.3 to enforce the required QoS and/or charging policy on the specified application flow(s). Depending on the *ApplicationFlowDescription* objects in the received Dynamic Policy Instance resource representation and the *filterMethod* indicated by each one, the Media AF shall populate for each one a *flowDescription* object and/or provide an Application Identifier referring to a *PFD* (Packet Flow Description) object containing the domain name of a Media AS instance.

NOTE: When the Media AF is deployed in an external Data Network, it is the responsibility of the NEF to map any external Application Identifier supplied by the Media AF into an internal Application Identifier that is known to the PCF.

If the supplied Dynamic Policy Instance is not acceptable to the Media AF, the create operation shall fail with an HTTP response status code of *400 (Bad Request)* and an error message body per clause 7.1.7. In this case, the Dynamic Policy Instance resource shall remain in an uncreated state in the Media AF.

If the request is acceptable but the Media AF forbids the instantiation of the referenced Policy Template, for example because the quota for Background Data Transfers has been exceeded or because the UE is not permitted in the charging specification, the create operation shall fail with an HTTP response status code of *403 (Forbidden)* and an error message body per clause 7.1.7. In this case, the Dynamic Policy Instance resource shall remain in an uncreated state in the Media AF.

If the request is acceptable but the Media AF is unable to provision the resources required by the supplied Dynamic Policy Instance, the create operation shall fail with an HTTP response status code of *500 (Internal Server Error)* and an error message body per clause 7.1.7. In this case, the Dynamic Policy Instance resource shall remain in an uncreated state in the Media AF.

If the Media Session Handler needs to instantiate several dynamic policies, it may invoke this operation as often as needed.

5.3.3.3 Retrieve Dynamic Policy Instance resource operation

This operation is used by the Media Session Handler to retrieve the current state of an existing Dynamic Policy Instance resource in the Media AF. HTTP `GET` method shall be used for this purpose, citing the resource identifier of the target Dynamic Policy Instance in the request URL.

If successful, the Media AF shall return a *200 (OK)* HTTP response message that includes a representation of the target Dynamic Policy Instance resource (see clause 9.3.3.1) in the response message body.

5.3.3.4 Update Dynamic Policy Instance resource operation

This operation is invoked by the Media Session Handler to entirely replace or modify certain properties of an existing Dynamic Policy resource. All available properties may be updated. The HTTP `PATCH` or HTTP `PUT` methods shall be used for this purpose, citing the resource identifier of an existing Dynamic Policy Instance in the request URL. The replacement Dynamic Policy Instance resource representation shall be provided in the body of the HTTP request message.

If all required information is set in the replacement Dynamic Policy Instance, the Media AF shall trigger the appropriate actions towards other Network Functions in the 5G System according to clause 5.5.3 to update the associated PCC rule in line with the modified QoS and charging policy.

If the HTTP request is acceptable but the operation results in no change to the resource representation, a *204 (No Content)* HTTP response message with an empty body should be returned.

If the operation is otherwise successful, the Media AF shall return a *200 (OK)* HTTP response message that includes a representation of the current state of the target resource in the message body to confirm successful update.

If the supplied Dynamic Policy Instance is not acceptable to the Media AF, the update operation shall fail with an HTTP response status code of *400 (Bad Request)* and an error message body per clause 7.1.7. In this case, the Dynamic Policy Instance resource shall remain in the state immediately prior to the update operation.

Attempts to modify read-only properties of the target Dynamic Policy Instance resource shall be rejected by the Media AF with a *403 (Forbidden)* HTTP response that includes an error message body per clause 7.1.7.

If the request is acceptable but the Media AF forbids the instantiation of the referenced Policy Template, for example because the UE is not permitted in the charging specification, the update operation shall fail with an HTTP response status code of *403 (Forbidden)* and an error message body per clause 7.1.7. In this case, the Dynamic Policy Instance resource shall remain in the state immediately prior to the update operation.

If the request is acceptable but the Media AF is unable to provision the resources required by the supplied Dynamic Policy Instance, the update operation shall fail with an HTTP response status code of *500 (Internal Server Error)* and an error message body per clause 7.1.7. In this case, the Dynamic Policy Instance resource shall remain in the state immediately prior to the update operation.

5.3.3.5 Destroy Dynamic Policy Instance resource operation

This operation is invoked by the Media Session Handler to destroy an existing Dynamic Policy Instance resource. The HTTP `DELETE` method shall be used for this purpose, citing the resource identifier of the target Dynamic Policy Instance in the request URL. As a result, the Media AF shall trigger the appropriate actions towards other Network Functions in the 5G System according to clause 11.2.x to remove the associated PCC rule and to revert the affected application flow(s) to a default QoS and charging policy.

If the operation is successful, the Media AF shall return a *204 (No Content)* HTTP response message with an empty message body.

No further MQTT notification messages shall be published by the Media AF on the sub-topic corresponding to the resource identifier of the destroyed resource. The Media Session Handler shall unsubscribe from the sub-topic corresponding to this resource identifier.

Any subsequent operations citing the resource identifier of a destroyed Dynamic Policy Instance should result in a *410 (Gone)* or else a *404 (Not Found)* HTTP response message that includes an error message body per clause 7.1.7.

5.3.4 Network Assistance invocation

5.3.4.1 Procedures

The following procedures are followed by the Media Session Handler to request Network Assistance from one of the Media AF instances listed in the *serverAddresses* property of the Network Assistance Configuration which is part of the Service Access Information that is either retrieved from the Media AF using the operation specified in clause 5.3.2.3 or else supplied via reference point M6.

1. The Media Client first creates a Network Assistance Session with its chosen Media AF instance. It provides information that will later be used by the Media AF to request a particular network QoS to be applied by the PCF to one or more application data flows, and to recommend a bit rate to the Media Client. The Media AF assigns a resource identifier to the Network Assistance Session at the point of creation. This procedure is further specified in clause 5.3.4.2.

2. The Network Assistance Session resource may be retrieved by the Media Session Handler using the procedure specified in clause 5.3.4.3.

3. At any time after the Network Assistance Session resource is created, the Media Client may use the Network Assistance Session resource identifier to explicitly request a bit rate recommendation by invoking a remote procedure call provided for this purpose by the Media AF. This procedure is further specified in clause 5.3.4.4.

4. Using the Network Assistance Session resource identifier, the Media Client may also request a delivery boost to be provided by the 5G System at any time by invoking a remote procedure call provided for this purpose by the Media AF. This procedure is further specified in clause 5.3.4.5.

5. The information provided when first creating a Network Assistance Session may be modified subsequently by the Media Session Handler using the session modification operation specified in clause 5.3.4.6.

6. In order to terminate a Network Assistance Session, the Media Client destroys the Network Assistance Session resource using the procedure specified in clause 5.3.4.7.

Details of the APIs supporting these procedures at reference point M5 are specified in clause 9.4.

HTTP responses for successful and operation-specific failure cases are specified in the following clauses. For all other failure cases, an HTTP response indicating a response code in accordance with clause 7.1.6 shall be returned to the API client. In all failure cases a message body in accordance with clause 7.1.7 shall be included in the response message.

5.3.4.2 Create Network Assistance Session resource operation

This operation is used by the Media Session Handler to create a Network Assistance Session in the Media AF. The `POST` HTTP method shall be used for this purpose and the request message body shall be a Network Assistance Session resource representation as specified in clause 9.4.3.1.

1. The *provisioningSessionId* property associates the Network Assistance Session with a Provisioning Session.

2. The *slice* property associates the Network Assistance Session with a specific network slice.

3. The *dataNetworkName* property associates the Network Assistance Session with a specific named Data Network.

4. The Media Session Handler shall populate the Network Assistance Session resource representation in the request with service data flow information and optionally the Policy Template identifier of the network QoS policy currently in force on the media streaming session for which Network Assistance operations are to be performed. (The Media AF subsequently uses this information to execute Network Assistance operations in the 5GC.)

   The *applicationFlowDescription* property of the Network Assistance Session resource representation shall be populated by the Media Session Handler and shall declare a Service Data Flow template according to TS 23.503 [33] that describes the application flow for which network assistance is sought. Exactly one of the

following filtering specifications shall be populated in the *ApplicationFlowDescription* object to identify traffic belonging to a media delivery application flow:

- a *packetFilter* object (including 5-tuples, Type of Service, Security Parameter Index, etc.). A Media Client shall not attempt to instantiate more than one Network Assistance Session at the same time that cites the same packet filter.

- a *domainName* populated with the fully-qualified Internet domain name of a Media AS at reference point M4. A Media Client shall not attempt to instantiate more than one Network Assistance Session at the same time that cites the same *domainName.*

In addition, the top-level media type of the application flow may be declared in the *mediaType* property.

The *mediaTransportParameters* property shall be omitted.

5. The *requestedQoS* property may be provided in the Network Assistance Session resource representation to specify an initial network QoS the Media Session Handler wishes to use for the media streaming session. If the *policyTemplateId* property is also populated in the Network Assistance Session resource representation, the Media AF shall return a *400 (Bad Request)* HTTP response message if the requested network QoS lies outside the limits specified in the referenced Policy Template.

   - If the *requestedQoS* property is omitted from the Network Assistance Session resource representation but the *policyTemplateId* is populated, the Media AF shall use the network QoS currently provisioned in the referenced Policy Template as the floor/ceiling for bit rate recommendations and delivery boosts within the scope of the Network Assistance Session.

   - If neither a *policyTemplateId* nor a *requestedQoS* are supplied when creating a Network Assistance Session, operations invoked on the Media AF within the scope of the Network Assistance session are constrained only by the policies of the PCF. Upon successful creation, the Media AF shall return a *201 (Created)* response message and the URL of the newly created resource, including its Network Assistance session resource identifier, shall be provided as the value of the `Location` HTTP header field. The response message body shall be a representation of the current state of the Network Assistance Session resource (see clause 9.4.3.1), including any properties assigned by the Media AF.

If the operation is successful, the Media AF shall create a new Network Assistance Session resource. In this case, the Media AF shall return a *201 (Created)* HTTP response message to the Media Session Handler, and the URL of the newly created Network Assistance Session resource, including its resource identifier, shall be provided as the value of the `Location` HTTP header field. The response message body shall be a representation of the current state of the Network Assistance Session resource (see clause 9.4.3.1), including any properties assigned by the Media AF.

Upon successful creation of the Network Assistance Session resource, notifications of updates to the resource may be notified asynchronously to the Media Session Handler:

- If the *notificationURL* property is present in the Service Access Information, the Media Session Handler shall subscribe to the MQTT sub-topic corresponding to the *resourceId* of the Network Assistance Session, and shall expect to receive asynchronous notifications published by the Media AF on the MQTT notification channel of type *NOTIFICATION_NETWORK_ASSISTANCE_SESSION* concerning changes to the Network Assistance Session, including an up-to-date bit rate recommendation whenever this changes.

- The Media AF shall use MQTT the notification channel signalled in the Service Access Information (if any, see clause 5.3.2.3) to notify subscribers of updates to the Network Assistance Session resource. A notification message of type *NOTIFICATION_NETWORK_ASSISTANCE_SESSION* shall be published to the MQTT sub-topic corresponding to the *resourceId* of the Network Assistance Session.

The usage and message formats for the MQTT notification channel are specified in clause 10.2.

When the Network Assistance Session is successfully instantiated, the Media AF triggers the creation of a corresponding PCC rule in the 5G System according to clause 5.5.4 to enforce the required QoS on the specified application flow(s). Depending on the *ApplicationFlowDescripton* objects in the received Network Assistance Session resource representation and the *filterMethod* indicated by each one, the Media AF shall populate for each one a

flowDescription object and/or provide an Application Identifier referring to a *PFD* (Packet Flow Description) object containing the domain name of a Media AS instance.

NOTE: When the Media AF is deployed in an external Data Network, it is the responsibility of the NEF to map any external Application Identifier supplied by the Media AF into an internal Application Identifier that is known to the PCF.

If the supplied Network Assistance Session is not acceptable to the Media AF, the create operation shall fail with an HTTP response status code of *400 (Bad Request)* and an error message body per clause 7.1.7. In this case, the Network Assistance Session resource shall remain in an uncreated state in the Media AF.

If the request is acceptable but the Media AF forbids the use of the referenced Policy Template in a Network Assistance Session, for example because the UE is not permitted in the charging specification, the create operation shall fail with an HTTP response status code of *403 (Forbidden)* and an error message body per clause 7.1.7. In this case, the Network Assistance Session resource shall remain in an uncreated state in the Media AF.

If the request is acceptable but the Media AF is unable to provision the resources required by the supplied Network Assistance Session, the create operation shall fail with an HTTP response status code of *500 (Internal Server Error)* and an error message body per clause 7.1.7. In this case, the Network Assistance Session resource shall remain in an uncreated state in the Media AF.

The Media Client uses the Network Assistance Session resource identifier (*naSessionId*) provided by the Media AF to refer all subsequent API calls to the Media AF instance responsible for that Network Assistance Session.

5.3.4.3 Retrieve Network Assistance Session resource operation

This operation is used by the Media Session Handler to retrieve the current state of a Network Assistance Session resource from the Media AF. The HTTP `GET` method shall be used for this purpose, citing the resource identifier of the target Network Assistance Session in the request URL.

If the operation is successful, the Media AF shall return *200 (OK)* and shall provide a representation of the requested resource in the HTTP message response body.

5.3.4.4 Bit rate recommendation request operation

This operation is used by the Media Session Handler to request a bit rate recommendation from the Media AF. the HTTP `GET` method shall be used for this purpose, citing the resource identifier of an existing Network Assistance Session in the request URL along with a sub-resource path indicting the bit rate recommendation operation.

If the operation is successful, the Media AF shall return a *200 (OK)* HTTP response message and shall provide the recommended bit rate(s) in an HTTP response message body containing an *M5QoSSpecification* object that is populated as follows:

- For a downlink media delivery session, the recommended minimum and maximum downlink bit rates shall be indicated in the properties *minimumRequestedBitRate* and *maximumBitRate* respectively of the *downlinkBitRates* object. If a unique downlink bit rate is recommended by the Media AF, then this value shall be set identically in both of these properties. The Media Session Handler shall ignore the mandatory properties related to uplink media delivery, i.e., *uplinkBitRates*.

- For an uplink media delivery session, the recommended minimum and maximum uplink bit rates shall be indicated in the properties *minimumRequestedBitRate* and *maximumRequestedBitRate* respectively of the *uplinkBitRates* object. If a unique uplink bit rate is recommended by the Media AF, then this value shall be set identically in both of these properties. The Media Session Handler shall ignore the mandatory properties related to downlink media delivery, i.e., *downlinkBitRates*.

The optional properties *minimumDesiredBitRate*, *desiredPacketLatency* and *desiredPacketLoss* shall not be included in the returned *M5QoSSpecification* object.

If the Media AF refuses to provide a bit rate recommendation, for example because the Provisioning Session in question currently lacks the rights required to receive this information, the operation shall fail with an HTTP response status code of *403 (Forbidden)* and an error message body per clause 7.1.7.

5.3.4.5 Delivery boost request operation

This operation is used by the Media Session Handler to request a delivery boost from the Media AF. the HTTP `POST` method shall be used for this purpose, citing the resource identifier of an existing Network Assistance Session in the request URL along with a sub-resource path indicating the delivery boost operation.

If the operation is successful, the Media AF shall return a *200 (OK)* HTTP response message and shall provide an *OperationSuccessResponse* object (see clause 7.3.3.9) in the message body indicating whether or not the delivery boost was successfully applied by the Media AF to the application data flow(s) described in the target Network Assistance Session.

If the Media AF refuses to provide the requested delivery boost, for example because the Provisioning Session in question currently lacks the rights required to receive a boost, the operation shall fail with an HTTP response status code of *403 (Forbidden)* and an error message body per clause 7.1.7.

5.3.4.6 Update Network Assistance Session resource operation

This operation is used by the Media Session Handler to replace the steaming session parameters in an existing Network Assistance Session resource with new values. The HTTP `PUT` or `PATCH` methods shall be used for this purpose, citing the resource identifier of an existing Network Assistance Session in the request URL. Any change to the Policy Template currently in force resulting from instantiation of a Dynamic Policy (see clause 5.3.3) should also be notified to the Media AF using this operation if a Network Assistance session has been created for the media delivery session in question.

If the HTTP request is acceptable but the operation results in no change to the resource representation, a *204 (No Content)* HTTP response message with an empty body should be returned.

If the operation is otherwise successful, the Media AF shall return a *200 (OK)* HTTP response message and shall provide a representation of the current state of the resource in the message body to confirm successful update.

If the supplied Network Assistance Session is not acceptable to the Media AF, the update operation shall fail with an HTTP response status code of *400 (Bad Request)* and an error message body per clause 7.1.7. In this case, the Network Assistance Session resource shall remain in the state immediately prior to the update operation.

Attempts to modify read-only properties of the target Network Assistance Session resource shall be rejected by the Media AF with a *403 (Forbidden)* HTTP response that includes an error message body per clause 7.1.7.

If the request is acceptable but the Media AF forbids the use of the referenced Policy Template in a Network Assistance Session, for example because the UE is not permitted in the charging specification, the update operation shall fail with an HTTP response status code of *403 (Forbidden)* and an error message body per clause 7.1.7. In this case, the Network Assistance Session resource shall remain in the state immediately prior to the update operation.

If the request is acceptable but the Media AF is unable to provision the resources required by the supplied Network Assistance Session, the update operation shall fail with an HTTP response status code of *500 (Internal Server Error)* and an error message body per clause 7.1.7. In this case, the Network Assistance Session resource shall remain in the state immediately prior to the update operation.

5.3.4.7 Destroy Network Assistance Session resource operation

This operation is used by the Media Session Handler to terminate a Network Assistance Session. The HTTP `DELETE` method shall be used for this purpose, citing the resource identifier of the target Network Assistance Session in the request URL.

If the operation is successful, the Media AF shall return a *204 (No Content)* HTTP response message with an empty message body.

No further MQTT notification messages shall be published by the Media AF to the sub-topic corresponding to the resource identifier of the destroyed resource. The Media Session Handler shall unsubscribe from the sub-topic corresponding to this resource identifier.

Any subsequent operations citing the resource identifier of a destroyed Network Assistance Session should result in a *410 (Gone)* or else a *404 (Not Found)* HTTP response message that includes an error message body per clause 7.1.7.

5.3.5 Metrics reporting

5.3.5.1 Procedures

These procedures are used by the Media AS at reference point M3 or else by the Metrics Reporting functions of the Media Client and subsequently by the Media Session Handler at reference point M5 to submit a metrics report to one of the Media AF instances listed in the client metrics reporting configuration of the Service Access Information resource previously retrieved using the procedure in clause 5.3.2.3.

When the metrics collection and reporting feature is provisioned for a media delivery session using the operations specified in clause 5.2.11, one or more client metrics reporting configurations, each associated with a provisioned Metrics Reporting Configuration, shall be provided to the Media Session Handler in the Service Access Information. A given client metrics reporting configuration contains information including:

1. The subset of metrics from the provisioned metrics scheme to be collected and reported by the Media Client;
2. The frequency at which these metrics are to be sampled by the Media Client;
3. The proportion of media delivery sessions for which reports are to be sent by the Media Session Handler;
4. The portion of the media session (represented by start offset and/or duration parameters) for which metrics reports are to be sent by the Media Session Handler if reporting is enabled for that media delivery session;
5. The interval at which metrics reports are to be sent by the Media Session Handler if reporting is enabled for a media delivery session; and
6. The Media AF address(es) to which metrics reports are to be sent.

Before a media delivery session is started, the Media Session Handler shall check if the Service Access Information includes any Metrics Reporting Configurations. If any such configurations are present, the Media Session Handler shall initiate the metrics reporting procedure for the media delivery session based on these configurations. During the media delivery session, the Media Session Handler shall periodically check if the Metrics Reporting Configurations are added to or removed from the Service Access Information and shall activate or deactivate the metrics reporting procedure as appropriate for the media delivery session in question. The Service Access Information indicating whether Metrics Reporting is provisioned for media delivery sessions is specified in clause 9.2.3.

Whenever a metrics report is produced for a given client metrics reporting configuration, the Media Session Handler shall reset its reporting interval timer for that configuration to the value of the c*lientMetricsReportingConfigurations[]. reportingInterval* property and it shall begin countdown of the timer again. Whenever the Media Session Handler terminates a media delivery session, it shall disable its reporting interval timer for all client metrics reporting configurations.

Details of the APIs supporting these procedures at reference points M3 and M5 are specified in clause 9.5.

HTTP responses for successful and operation-specific failure cases are specified in the following clauses. For all other failure cases, an HTTP response indicating a response code in accordance with clause 7.1.6 shall be returned to the API client. In all failure cases a message body in accordance with clause 7.1.7 shall be included in the response message.

5.3.5.2 Submit metrics report operation

This operation is used by the Media Session Handler or Media AS to submit a metrics report to the Media AF If several Media AF addresses are listed in the *serverAddresses* array of the client metrics reporting configuration (see table 9.2.3.1-1), the Media Session Handler shall choose one at random and shall send the metrics report to the selected server endpoint. The HTTP `POST` method shall be used for this purpose, citing the address of the chosen Media AF in the request URL. The request body shall be formatted according to the metrics scheme indicated in *scheme* property of one of the Client Metrics Reporting Configurations (see clause 5.3.2.3 and table 9.2.3.1-1) and the `Content-Type` HTTP request header set accordingly. Details of individual metrics reporting schemes and their corresponding metrics report formats are beyond the scope of the present document.

A reporting client identifier should be included in the metrics report if the metrics scheme supports carriage of this data. Metrics schemes designed for use with this operation should specify a means to convey a reporting client identifier. If available to the Media Session Handler, its value should be a GPSI value as defined by TS 23.003 [16]. Otherwise, the reporting client identifier should be represented by a stable and globally unique string.

If the HTTP request is acceptable but the Media AS has not yet fully processed the submitted metrics report, the Media AF may return a *202 (Accepted)* HTTP response message with an empty body and process the report later.

If the operation is otherwise successful, the Media AF shall return a *200 (OK)* HTTP response message with an empty body to acknowledge successful processing of the metrics report.

If metrics reporting is not provisioned for the Provisioning Session in question, the Media AF shall return a *403 (Forbidden)* HTTP response message with an error message body per clause 7.1.7 and the Media AF shall not process the submitted report.

If the HTTP request message indicates a MIME content type in the `Content-Type` request header that is not consistent with one of the provisioned metrics reporting schemes, the Media AF shall return a *415 (Unsupported Media Type)* HTTP response message with an error message body per clause 7.1.7 and shall not process the submitted metrics report.

If the target Media AF endpoint is temporarily unable to accept the submitted metrics report (e.g. because it is overloaded), it shall return a *503 (Service Unavailable)* HTTP response message with an empty body. The optional HTTP response header `Retry-After` should be included in such a response, indicating when the Media AS expects to be able to accept new submissions. In this case, the Media Client should store outstanding metrics reports and reattempt submission when the endpoint later becomes available. Details are left to implementation.

5.3.6 Consumption reporting

5.3.6.1 Procedures

These procedures are used by the Media AS at reference point M3 or else by the Consumption Reporting functions of the Media Client and subsequently by the Media Session Handler at reference point M5 to submit a consumption report to one of the Media AF instances listed in the Client Consumption Reporting Configuration of the Service Access Information resource previously retrieved using the procedure in clause 5.3.2.3.

When the Consumption Reporting feature is provisioned for a downlink media delivery session using the operations specified in clause 5.2.12, a Client Consumption Reporting Configuration shall be provided to the Media Session Handler in the Service Access Information.

Before a downlink media delivery session is started, the Media Session Handler shall check if the Service Access Information includes a Client Consumption Reporting Configuration. If such a configuration is present, the Media Session Handler shall initiate consumption reporting for the downlink media delivery session based on this configuration. During the course of the downlink media delivery session, the Media Session Handler shall periodically check if the Client Consumption Reporting Configuration is added to or removed from the Service Access Information and shall activate or deactivate the consumption reporting procedure as appropriate for the media delivery session in question.

The Service Access Information indicating whether Consumption Reporting is provisioned for a particular downlink media delivery session is specified in clause 9.2.3.

When the *samplePercentage* property in the Client Consumption Reporting Configuration has a value of 100 percent, the Media Session Handler shall activate the consumption reporting procedure. If the *samplePercentage* value is less than 100 percent, the Media Session Handler shall generate a random number which is uniformly distributed in the range of 0 to 100, and the Media Session Handler shall activate the consumption reporting procedure when the generated random number is of a lower value than the *samplePercentage* value.

If the consumption reporting procedure is activated, the Media Session Handler shall produce and submit a consumption report to the Media AF using the procedure specified in clause 5.3.6.2 when any of the following conditions are met:

- At the start of consumption of a downlink media delivery session;
- At the end of consumption of a downlink media delivery session;
- On determining the need to report ongoing content consumption at periodic intervals determined by the *reportingInterval* property in the Client Consumption Reporting Configuration.
- On detecting a location change, if the *locationReporting* property in the Client Consumption Reporting Configuration is set to *true*.

- On detecting a change of access network, if the *accessReporting* property in the Client Consumption Reporting Configuration is set to *true*.

Whenever a consumption report is produced, the Media Session Handler shall reset its consumption reporting interval timer to the value of the *reportingInterval* property of the Client Consumption Reporting Configuration and it shall begin countdown of the timer again. Whenever the Media Session Handler terminates a downlink media delivery session, it shall disable its consumption reporting interval timer.

Details of the APIs supporting these procedures at reference points M3 and M5 are specified in clause 9.6.

The consumption report shall comprise a time-ordered list of consumption reporting units. Each such unit shall describe the media selected for presentation during a continuous time period of a downlink media streaming session in terms of a start time and duration. The sequence of consumption reporting units shall be contiguous with no discontinuities in the reported timeline. When no media is being consumed (e.g., because the media streaming presentation is paused), the selected media shall still be indicated in the consumption reporting unit.

- A consumption reporting unit shall be included in exactly one consumption report, although delivery of this report may be attempted more than once by the Media Session Handler.

- A new consumption reporting unit shall be created when the media consumed changes or (if provisioned in the consumption reporting configuration per clause 4.3.8) when the network used to access media at reference point M4d changes, including a change of network slice and/or Data Network.

- The last (or only) consumption reporting unit in every consumption report describes the media currently being consumed in the media streaming session and indicates in the duration property how long this media has been consumed so far.

- If there is no change in the media consumed when the next consumption report is sent to the Media AF, this consumption reporting unit shall be repeated as the first (and possibly only) consumption reporting unit in the next report with the same start time but with its duration updated to reflect the period of time that the media has been consumed up to the point of reporting.

- The last (or only) consumption reporting unit in the final consumption report sent to the Media AF at the end of the downlink media streaming session therefore describes the last media consumed.

HTTP responses for successful and operation-specific failure cases are specified in the following clauses. For all other failure cases, an HTTP response indicating a response code in accordance with clause 7.1.6 shall be returned to the API client. In all failure cases a message body in accordance with clause 7.1.7 shall be included in the response message.

5.3.6.2 Submit consumption report operation

This operation is used by the Media Session Handler or Media AS to submit a consumption report to the Media AF If several Media AF addresses are listed in the *serverAddresses* array of the Client Consumption Reporting Configuration (see table 9.2.3.1-1), the Media Session Handler shall choose one at random and shall send the consumption report to the selected server endpoint. The HTTP `POST` method shall be used for this purpose, citing the address of the chosen Media AF in the request URL. The request body shall be a *ConsumptionReport* structure, as specified in clause 9.6.3.1.

A reporting client identifier shall be included in the consumption report. If available to the Media Session Handler, its value should be a GPSI value as defined by TS 23.003 [16]. Otherwise, the reporting client identifier should be represented by a stable and globally unique string.

The location(s) of the UE when the media was consumed shall be included in every *ConsumptionReportingUnit* (see clause 9.6.3.2) if the *locationReporting* property in the Client Consumption Reporting Configuration is set to *true*.

If the HTTP request is acceptable but the Media AS has not yet fully processed the submitted consumption report, the Media AF may return a *202 (Accepted)* HTTP response message with an empty body and process the report later.

If the operation is otherwise successful, the Media AF shall return a *200 (OK)* HTTP response message with an empty body to acknowledge successful processing of the consumption report.

If consumption reporting is not provisioned for the Provisioning Session in question, the Media AF shall return a *403 (Forbidden)* HTTP response message with an error message body per clause 7.1.7 and the Media AF shall not process the submitted consumption report.

If the target Media AF endpoint is temporarily unable to accept the submitted consumption report (e.g. because it is overloaded), it shall return a *503 (Service Unavailable)* HTTP response message with an empty body. The optional HTTP response header `Retry-After` should be included in such a response, indicating when the Media AS expects to be able to accept new submissions. In this case, the Media Client should store outstanding consumption reports and reattempt submission when the endpoint subsequently becomes available. Details are left to implementation.

5.4 UE media session handling (M6, M11) interactions

5.4.1 Overview

This clause specifies the interactions between the Media-aware Application and the Media Session Handler at reference point M6 and those between the Media Access Function and the Media Session Handler at reference point M11. Details are provided in clause 10.

5.4.2 Media delivery session life-cycle

5.4.2.1 Explicit media session handling initiation/termination

Media session handling of a new media delivery session may be explicitly initiated by a Media-aware Application or Media Access Function by invoking an appropriate API method on the Media Session Handler at reference points M6 or M11, respectively.

- An *external service identifier* shall be provided as input parameter to the API method.
- A Media Entry Point URL, for instance obtained from the Media Application Provider at reference point M8, may optionally be provided as an input parameter in order to initiate media delivery.

In response, the Media Session Handler shall allocate a globally unique *media delivery session identifier* for use by the Media Client in its subsequent interactions with the Media AF and Media AS.

If it does not already have a fresh copy cached, the Media Session Handler shall attempt to acquire full Service Access Information for the specified external service identifier from the Media AF using the operation defined in clause 5.3.2.3 and, if successful, shall return the media delivery session identifier to the invoker of the API method.

If invoked by a Media-aware Application at reference point M6, the Media Session Handler shall initialise a new Media Access Function instance on behalf of the invoker, and shall initiate media delivery by passing it the Media Entry Point URL.

Subsequent interactions between the Media-aware Application and the Media Session Handler at reference point M6 shall cite the relevant media delivery session identifier.

Subsequent interactions between the Media-aware Application and the Media Access Function at reference point M7 shall cite the relevant media delivery session identifier.

Subsequent interactions between the Media Session Handler and the Media Access Function at reference point M11 shall cite the relevant media delivery session identifier.

Subsequent interactions by the Media Access Client with the Media AS at reference point M4 shall cite the relevant media delivery session identifier to enable media access logged by the Media AS to be correlated with media session handling operations logged by the Media AF.

The Media-aware Application or Media Access Function may explicitly terminate media session handling of the media delivery session by invoking an appropriate API method on the Media Session Handler at reference point M6 or M11, respectively, citing the target media delivery session identifier as input parameter.

5.4.2.2 Implicit media session handling initiation/termination

A UE entity may use the implicit service launch mechanism specified in clause 6 to initiate media session handling of a new media delivery session.

In response, the Media Session Handler shall allocate a globally unique *media delivery session identifier* for use by the Media Client in its subsequent interactions with the Media AF and Media AS.

The Media Session Handler shall attempt to acquire full Service Access Information for the specified external service identifier from the Media AF using the operation defined in clause 5.3.2.3.

The implicit service launch mechanism does not typically yield a return value, in which case it is not possible to return a media delivery session identifier to the invoker. Hence, it is not possible for the UE entity invoking the implicit service launch mechanism to take advantage of the UE media session handling APIs specified in clause 10. Nevertheless, the Media Session Handler is able to support a limited subset of media session handling on behalf of the UE entity – such as dynamic policy instantiation based on a Service Operation Point – if additional launch parameters are embedded in the Service URL.

When it has been initiated using the implicit method described in this clause, the media delivery session is implicitly terminated by the Media Session Handler after it detects an implementation-specific period of inactivity.

5.4.3 Dynamic Policy invocation

At the start of a media delivery session, the Media Session Handler shall determine the external reference and target QoS parameters of the initial Service Operation Point by invoking an appropriate API method on the Media Session Handler at reference point M11. Based on the parameter values supplied, the Media Session Handler shall attempt to instantiate a Dynamic Policy satisfying the Media Access Function's requirements using the operation specified in clause 5.3.3.2 if the target QoS lies within the bounds of a Policy Template with the corresponding external reference advertised in the Service Access Information for the media delivery session.

The Media Session Handler shall subscribe to receive notifications from the Media Access Function at reference point M11 of changes to the Service Operation Point during the course of the media delivery session. When such a change occurs (e.g., when the Media Access Function selects a different MPEG-DASH Representation), the Media Access Function shall send a notification to the Media Session Handler at reference point M11 citing the external reference and target QoS parameters of the new Service Operation Point. If the QoS of the new Service Operation Point is not satisfied by the currently instantiated Dynamic Policy, the Media Session Handler shall attempt to instantiate a Dynamic Policy satisfying the Media Access Function's requirements using the operation specified in clause 5.3.3.2 if the target QoS lies within the bounds of a Policy Template with the corresponding external reference advertised in the Service Access Information for the media delivery session.

The Media-aware Application shall subscribe to receive notifications from the Media Session Handler at reference point M6 concerning Background Data Transfer opportunities. When such an opportunity is announced to the Media Session Handler by the Media AF at reference point M5, the Media Session Handler shall send a corresponding notification to the Media-aware Application at reference point M6 that includes an estimate of the opportunity window. If it wishes to avail itself of the Background Data Transfer opportunity, the Media-aware Application shall invoke a suitable API method on the Media Session Handler at reference point M6, providing an estimate of the data volume it intends to transfer over reference point M4. The Media Session Handler shall then attempt to instantiate a Dynamic Policy with Background Data Transfer network characteristics (including the data volume estimate supplied by the Media-aware Application) using the operation specified in clause 5.3.3.2 if a suitable Policy Template is advertised in the Service Access Information for the media delivery session.

5.4.4 Network Assistance invocation

At the start of a media delivery session, the Media Access Function shall determine an initial bit rate recommendation by invoking an appropriate API method on the Media Session Handler at reference point M11. The Media Session Handler shall invoke the operation specified in clause 5.3.4.2 to create a new Network Assistance Session and shall return to the Media Access Function the bit rate recommendation information in the response from the Media AF. The Media Access Function may use this information to set an initial bit rate for the media delivery session (e.g., by selecting an initial MPEG-DASH Representation or contribution media encoding bit rate). The Media Access Function may also use this bit rate recommendation to select a different Service Operation Point, resulting in a change to the current Dynamic Policy (see clause 5.4.2).

The Media Session Handler shall subscribe to receive notifications from the Media AF at reference point M11 of bit rate recommendation updates. When such an update is received from the Media AF, the Media Session Handler shall send a notification to the Media Access Function at reference point M11 providing the new bit rate recommendation. As a result, the Media Access Function may modify the bit rate of the media delivery session (e.g., by switching to a different MPEG-DASH Representation or by varying the rate controller of a contribution media encoder). The Media

Access Function may also use this bit rate recommendation to select a different Service Operation Point, resulting in a change to the current Dynamic Policy (see clause 5.4.2).

During the course of the media delivery session the Media Access Function may request a delivery boost by invoking an appropriate API method on the Media Session Handler at reference point M11 (see clause 10.4.1.2).

5.4.5 Metrics reporting

When metrics reporting is active for a given media delivery session, the Media Session Handler shall periodically sample the metrics required by the metrics scheme(s) configured per clause 5.3.5.1 by invoking an appropriate API method at reference point M11 on the Media Access Function and shall use these to populate metrics reports.

5.4.6 Consumption reporting

When consumption reporting is active for a given media delivery session, the Media Session Handler shall periodically determine the consumption reporting parameters required per clause 5.3.6.1 by invoking an appropriate API method on the Media Access Function at reference point M11 and shall use these to populate consumption reports.

5.5 5GC policy control (N5/N33) interactions

5.5.1 Overview

Certain features of the Media Delivery System rely on interfaces and APIs that are defined in the 5G Core. The interactions between the Media AF and the Network Functions in the 5GC to support these features are specified in the following clauses.

NOTE: The Media Delivery System architecture may be applied to an EPS although such application is not specified in the present document and is left to the discretion of deployments and implementations.

5.5.2 Policy control interactions for Policy Template provisioning

When a Policy Template is provisioned that specifies a new Background Data Transfer policy by including the *bdtSpecification* property, the Media AF shall invoke the *Npcf_BDTPolicyControl_Create* operation as specified in clause 4.2.2 of TS 29.554 [46] to create a separate Background Data Transfer policy in the PCF for each time window occurrence.

- The properties of the *bdtReqData* object shall be populated from the time window information in the *M1BDTSpecification* as follows:
  - *desTimeInt* shall be populated from *startTime* on the occurrence date and the *duration*.
  - *numOfUes* shall be populated from *numberOfUes*.
  - *volPerUe* shall be populated from *estimatedDataVolumePerUe*.
- The properties of the *bdtPolData* object shall be populated from the time window information in the *M1BDTSpecification* as follows:
  - *transfPolicies[ ].recTimeInt* shall be populated from *startTime* on the occurrence date and the *duration*.
  - *transfPolicies[ ].maxBitRateDl* shall be populated from *aggredatedDownlinkBitRateLimit*, if present.
  - *transfPolicies[ ].maxBitRateUl* shall be populated from *aggredatedUplinkBitRateLimit*, if present.

The Media AF is responsible for allocating downlink and uplink bit rates within the provisioned limits of the Background Data Transfer policy when a Policy Template with a Background Data Transfer specification is instantiated by the Media Client, such that the aggregate maximum bit rates (if any) are not exceeded at any point of time. How the Media AF polices this limit is out of scope of the present document.

The Media AF may use the values of the *numberOfUes* and *estimatedDataVolumePerUe* properties to estimate the total aggregate data volume that may be transferred by all Media Clients in a given time window.

When a Policy Template that specifies a Background Data Transfer policy by including the *bdtSpecification* property is destroyed, there is no operation specified by the *Npcf_BDTPolicyControl* service in TS 29.554 [46] to destroy the corresponding Background Data Transfer policy in the PCF.

5.5.3 Policy control interactions for Dynamic Policies

The Dynamic Policies feature operates at reference point M5 between the Media Session Handler in the Media Client and a Media AF that has been appropriately provisioned with Policy Templates (see clause 5.2.7). The Dynamic Policy API at reference point M5 (see clauses 5.3.3 and 9.3) is specified in a generic way such that the associated functionality in the 5GC may be realised by various means.

NOTE 1: This clause does not limit the possible set of 5G System exposure functionalities for realising dynamic policies.

In this release, the Media AF converts Dynamic Policies API invocations received at reference point M5 into direct or indirect invocations of the Policy Authorization Service exposed by the PCF, and converts responses from the PCF into their equivalents at reference point M5 for return to the Media Session Handler.

To realise dynamic policies, the Media AF shall interact with the PCF using one of the following methods:

A. If the Media AF is deployed in the Trusted DN, it may directly invoke the *Npcf_PolicyAuthorization* service at reference point N5, as specified in TS 29.514 [18].

NOTE 2: It is the responsibility of the Media AF in this case to discover and track changes to the PCF instance responsible for the PDU Session supporting the media streaming session at reference point M4 using the discovery services provided by the NRF and/or BSF.

B. If the Media AF is deployed outside the Trusted DN, or if it is more convenient for a Media AF deployed in the Trusted DN to do so, it invokes the *Nnef_AFSessionWithQoS* and/or *Nnef_ChargeableParty* services exposed by the NEF, as specified in clauses 4.4.9 and 4.4.8 respectively of TS 29.522 [19], to indirectly invoke the PCF at reference point N33.

NOTE 3: Per clause 4.4.9 of TS 29.522 [19], the *Nnef_AFSessionWithQoS* service is realised at reference point N33 by the *AsSessionWithQoS* exposure API. Similarly, the *Nnef_ChargeableParty* service is realised by the *ChargeableParty* exposure API per clause 4.4.8 of [19].

NOTE 4: Configuration of the NEF endpoint address and access credentials in the Media AF in this case is beyond the scope of the present document.

When the first Dynamic Policy is created by the Media Session Handler for a particular media delivery session (per clause 5.3.3.2), the Media AF shall create an *AF application session context* in the PCF responsible for the PDU Session corresponding to the M4 application flows indicated in the *DynamicPolicy.applicationFlowBindings* array.

If no corresponding AF application session context already exists, the Media AF shall use the *Npcf_PolicyAuthorization_Create* operation at reference point N5 (or, if deployed outside the Trusted DN, the equivalent *Nnef_AFsessionWithQoS* service operation) with the appropriate service information to create and provision a new AF application session context. The information in the *AppSessionContextReqData* shall be derived from the application flow descriptions in the Dynamic Policy Instance resource and/or the requested QoS.

The mapping of application flows listed in the *DynamicPolicy.applicationFlowBindings* array to media components and sub-components of the AF application session context is implementation-dependent.

[If the *pduSetQoSLimits* property is populated in *M1QoSSpecification.downlinkQoSSpecification*, then the *Media Component.pduSetQosDl* object shall be populated as follows by the Media AF:

- The *pduSetDelayBudget* property shall be set to the larger value of *pduSetQoSLimits.pduSetDelayBudget* and *desiredDownlinkPduSetQosParameters.pduSetDelayBudget*.

- The *pduSetErrorRate* property shall be set to the larger value of *pduSetQoSLimits.pduSetErrorRate* and *desiredDownlinkPduSetQosParameters.pduSetErrorRate*.

- The *pduSetHandlingInfo* property shall be set to the value of *pduSetQoSLimits.pduSetHandlingInfo*, ignoring the value of *desiredDownlinkPduSetQosParameters.pduSetHandlingInfo*, if any.

Otherwise, the *MediaComponent.pduSetQosDl* object shall be populated directly from the *desiredDownlinkPduSet QosParameters* object.

If the *pduSetQoSLimits* property is populated in *M1QoSSpecification.uplinkQoSSpecification*, then the *Media Component.pduSetQosUl* object shall be populated as follows by the Media AF:

- The *pduSetDelayBudget* property shall be set to the larger value of *pduSetQoSLimits.pduSetDelayBudget* and *desiredUplinkPduSetQosParameters.pduSetDelayBudget*.
- The *pduSetErrorRate* property shall be set to the larger value of *pduSetQoSLimits.pduSetErrorRate* and *desiredUplinkPduSetQosParameters.pduSetErrorRate*.
- The *pduSetHandlingInfo* property shall be set to the value of *pduSetQoSLimits.pduSetHandlingInfo*, ignoring the value of *desiredUplinkPduSetQosParameters.pduSetHandlingInfo*, if any.

Otherwise, the *MediaComponent.pduSetQosUl* object shall be populated directly from the *desiredUplinkPduSet QosParameters* object.]

For each of the Dynamic Policy Instances it is managing, the Media AF shall subscribe to the following PCF notifications on the corresponding AF application session context:

- Service Data Flow QoS notification control;
- Service Data Flow deactivation;
- Resources allocation outcome.

When requesting QoS provisioning for a media delivery session, the Media AF shall use the configured Policy Template of the Dynamic Policy Instance to determine the list of the QoS references within *altSerReqs*. The lowest priority index shall be assigned to the Policy Template with the lowest QoS requirement, and the highest priority shall be assigned to the Service Operation Point requested by the UE (if the UE is allowed to use that operation point).

When instantiating a Policy Template that has a Background Data Transfer policy, the Media AF needs to populate some of the properties in the *M5BDTSpecification* object specified in clause 9.3.3.3 for inclusion in the Dynamic Policy Instance resource returned to the Media Session Handler at reference point M5.

Where the Policy Template references an existing Background Data Transfer policy by including the *bdtPolicyId* property, in order to populate the properties of the *M5BDTSpecification* object the Media AF shall first retrieve the individual Background Data Transfer policy resource referenced by *bdtPolicyId* from the PCF. The *Npcf_BDTPolicy Control* service operation specified in clause 5.3.3.3.1 of TS 29.554 [46] shall be used for this purpose.

When a dynamic policy is subsequently destroyed by the Media Session Handler (per clause 4.7.3), the Media AF shall destroy the corresponding AF application session context in the relevant PCF instance.

5.5.4 Policy control interactions for AF-based Network Assistance

The AF-based Network Assistance feature operates at reference point M5 between the Media Session Handler in the Media Client and a Media AF that provides Network Assistance capabilities. The Network Assistance API at reference point M5 (see clauses 5.3.4 and 9.4) is specified in a generic way such that the associated Network Assistance functionality in the 5GC may be realised by various means.

NOTE 1: This clause does not limit the possible set of 5G System exposure functionalities for obtaining Network Assistance information.

In this release, the Media AF converts Network Assistance API invocations received at reference point M5 into direct or indirect invocations of the Policy Authorization Service exposed by the PCF, and converts responses and notifications from the PCF into their equivalents at reference point M5 for delivery to the Media Session Handler.

If it supports the Network Assistance feature, the Media AF shall offer the bit rate recommendation (throughput estimation) and delivery boost request API based on existing Policy Templates that match the filtering criteria for a media streaming session, and the Media AF shall interact with the PCF using one of the following methods:

A. If the Media AF is deployed in the Trusted DN, it may directly invoke the *Npcf_PolicyAuthorization* service at reference point N5, as specified in TS 29.514 [18].

NOTE 2: It is the responsibility of the Media AF in this case to discover and track changes to the PCF instance responsible for the PDU Session supporting the media streaming session at reference point M4 using the discovery services provided by the NRF and/or BSF.

B. If the Media AF is deployed outside the Trusted DN, or if it is more convenient for a Media AF deployed in the Trusted DN to do so, it invokes the *Nnef_AFSessionWithQoS* service exposed by the NEF, as specified in clause 4.4.9 of TS 29.522 [19], to indirectly invoke the PCF at reference point N33.

NOTE 3: Per clause 4.4.9 of TS 29.522 [19], the *Nnef_AFSessionWithQoS* service is realised at reference point N33 by the *AsSessionWithQoS* exposure API.

NOTE 4: Configuration of the NEF endpoint address and access credentials in the Media AF in this case is beyond the scope of the present document.

When the first Network Assistance Session is created by the Media Session Handler for a particular media delivery session (per clause 5.3.4.2), the Media AF shall create an *AF application session context* in the PCF responsible for the PDU Session corresponding to the M4 application flow indicated in the *NetworkAssistanceSession.applicartionFlow Description* property.

If no corresponding AF application session context already exists, the 5GMS AF shall use the *Npcf_Policy Authorization_Create* operation at reference point N5 (or, if deployed outside the Trusted DN, the equivalent *Nnef_AFsessionWithQoS* service operation) with the appropriate service information to create and provision a new AF application session context. The information in the *AppSessionContextReqData* shall be derived from the application flow descriptions in the Network Assistance Session resource, as well as from the referenced Policy Template (if any) and/or the requested QoS.

The mapping of application flows listed in the *DynamicPolicy.applicationFlowBindings* array to media components and sub-components of the AF application session context is implementation-dependent.

The *MediaComponent.pduSetQosDl* object shall not be populated by the Media AF

The *MediaComponent.pduSetQosUl* object shall not be populated by the Media AF

For each of the Network Assistance Sessions it is managing, the 5GMS AF shall subscribe to the following PCF notifications on the corresponding AF application session context:

- Service Data Flow QoS notification control;
- Service Data Flow deactivation;
- Resources allocation outcome.

When requesting QoS provisioning for a Network Assistance Session, the Media AF shall use the configured Policy Templates of the Provisioning Session to determine the list of the QoS references within *altSerReqs*. The lowest priority index shall be assigned to the Policy Template with the lowest QoS requirement, and the highest priority shall be assigned to the Service Operation Point requested by the UE (if the UE is allowed to use that operation point).

When a Network Assistance session is subsequently destroyed by the Media Session Handler (per clauses 5.3.4.7 and 11.6.4.6), the Media AF shall destroy the corresponding AF application session context in the relevant PCF instance.

5.6 UE modem interactions

5.6.1 Overview

Certain features of the Media Delivery System rely on interfaces and APIs that are essentially UE-internal. The interactions between the Media Session Handler and the UE modem to support these features are specified in the following clauses.

5.6.2 ANBR-based Network Assistance

If ANBR-based Network Assistance is supported, the Media Session Handler may use an interface to the Medium Access Control entity in the UE modem to send and receive bit rate recommendation messages. This interface may be based on the AT commands *+CGBRRREQ* and *+CGBRRREP* as defined in TS 27.007 [21].

Furthermore, messaging across that interface corresponds to the logical translations of the *Bit Rate Recommendation* and/or *Bit Rate Recommendation Query* messages, carried by the Recommended bit rate MAC CE, exchanged between the RAN Modem and the RAN, as specified in TS 38.321 [22] for 5G NR and TS 36.321 [22] for LTE. The association between the LCID for which the recommendation applies and the actual flow (including the intermediate RLC channel) is performed by the modem.

NOTE: The AT command *+C5GQOSRDP=?* may be used to obtain a list of CID values that are associated with QoS flows (both network and MT/TE initiated). When used to request a bit rate boost, the query shall not request a bit rate that may exceed the MFBR for the corresponding QoS Flow. Failure to ensure this may result in unexpected congestion-induced packet delays and packet dropping.

The *Bit Rate Recommendation Query* shall indicate the bit rate desired by the application, as described by [22] and [23]. This request may be used by the Media Session Handler to request for a temporary increase in bit rate for the corresponding flow ("delivery boost"). The RAN responds with a Bit Rate Recommendation message that confirms the recommended bit rate after the boost grant. Once the bit rate drops again after a boost grant, the network shall inform the Media Session Handler about the new recommended bit rate by means of an ANBR message.

Whenever the Media Session Handler receives a message from the UE modem corresponding to the logical translation of the *Bit Rate Recommendation* message for the associated RAN uplink or downlink, it shall notify the Media Access Function (via a suitable reference point M11 notification – see table 10.4.2-2) of the new bit rate recommendation associated with the indicated PDU session.

Furthermore, whenever the Media Session Handler receives a request for a delivery boost from the Media Access Function (via a suitable reference point M11 method invocation – see clause 10.4.1.2) relating to a PDU session in use for Media Delivery, it may send a delivery boost message to the UE modem. That delivery boost request is logically translated by the modem to the *Bit Rate Recommendation Query* message which is then sent to the RAN on the associated RAN uplink or downlink.

It is left to the implementer of the Media Access Function to decide how to best use the bit rate recommendation and the bit rate recommendation query information for individual media delivery sessions.

5.6.3 RAN-based metrics reporting

These procedures shall be used by the Media Session Handler to control metrics reporting when such reporting is configured by the OAM via the 5G control plane signalling.

As described in clause L.1 of TS 26.247 [8], the metrics configuration is delivered to the UE as a container from the OAM via RAN and the Media Session Handler should obtain its metrics configuration, e.g. using the AT Command *+CAPPLEVMC* or *+CAPPLEVMCNR*. This configuration may also include virtual reality metrics as specified in clause 9.3 of TS 26.118 [9]. When a metrics configuration is received, the Media Session Handler shall store this configuration and use it for all subsequent media delivery sessions.

When a media delivery session is started the Media Session Handler shall determine whether metrics from this session shall be reported. The determination shall be based on the *sample percentage*, *slice scope* and *streaming source filter* specified in the stored metrics configuration, according to clause 10.5 of TS 26.247 [8].

If metrics are to be reported for the session, the Media Session Handler shall request the Media Access Function to create a metrics collection job. The Media Access Function shall return a reference to the created job, which the Media Session Handler shall use in all subsequent actions related to this job.

The Media Session Handler shall configure the metrics collection job with the set of metrics to be collected during the media delivery session. The format of the configuration shall be according to clause L.2 of [8], but only the *metrics* attribute in the configuration shall be used for this purpose.

The Media Session Handler shall regularly request the collected metrics from the Media Access Function according to the *reportingInterval* specified in the metrics configuration. The metrics returned by the Media Access Function shall use the format as described in clause 10.6 of [8], and (for Virtual Reality media) in clause 9.4.3 of TS 26.118 [9]. The Media Session Handler should forward these to the UE modem, e.g. using the AT command *+CAPPLEVMR* or *+CAPPLEVMRNR*. As a result, the UE modem sends metrics reports to the RAN which are then forwarded to the OAM according to clause L.1 of [8].

When the media delivery session is finished the Media Session Handler shall destroy the metrics collection job.

6 3GPP Service URL

6.1 General

This clause defines the syntax for 3GPP Service URLs used to launch media delivery sessions and the associated URL handling.

The 3GPP Service URL may be provided to the application via reference point M8 if it is a Media-aware Application. In another variant, the application may generate a 3GPP Service URL based on information in its own configuration.

6.2 3GPP Service URL syntax

3GPP Service URLs used to initiate media delivery sessions shall take the following form:

`http`*[s]*`://launch.3gppservices.org/`*{service}/{service_id}*`?`*{query_parameters}*

The structure of the 3GPP Service URL is as follows:

- The *prefix part* starts with the scheme name `http:` or `https:` followed by a double-slash `//`, followed by the authority `launch.3gppservices.org`, followed by a service type discriminator path segment *service*, followed by a service identifier path segment *service_id* and optional further path segments. This is aligned with section 4.2 of RFC 9110 [24] with the restriction that the prefix part shall not contain the character "&".

- The *suffix part* is optional and consists of the character "?" followed by a *query part* specifying additional service launch parameters formatted as a set of *<key>=<value>* pairs or flags that do not contain the equals character "=" and, optionally, followed by a *fragment part*. The suffix part is terminated by the end of the URL.

The formal ABNF of the 3GPP Service URL, following the generic syntax of URIs specified in RFC 3986 [35] is specified in listing 6.2.1-1:

Listing 6.2-1: ABNF syntax of 3GPP Service URL for Media Delivery session launch

```
service-URI   = scheme ":" hier-part [ "?" query ] [ "#" fragment ]
scheme        = "http:" / "https:"
hier-part     = "//" 3gpp-domain "/" service [/ service_id] path-abempty
3gpp-domain   = "launch.3gppservices.org"
service       = service-label / "generic"
service-label = 1*uchar
service_id    = 1*uchar
path-abempty  = <path-abempty, see RFC3986 [35], section 3.3>
query         = <query, see RFC3986 [35], section 3.4>
fragment      = <fragment, see RFC3986 [35], section 3.5>
```

6.3 Handling of 3GPP Service URLs by Media Client

Requests for 3GPP Service URLs from Media-aware Applications shall be handled by the Media Session Handler at reference point M6.

Requests for 3GPP Service URLs from other applications shall be handled by the Media Session Handler.

To cater for cases where a Media Session Handler is not present in a UE, the network shall provide a resolution of the 3GPP Service URL, for example to a service endpoint provided by the Media AF, that provides the invoking application with a resource that it is able to process, for example a media file, the URL of an MPD, etc. If this resolution yields a new URL, the service endpoint shall respond with an HTTP redirect to this new URL; otherwise an appropriate HTTP error response shall be returned.

The Media Session Handler performs decomposition of the URI into the prefix and suffix. The following handling applies:

- If the URI handed to the Media Session Handler does not conform to a 3GPP Service URL, the Media Session Handler should return a proper error response to the invoking application.
- If the Media Session Handler does not support the service type discriminator *service* indicated in the prefix part, it should return a proper error response to the invoking application.
- Specific additional operations may be defined for each service.

7 General aspects of network APIs

7.1 Usage of HTTP

7.1.1 HTTP protocol version

Implementations of the Media AF shall expose both HTTP/1.1 [26] and HTTP/2 [27] endpoints at reference points M1, M3 and M5, including support for the HTTP/2 starting mechanisms specified in section 3 of RFC 9113 [27]. In both protocol versions, TLS version 1.3 [29] shall be supported and HTTPS interactions should be used on these interfaces in preference to cleartext HTTP.

- The Media Application Provider may use any supported HTTP protocol version at reference point M1.
- The Media AS may use any supported HTTP protocol version at reference point M3.
- The Media Session Handler may use any supported HTTP protocol version at reference point M5.

All responses from the Media AF that carry a message body shall provide a strong entity tag in the form of an `ETag` response header and a modification timestamp in the form of a `Last-Modified` response header.

All endpoints exposed at reference points M1, M3 and M5 shall support the conditional HTTP requests `If-None-Match` and `If-Modified-Since`.

7.1.2 HTTP endpoint addresses

7.1.2.1 Default Media AF endpoint address at reference point M1

The present document does not specify a default endpoint address for the *Maf_Provisioning* service exposed to the Media Application Provider at reference point M1.

7.1.2.2 Default Media AF endpoint address at reference point M3

The present document does not specify a default endpoint address for the *Maf_SessionHandling* service exposed to the Media AS at reference point M3.

7.1.2.3 Default Media AF endpoint address at reference point M5

The present document does not specify a default endpoint address for the *Maf_SessionHandling* service exposed to the Media Session Handler at reference point M5.

7.1.3 HTTP resource URIs and paths

The resource URI used in each HTTP request to the API provider shall have the structure defined in subclause 4.4.1 of TS 29.501 [31], i.e.:

{apiRoot}/{apiName}/{apiVersion}/{apiSpecificResourceUriPart}

with the following components:

- *{apiRoot}* shall be set as described in [31].
- *{apiName}* shall be set as defined by the following clauses.
- *{apiVersion}* shall be set to "v1" in this release of the specification.
- *{apiSpecificResourceUriPart}* shall be set as described in the following clauses.

7.1.4 Usage of HTTP headers

7.1.4.1 General

Standard HTTP headers shall be used in accordance with clause 5.2.2 of TS 29.500 [30] for both HTTP/1.1 [26] and HTTP/2 [27] messages.

7.1.4.2 User Agent identification

Individual specifications referencing the present document shall specify the `User-Agent` HTTP request header used by the Media Session Handler to identify itself.

7.1.4.3 Server identification

Individual specifications referencing the present document shall specify the `Server` HTTP response header used by the Media AF to identify itself.

7.1.4.2 Cache control

All responses from the Media AF that carry a resource message body shall include:

- a strong entity tag for the resource, conveyed in an `ETag` response header as specified in section 8.8.3 of RFC 9110 [24],
- a resource modification timestamp, conveyed in a `Last-Modified` response header as specified in section 8.8.2 of [24], and
- a predicted time-to-live period for the resource, conveyed in a `Cache-Control: max-age` response header as specified in section 5.2.1.1 of RFC 9111 [25].

7.1.4.3 Support for conditional HTTP GET requests

All API endpoints on the Media AF that expose the HTTP `GET` method shall support conditional requests using the `If-None-Match` and `If-Modified-Since` request headers as specified in section 13 of [24]. API clients should not attempt to revalidate their cached copy of a resource using a conditional `GET` request before the indicated time-to-live period has elapsed.

7.1.4.4 Support for conditional HTTP POST, PUT, PATCH and DELETE requests

All API endpoints on the Media AF that expose the HTTP `POST`, `PUT`, `PATCH` or `DELETE` methods shall support conditional requests using the `If-Match` request header as specified in section 13.1.1 of RFC 9110 [24]. The API client should supply a strong entity tag previously obtained per clause 7.1.4.2 in this request header when invoking any of these HTTP methods.

7.1.5 HTTP message bodies for API resources

The OpenAPI [OpenAPI300] specification of request and response HTTP messages is contained in annex A.

7.1.6 HTTP response codes

Guidelines for error responses to the invocation of APIs of NF services are specified in clause 4.8 of TS 29.501 [31]. API-specific error responses are specified in the respective clauses of the present document.

7.1.7 HTTP error response message bodies

Error messages shall be conveyed to the API client using an HTTP response body of type *ProblemDetails*, as specified in clause 4.8.2 of TS 29.501 [31].

7.2 Explanation of API data model notation

The data models in the following API clauses are specified using the following notational conventions:

1. Data models are expressed as an unordered list of JSON properties [37] with one property defined in each row of the data model table.

2. The *Data type* column defines the type of the property, according to JSON [37].

3. The keyword "Array" in the *Data type* column indicates that zero or more elements of the data type in brackets are included. The number of elements in the array may additionally be constrained by normative text in the *Description* column.

4. The *Cardinality* column defines whether a property is optional or mandatory. An array with cardinality 0 indicates that the array property is optional in the data structure. An array with cardinality 1 indicates that the property is mandatory in the data structure, even when the array is empty.

5. The keyword "Object" in the *Data type* column indicates a structured sub-object of an unnamed type whose properties are defined inline in the indented table rows immediately afterwards. The "Object" type may be combined with the "Array" type.

6. In the case of data types specifying RESTful resources, the additional *Usage* column defines the property behaviour for each CRUD Operation as follows:

   - "C" (Create), "R" (Retrieve) and "U" (Update) refers to the CRUD procedure during which the property is present in the resource type. (A Destroy operation never takes any input data type in the body of the request message, nor does it return any response message body.)

   - "RO" signifies a read-only property. Only the API provider function is permitted to modify the property value. Any value supplied by the API invoker in the body of a request message is ignored by the API provider function. The property may be omitted from request message bodies, even if it is a mandatory property of the data type, i.e., the property is only mandatory in response message bodies.

   - "WO" signifies a write-only property. Only the API invoker is permitted to modify the property value. The API provider function should not populate this property in the body of any response message returned to the API invoker, even if it is a mandatory property of the data type, i.e., the property is only mandatory in request message bodies.

   - "RW" signifies a read/write property. The API provider and API invoker may both modify the property value.

7. An additional read-only property is included at the start of all data models defining resources that are members of a RESTful collection. This property is populated with the unique identifier of the resource within its parent collection, and corresponds to the leaf path element in the RESTful URL of that resource.

7.3 Common OpenAPI data types

7.3.1 General

The data types defined in this clause are intended to be used by more than one of the Media Delivery APIs.

7.3.2 Simple data types

Table 7.3.2-1 below specifies common simple data types used within the Media Delivery APIs, including a short description of each. In cases where types from other specifications are reused, a reference is provided.

Table 7.3.2-1: Simple data types

Type name	Type definition	Description	Reference
Uint6	`integer`	Integer where the allowed values correspond to the value range of an unsigned 6-bit integer.	Clause A.2
Uint8	`integer`	Integer where the allowed values correspond to the value range of an unsigned 8-bit integer.	Clause A.2
Uint20	`integer`	Integer where the allowed values correspond to the value range of an unsigned 20-bit integer.	Clause A.2
ResourceId	`string`	String chosen by the 5GMS AF to serve as an identifier in a resource URL.	Clause A.2.
Percentage	`number`	A percentage expressed as a floating-point value between 0.0 and 100.0 (inclusive).	Clause A.2.
DurationSec	`integer`	An unsigned integer identifying a period of time expressed in units of seconds.	TS 29.571 [12] table 5.2.2-1
Duration	`string`	A period of time expressed as a string compliant with the *duration* format specified in section 7.3.1 of the JSON Schema specification [38].	Clause A.2. IETF RFC 3339 [34] appendix A.
DateTime	`string`	An absolute date and time expressed using the OpenAPI *date-time* string format.	TS 29.571 [12] table 5.2.2-1
Uri	`string`	Uniform Resource Identifier conforming with the *URI-reference* production of the URI Generic Syntax.	TS 29.571 [12] table 5.2.2-1
Url	`string`	Uniform Resource Locator, conforming with the URI Generic Syntax.	RFC 3986 [41] section 4.1
RelativeUrl	`string`	Relative Uniform Resource Locator, conforming with the *relative-ref* production of the URI Generic Syntax. Both *query* and *fragment* suffixes are permitted.	RFC 3986 [35], section 4.2
AbsoluteUrl	`string`	Absolute Uniform Resource Locator, conforming with the *absolute-URI* production of the URI Generic Syntax in which the scheme part is *http* or *https*. The *query* suffix is permitted but the *fragment* suffix is not.	RFC 3986 [35], section 4.3
IPv4Addr	`string`	IPv4 address formatted in "dotted decimal" notation	TS 29.571 [33] table 5.2.2-1.
IPv6Addr	`string`	IPv6 address formatted in colon-separated hexadecimal quartet notation.	TS 29.571 [33] table 5.2.2-1.
Uinteger	`integer`	Unsigned integer.	TS 29.571 [33] table 5.2.2-1.
UintegerRm	`integer`	Unsigned integer (nullable).	TS 29.571 [33] table 5.2.2-1.
Uint32	`integer`	Unsigned integer between 0 and 4294967295 ($2^{32} - 1$).	TS 29.571 [33] table 5.2.2-1.
Dnn	`string`	Data Network Name.	TS 29.571 [33] table 5.3.2-1.
BitRate	`string`	A bit rate expressed as a string-encoded decimal value and unit.	TS 29.571 [33] table 5.5.2-1.
PacketLoss Rate	`integer`	An integer between 0 and 1000 encoding tenths of a percent	TS 29.571 [33] table 5.5.2-1.
MediaDelivery SessionId	`string`	A unique identifier for a media delivery session. This should not contain any user-identifiable data.	Clause A.2.

7.3.3 Structured data types

7.3.3.1 IpPacketFilterSet type

Table 7.3.3.1-1: Definition of type IpPacketFilterSet

Property name	Data type	Cardinality	Description
direction	`string`	1..1	Packet Filter Set direction.
sourceAddress	`IpAddr`	0..1	Source IP address or IPv6 prefix.
destinationAddress	`IpAddr`	0..1	Destination IP address or IPv6 prefix.
protocolNumber	`Uint8`	0..1	Transport layer protocol number as it appears in the Protocol field of the IPv4 packet header or the Next Header field of the IPv6 packet header.
sourcePort	`Uint16`	0..1	Source port number.
destinationPort	`Uint16`	0..1	Destination port number.
differentiatedServicesCode Point	`Uint6`	0..1	Differentiated Services Code Point.
flowLabel	`Uint20`	0..1	IPv6 Flow Label.
securityParametersIndex	`Uint32`	0..1	IPv6 Security Parameters Index (SPI).

7.3.3.2 ApplicationFlowDescription type

This data type is used to declare the properties of an application data flow to the Media AF during the course of a media delivery session. Its properties are used to describe a Service Data Flow to the 5G Core for the purpose of application traffic detection.

Table 7.3.3.2-1: Definition of type ApplicationFlowDescription

Property name	Data type	Cardinality	Description
filterMethod	`SdfMethod`	1..1	The filtering method used to identify packets belonging to this application flow (see clause 7.3.4.2).
packetFilter	`IpPacketFilterSet`	0..1	Description of the application flow in terms of packet header field values (see below).
domainName	`string`	0..1	Description of the application flow in terms of the Fully-Qualified Domain Name (FQDN) of the Media AS targeted at reference point M4 (see below).
mediaType	`MediaType`	0..1	The type of media carried by this application flow (see NOTE 1).
mediaTransportParameters	`Protocol Description`	0..1	The set of media transport protocol parameters to be used by the 5G Core for the purpose of PDU Set identification and/or end of data burst detection on this application flow (see NOTE 2).
NOTE 1: Enumeration *MediaType* is specified in clause 5.6.3.3 of TS 29.514 [18]. NOTE 2: Data type *ProtocolDescription* is specified in clause 5.5.4.13 of TS 29.571 [33].			

Exactly one of the following properties shall be populated: *packetFilter*, *domainName*.

7.3.3.3 M1UnidirectionalQoSSpecification type

Table 7.3.3.3-1: Definition of type M1UnidirectionalQoSSpecification

Property name	Data type	Cardinality	Usage	Description
maximumBitRate	`BitRate`	1..1	RO	Maximum bit rate supported by the 5G System. Populated by the Media AF.
maximumAuthorisedBitRate	`BitRate`	0..1	RW	Maximum bit rate authorised by the Media Application Provider.
minimumPacketLossRate	`PacketLossRate`	0..1	RW	Minimum packet loss rate permitted by the Media Application Provider, expressed in tenths of a percent.
pduSetQosLimits	`PDUSetQosPara`	0..1	RW	QoS limits for PDU Sets, including minimum delay budget and minimum error rate permitted by the Media Application Provider, and an indication by the Media Application Provider of whether the receiving application is sensitive to the loss of individual PDUs in a PDU Set (see NOTE).
NOTE: Data type *PDUSetQosPara* is specified in clause 5.5.4.11 of TS 29.571 [33].				

7.3.3.4 M1QoSSpecification type

Table 7.3.3.4-1: Definition of type M1QoSSpecification

Property name	Data type	Cardinality	Description
componentReference	`string`	1..1	A unique string identifying this QoS specification within the scope of its parent.
qosReference	`string`	0..1	As specified in clause 5.6.2.7 of TS 29.514 [18].
downlinkQosSpecification	`M1UnidirectionalQoSSpecification`	0..1	QoS specification in the downlink direction (see below and clause 7.3.3.3).
uplinkQosSpecification	`M1UnidirectionalQoSSpecification`	0..1	QoS specification in the uplink direction (see below and clause 7.3.3.3).
pduSetMarking	`boolean`	0..1	Indicates that packets at reference point M4 are required to include PDU Set marking if the media transport protocol supports this. Default value *false* if omitted.

At least one of the following properties shall be populated: *qosReference*, *downlinkQosSpecification*, *uplinkQosSpecification*.

7.3.3.5 M5BitRateSpecification type

Table 7.3.3.5-1: Definition of type M5BitRateSpecification

Property name	Data type	Cardinality	Description
maximumRequestedBitRate	`BitRate`	1..1	Maximum requested bit rate.
minimumDesiredBitRate	`BitRate`	0..1	Minimum desired bit rate.
minimumRequestedBitRate	`BitRate`	1..1	Minimum requested bit rate.

7.3.3.6 M5QoSSpecification type

Table 7.3.3.6-1: Definition of type M5QoSSpecification

Property name	Data type	Cardinality	Description
downlinkBitRates	`M5BitRate Specification`	1..1	Bit rate specification for the downlink direction (see clause 7.3.3.5).
uplinkBitRates	`M5BitRate Specification`	1..1	Bit rate specification for the uplink direction (see clause 7.3.3.5).
desiredPacketLatency	`number`	0..1	Desired packet latency in milliseconds, expressed as a positive floating-point value (see NOTE 1).
desiredPacketLossRate	`PacketLossRate`	0..1	Desired packet loss rate expressed in tenths of a percent (see NOTE 1).
desiredDownlinkPduSetQosParameters	`PDUSetQosPara`	0..1	Desired PDU Set QoS parameters for the downlink direction (see NOTE 2).
desiredUplinkPduSetQosParameters	`PDUSetQosPara`	0..1	Desired PDU Set QoS parameters for the uplink direction (see NOTE 2).
NOTE 1: Clause 5.6.2.7 of TS 29.514 [18] restricts packet latency and packet loss to be the same in the downlink and uplink directions for a given *MediaComponent* when the CHEM feature is not supported by the PCF. NOTE 2: Data type *PDUSetQosPara* is specified in clause 5.5.4.11 of TS 29.571 [33].			

7.3.3.7 ChargingSpecification type

Table 7.3.3.7-1: Definition of type ChargingSpecification

Property name	Data type	Cardinality	Description
sponsorId	`SponId`	0..1	As defined in clause 5.6.2.3 of TS 29.514 [18].
sponsoringStatus	`SponsoringStatus`	0..1	
permittedUes	`array(Gpsi)`	0..1	Set of UEs permitted to instantiate the parent Policy Template. If present, the array shall contain at least one member. If absent, all UEs are permitted.

7.3.3.8 TypedLocation type

Table 7.3.3.8-1: Definition of TypedLocation type

Property name	Data type	Cardinality	Description
locationIdentifierType	`CellIdentifierType`	1..1	The type of cell location present in the *location* property.
location	`string`	1..1	Identifies the cell location.

7.3.3.9 OperationSuccessResponse type

Table 7.3.3.9-1: Definition of OperationSuccessResponse type

Property name	Type	Cardinality	Description
success	`boolean`	1..1	Indicates whether an operation was successful (*true*) or not (*false*).
reason	`string`	0..1	Optional explanation of the success or otherwise of the operation.

7.3.3.10 EdgeProcessingEligibilityCriteria type

Table 7.3.3.10-1: Definition of EdgeProcessingEligibilityCriteria type

Property name	Type	Cardinality	Description
applicationFlow Descriptions	`array(Application FlowDescription)`	1..1	A set of application flow descriptions that are to be used as triggers for invoking edge media processing (see NOTE 1). If the set is empty, edge media processing may be invoked for an otherwise eligible media stream session on any service data flow. Valid *ApplicationFlowDescription* elements: - *domainName* - *packetFilter.destinationAddress* and *packetFilter.destinationPort* - *packetFilter.differentiatedServices* - *packetFilter.flowLabel* Other *ApplicationFlowDescription* settings shall be rejected by the Media AF.
ueLocations	`array(Location Area5G)`	1..1	A set of geographical areas in which edge media processing is to be triggered when a UE is present (see NOTE 2). If the set is empty, edge media processing may be invoked for an otherwise eligible media stream session in any location.
timeWindows	`array(TimeWindow)`	1..1	Edge media processing is triggered when the media streaming session is taking place during one of the indicated time windows (see NOTE 2). If the set is empty, edge media processing may be invoked for an otherwise eligible media stream session at any time.
appRequest	`boolean`	1..1	When set *true*, edge media processing is to be triggered based on application request only.
NOTE 1: The usage of these fields to influence route selection and EAS reselection are for future study. NOTE 2: Data types *LocationArea5G* and *TimeWindow* are defined in TS 24.558 [14].			

7.3.3.11 EndpointAddress type

Table 7.3.3.11-1: Definition of EndpointAddress type

Property name	Type	Cardinality	Description
domainName	`string`	1..1	Internet domain name of the endpoint.
portNumbers	`array(Uint16)`	1..1	Port number of each endpoint. The array shall contain at least one member.

7.3.3.12 M1MediaEntryPoint type

Table 7.3.3.12-1: Definition of type M1MediaEntryPoint

Property name	Data type	Cardinality	Description
relativePath	`RelativePath`	1..1	A relative path (i.e., without a scheme or any leading forward slash characters) to the Media Entry Point document resource. The semantics are dependent on the value of the *contentType* or *protocol* property.
contentType	`string`	1..1	The MIME content type of this Media Entry Point. This property shall be mutually exclusive with *protocol*.
protocol	`Uri`	1..1	A fully-qualified term identifier URI that identifies the media delivery protocol at reference point M4 for this Media Entry Point. This property shall be mutually exclusive with *contentType*.
profiles	`array(Uri)`	0..1	An optional list of conformance profile identifiers associated with this Media Entry Point, each one expressed as a URI. A profile URI may indicate an interoperability point, for example. If present, the array shall contain at least one item.

7.3.3.13 CachingConfiguration type

Table 7.3.3.13-1: Definition of type CachingConfiguration

Property name	Data type	Cardinality	Description
urlPatternFilter	`string`	1..1	A pattern used to match media resource URLs to determine whether a given media resource falls within the scope of this caching configuration. The format of the pattern shall be a regular expression as specified in [36].
cachingDirectives	`object`	1..1	If a *urlPatternFilter* applies to a resource, then the provided *cachingDirectives* shall be applied by the Media AS.
statusCodeFilters	`array(integer)`	0..1	The set of HTTP response status codes to which these caching directives apply. If the property is present, the array shall contain at least one item. If absent, the caching directives shall apply to all HTTP response status codes.
noCache	`boolean`	0..1	If set to *true*, media resources falling within the scope of these caching directives shall be marked as not to be cached when served by the Media AS. Default value if omitted: *false*.
maxAge	`Unit32`	0..1	The caching time-to-live period, expressed in seconds, of media resources falling within the scope of these caching directives. This determines the minimum period for which the Media AS shall cache matching media resources. If *noCache* is *false*, it also determines the time-to-live period signalled by the Media AS when it serves such media resources.

7.3.3.14 BDTWindow type

Table 7.3.3.14-1: Definition of BDTWindow type

timeWindow	`TimeWindow`	1..1	The absolute start date-time and stop date-time of the Background Data Transfer window populated by the Media AF (see NOTE).
maximumDownlink BitRate	`BitRate`	0..1	The maximum bit rate that the Media Client requests or is authorised to use in the downlink direction during *timeWindow*.
maximumUplink BitRate	`BitRate`	0..1	The maximum bit rate that the Media Client requests or is authorised to use in the uplink direction during *timeWindow*.
NOTE: Data type *TimeWindow* is defined in TS 29.122 [20].			

7.3.4 Enumerated data types

7.3.4.1 CellIdentifierType enumeration

The *CellIdentifierType* enumeration indicates the type of cell identifier as defined in TS 23.003 [7].

Table 7.3.4.1-1: Definition of CellIdentifierType enumeration

Enumeration value	Description
CGI	Cell Global Identification.
ECGI	E-UTRAN Cell Global Identification.
NCGI	NR Cell Global Identity.

7.3.4.2 SdfMethod enumeration

Table 7.3.4.2-1: Definition of SdfMethod enumeration

Enumeration value	Description
5_TUPLE	Service Data Flow described by source (Media Access Function) IP address and port number, destination (Media AS) IP address and port number and protocol number. Wildcard values are not permitted in the Service Data Flow description.
2_TUPLE	Service Data Flow described by source (Media Access Function) IP address and destination (Media AS) IP address duple.
TYPE_OF_SERVICE_MARKING	Service Data Flow described by Type of Service (TOS) marking.
FLOW_LABEL	Service Data Flow described by Ipv6 flow label marking.
DOMAIN_NAME	Service Data Flow described by a domain name.

7.3.4.3 ProvisioningSessionType enumeration

Table 7.3.4.3-1: Definition of ProvisioningSessionType enumeration

Enumeration value	Description
MS_DOWNLINK	Downlink media streaming
MS_UPLINK	Uplink media streaming
RTC	Real-time media communication (RTC)

7.3.4.4 EASRelocationTolerance enumeration

Table 7.3.4.4-1: Definition of EASRelocationTolerance enumeration

Enumeration value	Description
RELOCATION_UNAWARE	The application is not aware of any EAS relocation that may happen. Relocation procedures may be executed without any restrictions.
RELOCATION_TOLERANT	The application may tolerate EAS relocation, but requirements for the relocation procedure must be met. An application context may need to be transferred.
RELOCATION_INTOLERANT	The application does not tolerate relocation.

7.3.4.5 ContentTransferMode enumeration

Table 7.3.4.5-1: Definition of ContentTransferMode enumeration

Enumeration value	Description
PULL	Content is pulled by or from the Media AS.
PUSH	Content is pushed into or by the Media AS.

7.4 Security

7.4.1 General

The Media AF shall enable secure provision of information in the Media Delivery System by authenticated and authorised Media-aware Applications or Media Application Providers.

7.4.2 Authorising Media Application Provider access to the Media AF at reference point M1

When a Media Application Provider deployed outside the Trusted DN attempts to access a Media AF deployed inside the Trusted DN, the Media Delivery System shall authenticate and authorise the Media Application Provider.

Access to the *Maf_Provisioning* API of the Media AF by the Media Application Provider at reference point M1 may be authorised by means of the OAuth 2.0 protocol specified in RFC 6749 [47]), using the *client credentials* authorization grant.

NOTE: The provisioning and negotiation of the security method is not specified in this release.

When CAPIF (see TS 29.222 [48]) is used for external API exposure:

- The CAPIF core function shall play the role of authorization server, the Media AF shall play the role of resource server and the Media Application Provider shall play the role of client.

- Before invoking any service operation exposed by the Media AF, the Media Application Provider shall negotiate the security method (PKI, TLS-PSK or OAuth 2.0) with the CAPIF core function and shall ensure that the Media AF has the required credentials to authenticate access tokens subsequently presented by the Media Application Provider (see clauses 5.6.2.2 and 6.2.2.2 of TS 29.222 [48]).

- If PKI or TLS-PSK is the selected security method between the Media Application Provider and the Media AF shall, upon invocation of a *Maf_Provisioning* service operation by the Media Application Provider at reference point M1, retrieve the authorisation information from the CAPIF core function as described in clause 5.6.2.4 of TS 29.222 [48].

- If OAuth 2.0 [47] is the selected security method between the Media Application Provider and the Media AF, the Media Application Provider shall, prior to invoking *Maf_Provisioning* service operations on the Media AF at reference point M1, obtain an access token from the authorization server (CAPIF core function) by invoking the *Obtain_Authorization* service operation specified in clause 5.6.2.3.2 of TS 29.222 [48].

Otherwise:

- The Media AF shall play the role of both authorization server and resource server, and the Media Application Provider shall play the role of client.
- The Media Application Provider shall obtain an access token from the authorization server (Media AF) using the client credentials authorization grant specified in section 4.4 of RFC 6749 [47] prior to invoking *Maf_Provisioning* service operations on the resource server (Media AF) at reference point M1.

7.4.3 Authorising Media Session Handler access to the Media AF at reference point M5

When a Media Session Handler deployed in a Media Client attempts to access a Media AF deployed inside the Trusted DN, the Media Delivery System shall authenticate and authorise the Media Session Handler.

Access to the *Maf_SessionHandling* API of the Media AF by the Media Session Handler at reference point M5 shall be authorised by means of the OAuth 2.0 protocol specified in RFC 6749 [47], using the *client credentials* or *authorization code* flow grant types.

NOTE: The provisioning and negotiation of the security method is not specified in this release.

When CAPIF (see TS 29.222 [48]) is used for external API exposure:

- The CAPIF core function shall play the role of authorization server, the Media AF shall play the role of resource server and the Media Session Handler shall play the role of client.
- Before invoking any service operation exposed by the Media AF, the Media Application Provider shall negotiate the security method (PKI, TLS-PSK or OAuth 2.0) with the CAPIF core function and shall ensure that the Media AF has the required credentials to authenticate access tokens subsequently presented by the Media Application Provider (see clauses 5.6.2.2 and 6.2.2.2 of TS 29.222 [48]).
- If PKI or TLS-PSK is the selected security method between the Media Session Handler and the Media AF shall, upon invocation of a *Maf_SessionHandling* service operation by the Media Session Handler at reference point M1, retrieve the authorisation information from the CAPIF core function as described in clause 5.6.2.4 of TS 29.222 [48].
- If OAuth 2.0 [47] is the selected security method between the Media Session Handler and the Media AF, the Media Session Handler shall, prior to invoking *Maf_SessionHandling* service operations on the Media AF at reference point M5, obtain an access token from the authorization server (CAPIF core function) by invoking the *Obtain_Authorization* service operation specified in clause 5.6.2.3.2 of TS 29.222 [48].

Otherwise:

- Either the Media AF shall play the role of both authorization server and resource server, or the Media AF shall play the role of the resource server and the Media Application Provider plays the role of the authorization server. The Media Session Handler shall play the role of client.
- The Media Session Handler shall obtain an access token from the authorization server using either the client credentials grant specified in section 4.4 of RFC 6749 [47] or the authorization code grant specified in section 4.1 of [47] prior to invoking *Maf_Provisioning* service operations on the resource server (Media AF) at reference point M5.

8 Maf_Provisioning service

8.1 Overview

This clause defines the provisioning API used by a Media Application Provider at reference point M1 to configure downlink or uplink Media Delivery services. The corresponding OpenAPI definitions for the *Maf_Provisioning* service are specified in clause A.3. A summary of the resource structure is shown in table 8.1-1 below. The default endpoint address of the Media AF at reference point M1 is specified in clause 7.1.2.1.

Table 8.1-1: Resource structure of Maf_Provisioning APIs

HTTP request path element hierarchy	Description	Allowed HTTP methods					Resource structure definition clause	OpenAPI definition clause
		Create	Retrieve	Update	Destroy	Non-RESTful operation		
`provisioning-sessions`	Provisioning Sessions collection	POST	GET				8.2.2	A.3.1
{provisioningSessionId}	Provisioning Session resource		GET		DELETE			
`content-protocols`	Content Protocols resource		GET				8.3.2	A.3.2
`certificates`	Server Certificates collection	POST					8.4.2	A.3.3
{certificateId}	Server Certificate resource		GET	PUT	DELETE			
`content-preparation-templates`	Content Preparation Templates collection	POST					8.5.2	A.3.4
{contentPreparationTemplateId}	Content Preparation Template resource		GET	PUT, PATCH	DELETE			
`edge-resources-configurations`	Edge Resources Configurations collection	POST					8.6.2	A.3.5
{edgeResourcesConfigurationId}	Edge Resources Configuration resource		GET	PUT, PATCH	DELETE			
`policy-templates`	Policy Templates collection	POST					8.7.2	A.3.6
{policyTemplateId}	Policy Template resource		GET	PUT, PATCH	DELETE			
`content-hosting-configuration`	Content Hosting Configuration resource	POST	GET	PUT, PATCH	DELETE		8.8.2	A.3.7
`purge`	Content Hosting cache purge operation					POST		

content-publishing-configuration	Content Publishing Configuration resource	POST	GET	PUT, PATCH	DELETE		8.9.2	A.3.8
purge	Content Publishing cache purge operation					POST		
rtc-configuration	RTC Configuration resource	POST	GET	PUT, PATCH	DELETE		8.10.2	A.3.9
metrics-reporting-configuration	Metrics Reporting Configuration collection	POST					8.11.2	A.3.10
{metricsReportingConfigurationId}	Metrics Reporting Configuration resource		GET	PUT, PATCH	DELETE			
consumption-reporting-configuration	Consumption Reporting Configuration resource	POST	GET	PUT, PATCH	DELETE		8.12.2	A.3.11
event-data-processing-configurations	Event Data Processing Configuration collection	POST					8.13.2	A.3.12
{eventDataProcessingConfigurationId}	Event Data Processing Configuration resource		GET	PUT, PATCH	DELETE			

8.2 Provisioning Sessions API

8.2.1 Overview

The Provisioning Sessions API is used by the 5GMS Application Provider to instantiate and manipulate Provisioning Sessions in the 5GMS System, as described in clause 5.2.2. Having created a Provisioning Session, the 5GMS Application Provider then goes on to discover the content protocols it supports using the API specified in clause 8.3, and to provision other Media Delivery features in the context of that Provisioning Session, using the APIs specified in clause 8.4 *et seq*. Certain of these features are only applicable to the type of Provisioning Session created.

8.2.2 Resource structure

The Provisioning Sessions API is accessible through the following URL base path:

{apiRoot}`/3gpp-maf-provisioning/`*{apiVersion}*`/provisioning-sessions/`

Table 8.2.2-1 specifies the operations and the corresponding HTTP methods that are supported by this API. In each case, the sub-resource path specified in the second column of the table shall be appended to the above URL base path.

Table 8.2.2-1: Operations supported by the Provisioning Sessions API

Operation name	Sub-resource path	Allowed HTTP method(s)	Description
Enumerate Provisioning Sessions		`GET`	Enumerate the resource identifiers of the Provisioning Session collection.
Create Provisioning Session		`POST`	Create a new Provisioning Session resource.
Retrieve Provisioning Session	*{provisioningSessionId}*	`GET`	Retrieve an existing Provisioning Session resource for inspection.
Destroy Provisioning Session		`DELETE`	Destroy an existing Provisioning Session resource.

8.2.3 Data model

8.2.3.1 ProvisioningSession resource

Different properties are present in the *ProvisioningSession* resource depending on the type of Provisioning Session indicated in the *provisioningSessionType* property, and this is specified in the *Applicability* column.

Table 8.2.3.1-1: Definition of ProvisioningSession resource

Property name	Type	Cardinality	Usage	Description	Applicability
provisioningSessionId	`ResourceId`	1..1	C: RO R: RO U: –	A unique identifier assigned to this Provisioning Session by the Media AF on creation.	All types.

provisioningSessionType	`ProvisioningSessionType`	1..1	C: RW R: RW U: –	The type of Provisioning Session.	All types.
externalServiceId	`string`	1..1	C: RW R: RO U: RW	An identifier, nominated by the Media Application Provider, that identifies this Provisioning Session to the Media Client. Every Provisioning Session in a Media Delivery System shall have a different service identifier. Takes the form of a reverse FQDN e.g., `com.provider.service` to ensure global applicability across different Media Delivery Systems. Used by the Media Session Handler to invoke the network media session handling operations specified in clause 5.3 and clause 9, in particular to fetch full Service Access Information from the Media AF (see clauses 5.3.2 and 9.2).	All types.
aspId	`AspId`	0..1	C: RW R: RW U: –	The identity of the Application Service Provider responsible for this Provisioning Session, as specified in clause 5.6.2.3 of TS 29.514 [18].	All types.
appId	`ApplicationId`	1..1	C: RW R: RW U: –	The Application Identifier (see table 5.4.2-1 of TS 29.571 [33]) to which this Provisioning Session pertains. The same <*aspId, appId*> duple may be present in several Provisioning Sessions in a given 5GMS System. Used as the AF Application identifier (see clause 5.6.2.3 of TS 29.514 [18]) for PCF interactions. When a 5GMS AF in the Trusted DN is provisioned from outside the Trusted DN, the NEF is responsible for mapping an external Application Identifier to the corresponding internal AF Application Identifier known to the PCF.	All types.
locationReporting	`boolean`	0..1		If *true*, the Media Session Handler is required to populate UE location information in Dynamic Policy interactions (see clause 9.3.3.1), Network Assistance interactions (see clause 9.4.3.1), QoE metrics reporting interactions (see clause 9.5.3) and consumption reporting interactions (see clause 9.6.3.2) with a Media AF deployed by an MNO or trusted third party. If *false* or omitted, UE location shall not be populated by the Media Session Handler in any of the abovementioned interactions.	All types.
serverCertificateIds	`array(ResourceId)`	0..1	C: RO R: RO U: –	A list of Server Certificate identifiers currently associated with this Provisioning Session.	*MS_DOWNLINK, MS_UPLINK RTC*
contentPreparationTemplateIds	`array(ResourceId)`	0..1	C: RO R: RO U: –	A list of Content Preparation Template identifiers currently associated with this Provisioning Session.	*MS_DOWNLINK, MS_UPLINK*
metricsReportingConfigurationIds	`array(ResourceId)`	0..1	C: RO R: RO U: –	A list of Metrics Reporting Configuration identifiers currently associated with this Provisioning Session.	*MS_DOWNLINK, MS_UPLINK, RTC*
policyTemplateIds	`array(ResourceId)`	0..1	C: RO R: RO U: –	A list of Policy Template identifiers currently associated with this Provisioning Session.	*MS_DOWNLINK, MS_UPLINK, RTC*

edgeResources ConfigurationIds	`array(ResourceId)`	0..1	C: RO R: RO U: –	A list of Edge Resources Configuration identifiers currently associated with this Provisioning Session.	*MS_DOWNLINK, MS_UPLINK, RTC*
eventDataProcessing ConfigurationIds	`array(ResourceId)`	0..1	C: RO R: RO U: –	A list of Event Data Processing Configuration identifiers currently associated with this Provisioning Session.	*MS_DOWNLINK, MS_UPLINK*

8.3 Content Protocols Discovery API

8.3.1 Overview

The Content Protocols Discovery API is used by a 5GMS Application Provider to find out which content ingest or egest protocols are supported by the Media AS instance(s) associated with a Provisioning Session in the Media AF.

8.3.2 Resource structure

The Content Protocols Discovery API is accessible through the following URL base path:

{apiRoot}`/3gpp-maf-provisioning/`*{apiVersion}*`/provisioning-sessions/`*{provisioningSessionId}*`/`

Table 8.3.2-1 below specifies the operations and the corresponding HTTP methods that are supported by this API. In each case, the Provisioning Session identifier shall be substituted into *{provisioningSessionId}* in the above URL template and the sub-resource path specified in the second column of the table shall be appended to the URL base path.

Table 8.3.2-1: Operations supported by the Ingest Protocols Discovery API

Operation name	Sub-resource path	Allowed HTTP method(s)	Description
Fetch list of supported content protocols	`content-protocols`	`GET`	Retrieve a list of supported content protocols.

8.3.3 Data model

8.3.3.1 ContentProtocols resource

Table 8.3.3.1-1: Definition of ContentProtocols resource

Property name	Data Type	Cardinality	Description
downlinkIngestProtocols	`array(Content ProtocolDescriptor)`	0..1	An set of *ContentProtocolDescriptor* objects, as specified in clause 8.3.3.2, each one uniquely identifying a content ingest protocol supported at reference point M2 by the Media AS instance(s) associated with the parent Provisioning Session. If present, the array shall contain at least one member.
uplinkEgestProtocols	`array(Content ProtocolDescriptor)`	0..1	An set of *ContentProtocolDescriptor* objects, as specified in clause 8.3.3.2, each one uniquely identifying a content egest protocol supported at reference point M2 by the Media AS instance(s) associated with the parent Provisioning Session. If present, the array shall contain at least one member.
downlinkDistribution Protocols	`array(Content ProtocolDescriptor`	0..1	A set of *ContentProtocolDescriptor* objects, as specified in clause 8.3.3.2, each one uniquely identifying a distribution protocol supported at reference point M4 by the Media AS instance(s) associated with the parent Provisioning Session. If present, the array shall contain at least one member.
uplinkContribution Protocols	`array(Content ProtocolDescriptor`	0..1	A set of *ContentProtocolDescriptor* objects, as specified in clause 8.3.3.2, each one uniquely identifying a contribution protocol supported at reference point M4 by the Media AS instance(s) associated with the parent Provisioning Session. If present, the array shall contain at least one member.
geoFencingLocatorTypes	`array(Uri)`	0..1	A set of fully-qualified term identifiers, each one indicating a content geo-fencing locator type supported at reference point M2 by the Media AS instance(s) associated with the parent Provisioning Session. (See clause B.1.) If present, the array shall contain at least one member.

8.3.3.2 ContentProtocolDescriptor type

Table 8.2.3.2-1: Definition of ContentProtocolDescriptor type

Property name	Data Type	Cardinality	Description
termIdentifier	`Uri`	1..1	A fully-qualified term identifier indicating support for a content protocol (see NOTE).
descriptionLocator	`AbsoluteUrl`	0..1	The location of a description of the content protocol, for example the public web URL of its specification.
NOTE: The controlled vocabulary of terms identifying 5G Media Streaming content ingest and content egest protocols at reference point M2 is specified in clause 8 of TS 26.512 [6]. The controlled vocabulary of terms identifying 5G Media Streaming content distribution and content contribution protocols at reference point M4 is specified in clause 10 of TS 26.512 [6].			

8.4 Server Certificates provisioning API

8.4.1 Overview

The Server Certificates Provisioning API is used to provision X.509 [10] server certificates that can be referenced by a Content Hosting Configuration and subsequently presented by the Media AS when it distributes content to Media Clients at reference point M4 using Transport Layer Security [13]. Server Certificate resources are provisioned within the scope of an enclosing Provisioning Session.

8.4.2 Resource structure

The Server Certificates Provisioning API is accessible through the following URL base path:

{apiRoot}/3gpp-maf-provisioning/*{apiVersion}*/provisioning-sessions/*{provisioningSessionId}*/

Table 8.4.2-1 specifies the operations and the corresponding HTTP methods that are supported by this API. In each case, the Provisioning Session identifier shall be substituted into *{provisioningSessionId}* in the above URL template and the sub-resource path specified in the second column shall be appended to the URL base path.

Table 8.4.2-1: Operations supported by the Server Certificates Provisioning API

Operation name	Sub-resource path	Allowed HTTP method(s)	Description
Create Server Certificate	certificates	POST	Invoked on the Server Certificates collection associated with a Provisioning Session to request that the Media AF creates a new Server Certificate on behalf of the Media Application Provider.
Reserve Server Certificate	certificates?csr	POST	Invoked on the Server Certificates collection associated with a Provisioning Session to solicit a Certificate Signing Request for a new Server Certificate.
Retrieve Server Certificate	certificates/*{certificateId}*	GET	Used to retrieve a previously created or uploaded Server Certificate.
Upload Server Certificate		PUT	Used by the Media Application Provider to supply a new Server Certificate in response to a Certificate Signing Request previously solicited from the Media AF.
Destroy Server Certificate		DELETE	Removes the specified Server Certificate from the set of certificates associated with the Provisioning Session.
NOTE: The Server Certificate resource identifier *{certificateId}* may differ from the serial number of the X.509 certificate.			

8.4.3 Data model

8.4.3.1 Certificate Signing Request

The Certificate Signing Request shall comply with the Privacy-Enhanced Mail (PEM) textual format specified in RFC 7468 [12], i.e. a Base64-encoded DER certificate request or certificate, including leading and trailing encapsulation boundary lines.

The MIME content type shall be *application/x-pem-file*.

8.4.3.2 Server Certificate resource

The Server Certificate resource shall comply with the Privacy-Enhanced Mail (PEM) textual format specified in RFC 7468 [12], i.e. a Base64-encoded DER certificate, including leading and trailing encapsulation boundary lines. The resource shall include only the public parts of the X.509 certificate [10]. In particular, the private key shall not be included.

The MIME content type shall be *application/x-pem-file*.

8.5 Content Preparation Templates provisioning API

8.5.1 Overview

Content Preparation Templates are used to specify manipulations applied by a Media AS to downlink media resources ingested at reference point M2 for distribution at reference point M4, or to uplink media resources contributed at reference point M4 for egest at reference point M2. The Content Preparation Templates Provisioning API is used to provision a Content Preparation Template within the scope of a Provisioning Session that can subsequently be referenced from a Content Hosting Configuration or Content Publishing Configuration.

8.5.2 Resource structure

The Content Preparation Templates Provisioning API is accessible through the following URL base path:

{apiRoot}`/3gpp-maf-provisioning/`*{apiVersion}*`/provisioning-sessions/`*{provisioningSessionId}*`/`

Table 8.5.2-1 specifies the operations and the corresponding HTTP methods that are supported by this API. In each case, the Provisioning Session identifier shall be substituted into *{provisioningSessionId}* in the above URL template and the sub-resource path specified in the second column shall be appended to the URL base path.

Table 8.5.2-1: Operations supported by the Content Preparation Templates Provisioning API

Operation name	Sub-resource path	Allowed HTTP method(s)	Description
Create Content Preparation Template	`content-preparation-templates`	`POST`	Invoked on a Content Preparation Templates collection when supplying a new Content Preparation Template resource.
Retrieve Content Preparation Template	`content-preparation-templates/` *{contentPreparationTemplateId}*	`GET`	Retrieve a specific Content Preparation Template resource.
Update Content Preparation Template		`PUT, PATCH`	Modify an existing Content Preparation Template resource.
Destroy Content Preparation Template		`DELETE`	Destroy an existing Content Preparation Template resource.

8.5.3 Data model

The data model of the Content Preparation Template resource shall be determined by its MIME content type.

8.6 Edge Resources provisioning API

8.6.1 General

The Edge Resources Provisioning API is used by the Media Application Provider to provision edge resource usage for media streaming sessions associated with the parent Provisioning Session. The information serves as a template to select or instantiate the appropriate Media AS EAS instance that will serve the media session to the UE.

8.6.2 Resource structure

The Edge Resources API is accessible through the following URL base path:

{apiRoot}`/3gpp-maf-provisioning/`*{apiVersion}*`/provisioning-sessions/`*{provisioningSessionId}/*

Table 8.6.2-1 specifies the operations and the corresponding HTTP methods that are supported by the Edge Resources API. In each case, the Provisioning Session identifier shall be substituted into ***{provisioningSessionId}*** in the above URL template and the sub-resource path indicated by the second column of the table shall be appended to the resulting URL base path.

Table 8.6.2-1: Operations supported by the Edge Resources API

Operation name	Sub-resource path	Allowed HTTP method(s)	Description
Configure Edge Resources	`edge-resources-configurations`	`POST`	Invoked on the Edge Resources Configurations collection to create a new Edge Resources Configuration resource.
Retrieve Edge Resources Configuration	`edge-resources-configurations/` *{edgeResourcesConfigurationId}*	`GET`	Retrieve a specific Edge Resources Configuration resource.
Modify Edge Resources Configuration		`PUT`, `PATCH`	Modify or replace an existing Edge Resources Configuration resource.
Destroy Edge Resources Configuration		`DELETE`	Destroy an existing Edge Resources Configuration resource.

8.6.3 Data model

8.6.3.1 EdgeResourcesConfiguration resource type

Table 8.6.3.1-1: Definition of EdgeResourcesConfiguration resource type

Property name	Type	Cardinality	Description
edgeResourcesConfigurationId	`ResourceId`	1..1	A resource identifier for this Edge Resources Configuration assigned by the Media AF when the resource is created that is unique within the scope of the enclosing Provisioning Session.
edgeManagementMode	`EdgeManagementMode`	1..1	Indicates whether the management of edge resources is client-driven or AF-driven. (See clause 8.6.3.2.)
eligibilityCriteria	`EdgeProcessing EligibilityCriteria`	0..1	Condition to activate edge resources for this Provisioning Session. If omitted, it shall be assumed that all media delivery sessions related to the parent Provisioning Session use edge resources. (See clause 7.3.3.10.)
easRequirements	`EASRequirements`	1..1	Requirements on the EAS Profile used by the Media AF or by the EEC to discover and select one or more Media EAS instances to serve media streaming sessions. (See clause 8.6.3.3.)
easRelocationRequirements	`M1EASRelocation Requirements`	0..1	EAS relocation tolerance and requirements. If not present, the Media AF shall assume that the application is unaware of context transfer and that transfers to a target Media EAS are allowed. (See clause 8.6.3.4.)

8.6.3.2 EdgeManagementMode enumeration

Table 8.6.3.2-1: Definition of EdgeManagementMode enumeration

Enumeration value	Description
EM_AF_DRIVEN	The Media AF, in coordination with the Media Session Handler, assigns edge resources and directs application traffic to the Media EAS instance transparently to the application running on the UE.
EM_CLIENT_DRIVEN	An Application Client running on the UE explicitly manages edge resources via the EES at reference point EDGE-1.

8.6.3.3 EASRequirements type

Table 8.6.3.3-1: Definition of EASRequirements type

Property name	Type	Cardinality	Description
easProviderIds	`array(string)`	0..1	The set of acceptable providers of Media EAS instances associated with this Provisioning Session. If empty, EAS instances from any provider are acceptable.
easId	`string`	0..1	The Application Identifier (e.g., in the form of a URI or Fully-Qualified Domain Name) of a set of EAS instances, or of a particular EAS instance associated with this Provisioning Session.
easType	`string`	0..1	The type of Media EAS instances associated with this Provisioning Session.
easFeatures	`array(string)`	0..1	Media EAS service features required to be supported by EAS instances associated with this Provisioning Session. If empty, Media EAS instances of the specified *easType* with any feature set are acceptable.
serviceKpi	`EASServiceKPI`	0..1	Media AS service characteristics required to be satisfied by EAS instances associated with this Provisioning Session. If omitted, Media EAS instances with any service characteristics are acceptable.
serviceArea	`Geographical ServiceArea`	0..1	The list of geographical areas that Media EAS instances associated with this Provisioning Session are required to serve. If omitted, Media EAS instances shall serve all geographical areas whenever possible.
serviceAvailability Schedule	`array(Scheduled CommunicationTime)`	0..1	The required availability schedule for Media EAS instances associated with this Provisioning Session. If omitted, Media EAS instances are required to be available at all times.
serviceContinuity Scenarios	`array(ACRScenario)`	0..1	The Application Context Relocation scenarios that Media EAS instances associated with this Provisioning Session are required to support for service continuity. If omitted Media EAS instances are not required to support service continuity across EAS relocation.
NOTE: Data types *ScheduledCommunicationTime*, *GeographicalServiceArea*, *EASServiceKPI*, and *ACRScenario* are defined in TS 29.558 [15].			

8.6.3.4 M1EASRelocationRequirements type

Table 8.6.3.4-1: Definition of M1EASRelocationRequirements type

Property name	Type	Cardinality	Description
tolerance	`EASRelocation Tolerance`	1..1	Indicates whether the Media EAS instance tolerates Application Context Relocation. (See clause 7.3.4.4.) If set to *RELOCATION_INTOLERANT*, the other properties in this data type shall be ignored.
maxInterruption Duration	`UintegerRm`	0..1	The maximum downtime (expressed in milliseconds) that an application can tolerate during Media EAS relocation. If the expected downtime of the application is expected to exceed this duration, relocation of the Media EAS instance shall not be performed.
maxResponseTime Difference	`UintegerRm`	0..1	The maximum allowed difference between the previously experienced average User Plane network latency to the source Media EAS instance and the expected latency to the target Media EAS instance, expressed in milliseconds.

8.7 Policy Templates provisioning API

8.7.1 Overview

The Policy Templates Provisioning API allows a Media Application Provider to configure a set of Policy Templates within the scope of a Provisioning Session that may subsequently be applied to downlink or uplink media delivery sessions belonging to that Media Application Provider using the Dynamic Policy API specified in clause 9.3. A Policy Template is used to specify the traffic shaping and charging policies to be applied to these media streaming sessions, as specified in clause 5.2.7.1.

8.7.2 Resource structure

The Policy Templates Provisioning API is accessible through the following URL base path:

{apiRoot}/3gpp-maf-provisioning/*{apiVersion}*/provisioning-sessions/*{provisioningSessionId}*/

Table 8.7.2-1 below specifies the operations and the corresponding HTTP methods that are supported by this API. In each case, the Provisioning Session identifier shall be substituted into *{provisioningSessionId}* in the above URL template and the sub-resource path specified in the second column shall be appended to the URL base path.

Table 8.7.2-1: Operations supported by the Policy Template Provisioning API

Operation name	Sub-resource path	Allowed HTTP method(s)	Description
Create Policy Template	policy-templates	POST	Create a new Policy Template resource within the scope of a Provisioning Session.
Fetch Policy Template	policy-templates/*{policyTemplateId}*	GET	Retrieve an existing Policy Template resource.
Update Policy Template		PUT, PATCH	Modify the configuration of an existing Policy Template.
Destroy Policy Template		DELETE	Destroy an existing Policy Template resource.

8.7.3 Data model

8.7.3.1 PolicyTemplate resource

Table 8.7.3.1-1: Definition of PolicyTemplate resource

Property	Type	Cardinality	Usage	Description
policyTemplateId	ResourceId	1..1	C: RO R: RO U: RO	Resource identifier of this Policy Template assigned by the Media AF that is unique within the scope of the Provisioning Session.
state	string enum	1..1	C: RO R: RO U: RO	Current state of this Policy Template (see clause 5.2.7.2) exposed to the 5GMS Application Provider by the Media AF. Only a Policy Template in the *READY* state may be instantiated as a Dynamic Policy Instance and applied to media streaming sessions.

Property	Type	Cardinality	Usage	Description
stateReason	`Problem Details`	1..1	C: RO R: RO U: RO	Additional details about the current state of this Policy Template exposed to the Media Application Provider by the Media AF. The *instance* sub-property shall be present and shall indicate the URL of this Policy Template resource at reference point M1. The *title* sub-property shall be present and shall indicate a human-readable representation of the *state* property specified above, e.g., "Policy Template ready for use" or "Policy Template invalid". The *detail* sub-property shall be present and shall indicate a human-readable status/error message. All other properties shall be omitted.
externalReference	`string`	1..1	C: RW R: RW U: RW	Additional identifier for this Policy Template, unique within the scope of its Provisioning Session, that may be cross-referenced with external metadata about a media delivery session. Example: "HD_Premium".
applicationSessionContexts	`array(object)`	0..1	C: RW R: RW U: RW	Exactly one application session context at reference point M4 to which this Policy Template may be applied. Each object in the array shall specify at least one property. If more than one property is specified, instantiation of the Policy Template is restricted to the conjunction of all the object's properties.
sliceInfo	`Snssai`	0..1	C: RW R: RW U: RW	A Network Slice on which this Policy Template may be instantiated. (See clause 5.4.4.2 of TS 29.571 [33].)
dnn	`Dnn`	0..1	C: RW R: RW U: RW	A Data Network on which this Policy Template may be instantiated. (See clause 7.3.2.)
qoSSpecifications	`array(M1QoS Specification)`	0..1	C: RW R: RW U: RW	The network Quality of Service policy envelopes to be applied to the application service component(s) of media delivery sessions that instantiate this Policy Template (see NOTE and clause 7.3.3.4). Each member of the array is identified by a component reference that is unique in this array. If present, the array shall contain at least one object.
chargingSpecification	`Charging Specification`	0..1	C: RW R: RW U: RW	The charging policy to be applied to media delivery sessions that instantiate this Policy Template is instantiated (see NOTE and clause 7.3.3.7).
bdtPolicyId	`BdtReferenceId`	0..1	C: RW R: RO U: RW	A reference to an existing Background Data Transfer policy in the PCF (see NOTE). Mutually exclusive with *bdtSpecification*.
bdtSpecification	`M1BDT Specification`	0..1	C: RW R: RO U: RW	The Background Data Transfer policy specification to be associated with media delivery sessions that instantiate this Policy Template (see clause 8.7.3.2). Mutually exclusive with *bdtPolicyId* property.
NOTE: Data type *BdtReferenceId* is specified in TS 29.122 [20].				

At least one of the following properties shall be present: *qosSpecification*, *chargingSpecification*, *bdtPolicyId*, *bdtSpecification*.

8.7.3.2 M1BDTSpecification type

Table 8.7.3.2-1: Definition of M1BDTSpecification type

Property name	Type	Cardinality	Description
startDate	`DateTime`	0..1	The start date of the Background Data Transfer policy. The time data part of this value shall not be present.
endDate	`DateTime`	0..1	The last date of the Background Data Transfer policy. The time data part of this value shall not be present.
windows	`array(object)`	1..1	The windows when Background Data Transfers are permitted. The array shall contain at least one window specification.
startTime	`TimeOfDay`	0..1	The starting time of a Background Data Transfer window (see NOTE).
duration	`DurationMin`	0..1	The duration of the Background Data Transfer window in minutes with the maximum value of 1339 (see NOTE). The duration may result in a window ending on the next calendar day.
daysInWeek	`array(DayOfWeek)`	0..1	The days of the week that the Background Data Transfer window is in effect. A maximum of seven occurrences can be provided. No two occurrences of array shall have the same value. If omitted, the Background Data Transfer window is applicable on all days of the week.
numberOfOccurrences	`integer`	0..1	The number of days that the Background Data Transfer window is in effect. Each *daysInWeek* reccurrence is counted as one occurrence. The Background Data Transfer specification ends when either the *endDate* or the *numberofOccurrences* is reached, whichever sooner.
numberOfUes	`integer`	0..1	The maximum number of UEs permitted to use the Background Data Transfer policy in each occurrence of the Background Data Transfer window. Minimum value: 1.
estimatedDataVolumePerUe	`UsageThreshold`	0..1	An estimate of the data volume an average UE is expected to transfer (in both the downlink and uplink directions) when applying this Background Data Transfer policy in each occurrence of the Background Data Transfer window (see NOTE).
aggregateDownlinkBitRate Limit	`BitRate`	0..1	A limit on the total uplink bit rate of concurrent instances of the parent Policy Template to be enforced by the Media AF.
aggregateUplinkBitRateLimit	`BitRate`	0..1	A limit on the total downlink bit rate of concurrent instances of the parent Policy Template to be enforced by the Media AF.

NOTE: Data types *TimeOfDay*, *DurationMin DayOfWeek* and *UsageThreshold* are defined in TS 29.122 [20].

8.8 Content Hosting provisioning API

8.8.1 Overview

This clause specifies the API that a Media Application Provider uses by interacting with a Media AF at reference point M1 to provision and manage Media AS Content Hosting Configurations for downlink media delivery. Each such configuration is represented by a *ContentHostingConfiguration*, the data model for which is specified in clause 8.8.3 below. The RESTful resources for managing Content Hosting Configurations are specified in clause 8.8.2 and the operations on these resources are further elaborated in clause 5.2.8.

8.8.2 Resource structure

The Content Hosting Provisioning API is accessible through this URL base path:

{apiRoot}`/3gpp-maf-provisioning/`*{apiVersion}*`/provisioning-sessions/`*{provisioningSessionId}*`/`

Table 8.8.2-1 below specifies the operations and the corresponding HTTP methods that are supported by this API. In each case, the Provisioning Session identifier shall be substituted into *{provisioningSessionId}* in the above URL template and the sub-resource path specified in the second column shall be appended to the URL base path.

Table 8.8.2-1: Operations supported by the Content Hosting Provisioning API

Operation name	Sub-resource path	Allowed HTTP method(s)	Description
Create Content Hosting Configuration	`content-hosting-configuration`	`POST`	Create the Content Hosting Configuration resource within the context of a parent Provisioning Session.
Retrieve Content Hosting Configuration		`GET`	Retrieve an existing Content Hosting Configuration resource.
Update Content Hosting Configuration		`PUT, PATCH`	Modify an existing Content Hosting Configuration resource.
Destroy Content Hosting Configuration		`DELETE`	Destroy an existing Content Hosting Configuration resource.
Purge Content Hosting Configuration cache	`content-hosting-configuration/purge`	`POST`	Invalidate some or all cached media resources associated with the specified Content Hosting Configuration.

8.8.3 Data model

8.8.3.1 ContentHostingConfiguration resource

Table 8.8.3.1-1: Definition of ContentHostingConfiguration resource

Property name	Data Type	Cardinality	Description
name	`string`	1..1	A name for this Content Hosting Configuration.
ingestConfiguration	`IngestConfiguration`	1..1	Parameters for ingesting media content into the Media AS at reference point M2.
mode	`ContentTransferMode`	1..1	Indicates whether media content is pulled by the Media AS from the Media Application Provider's origin server or pushed into the Media AS by the Media Application Provider (see clause 7.3.4.5).
protocol	`Uri`	1..1	A fully-qualified term identifier URL that identifies the content ingest protocol. The controlled vocabulary of content ingest protocols is specified in clause 8 of TS 26.512 [6].
baseURL	`AbsoluteUrl`	0..1	A base URL (i.e., one that includes a scheme, authority and, optionally, path segments) from which content is ingested at reference point M2 for this ingest configuration. In the case of pull-based content ingest (*mode* is set to *PULL*), the base URL shall be provided to the Media AF to indicate the location from which content is to be pulled. A request received at reference point M4 is mapped by the Media AS to a URL at reference point M2 whose base is the value of this property. In the case of push-based content ingest (*method* is set to *PUSH*), this property shall be populated by the Media AF and returned to the Media Application Provider to indicate the base URL to which content for this Content Hosting Configuration is to be published.
distributionConfigurations	`array(DistributionConfiguration)`	1..1	Specifies the distribution method and configuration for the ingested content. The array shall contain at least one member. Hence, more than one distribution may be configured for the same ingested content, e.g. to offer different distribution configurations such as DASH and HLS.
supplementaryDistributionNetworks	`array(<DistributionNetworkType, DistributionMode>`	0..1	Indicates that the content for this distribution configuration is also to be distributed via one or more supplementary networks. Each member of the array is a duple mapping a type of distribution network to a mode of distribution. The same *DistributionNetworkType* value shall appear at most once in this array.
edgeResourcesConfigurationId	`ResourceId`	0..1	A reference to an Edge Resources Configuration resource (see clause 8.6.2). When present, indicates that the Media AS supporting this content distribution shall be realised as a set of one or more EAS instances configured per the referenced resource.
contentPreparationTemplateId	`ResourceId`	0..1	A reference to a Content Preparation Template resource (see clause 8.5.2). Indicates that the referenced content preparation is required prior to distribution.
certificateId	`ResourceId`	0..1	A reference to a Server Certificate resource (see clause 8.4.3.2). When content is distributed using TLS [29], the referenced X.509 [10] certificate for the origin domain is presented by the Media AS in the TLS handshake at reference point M4. This attribute indicates the identifier of the certificate to use.
canonicalDomainName	`string`	1..1	All resources exposed at reference point M4 shall be accessible through this default Fully-Qualified Domain Name assigned by the Media AF.

	domainNameAlias	`string`	0..1	The Media Application Provider may assign another Fully-Qualified Domain Name (FQDN) through which media resources within the scope of this distribution configuration are additionally accessible from the Media AS at reference point M4. This domain name is used by the Media AS to set appropriate CORS HTTP response headers at reference point M4. If this property is present, the Media Application Provider is responsible for providing in the DNS a *CNAME* record that resolves *domainNameAlias* to *canonicalDomainName*. If the *certificateId* property is also present in this distribution configuration, the provided domain name alias shall match one of the *subjectAltName* extension fields in the referenced Server Certificate resource, allowing for wildcard matching.
	baseURL	`AbsoluteUrl`	1..1	A base URL (i.e., one that includes a scheme, authority and, optionally, path segments) from which content is made available to Media Clients at reference point M4 for this distribution configuration. The value is chosen by the Media AF when the Content Hosting Configuration is provisioned. It is an error for the Media Application Provider to set this.
	entryPoint	`M1MediaEntryPoint`	0..1	The Media Entry Point nominated by the Media Application Provider for this distribution configuration when it is used to describe a single content item (see clause 7.3.3.12). Omitted when this distribution configuration describes multiple content items.
		relativePath	`RelativeUrl`	1..1	A relative path (i.e., without a scheme or any leading forward slash characters) to the Media Entry Point document resource. The semantics are dependent on the value of *ingestConfiguration.protocol*. The path shall be valid at reference point M2 when appended to the ingest base URL and at reference point M4 when appended to the distribution base URL.
		contentType	`string`	1..1	The MIME content type of the Media Entry Point. Used by the Media Client to select a Media Entry Point.
		protocol	`Uri`	0..0	This property shall not be present in a distribution configuration.
		profiles	`array(Uri)`	0..1	An optional list of conformance profile identifiers associated with the Media Entry Point, each one expressed as a URI. A profile URI may indicate an interoperability point, for example. Used by the Media Client to select a Media Entry Point. If present, the array shall contain at least one item.
	pathRewriteRules	`array(PathRewrite Rule)`	0..1	An ordered list of rules for rewriting the request URL paths of media resource requests handled by the Media AS at reference point M4 and translating them to URL paths at reference point M2. If multiple rules match a particular resource's path, only the first matching rule, in order of appearance in this array, shall be applied.
		requestPathPattern	`string`	1..1	A regular expression [36] against which the path part of each Media AS request URL, including the leading "/", and up to and including the final "/", shall be compared. (Any leaf path element following the final "/" shall be excluded from this comparison.) In the case of pull-based content ingest, the M4 download request path is used in the comparison. In the case of push-based content ingest, the M2 upload request path is used in the comparison. In either case, if the request path matches this pattern, the path mapping specified in the corresponding *mappedPath* shall be applied.
		mappedPath	`string`	1..1	A replacement for the portion of the Media AS request path that matches *requestPathPattern*.

					In the case of pull-based content ingest, *ingestConfiguration.entryPoint* is concatenated with the mapped path and any leaf path element from the original M4 download request to form the M2 origin request URL. In the case of push-based content ingest, *canonicalDomainName* (and, optionally, *domainName Alias*) are concatenated with the mapped path and any leaf path element from the original M2 upload request to form the distribution URL(s) exposed over reference point M4.
	cachingConfigurations		`array(Caching Configuration)`	0..1	A set of configurations of the Media AS content cache nominated by the Media Application Provider, each one affecting a matching subset of media resources ingested in relation to this Content Hosting Configuration. (See clause 7.3.3.13.) If present, the array shall have at least one member.
		urlPatternFilter	`string`	1..1	A pattern used to match media resource URLs at reference point M2 to determine whether a given media resource ingested by the Media AS is eligible to be cached by it. The format of the pattern shall be a regular expression as specified in [36].
		cachingDirectives	`object`	1..1	If a *urlPatternFilter* applies to a resource, then the provided *cachingDirectives* shall be applied by the Media AS at reference point M4, potentially overwriting any origin caching directives provided by the Media Application Provider when that resource is ingested at reference point M2.
			statusCodeFilters	`array(integer)`	0..1	The set of HTTP origin response status codes at reference point M2 to which these *cachingDirectives* apply. If the property is present, the array shall contain at least one item. If absent, the enclosing *cachingDirectives* shall apply to all HTTP origin response status codes.
			noCache	`boolean`	0..1	If set to *true*, indicates that the media resources matching the filters shall be marked by the Media AS as not to be cached when it servers such media resources at reference point M4. Default value if omitted: *false*.
			maxAge	`Uint32`	0..1	The caching time-to-live period, expressed in seconds, of ingested media resources matching the filters. This determines the minimum period for which the Media AS shall cache matching media resources. If *noCache* is *false*, it also determines the time-to-live period signalled by the Media AS at reference point M4 when it serves such media resources. The time-to-live for a given media resource shall be calculated relative to the time it was ingested by the Media AS.
	geoFencing		`object`	0..1	Directives limiting access to the content to the indicated geographic areas (see NOTE 1).
		locatorType	`Uri`	1..1	The type of the members of the *locators* array shall be indicated using a fully-qualified term identifier URI from the controlled vocabulary specified in clause B.1, or else from a vendor-specific vocabulary.
		locators	`array(string)`	1..1	Array of locators from which access to the resources is to be allowed. The format of the locator strings shall be determined by the semantics of the term identifier indicated in *locatorType*.
	urlSignature		`object`	0..1	Defines the URL signing scheme to be enforced by the Media AS at reference point M4 (see NOTE 2). When present, only correctly signed and valid URLs are permitted to access the content resources within the scope of the enclosing distribution configuration.
		urlPattern	`string`	1..1	A pattern that shall be used by the Media AS to match M4 media resource request URLs. The Media AS shall not serve a matching media resource at reference point M4 unless it includes a valid authentication token calculated over the portion of the M4 request URL that matches this pattern. The format of the pattern shall be a regular expression as specified in [36].
		tokenName	`string`	1..1	The name of the query parameter that the Media Access Function shall use to present the authentication token in the M4 request URL when required to do so.
		passphraseName	`string`	1..1	The name of the token parameter to be used to refer to the passphrase when constructing the M4 authentication token.

		passphrase	`string`	1..1	A string of between 6 and 50 characters to be used as the shared secret between the Media Application Provider and the Media AS for this *distributionConfiguration*. (This secret is used in the computation and verification of the M4 authentication token but is never sent in the cleartext part of the M4 request URL.)
		tokenExpiryName	`string`	1..1	The name of the token parameter to be used to refer to the token expiry time point when constructing the M4 authentication token. The name of the query parameter that the Media Access Function shall use to present the token expiry time point in the cleartext part of the M4 request URL.
		useIPAddress	`boolean`	1..1	If set to *true*, the IP address of the Media Access Function is included in the computation of the authentication token for resources that match *urlPattern* and access to matching media resources shall be allowed by the Media AF only when the M4 request is made from this IP address.
		ipAddressName	`string`	0..1	The name of the token parameter that is encoded as part of the M4 authentication token if the *useIPAddress* flag is set to *true*. (The IP address is not passed in the cleartext part of the M4 request URL.)
NOTE 1: The geofencing feature used to restrict content requests to the Media AS at reference point M4 is specified in clause 7.6.4.6 of TS 26.512 [6]. NOTE 2: The format of the authentication token used to sign content requests to the Media AS at reference point M4 is specified in clause 7.6.4.5 of TS 26.512 [6].					

8.8.3.2 DistributionNetworkType enumeration

Table 8.8.3.2-1: Definition of DistributionNetworkType enumeration

Enumeration value	Description
DISTRIBUTION_NETWORK_EMBMS	Downlink media streaming via eMBMS.
DISTRIBUTION_NETWORK_MBS	Downlink media streaming via MBS.

8.8.3.3 DistributionMode enumeration

Table 8.8.3.3-1: Definition of DistributionMode enumeration

Enumeration value	Description
MODE_EXCLUSIVE	Content ingested by the Media AS is distributed exclusively via a supplementary network and is not available at reference point M4.
MODE_HYBRID	Content ingested by the Media AS is available at reference point M4 and is additionally distributed via a supplementary network.
MODE_DYNAMIC	Content ingested by the Media AS is available at reference point M4 and is additionally distributed via a supplementary network only when reported client demand exceeds a configured threshold.

8.9 Content Publishing provisioning API

8.9.1 Overview

This clause specifies the API that a Media Application Provider uses with a Media AF at reference point M1 to provision and manage Media AS Content Publishing Configurations for uplink media delivery. Each such configuration is represented by a *ContentPublishingConfiguration*, the data model for which is specified in clause 8.9.3 below. The RESTful resources for managing Content Publishing Configurations are specified in clause 8.9.2 and the operations on these resources are further elaborated in clause 5.2.9.

8.9.2 Resource structure

The Content Publishing Provisioning API is accessible through this URL base path:

{apiRoot}/3gpp-maf-provisioning/*{apiVersion}*/provisioning-sessions/*{provisioningSessionId}*/

Table 8.9.2-1 below specifies the operations and the corresponding HTTP methods that are supported by this API. In each case, the Provisioning Session identifier shall be substituted into *{provisioningSessionId}* in the above URL template and the sub-resource path specified in the second column shall be appended to the URL base path.

Table 8.9.2-1: Operations supported by the Content Publishing Provisioning API

Operation	Sub-resource path	Allowed HTTP method(s)	Description
Create Content Publishing Configuration	content-publishing-configuration	POST	Create the Content Publishing Configuration resource within the context of a parent Provisioning Session.
Retrieve Content Publishing Configuration		GET	Retrieve an existing Content Publishing Configuration resource.
Update Content Publishing Configuration		PUT, PATCH	Modify an existing Content Publishing Configuration resource.
Destroy Content Publishing Configuration		DELETE	Destroy an existing Content Publishing Configuration resource.
Purge Content Publishing Configuration cache	content-publishing-configuration/purge	POST	Invalidate some or all cached media resources associated with the specified Content Publishing Configuration. Applicable to pull-based content egest only.

8.9.3 Data model

8.9.3.1 ContentPublishingConfiguration resource

Table 8.9.3.1-1: Definition of ContentPublishingConfiguration resource

Property name	Data type	Cardinality	Description
name	`string`	1..1	A name for this Content Publishing Configuration.
contributionConfigurations	`array(ContributionConfiguration)`	1..1	Specifies the Media Entry Point and content preparation required for the egested content. The array shall contain at least one member. Hence, more than one contribution may be configured for different content types.
edgeResourcesConfigurationId	`ResourceId`	0..1	A reference to an Edge Resources Configuration resource (see clause 8.6.2). When present, indicates that the Media AS supporting this content contribution shall be realised as a set of one or more EAS instances configured per the referenced resource.
contentPreparationTemplateId	`ResourceId`	0..1	A reference to a Content Preparation Template resource (see clause 8.5.2). Indicates that the referenced content preparation is required prior to egest.
certificateId	`ResourceId`	0..1	A reference to a Server Certificate resource (see clause 8.4.3.2). When content is contributed using TLS [29], the referenced X.509 [10] certificate for the origin domain is presented by the Media AS in the TLS handshake at reference point M4. This attribute indicates the identifier of the certificate to use.
canonicalDomainName	`string`	1..1	All resources exposed at reference point M4 shall be accessible through this default Fully-Qualified Domain Name assigned by the Media AF.
domainNameAlias	`string`	0..1	The Media Application Provider may assign another Fully-Qualified Domain Name (FQDN) through which media resources within the scope of this contribution configuration are additionally accessible from the Media AS at reference point M4. This domain name is used by the Media AS to set appropriate CORS HTTP response headers at reference point M4. If this property is present, the Media Application Provider is responsible for providing in the DNS a *CNAME* record that resolves *domainNameAlias* to *canonicalDomainName*. If the *certificateId* property is also present in this contribution configuration, the provided domain name alias shall match one of the *subjectAltName* extension fields in the referenced Server Certificate resource, allowing for wildcard matching.
baseURL	`AbsoluteUrl`	1..1	A base URL (i.e. one that includes a scheme, authority, and, optionally, path segments) to which content is contributed by Media Clients at reference point M4 for this contribution configuration. Nominated by the Media AF when the Content Publishing Configuration is provisioned. It is an error for the Media Application Provider to set this.
entryPoint	`M1MediaEntryPoint`	1..1	The Media Entry Point nominated by the Media Application Provider for this contribution configuration (see clause 7.3.3.12).
relativePath	`RelativeUrl`	1..1	A relative path (i.e., without a scheme or any leading forward slash characters) for this Media Entry Point which may point to a document resource. Nominated by the Media AF.
contentType	`string`	1..1	The MIME content type of this Media Entry Point. This property shall be mutually exclusive with *protocol*.

					Used by the Media Client to select a contribution configuration. Nominated by the Media Application Provider.
		protocol	`Uri`	1..1	A fully-qualified term identifier URI that identifies the media contribution protocol at reference point M4 for this Media Entry Point. This property shall be mutually exclusive with *contentType*. Nominated by the Media Application Provider. The controlled vocabulary of media contribution protocols is specified in clause 10 of TS 26.512 [6].
		profiles	`array(Uri)`	0..1	An optional list of conformance profile identifiers associated with this Media Entry Point, each one expressed as a URI. A profile URI may indicate an interoperability point, for example. Used by the Media Client to select a contribution configuration. Nominated by the Media Application Provider and, if present, the array shall contain at least one item.
egestConfiguration			`EgestConfiguration`	1..1	Parameters for egesting media content from the Media AS at reference point M2.
	mode		`ContentTransferMode`	1..1	Indicates whether content is pulled from the Media AS by the Media Application Provider or pushed to the Media Application Provider by the Media AS (see clause 7.3.4.5). Nominated by the Media Application Provider.
	protocol		`Uri`	1..1	A fully-qualified term identifier URI that identifies the content egest protocol. Nominated by the Media Application Provider. The controlled vocabulary of content egest protocols is specified in clause 8 of TS 26.512 [6].
	baseURL		`AbsoluteURL`	0..1	A base URL (i.e., one that includes a scheme, authority, and, optionally, path segments) to which content is published at reference point M2 for this publishing configuration. In the case of pull-based content egest (*mode* is set to *PULL*), this property shall be populated by the Media AF to indicate the location on the Media AS from which content is to be pulled. An uplink media streaming request received at reference point M4 is mapped by the Media AS to a URL at reference point M2 whose base is the value of this property. In the case of push-based content egest (*mode* is set to *PUSH*), this property shall be provided to the Media AF and indicates the base URL to which content for this Content Publishing Configuration is to be published.
	entryPoint		`M1MediaEntryPoint`	0..1	The Media Entry Point for content egest used by the Media Application Provider at reference point M2. In the case of pull-based content egest (*mode* is set to *PULL*), this object shall be provided by the Media AF. In the case of push-based content egest (*mode* is set to *PUSH*), this object may be provided by the Media Application Provider. The semantics of the entry point are dependent on the value of the *contentType* property.
		relativePath	`RelativeURL`	1..1	A relative path (i.e., without a scheme or any leading forward slash characters) to the Media Entry Point document resource. Nominated by the Media AF for pull-based content egest. Nominated by the Media Application Provider for Push-based content egest.
		contentType	`string`	1..1	The MIME content type of this Media Entry Point. Nominated by the Media Application Provider.
		protocol	`Uri`	1..1	A fully-qualified term identifier URI that identifies the media egest protocol at reference point M2 for this Media Entry Point.

					This property shall be mutually exclusive with *contentType*. Nominated by the Media Application Provider. The controlled vocabulary of media contribution protocols is specified in clause 10 of TS 26.512 [6].
		profiles	`array(Uri)`	0..1	An optional list of conformance profile identifiers associated with this Media Entry Point, each one expressed as a URI. A profile URI may indicate an interoperability point, for example. Nominated by the Media Application Provider and, if present, the array shall contain at least one item.
	cachingConfigurations		`array(Caching Configuration)`	0..1	A set of configurations of the Media AS cache nominated by the Media Application Provider, each one affecting a matching subset of media resources intended for pull-based egest at reference point M2 in relation to this Content Publishing Configuration. (See clause 7.3.3.13.) Applicable only for pull-based content egest (*mode* is set to *PULL*). For Push-based egest (*method* is set to *PUSH*), this property shall not be present. If present, the array shall have at least one member.
		urlPatternFilter	`string`	1..1	A pattern used to match media resource URLs to determine whether a given media resource is eligible for caching by the Media AS. The format of the pattern shall be a regular expression as specified in [36].
		cachingDirectives	`object`	1..1	If a *urlPatternFilter* applies to a resource, then the provided *cachingDirectives* shall be applied by the Media AS at reference point M2. Any caching directives set by the Media Streamer on content contributed at reference point M4 which define a shorter lifetime for the content shall take precedence over these parameters.
			statusCodeFilters	`array(integer)`	0..1	The set of Media AS response status codes at reference point M2 to which these *cachingDirectives* apply. If the property is present, the array shall contain at least one item. If absent, the enclosing *cachingDirectives* shall apply to all Media AS responses.
			noCache	`boolean`	0..1	If set to *true*, this indicates that the media resources matching the filters shall be marked by the Media AS as not to be cached when it serves such media resources at reference point M2. Default value if omitted: *false*.
			maxAge	`Uint32`	0..1	The caching time-to-live period, expressed in seconds, of media resources matching the filters. This determines the minimum period for which the Media AS shall cache matching media resources. If *noCache* is *false*, it also determines the time-to-live period signalled by the Media AS at reference point M2 when it serves such media resources. The time-to-live for a given media resource shall be calculated relative to the time it was contributed to the Media AS.

8.10 Real-time Media Communication provisioning API

8.10.1 Overview

The Real-time Media Communication provisioning API is used by the Media Application Provider to supply configuration information, in the form of an *RTCConfiguration* resource (specified in clause 8.10.3) that is used by the Media Client to gain access to real-time media communication (RTC) functionality of the Media AS. The provisioning API allows for the enablement and/or advertisement of ICE (STUN, TURN, and/or SWAP) services to facilitate communication between Media Clients in an RTC-based media delivery session. These facilitation services may either be provided by the Media AS itself or provisioned by the Media AF.

8.10.2 Resource structure

The RTC Configuration API is accessible through this URL base path:

{apiRoot}/3gpp-maf-provisioning/*{apiVersion}*/provisioning-sessions/*{provisioningSessionId}*/

Table 8.10.2-1 below specifies the operations and the corresponding HTTP methods that are supported by this API. In each case, the Provisioning Session identifier shall be substituted into *{provisioningSessionId}* in the above URL template and the sub-resource path specified in the second column shall be appended to the URL base path.

Table 8.10.2-1: Operations supported by the Real-Time Communication Configuration API

Operation name	Sub-resource path	Allowed HTTP method(s)	Description
Create RTC Configuration	rtc-configuration	POST	Create the RTC Configuration resource within the context of a parent Provisioning Session.
Retrieve RTC Configuration		GET	Retrieve an existing RTC Configuration resource.
Update RTC Configuration		PUT, PATCH	Modify an existing RTC Configuration resource.
Destroy RTC Configuration		DELETE	Destroy an existing RTC Configuration resource.

8.10.3 Data model

8.10.3.1 RTCConfiguration resource

Table 8.10.3.1-1: Definition of RTCConfiguration resource

Property name	Data Type	Cardinality	Description
edgeResources ConfigurationId	`ResourceId`	0..1	A reference to an Edge Resources Configuration resource (see clause 8.6.2). When present, indicates that the Media AS supporting this RTC Configuration shall be realised as a set of one or more EAS instances configured per the referenced resource.
enableStunService	`boolean`	0..1	If *true*, the Media AS shall provide a STUN service to the Media Session Handler for use in RTC-based media delivery sessions initiated in the context of the parent Provisioning Session. If *false* the Media Application Provider may populate the *stunEndpoints* property. If omitted, the default value shall be *false*.
stunEndpoints	`array(M1Endpoint Access)`	0..1	A list of one or more trusted STUN server endpoints populated by the Media Application Provider or else by the Media AF that may be used as ICE candidates for RTC-based media delivery sessions.
enableTurnService	`boolean`	0..1	If *true*, the Media AS shall provide a TURN service to the Media Session Handler for use in RTC-based media delivery sessions initiated in the context of the parent Provisioning Session. If *false* the Media Application Provider may populate the *turnEndpoints* property. If omitted, the default value shall be *false*.
turnEndpoints	`array(M1Endpoint Access)`	0..1	A list of one or more trusted TURN server endpoints populated by the Media Application Provider or else by the Media AF that may be used as ICE candidates for RTC-based media delivery sessions.
enableSwapService	`boolean`	0..1	If *true*, the Media AS shall provide a SWAP service to the Media Session Handler for use in RTC-based media delivery sessions initiated in the context of the parent Provisioning Session. If *false* the Media Application Provider may populate the *swapEndpoints* property. If omitted, the default value shall be *false*.
swapEndpoints	`array(M1Endpoint Access)`	0..1	A list of one or more trusted WebRTC Signalling Server endpoints populated by the Media Application Provider or else by the Media AF that support the SWAP protocol that may be used by the application for RTC-based media delivery sessions in the context of the parent Provisioning Session.

8.10.3.2 M1EndpointAccess

This data type is derived by extension from *EndpointAddress* (see clause 7.3.3.11).

Table 8.10.3.2-1: Definition of M1EndpointAccess data type

Property name	Data Type	Cardinality	Description
domainName	`string`	1..1	The Fully-Qualified Domain Name of this service endpoint.
portNumbers	`array(Uint16)`	1..1	A list of listening ports over which the service is accessible (e.g. STUN servers must offer two listening ports). The array shall contain at least one member.
credentials	`object`	0..1	Authentication information to be presented to this service endpoint by all clients.
username	`string`	1..1	A username that is authorized to access the server.
passphrase	`string`	1..1	The passphrase associated with *username*.
certificateId	`ResourceId`	0..1	Identifies the server certificate resource to be presented to Media Clients by this service endpoint.

8.11 Metrics Reporting provisioning API

8.11.1 Overview

The Metrics Reporting Provisioning API allows a Media Application Provider to configure the Metrics Collection and Reporting procedure for a particular downlink or uplink media delivery Provisioning Session at reference point M1.

8.11.2 Resource structure

The Metrics Reporting Provisioning API is accessible through the following URL base path:

{apiRoot}/3gpp-maf-provisioning/*{apiVersion}*/provisioning-sessions/*{provisioningSessionId}*/

Table 8.11.2-1 below specifies the operations and the corresponding HTTP methods that are supported by this API. In each case, the Provisioning Session identifier shall be substituted into *{provisioningSessionId}* in the above URL template and the sub-resource path specified in the second column of the table shall be appended to the URL base path.

Table 8.11.2-1: Operations supported by the Metrics Reporting Provisioning API

Operation name	Sub-resource path	Allowed HTTP method(s)	Description
Create Metrics Reporting Configuration	metrics-reporting-configurations	POST	Create and provide a metrics reporting configuration.
Read Metrics Reporting Configuration	metrics-reporting-configurations/ *{metricsReportingConfigurationId}*	GET	Retrieve the values of an existing Metrics Reporting Configuration.
Update Metrics Reporting Configuration		PUT, PATCH	Modify or replace an existing metrics reporting configuration.
Destroy Metrics Reporting Configuration		DELETE	Destroy a metrics reporting configuration.

8.11.3 Data model

8.11.3.1 MetricsReportingConfiguration resource

Table 8.11.3-1: Definition of MetricsReportingConfiguration resource

Property name	Type	Cardinality	Description
metricsReportingConfigurationId	`ResourceId`	1..1	An identifier for this Metrics Reporting Configuration assigned by the Media AF when the resource is created that is unique within the scope of the enclosing Provisioning Session.
sliceScope	`array(Snssai)`	0..1	The set of network slice(s) for which metrics collection and reporting shall be executed in connection with this metrics reporting configuration (see NOTE). If present, the array shall identify at least one network slice. If absent, metrics shall be collected and reported for media delivery sessions within the scope of the parent Provisioning Session regardless of network slice.
scheme	`Uri`	0..1	The QoE metrics scheme associated with this Metrics Reporting Configuration which indicates the required format of metrics reports. The set of QoE metrics schemes valid for use in 5G Media Streaming along with their respective scheme identifiers is specified in clauses 4.7.5 and 7.8.1 of TS 26.512 [6]. The QoE metrics scheme valid for use in RTC along with its respective scheme identifier is specified in clause 15 of TS 26.113 [7]. Omitting this property signals to the Media AF that metrics reporting is currently disabled for the Provisioning Session in question.
dataNetworkName	`Dnn`	0..1	Identifies the Data Network which shall be used when sending metrics reports. If not specified, the default Data Network shall be used.
reportingStartOffset	`DurationSec`	0..1	The time offset (expressed in seconds) from the start of a media delivery session when the Media Client is required to begin submitting metrics reports. The value shall not be negative. If omitted, the value of this parameter is assumed to be zero, i.e., directing the Media Client to start reporting metrics from the start of the media delivery session.
reportingDuration	`DurationSec`	0..1	The period of time (expressed in seconds) measured relative to the reporting start point, after which the Media Client is required to stop reporting metrics. The value shall not be negative. If set to zero, a single report shall be sent at *reportingStartOffset*. If omitted, reporting is required to continue until the end of the media delivery session.
reportingInterval	`DurationSec`	0..1	The time interval between successive metrics reports to be sent by the Media Session Handler. The value shall be greater than zero. If not specified, a single final report shall be sent after the media delivery session has ended.
samplePercentage	`Percentage`	0..1	The proportion of media delivery sessions for which QoE metrics shall be reported, expressed as a floating-point value between 0.0 and 100.0. If not specified, reports shall be sent for all media delivery sessions.
urlFilters	`array(string)`	0..1	If present, a non-empty list of Media Entry Point URL patterns for which QoE metrics shall be reported. If not specified, reporting shall be done for all media delivery sessions initiated within the scope of the parent Provisioning Session.

Property name	Type	Cardinality	Description
samplingPeriod	`DurationSec`	1..1	The time interval the Media Client should wait between sampling the QoE metrics specified by this Metrics Reporting Configuration. The value shall be greater than zero.
metrics	`array(Uri)`	0..1	If present, a non-empty list of QoE metrics, each indicated using a fully-qualified term identifier from a controlled vocabulary, which shall be collected and reported by the Media Client. A controlled vocabulary of QoE metrics shall be specified by each QoE metrics scheme for use with this property. If omitted, the complete (or default, as applicable) set of metrics associated with the specified metrics scheme shall be collected and reported.
NOTE: The *Snssai* data type is specified in TS 29.571 [33].			

8.12 Consumption Reporting provisioning API

8.12.1 Overview

The Consumption Reporting Provisioning API is a RESTful API that allows a Media Application Provider to configure the Consumption Reporting Procedure for a particular downlink media delivery Provisioning Session at reference point M1. The different interactions are described in clause 5.2.12. The Consumption Reporting Configuration is represented by a *ConsumptionReportingConfiguration* resource, the data model for which is specified in clause 8.12.3 below. The RESTful resources for managing the Consumption Reporting Configuration are specified in clause 8.12.2.

8.12.2 Resource structure

The Consumption Reporting Provisioning API is accessible through the following URL base path:

{apiRoot}/3gpp-maf-provisioning/*{apiVersion}*/provisioning-sessions/*{provisioningSessionId}*/

Table 8.12.2-1 below specifies the operations and the corresponding HTTP methods that are supported by this API. In each case, the Provisioning Session identifier shall be substituted into *{provisioningSessionId}* in the above URL template and the sub-resource path specified in the second column shall be appended to the URL base path.

Table 8.12.2-1: Operations supported by the Consumption Reporting Provisioning API

Operation name	Sub-resource path	Allowed HTTP method(s)	Description
Create Consumption Reporting Configuration resource	consumption-reporting-configuration	POST	Activate the consumption reporting procedure for a Provisioning Session by providing the Consumption Reporting Configuration.
Retrieve Consumption Reporting Configuration resource		GET	Retrieve an existing Consumption Reporting Configuration.
Update Consumption Reporting Configuration resource		PUT, PATCH	Replace or modify an existing Consumption Reporting Configuration.
Destroy Consumption Reporting Configuration resource		DELETE	Deactivate the consumption reporting procedure for the parent Provisioning Session.

8.12.3 Data model

8.12.3.1 ConsumptionReportingConfiguration resource

Table 8.12.3.1-1: ConsumptionReportingConfiguration resource

Property name	Type	Cardinality	Description
reportingInterval	`DurationSec`	0..1	The interval between two consecutive consumption reports. The value shall be greater than zero. If absent, a single final report shall be sent immediately after the media streaming session has ended.
samplePercentage	`Percentage`	0..1	The proportion of media streaming clients that shall report media consumption, expressed as a floating-point value between 0.0 and 100.0. If not specified, all clients shall send consumption reports.
accessReporting	`boolean`	0..1	Stipulates whether the Media Session Handler is required to provide consumption reporting messages to the Media AF when the access network changes during a media streaming session. If omitted, access network change reporting is disabled.

8.13 Event Data Processing provisioning API

8.13.1 General

The Event Data Processing Provisioning API is used by a Media Application Provider to provide Event Data Processing Configurations to the Data Collection AF instantiated in the Media AF. Each such configuration is represented by an *EventDataProcessingConfiguration* resource, the data model of which is specified in clause 8.13.3 below. It comprises processing rules and parameters expressed using Data Access Profiles (specified in TS 26.532 [40]), each of which defines a level of access by Event consumers to the UE data collected by the Data Collection AF. The RESTful structure of the Data Exposure Restriction Configuration resource collection, along with the operations and corresponding HTTP methods for managing resources of this type are defined in clause 8.13.2.

8.13.2 Resource structure

The Event Data Processing Provisioning API is accessible through the following URL base path:

{apiRoot}/3gpp-maf-provisioning/*{apiVersion}*/provisioning-sessions/*{provisioningSessionId}*/

Table 8.13.2-1 below specifies the operations and the corresponding HTTP methods that are supported by this API. In each case, the Provisioning Session identifier shall be substituted into *{provisioningSessionId}* in the above URL template and the sub-resource path specified in the second column shall be appended to the URL base path.

Table 8.13.2-1: Operations supported by the Data Exposure Restriction API

Operation name	Sub-resource path	Allowed HTTP method(s)	Description
Create Event Data Processing Configuration	event-data-processing-configurations	POST	Create a new Event Data Processing Configuration resource.
Retrieve Event Data Processing Configuration	event-data-processing-configurations/ *{eventDataProcessingConfigurationId}*	GET	Retrieve an existing Event Data Processing Configuration.
Update Event Data Processing Configuration		PUT, PATCH	Modify or replace an existing Event Data Processing Configuration.
Destroy Event Data Processing Configuration		DELETE	Destroy an existing Event Data Processing Configuration.

8.13.3 Data model

8.13.3.1 EventDataProcessingConfiguration resource type

Table 8.13.3-1: Definition of EventDataProcessingConfiguration resource

Property name	Type	Cardinality	Description
eventDataProcessingConfigurationId	`ResourceId`	1..1	An identifier for this Event Data Processing Configuration assigned by the Media AF that is unique within the scope of the enclosing Provisioning Session.
eventId	`AfEvent`	1..1	One of the enumerated values specified in clause 5.6.3.3 of TS 29.517 [39] relating to Media Delivery.
authorizationUrl	`AbsoluteUrl`	0..1	A URL that may be used to authorize the Event consumer entity to enable its subscription to the Data Collection AF for event notification, subject to the data access restrictions of a Data Access Profile.
dataAccessProfiles	`array(DataAccessProfile)`	1..1	One or more Data Access Profile definitions, each one describing a set of data processing instructions to be applied by the Data Collection AF when exposing events to an associated Event consumer entity. (See clause 6.3.3.2 of TS 26.532 [40].) The controlled vocabularies to be used with *DataAccessProfile.parameters* are not specified in the present document.

9 Maf_SessionHandling service

9.1 Overview

This clause defines the network media session handling API used by a Media AS at reference point M3 or by a Media Session Handler at reference point M5 to access Service Access Information and to invoke the session handling features provisioned using the APIs in clause 8. The corresponding OpenAPI definitions for the *Maf_SessionHandling* service are specified in clause A.4. A summary of the resource structure is shown in table 9.1-1 below. The default endpoint address of the Media AF at reference point M3 is specified in clause 7.1.2.2 and that at reference point M5 is specified in clause 7.1.2.3.

Table 9.1-1: Resource structure of Maf_SessionHandling APIs

HTTP request path element hierarchy	Description	Allowed HTTP methods					Resource structure definition clause	OpenAPI definition clause
		Create	Retrieve	Update	Destroy	Non-RESTful operation		
service-access-information	Service Access Information collection						9.2.2	A.4.1
{externalServiceId}	Service Access Information resource		GET					
provisioning-sessions	Provisioning Sessions collection							
{provisioningSessionId}	Provisioning Session path element							
dynamic-policies	Dynamic Policies collection	POST					9.3.2	A.4.2
{dynamicPolicyId}	Dynamic Policy resource		GET	PUT, PATCH	DELETE			
network-assistance-sessions	Network Assistance Sessions collection	POST					9.4.2	A.4.3
{naSessionId}	Network Assistance Session resource		GET	PUT, PATCH	DELETE			
recommendation	Bit rate recommendation request operation					GET		
boost	Delivery boost request operation					POST		
metrics-reporting	Metrics Reporting endpoint collection						9.5.2	A.4.4
{metricsReportingConfgurationId}	Metrics Reporting operation					POST		
consumption-reporting	Consumption Reporting endpoint					POST	9.6.2	A.4.5

9.2 Service Access Information API

9.2.1 General

The Service Access Information API is used by the Media Session Handler to obtain configuration information from the Media AF that enables it to use the other Media Session Handling APIs specified in clause 9.3 *et seq.*

9.2.2 Resource structure

The Service Access Information API is accessible through the following URL base path:

{apiRoot}/3gpp-maf-session-handling/*{apiVersion}*/service-access-information/

The operations and the corresponding HTTP methods in table 9.2.2-1 are supported. In each case, the sub-resource path specified in the second column shall be appended to the URL base path.

Table 11.2.2-1: Operations supported by the Service Access Information API

Operation name	Sub-resource path	Allowed HTTP method(s)	Description
Retrieve Service Access Information	*{externalServiceId}*	GET	Acquire the Service Access Information resource for the specified Provisioning Session.

9.2.3 Data model

9.2.3.1 ServiceAccessInformation resource type

The data model for the *ServiceAccessInformation* resource is specified in table 9.2.3.1-1 below. Different properties are present in the resource depending on the type of Provisioning Session from which the Service Access Information is derived (as indicated in the *provisioningSessionType* property) and this is specified in the *Applicability* column.

Table 9.2.3.1-1: Definition of ServiceAccessInformation resource

Property name	Type	Cardinality	Description	Applicability
provisioningSessionId	`ResourceId`	1..1	Unique identification of the M1 Provisioning Session.	All types
provisioningSessionType	`ProvisioningSessionType`	1..1	The type of Provisioning Session.	All types.
locationReporting	`boolean`	1..1	If *true*, the Media Session Handler is required to provide UE location data in Dynamic Policy interactions (see clause 9.3.3.1), Network Assistance interactions (see clause 9.4.3.1), QoE metrics reporting interactions (see clause 9.5.3) and consumption reporting interactions (see clause 9.6.3.2). Shall be set *false* if the *locationReporting* parameter is omitted from the *ProvisioningSession*, as specified in table 8.2.3.1-1.	All types.
notificationURL	`AbsoluteURL`	0..1	A URL to the MQTT channel, nominated by the Media AF, over which notifications are to be sent by the Media AF (see clause 10.2).	All types.
streamingAccess	`object`	0..1	Present if Content Hosting or Content Publishing is provisioned in the parent Provisioning Session.	*MS_DOWNLINK, MS_UPLINK*
entryPoints	`array(M5MediaEntryPoint)`	0..1	A list of alternative Media Entry Points for the Media Client to choose between.	
locator	`AbsoluteUrl`	1..1	Populated from information in the Content Hosting Configuration or Content Publishing Configuration as specified in clause 8 of TS 26.512 [6]. For downlink media streaming, either a pointer to a document at reference point M4 that defines a media presentation (e.g. a DASH MPD) whose resources are mapped to a content ingest configuration at reference point M2, or else the URL of a single media resource (e.g. an MP4 asset) available for download at reference point M4 that is mapped to reference point M2 by a Content Hosting Configuration. In both cases, the *contentType* property shall also be present. For uplink media streaming, either a pointer to a document at reference point M4 that defines a media presentation (e.g. a DASH MPD) whose resources are mapped to an egest configuration at reference point M2 (in which case the *contentType* property shall also be present), or else the URL of a path at reference point M4 the sub-resources of which are mapped to reference point M2 by a Content Publishing Configuration (in which case the *protocol* property shall also be present).	
contentType	`string`	1..1	The MIME content type of resource at *locator*. This property shall be mutually exclusive with *protocol*.	

Property name	Type	Cardinality	Description	Applicability
protocol	Uri	1..1	A fully-qualified term identifier URI that identifies the media delivery protocol at reference point M4 for this Media Entry Point. This property shall be mutually exclusive with *contentType*. The controlled vocabulary of media delivery protocols at this reference point is specified in clause 10 of TS 26.512 [6].	
profiles	array(Uri)	0..1	An optional list of conformance profile URIs with which this Media Entry Point is compliant. If present, the array shall contain at least one item.	
eMBMSService AnnouncementLocator	AbsoluteUrl	0..1	A pointer to an eMBMS User Service Announcement document.	
mbsExternalService Identifier	string	0..1	The external service identifier of an MBS User Service.	
rtcClientConfiguration	object	0..1	Present if real-time media communication (RTC) is provisioned.	*RTC*
stunEndpoints	array(M5Endpoint Access)	0..1	An array of one or more trusted STUN service endpoints for use as ICE candidates. If present, the RTC Client shall use one of the listed servers for RTC-based media delivery sessions within the scope of *provisioningSessionId*. If the *credentials* sub-property was not provisioned at reference point M1, the Media AF shall populate this with a set of credentials unique to the requesting Media Client.	
turnEndpoints	array(M5Endpoint Acess)	0..1	An array of one or more trusted TURN service endpoints for use as ICE candidates. If present, the RTC Client shall use one of the listed servers for RTC-based media delivery sessions within the scope of *provisioningSessionId*. If the *credentials* sub-property was not provisioned at reference point M1, the Media AF shall populate this with a set of credentials unique to the requesting Media Client.	
swapEndpoints	array(M5Endpoint Access)	0..1	An array of one or more trusted WebRTC Signalling Function service endpoints that support the SWAP protocol. If present, the RTC Client shall use one of the listed servers for RTC-based media delivery sessions within the scope of *provisioning SessionId*. If the *credentials* sub-property was not provisioned at reference point M1, the Media AF shall populate this with a set of credentials unique to the requesting Media Client.	
clientConsumptionReporting Configuration	object	0..1	Present if consumption reporting is activated for this Provisioning Session.	*MS_DOWNLINK, RTC*
reportingInterval	DurationSec	0..1	The time interval, expressed in seconds, between consumption report messages being sent by the Media Session Handler. The value shall be greater than zero. When this property is omitted, a single final report shall be sent immediately after the media streaming session has ended.	
serverAddresses	array(AbsoluteUrl)	1..1	A list of Media AF addresses (URLs) where the consumption reporting messages are sent by the Media Session Handler. (See NOTE 1). Each address shall be an opaque base URL, following the format specified in clause 7.1.3 up to and including the *{apiVersion}* path element.	

Property name	Type	Cardinality	Description	Applicability
accessReporting	`boolean`	1..1	Indicates whether the Media Session Handler is required to supply consumption reporting units whenever the access network changes during a media delivery session. Shall be set *false* if the *accessReporting* parameter is omitted from the *ConsumptionReportingConfiguration*, as specified in table 8.12.3.1-1.	
samplePercentage	`Percentage`	1..1	The percentage of media delivery sessions that shall send consumption reports, expressed as a floating-point value between 0.0 and 100.0. Shall be set to 100.0 if the *samplePercentage* parameter is omitted from the *ConsumptionReportingConfiguration*, as specified in table 8.12.3.1-1.	
dynamicPolicyInvocationConfiguration	`object`	0..1	Present if Policy Templates have been provisioned in the parent Provisioning Session and at least one of them is in the *READY* state.	*MS_DOWNLINK, MS_UPLINK, RTC*
serverAddresses	`array(AbsoluteUrl)`	1..1	A list of Media AF addresses (URLs) which offer the APIs for dynamic policy invocation sent by the Media Session Handler. (See NOTE 1.) Each address shall be an opaque base URL, following the format specified in clause 7.1.3 up to and including the *{apiVersion}* path element.	
policyTemplateBindings	`array(object)`	1..1	A list of duples, each one binding an external reference to a Policy Template resource identifier.	
externalReference	`string`	1..1	Additional identifier for this Policy Template, unique within the scope of its Provisioning Session, that can be cross-referenced with external metadata about the media streaming session. Example: "HD_Premium".	
policyTemplateId	`ResourceId`	1..1	The resource identifier of a Policy Template tagged with *externalReference* that is in the *READY* state.	
pduSetMarking	`boolean`	0..1	If *true*, indicates that PDU Set marking applies to Dynamic Policy Instances based on *policyTemplateId*. Default value *false* if omitted.	
bdtWindows	`array(BDTWindow)`	0..1	A list of Background Data Transfer time windows during which the application may request the activation of a Background Data Transfer policy by instantiating the Policy Template identified by *policyTemplateId*. The actual usage quotas for data volume and bit rate are determined by the Media AF upon instantiation of the Policy Template by the Media Session Handler. *BDTWindow* is specified in clause 7.3.3.14.	
sdfMethods	`array(SdfMethod)`	1..1	A list of Service Data Flow description methods, e.g. 5-tuple, TOS, 2-tuple, etc., which should be used by the Media Session Handler to describe the Service Data flows at reference point M2 for media delivery sessions.	

Property name	Type	Cardinality	Description	Applicability
clientMetricsReporting Configurations	`array(object)`	0..1	Present if QoE metrics reporting is provisioned in the parent Provisioning Session. If present, contains one or more client metrics reporting configurations.	*MS_DOWNLINK, MS_UPLINK, RTC*
metricsReporting ConfigurationId	`ResourceId`	1..1	The identifier of this metrics reporting configuration, unique within the scope of the parent Provisioning Session. The value shall be the same as the corresponding identifier provisioned at reference point M1 (see clause 8.11.3.1).	
serverAddresses	`array(AbsoluteUrl)`	1..1	A list of Media AF addresses to which metrics reports shall be sent. (See NOTE 1). Each address shall be an opaque base URL, following the format specified in clause 7.1.3 up to and including the *{apiVersion}* path element.	
sliceScope	`array(Snssai)`	0..1	The set of network slice(s) for which metrics collection and reporting shall be executed in connection with this metrics reporting configuration (see NOTE 2). If present, the array shall identify at least one network slice. If absent, metrics shall be collected and reported for media delivery sessions within the scope of the parent Provisioning Session regardless of network slice.	
scheme	`Uri`	1..1	A URI identifying the metrics scheme that metrics reports shall use (see clause 5.2.11). The set of QoE metrics schemes valid for use in 5G Media Streaming along with their respective scheme identifiers is specified in clauses 4.7.5 and 7.8.1 of TS 26.512 [6]. The QoE metrics scheme valid for use in RTC along with its respective scheme identifier is specified in clause 15 of TS 26.113 [7].	
dataNetworkName	`Dnn`	0..1	The name of the Data Network which shall be used to send metrics reports. If not specified, the default Data Network shall be used.	
reportingStartOffset	`DurationSec`	0..1	The time offset (expressed in seconds) from the start of a media delivery session when the Media Client is required to begin submitting metrics reports. If omitted, the value of this parameter is assumed to be zero, i.e., directing the Media Client to start reporting metrics from the start of the media delivery session.	
reportingDuration	`DurationSec`	0..1	The period of time (expressed in seconds) measured relative to the reporting start point, after which the Media Client is required to stop reporting metrics. If omitted, reporting is required to continue until the end of the media delivery session.	
reportingInterval	`DurationSec`	0..1	The time interval, expressed in seconds, between metrics reports being sent by the Media Session Handler. The value shall be greater than zero. When this property is omitted, a single final report shall be sent immediately after the media streaming session has ended.	
samplePercentage	`Percentage`	1..1	The percentage of media delivery sessions that shall report QoE metrics, expressed as a floating-point value between 0.0 and 100.0.	
urlFilters	`array(string)`	0..1	A non-empty list of Media Entry Point URL patterns for which QoE metrics shall be reported. The format of each pattern shall be a regular expression as specified in [36]. If not specified, reporting shall be done for all media delivery sessions.	

Property name	Type	Cardinality	Description	Applicability
samplingPeriod	`DurationSec`	1..1	The time interval the Media Client should wait between sampling the QoE metrics specified by this metrics reporting configuration.	
metrics	`array(Uri)`	0..1	A list of one or more QoE metrics, each indicated by a fully-qualified term from a controlled vocabulary, which shall be reported. If omitted, the complete (or default if applicable) set of metrics associated with the specified *scheme* shall be collected and reported.	
networkAssistance Configuration	`object`	0..1	Present if Network Assistance is provisioned in the parent Provisioning Session.	*MS_DOWNLINK, MS_UPLINK, RTC*
serverAddresses	`array(AbsoluteUrl)`	1..1	A list of Media AF addresses (URLs) that offer the APIs for AF-based Network Assistance at reference point M5. (See NOTE 1.) Each address shall be an opaque URL, following the format specified in clause 7.1.3 up to and including the *{apiVersion}* path element.	
clientEdgeResources Configuration	`object`	0..1	Present only for Provisioning Sessions with client-driven edge computing management mode provisioned.	*MS_DOWNLINK, MS_UPLINK, RTC*
eligibilityCriteria	`EdgeProcessing EligibilityCriteria`	0..1	Conditions for activating edge resources for media delivery sessions in the scope of the parent Provisioning Session. (See clause 7.3.3.10.)	
easDiscoveryTemplate	`EASDiscoveryTemplate`	1..1	A template for the EAS discovery filter that shall be used by the EEC to discover and select a Media EAS instance to serve media delivery sessions at reference point M4 in the scope of the parent Provisioning Session. (See clause 9.2.3.3.)	
easRelocation Requirements	`M5EASRelocation Requirements`	0..1	EAS relocation tolerance and requirements. If absent, the EEC shall assume that relocation is tolerated by all Media EAS instances in the scope of the parent Provisioning Session. (See clause 9.2.3.4.)	

NOTE 1: In deployments where multiple instances of the Media AF expose the Media Session Handling APIs at reference point M5, the 5G System may use a suitable mechanism (e.g., HTTP load balancing or DNS-based host name resolution) to direct requests to a suitable Media AF instance.
NOTE 2: The *Snssai* data type is specified in TS 29.571 [33].

9.2.3.2 M5EndpointAccess

This data type is derived by extension from *EndpointAddress* (see clause 7.3.3.11).

Table 9.2.3.2-1: Definition of M5EndpointAccess data type

Property name	Data Type	Cardinality	Description
domainName	`string`	1..1	The Fully-Qualified Domain Name of this service endpoint.
portNumbers	`array(Uint16)`	1..1	A list of listening ports over which the service is accessible (e.g. a STUN service offers two listening ports). The array shall contain at least one member.
credentials	`object`	0..1	Authentication information to be presented to this service endpoint.
username	`string`	1..1	A username that is authorised to access the service.
passphrase	`string`	1..1	The passphrase associated with *username*.

9.2.3.3 EASDiscoveryTemplate type

Table 9.2.3.3-1: Definition of EASDiscoveryTemplate type

Property name	Type	Cardinality	Description
easId	`string`	0..1	The application identifier of the EAS, e.g. reverse FQDN, URI. If omitted, any Media EAS instance matching the other criteria specified in the template are acceptable. Corresponding to *EasCharacteristics.easId*, as specified in clause 6.3.5.2.7 of TS 24.558 [14].
easType	`string`	0..1	If present, a non-empty string indicating the type of Media EAS required to support media delivery sessions in the scope of this discovery template. Corresponding to *EasCharacteristics.easType*, as specified in clause 6.3.5.2.7 of TS 24.558 [14].
easProviderIds	`array(string)`	0..1	The non-empty set of acceptable Media EAS provider identifiers, each expressed as a non-empty string. If omitted, Media EAS instances of the specified *easType* from any provider are acceptable. Corresponding to *EasCharacteristics.easProvId*, as specified in clause 6.3.5.2.7 of TS 24.558 [14].
easFeatures	`array(string)`	0..1	The non-empty set of required service features for the Media EAS to serve media delivery sessions, each expressed as a non-empty string. If omitted, Media EAS instances of the specified *easType* with any feature set are acceptable. Corresponding to *EasCharacteristics.svcFeats*, as specified in clause 6.3.5.2.7 of TS 24.558 [14].

At least one of the properties shall be populated:

9.2.3.4 M5EASRelocationRequirements type

Table 9.2.3.4-1: Definition of M5EASRelocationRequirements type

Property name	Type	Cardinality	Description
tolerance	`EASRelocationTolerance`	1..1	Indicates whether the Media EAS instance tolerates relocation. (See clause 7.3.4.4.)
maxInterruptionDuration	`UintegerRm`	0..1	The maximum downtime (expressed in milliseconds) that an application can tolerate during EAS relocation. If the expected downtime of the application is expected to exceed this duration, relocation of the Media EAS instance shall not be performed.

9.3 Dynamic Policy API

9.3.1 Overview

The Dynamic Policy API allows the Media Session Handler to request a specific policy and charging treatment to be applied to a particular application data flow of a downlink or uplink media delivery session by invoking RESTful operations on the Media AF at reference point M5. The API defines a set of data models, resources and the related operations for the creation and management of the dynamic policy request.

9.3.2 Resource structure

The Dynamic Policies API is accessible through the following URL base path:

{apiRoot}/3gpp-maf-session-handling/*{apiVersion}*/provisioning-sessions/*{provisioningSessionId}*/dynamic-policies/

where the first three path elements shall be substituted by the Media Session Handler with one of the URLs selected from the *dynamicPolicyInvocation Configuration.serverAddresses* array of the *ServiceAccessInformation* resource (see clause 9.2.3.1) and the fifth path element shall be substituted with the value of the relevant Provisioning Session identifier obtained from the same resource.

Table 9.3.2-1 below specifies the operations and the corresponding HTTP methods that are supported by this API. The sub-resource path specified in the second column shall be appended to the URL base path.

Table 9.3.2-1: Operations supported by the Dynamic Policies API

Operation name	Sub-resource path	Allowed HTTP methods	Description
Create Dynamic Policy resource		POST	Create a new Dynamic Policy resource.
Retrieve Dynamic Policy resource	*{dynamicPolicyId}*	GET	Retrieve an existing Dynamic Policy resource.
Update Dynamic Policy resource		PUT	Replace an existing Dynamic Policy resource.
		PATCH	Modify an existing Dynamic Policy resource.
Destroy Dynamic Policy resource		DELETE	Remove an existing Dynamic Policy resource.

9.3.3 Data model

9.3.3.1 DynamicPolicy resource

Table 9.3.3.1-1: Definition of Dynamic Policy Instance resource

Property name	Data type	Cardinality	Usage	Description
dynamicPolicyId	`ResourceId`	1..1	RO	Unique identifier for this Dynamic Policy Instance assigned by the Media AF when the resource is created.
provisioningSessionId	`ResourceId`	1..1	C: RO R: RO U: RO	Uniquely identifies the parent Provisioning Session, which is linked to the Application Service Provider.
sessionId	`MediaDelivery SessionId`	1..1	C: RW R:RO U: RO	Unique identifier of the current media delivery session assigned by the Media Session Handler.
policyTemplateId	`ResourceId`	1..1	C: RW R: RO U: RW	Identifies the Policy Template to be applied to the application flow(s) that fall within the scope of this Dynamic Policy Instance.
sliceInfo	`Snssai`	0..1	C: RW R: RO U: RW	Identifying the target slice in which the Policy Template is instantiated.
dataNetworkName	`Dnn`	0..1	C: RW R: RO U: RW	The name of the target Data Network in which the Policy Template is instantiated.
location	`TypedLocation`	0..1	C: RW R: RO U: RW	The location of the UE when the Dynamic Policy was created or last updated.
applicationFlowBindings	`array(Application FlowBinding)`	1..1	C: RW R: RO U: RW	The bindings between application flows at reference point M4 managed within the scope of this Dynamic Policy Instance and their network Quality of Service requirements (see clause 9.3.3.2). The array shall contain at least one member.
componentIdentifier	`string`	1..1	C: RW R: RO U: RW	References a particular service component in the Policy Template.
applicationFlowDescription	`ApplicationFlow Description`	1..1	C: RW R: RO U: RW	The Media Client's specification of an application flow managed by this Dynamic Policy to be used for application traffic identification purposes in the 5G Core (see clause 7.3.3.2). When PDU Set handling is enabled for the Policy Template identified by *policyTemplateId*, this property shall also specify the media transport protocol parameters to be used by the Media Access Function for PDU Set signalling purposes.
qosSpecification	`M5QoSSpecification`	0..1	C: RW R: RO U: RW	The Media Client's network Quality of Service requirements of the application flow described by *applicationFlowDescription*. If omitted, the default provisioned network Quality of Service requirements of the Policy Template indicated in *policyTemplateId* shall apply to *applicationFlowDescription*.
bdtSpecification	`M5BDTSpecification`	0..1	C: RW R: RO	The Background Data Transfer time windows and traffic limits that apply to this Dynamic Policy (see clause 9.3.3.3).

			U: RW	
qosEnforcement	`boolean`	1..1	C: RO R: RO U: RO	Indication that the Quality of Service described in *qosSpecification* is being enforced by the 5G System. Populated by the Media AF.

9.3.3.2 ApplicationFlowBinding

Table 9.3.3.2-1: ApplicationFlowBinding type

Property name	Data type	Cardinality	Description
componentReference	`string`	1..1	References a particular service component in the Policy Template.
applicationFlowDescription	`ApplicationFlowDescription`	1..1	The specification of an application flow to be used by the 5G Core for application traffic identification purposes (see clause 7.3.3.2).
qosSpecification	`M5QoSSpecification`	0..1	The network Quality of Service requirements of the application flow(s) described by *applicationFlowDescription*.

9.3.3.3 M5BDTSpecification type

Table 9.3.3.3-1: M5BDTSpecification type

Property name	Type	Cardinality	Description
estimatedDataTransferVolume	`UsageThreshold`	0..1	The data volume that the Media Client expects to transfer in both directions at reference point M4 during the current time window (see NOTE). Value provided by the Media Session Handler to the Media AF.
windows	`array(BDTWindow)`	1..1	Windows when Background Data Transfers are requested by the Media Session Handler or granted by the Media AF. The array shall contain at least one window specification.
NOTE: Data type *UsageThreshold* is defined in TS 29.122 [20].			

9.4 Network Assistance API

9.4.1 Overview

If AF-based Network Assistance is supported by the Media AF, then the Network Assistance API, as defined in the present sub-clause, is first used to provision a Network Assistance Session resource at reference point M5. The Network Assistance resource can then be used to obtain bit rate recommendations and to issue delivery boost requests during the ongoing media delivery session.

9.4.2 Resource structure

The Network Assistance API is accessible via the following URL base path:

{apiRoot}/3gpp-maf-session-handling/*{apiVersion}*/provisioning-sessions/*{provisioningSessionId}*/network-assistance-sessions/

where the first three path elements shall be substituted by the Media Session Handler with one of the URLs selected from the *networkAssistanceConfiguration.serverAddresses* array of the *ServiceAccessInformation* resource (see clause 9.2.3.1) and the fifth path element shall be substituted with the relevant Provisioning Session identifier obtained from the same resource.

Table 9.4.2-1 below specifies the operations and the corresponding HTTP methods that are supported by this API. In each case, the sub-resource path specified in the second column of the table shall be appended to the URL base path.

Table 9.4.2-1: Operations supported by the Network Assistance API

Operation name	Sub-resource path	Allowed HTTP method(s)	Description
Create Network Assistance Session resource		POST	Provision a new Network Assistance Session. If the operation succeeds, the URL of the created Network Assistance Session resource shall be returned in the Location header of the response.
Retrieve Network Assistance Session resource	*{naSessionId}*	GET	Fetch the properties of an existing Network Assistance Session.
Bit rate recommendation request	*{naSessionId}*/recommendation	GET	Obtain a bit rate recommendation.
Delivery boost request	*{naSessionId}*/boost-request	POST	Request a delivery boost.
Update Network Assistance Session resource	*{naSessionId}*	PUT, PATCH	Update the properties of an existing Network Assistance Session.
Destroy Network Assistance Session	*{naSessionId}*	DELETE	Terminate a Network Assistance session.

9.4.3 Data model

9.4.3.1 NetworkAssistanceSession resource

Table 9.4.3.1-1: Definition of NetworkAssistanceSession resource

Property name	Type	Cardinality	Usage	Description
naSessionId	`ResourceId`	1..1	C: RO R: RO U: RO	Unique identifier for this Network Assistance Session assigned by the Media AF when the resource is created.
provisioningSessionId	`ResourceId`	1..1	C: RO R: RO U: RO	Uniquely identifies the parent Provisioning Session, which is linked to the Application Service Provider.
sessionId	`MediaDelivery SessionId`	1..1	C: RW R:RO U: RO	Unique identifier of the current media delivery session assigned by the Media Session Handler.
sliceInfo	`Snssai`	0..1	C: RW R: RO U: RW	Identifying the target network slice in which Network Assistance is sought.
dataNetworkName	`Dnn`	0..1	C: RW R: RO U: RW	The name of the target Data Network in which Network Assistance is sought.
location	`TypedLocation`	0..1	C: RW R: RO U: RW	The location of the UE when the Network Assistance Session was created or last updated.
policyTemplateId	`ResourceId`	0..1	C: RW R: RO U: RW	Identification of the policy (if any) that is currently in force for the media delivery session.
componentReference	`string`	0..1	C: RW R: RO U: RW	References a particular service component in the Policy Template. This property shall be present if *policyTemplate* is present.
applicationFlowDescription	`ApplicationFlow Description`	1..1	C: RW R: RO U: RW	Identifying the application flow for which Network Assistance is sought, e.g. 2-tuple (IP address pair) or 5-tuple (IP address pair, port pair and protocol).
requestedQoS	`M5QoSSpecification`	0..1	C: RW R: RO U: RW	The QoS parameters requested by the Media Session Handler.
recommendedQoS	`M5QoSSpecification`	0..1	C: RO R: RO U: RO	The QoS parameters currently recommended by the Media AF.

9.5 Metrics Reporting API

9.5.1 General

The Metrics Reporting API allows the Media Session Handler to send QoE metrics reports to the Media AF. This procedure is configured by the *ServiceAccessInformation* resource, as defined in clause 9.2.3.1. Multiple metrics reporting configurations may be active at the same time, each identified by a unique *metricsReportingConfigurationId*.

9.5.2 Reporting procedure

Metrics reports related to a specific *metricsReportingConfigurationId* shall be submitted according to the following general format:

{apiRoot}`/3gpp-maf-session-handling/`*{apiVersion}*`/provisioning-sessions/`*{provisioningSessionId}*`/metrics-reporting/`*{metricsReportingConfigurationId}*

where the first three path elements shall be substituted by the Media Session Handler with one of the base URLs selected from the *clientMetricsReportingConfigurations.serverAddresses* array of the *ServiceAccessInformation* resource (see clause 9.2.3.1), the fifth path element shall be substituted with the relevant Provisioning Session identifier obtained from the same resource and *{metricsReportingConfigurationId}* shall be substituted with the relevant Metrics Reporting Configuration identifier.

The only HTTP method supported by this endpoint is `POST`.

9.5.3 Report format

Metrics reports shall be submitted by the Media Session Handler in a format specified by the metrics scheme in question. The `Content-Type` HTTP request header shall be set in accordance with the specification of the relevant metrics scheme.

Metrics schemes specified by 3GPP shall make provision to convey the media delivery session identifier in their metrics reports. For metrics reporting formats specified elsewhere, the 3GPP specification referencing the metrics scheme should specify a means to convey the media delivery session identifier in metrics reports where practicable.

9.6 Consumption Reporting API

9.6.1 General

The Consumption Reporting API allows the Media Session Handler to report downlink media consumption to the Media AF. The API defines data models, resources and the related operations for the creation and management of the consumption reporting procedures. This feature is configured by the *ServiceAccessInformation* resource, as defined in clause 9.2.3.1.

9.6.2 Reporting procedure

Consumption reports shall be submitted to a Media AF endpoint according to the following general URL format:

{apiRoot}`/3gpp-maf-session-handling/`*{apiVersion}*`/provisioning-sessions/`*{provisioningSessionId}*`/consumption-reporting/`

Where the first three path elements shall be substituted by the 5GMSd Client with one of the base URLs selected from the *clientConsumptionReportingConfiguration.serverAddresses* array of the *ServiceAccessInformation* resource (see clause 9.2.3.1) and the fifth path element shall be substituted with the relevant Provisioning Session identifier obtained from the same resource.

The only HTTP method supported by this endpoint is `POST`.

9.6.3 Report format

9.6.3.1 ConsumptionReport type

This data type specifies the root object of a consumption report instance document used by the Media Session Handler to report media consumption.

Table 9.6.3.1-1: Definition of ConsumptionReport format

Property name	Data type	Cardinality	Description
reportingClientId	`string`	1..1	Identifier of the reporting client that consumed the streaming media service associated with this consumption report. If available to the Media Session Handler, a GPSI value (see clause 28.8 of TS 23.003 [7]); otherwise, a stable and globally unique string.
sessionId	`MediaDelivery SessionId`	1..1	Unique identifier of the current media delivery session assigned by the Media Session Handler.
mediaPlayerEntry	`AbsoluteUrl`	1..1	Identifies the Media Entry Point. The content of this property is not specified in the present document.
consumptionReportingUnits	`array(Consumption ReportingUnit)`	1..1	A list of consumption reporting units, ordered by start time. The content of this property is not specified in the present document.

9.6.3.2 ConsumptionReportingUnit type

This data type represents a single consumption reporting unit.

Table 9.6.3.2-1: Definition of type ConsumptionReportingUnit

Property name	Data type	Cardinality	Description
mediaConsumed	`string`	1..1	Identifies the media consumed. The content of this property is not specified in the present document.
clientEndpointAddress	`EndpointAddress`	0..1	The IP address and port number of the Media Access Function endpoint used to access the media consumed (see clause 7.3.3.11). Present only if access reporting is enabled in the Consumption Reporting Configuration.
serverEndpointAddress	`EndpointAddress`	0..1	The IP address, port number and host name of the Media AS endpoint used to access the media consumed (see clause 7.3.3.11). Present only if access reporting is enabled in the Consumption Reporting Configuration.
startTime	`DateTime`	1..1	The time when this consumption reporting unit started.
duration	`DurationSec`	1..1	The duration of this consumption reporting unit relative to *startTime*. The value shall not be negative. Media consumed for less than 1 second should be reported with *duration* = 0. For consumption reporting units describing the currently consumed media, this shall indicate the duration so far.
sliceInfo	`Snssai`	0..1	Identifying the network slice in which the media was consumed.
dataNetworkName	`Dnn`	0..1	The name of the Data Network in which the media was consumed.
locations	`array(TypedLocation)`	0..1	A time-ordered list of one or more UE location(s) where the media was consumed during the period of this consumption reporting unit (see clause 7.3.3.8). Present only if location reporting is enabled in the Consumption Reporting Configuration (only for trusted Media AF).

10 Ancillary network media session handling services

10.1 Overview

This clause specifies ancillary network media session handling services used by a Media AS at reference point M3 or by a Media Session Handler at reference point M5 to interact with the Media AF.

10.2 Resource update notification channel

10.2.1 General

The Media Session Handler and the Media AF shall support the usage of an MQTT notification channel that is used by the Media AF to notify the Media Session Handler about updates to certain resources specified in clause 9. For this purpose, the MQTT protocol [50] shall be used over TLS as specified below.

The Media AF shall use a trusted MQTT broker for the exchange of these notifications. The MQTT broker shall use appropriate client authentication mechanisms, such as a token-based authentication mechanism (JWT) [51] to secure access to the notification channel.

The MQTT Topics shall be structured as defined in clause 10.2.2.

The MQTT Messages shall be formated according to clause 10.2.3.

10.2.2 Topic structure of notification channel

The MQTT Topics on the resource update notification channel shall be formatted according to the following ABNF syntax:

Topic = externalServiceId ["/" resourceId]

Where:

- *externalServiceId* is the external service identifier that is associated with a Provisioning Session.
- The *resourceId* if present, identifies a sub-topic associated with a resource previously created at reference point M5 by a particular Media Session Handler and having *resourceId* as its resource identifier.

The Media Session Handler shall at least subscribe to the *{externalServiceId}* topic, enabling the reception of all common notification messages relating to a particular Provisioning Session. Service Access Information update notification messages shall be published to the common topic. Dynamic Policy Instance and Network Assistance Session update notification messages shall be published to the sub-topics using their respective *resourceId* values, and are specific to a particular media delivery session.

10.2.3 Notification message format

The Media AF shall format each notification it publishes to the notification channel as an MQTT Application Message conveyed as the payload of an MQTT *PUBLISH* message.

- The *Topic* property of the Variable Header shall be as specified in clause 10.2.2.
- The *Payload Format Indicator* property of the Variable Header shall indicate UTF-8 encoding of the *Payload* field.

The notification message shall be conveyed in the *Payload* field which shall be a formatted as a *NotificationMessage* JSON [37] object as specified in table 10.2.3-1 using the UTF-8 character encoding.

Table 10.2.3-1: NotificationMessage data type

Property name	Type	Cardinality	Description
type	*NotificationMessageType*	1..1	The type of notification message (see table 10.2.3-2).
serviceAccessInformation	*ServiceAccessInformation*	0..1	Present if *type* is *NOTIFICATION_SERVICE_ACCESS_INFORMATION*.
dynamicPolicy	*DynamicPolicy*	0..1	Present if *type* is *NOTIFICATION_DYNAMIC_POLICY_INSTANCE*.
networkAssistanceSession	*NetworkAssistanceSession*	0..1	Present if *type* is *NOTIFICATION_NETWORK_ASSISTANCE_SESSION*.

Exactly one of the following properties shall be present: *serviceAccessInformation*, *dynamicPolicy*, *networkAssistance Session*.

The type of the notification message shall be indicated using one of the values in table 10.2.3-2.

Table 10.2.3-2: NotificationMessageType enumeration

Enumeration value	Description
NOTIFICATION_SERVICE_ACCESS_INFORMATION	Notification of a change to a Service Access Information resource.
NOTIFICATION_DYNAMIC_POLICY_INSTANCE	Notification of a change to a Dynamic Policy Instance resource.
NOTIFICATION_NETWORK_ASSISTANCE_SESSION	Notification of a change to a Network Assistance Session resource.

11 UE media session handling APIs

11.1 Introduction

This clause defines the abstract client APIs exposed by the Media Session Handler to the Media-aware Application at reference point M6 and to the Media Access Function at reference point M11. The APIs may be used to query a subset of information from Service Access Information and its updates as well as to control the life-cycle of a media delivery session and to receive notifications of various events occurring during that media delivery session.

NOTE: Client-driven management of edge processing resources via reference point EDGE-5 is not specified in this release.

11.2 Media Session Handler client API

11.2.1 Media Session Handler internal properties

The Media Session Handler maintains internal properties as defined table 11.2.1-1. Note that the parameters are conceptual and internal. They serve only for the purpose of defining the media session handling APIs.

Table 11.2.1-1: Parameters of Media Session Handler internal data model

States and Parameters	Definition
_Configuration[externalServiceId]	The Media Session Handler maintains a separate configuration for each set of Service Access Information it has knowledge of, indexed by its external service identifier.
_streamingAccess	Streaming access configuration.
_rtcClient	RTC Client configuration.
_networkAssistance	Network Assistance configuration.
_policyTemplate	Policy Template configuration.
_consumptionReporting	Consumption reporting configuration.
_metricsReporting	Metrics reporting configuration.
_edgeResourcesConfiguration	Edge resources configuration.
_status[mediaDeliverySessionId]	The Media Session Handler maintains a separate status record for each currently active media delivery session, indexed by media delivery session identifier.
_generalStatus	General status information. (See table 11.2.3-1.)
_dynamicPolicyStatus	Dynamic Policy status information. (See table 11.3.2-1)
_networkAssistanceStatus	Network Assistance status information. (See table 11.4.2-1)
_consumptionReportingStatus	Consumption Reporting status information. (See table 11.5.2-1.)
_metricsReportingStatus	Metrics Reporting status information. (See table 11.6.2-1.)

A subset of the above information which is needed by the 5GMS-Aware Application and/or by the Media Access Function is accessible through reference points M6 and M11, respectively, using the methods specified in the following clauses.

11.2.2 General Media Session Handler methods

11.2.2.1 Create a media delivery session

A 3GPP Service URL (see clause 6) may be used to implicitly trigger the creation of a new media delivery session with the Media Session Handler.

The Media Session Handler also offers the explicit `createMediaDeliverySession()` method, which is used to create a new media delivery session in the Media Session Handler.

The input parameters of the method are specified in table 11.2.2.1-1:

Table 11.2.2.1-1: Input parameters for `createMediaDeliverySession()` method

Name	Type	O	Description
serviceId	`string`	M	The external service identifier (see table 8.2.3.1-1) of the Provisioning Session that this media delivery session pertains to.
entryPoint	`Url`	O	The location of a Media Entry Point document or media resource.
domainNames	`array(string)`	O	A set of Fully-Qualified Domain Name (FQDN) of the Media AS endpoint(s) supporting the media delivery session at reference point M4.
accessToken	`string`	O	An access token that the Media Session Handler presents to the Media AF to authorise invocation of media session handling operations at reference point M5.

If it does not already have a fresh copy cached, the Media Session Handler shall attempt to retrieve a copy of the full Service Access Information from the Media AF at reference point M5 using the procedure specified in clause 5.3.2.

If successful, the Media Session Handler shall assign a new media delivery session identifier to the media delivery session and shall create an entry in its *_status* array indexed by the media delivery session identifier.

If the *entryPoint* input parameter is provided, the Media Session Handler shall attempt to initialise the Media Access Function using an appropriate method, and shall pass the Media Entry Point URL to it (as well as the media delivery session identifier) in order to initiate media access.

If the *entryPoint* input parameter is provided, and indicates a Service Operation Point, the Media Session Handler shall create a Dynamic Policy Instance using the procedure specified in clause 5.3.3 using the Service Operation Point reference as the external reference. The Dynamic Policy Instance shall include a Policy Template binding for each of the domain names listed in the *domainNames* input parameter, if present.

If all of the above actions are successful, the Media Session Handler shall set *sessionHandlingState* to *ACTIVE* (see table 11.2.3-1) and shall send a *SESSION_HANDLING_ ACTIVATED* notification (see table 11.2.3-2). If any of the above actions fail, the Media Session Handler shall set *sessionHandlingState* to *ERRORED* (see table 11.2.3-1).

The return value of the method is specified in table 11.2.2.1-2.

Table 11.2.2.1-2: Return value for `createMediaDeliverySession()` method

Type	Description
`string`	The media delivery session identifier.

11.2.2.2 Destroy a media delivery session

The `destroyMediaDeliverySession()` method is used to end the media delivery session and release the allocated resources by the Media Session Handler. As a result, the Media Session Handler does no longer maintains the internal properties corresponding to the media delivery session identifier.

The input parameters of the method are specified in table 11.2.2.2-1.

Table 11.2.2.2-1: Input parameters for `destroyMediaDeliverySession()` method

Name	Type	Description
mediaDeliverySessionIdentifier	`string`	The media delivery session identifier.

The Media Session Handler shall remove the entry from its *_status* array indexed by the media delivery session identifier.

The Media Session Handler shall send a *SESSION_HANDLING_TERMINATED* notification (see table 10.2.3-2).

11.2.3 General Media Session Handler information

Table 11.2.3-1 specifies the status information that can be obtained from the Media Session Handler.

Table 11.2.3-1: General Media Session Handler Status Information

Status	Type	Parameter	Definition
sessionHandlingState	`string enum`	Media delivery session identifier	The status of media delivery session: *ACTIVE*: The media delivery session is being handled by the Media Session Handler. *ERRORED*: An error has occurred, and the Media Session Handler is no longer able to handle it.

Table 11.2.3-2 provides a list of general notification events exposed by the Media Session Handler through reference points M6 and M11.

Table 11.2.3-2: General Media Session Handler Notification Events

Event	Definition	Payload
SESSION_HANDLING_ACTIVATED	Triggered when media session handling was activated for a specific Media Entry Point.	Media delivery session identifier, Media Entry Point URL.
SESSION_HANDLING_TERMINATED	Triggered when media session handling is terminated for a specific Media Entry Point.	Media delivery session identifier, Media Entry Point URL.
STREAMING_ACCESS_UPDATED	Triggered when an update to the stream access is available for the Provisioning Session associated with the external service identifier supplied when the media delivery session was created (see clause 10.2.2.1).	Media delivery session identifier, Streaming access.
RTC_CLIENT_CONFIGURATION_UPDATED	Triggered when an update to the RTC Client configuration is available for the Provisioning Session associated with the external service identifier supplied when the media delivery session was created (see clause 10.2.2.1).	Media delivery session identifier, RTC Client configuration.

Table 11.3.3-3 provides a list of general error events exposed by the Media Session Handler through reference points M6 and M11.

Table 11.2.3-3: General Media Session Handler Error Events

Status	Definition	Payload
ERROR_SESSION_HANDLING	Triggered when there is an error in the media session handling.	Media delivery session identifier.

11.3 Dynamic Policy client API

11.3.1 Dynamic Policy methods

11.3.1.1 Retrieve Background Data Transfer information

The `getBDTInfo()` method is used to retrieve information about the next Background Data Transfer opportunity window at one of the Service Operation Points that are available in the context of a particular media delivery session.

The input parameters of the method are specified in tables 11.3.1.1-1.

Table 11.3.1.1-1: Input parameters for `getBDTInfo()` method

Name	Type	Description
sessionId	`string`	The media delivery session identifier (as specified in clause 7.3.2) of an initialised media delivery session in the Media Session Handler.
serviceOperationPointReference	`string`	The external reference identifier of a Service Operation Point that uniquely identifies a Policy Template within the context of *sessionId*.

The return value of the method is specified in table 11.3.1.2-1.

Table 11.3.1.2-1: Return value for `getBDTInfo()` method

Type		Description
	`object`	Information about a Background Data Transfer opportunity.
windowStart	`dateTime`	The start date–time of the Background Data Transfer window.
windowEnd	`dateTime`	The end date–time of the Background Data Transfer window.
maximumDataTransferVolume	`integer`	The maximum volume of data (expressed in bytes) that a Media Access Function is permitted to transfer during the Background Data Transfer window.

11.3.1.2 Activate Dynamic Policy

The activatePolicy() method is employed to request the application of a dynamic policy to a media delivery session that is configured at the Media Session Handler. The scope of the dynamic policy is all application flows that match the Media AS domain name declared when the media delivery session was created (see table 11.2.2.1-1). The application may also provide the estimated transfer volume if the media delivery session is expected to be within the bounds of a Background Data Transfer time window. The Media Session Handler convey the request to the Media AF and provides the corresponding response to the invoker of the method. The input parameters of the method are specified in table 11.3.1.2-1.

Table 11.3.1.2-1: Input parameters for `activatePolicy()` method

Name	Type	O	Description
sessionId	`string`	M	The media delivery session identifier (as specified in clause 7.3.2) of an initialised media delivery session in the Media Session Handler.
serviceOperationPointReference	`string`	M	The external reference identifier of a Service Operation Point that uniquely identifies a Policy Template within the context of *sessionId*.
estimatedTransferVolume	`integer`	C	The estimated volume of data to be transferred, expressed in bytes. Minimum value 1 byte. Required to be populated when the Policy Template corresponding to the referenced Service Operation Point declares a Background Data Transfer policy.

The return value of the method is specified in table 11.3.1.2-2.

Table 11.3.1.2-2: Return value for `activatePolicy()` method

Type		Description
	`object`	
`recommendedDownlinkBitRate`	`dateTime`	The recommended downlink bit rate for the requested Service Operation Point.
`recommendedUplinkBitRate`	`dateTime`	The recommended uplink bit rate for the requested Service Operation Point.
`backgroundDataTransferActivated`	`integer`	Indicates whether Background Data Transfer has been successfully activated for the media delivery session for the duration of the indicated time window.

11.3.2 Dynamic Policy information

Table 11.3.2-1 specifies the status information that can be obtained from the Media Session Handler.

Table 11.3.2-1: Status Information relating to Dynamic Policies

Status	Type	Parameter	Definition
currentDynamicPolicies[mediaDeliverySession]	`object`		Descriptions of the Dynamic Policies currently instantiated for each current media delivery session, including the external reference identifier of its Service Operation Point and details of applicable Background Data Transfer quotas, if any.

Table 11.3.2-2 provides a list of general notification events exposed by the Media Session Handler.

Table 11.3.2-2: Notification Events relating to Dynamic Policies

Event	Definition	Payload
POLICY_ACTIVATED	Triggered when a new Dynamic Policy is successfully activated for the media delivery session.	Media delivery session identifier, Recommended downlink bit rate, Recommended uplink bit rate.
POLICY_DEACTIVATED	Triggered when the Dynamic Policy for this media delivery session is deactivated.	Media delivery session identifier.

Table 11.3.3-3 provides a list of error events exposed by the Media Session Handler through reference points M6 and M11 in relation to Dynamic Policies.

Table 11.3.2-3: Error Events relating to Dynamic Policies

Status	Definition	Payload
ERROR_INVALID_SERVICE_OPERATION_POINT	Triggered when the provided Service Operation Point reference is not valid for the media delivery session.	Media delivery session identifier, Service Operation Point reference.
ERROR_UNAUTHORISED	Triggered when the application is not authorised to instantiate a dynamic policy for the provided Service Operation Point reference.	Media delivery session identifier, Service Operation Point reference.
ERROR_BACKGROUND_DATA_TRANSFER	Triggered when there is an error during a Background Data Transfer, for example if it is cancelled before the end of the advertised opportunity window.	Media delivery session identifier, Error reason.

11.4 Network Assistance client API

11.4.1 Network Assistance methods

11.4.1.1 Bit rate recommendation request

The specification of this method is for further study.

11.4.1.2 Delivery boost request

The specification of this method is for further study.

NOTE: The duration and network QoS of the delivery boost is at the discretion of the Media Delivery System.

11.4.2 Network Assistance information

Table 11.4.2-1 specifies the status information that can be obtained from the Media Session Handler.

Table 11.4.2-1: Status Information relating to Network Assistance

Status	Type	Parameter	Definition

Table 10.4.2-2 provides a list of general notification events exposed by the Media Session Handler.

Table 11.4.2-2: Notification Events relating to Network Assistance

Event	Definition	Payload
BIT_RATE_RECOMMENDATION	Triggered when a bit rate recommendation is received from the Media AF or from the UE modem. The source of the recommendation is indicated as a parameter of the payload.	Media delivery session identifier, Recommendation source, Recommended downlink bit rate, Recommended uplink bit rate.

Table 11.4.2-3 provides a list of general error events exposed by the Media Session Handler.

Table 11.4.2-3: Error Events relating to Network Assistance

Status	Definition	Payload
ERROR_BIT_RATE_RECOMMENDATION	Triggered when there is an error when requesting a bit rate recommendation from the Media AF or from the UE modem. The source of the recommendation is indicated as a parameter of the payload.	Media delivery session identifier, Recommendation source.
ERROR_DELIVERY_BOOST	Triggered when there is an error when requesting a delivery boost from the Media AF or from the UE modem. The source of the recommendation is indicated as a parameter of the payload.	Media delivery session identifier, Recommendation source.

11.5 Consumption Reporting API

11.5.1 Consumption Reporting methods

No methods are exposed by the Media Session Handler in this release to control Consumption Reporting by the Media Session Handler.

11.5.2 Consumption Reporting information

Table 11.5.2-1 specifies the status information relating to Consumption Reporting that is exposed by the Media Session Handler.

Table 11.5.2-1: Status Information relating to Consumption Reporting

Status	Type	Parameter	Definition
consumptionReport	`object`		The most recently sent consumption report.

Table 11.5.2-2 provides a list of general notification events exposed by the Media Session Handler.

Table 11.5.2-2: Notification Events relating to Consumption Reporting

Status	Definition	Payload
CONSUMPTION_REPORTING_ACTIVATED	Consumption reporting has been activated.	Media delivery session identifier.
CONSUMPTION_REPORTING_STOPPED	Consumption reporting has been stopped.	Media delivery session identifier.
NEW_CONSUMPTION_REPORT	A new consumption report is available and has been sent.	Media delivery session identifier.

Table 11.5.2-3 provides a list of general error events exposed by the Media Session Handler.

Table 11.5.2-3: Error Events relating to Consumption Reporting

Status	Definition	Payload
ERROR_CONSUMPTION_REPORTING	Error in consumption reporting occurred.	Media delivery session identifier, Provisioning Session Id, Server address, HTTP response code Error message.

11.6 Metrics Reporting client API

11.6.1 Consumption Reporting methods

No methods are exposed by the Media Session Handler in this release to control Metrics Reporting by the Media Session Handler.

11.6.2 Metrics Reporting information

Table 10.6.2-1 specifies the status information relating to Metrics Reporting that is exposed by the Media Session Handler.

Table 11.6.2-1: Status Information relating to Metrics Reporting

Status	Type	Definition
lastMetricsReport	`object`	Status information relating to the last sent metrics report.
provisioningSessionId	`ResourceId`	The Provisioning Session identifier for this metrics report.
metricsReportingConfigurationId	`ResourceId`	The metrics reporting configuration identifier for this report.
scheme	`Uri`	The metrics reporting scheme used by this metrics report (see clause 5.3.5).
metricsReport	`object`	The most recently sent metrics report.

Table 11.6.2-2 provides a list of general notification events exposed by the Media Session Handler.

Table 11.6.2-2: Notification Events relating to Metrics Reporting

Event	Definition	Payload
METRICS_REPORTING_ACTIVATED	Metrics reporting has been activated.	Media delivery session identifier.
METRICS_REPORTING_STOPPED	Metrics reporting has been stopped.	Media delivery session identifier.
NEW_METRICS_REPORT	A new metrics report is available and has been sent.	Media delivery session identifier.

Table 11.6.2-3 provides a list of general error events exposed by the Media Session Handler.

Table 11.6.2-3: Error Events relating to Metrics Reporting

Error event	Definition	Payload
ERROR_METRICS_REPORTING	Error in metrics reporting occurred.	Media delivery session identifier, Provisioning Session Id, Server address, Metrics Reporting Configuration Id, HTTP response code Error message.

Details of status information for RAN-based metrics reporting are for further study.

Annex A (normative): OpenAPI representation of HTTP REST APIs

A.1 General

The normative code specifying the APIs defined in clauses 7.3, 8 and 9 of the present document, including JSON Schema [38] representations of HTTP message bodies to be used with these APIs, is published on 3GPP Forge according to the OpenAPI 3.0.0 specification [32]. The YAML files corresponding to this version of the present document shall be published to the following location:

```
https://forge.3gpp.org/rep/all/5G_APIs/-/tags/TSG104-Rel18
```

Informative copies of these YAML files shall be distributed with the present document for the convenience only. Where any discrepancy exisits, the version on 3GPP Forge shall be considered definitive.

A.2 Data Types applicable to several APIs

For the purpose of referencing entities specified in this clause, it shall be assumed that the OpenAPI definitions are contained in a physical file named "TS26510_CommonData.yaml".

A.3 OpenAPI representation of Maf_Provisioning APIs

A.3.1 Maf_Provisioning_ProvisioningSessions API

For the purpose of referencing entities specified in this clause, it shall be assumed that the OpenAPI definitions are contained in a physical file named "TS26510_Maf_Provisioning_ProvisioningSessions.yaml".

A.3.2 Maf_Provisioning_ContentProtocols API

For the purpose of referencing entities specified in this clause, it shall be assumed that the OpenAPI definitions are contained in a physical file named "TS26510_Maf_Provisioning_ContentProtocols.yaml".

A.3.3 Maf_Provisioning_ServerCertificates API

For the purpose of referencing entities specified in this clause, it shall be assumed that the OpenAPI definitions are contained in a physical file named "TS26510_Maf_Provisioning_ServerCertificates.yaml".

A.3.4 Maf_Provisioning_ContentPreparationTemplates API

For the purpose of referencing entities specified in this clause, it shall be assumed that the OpenAPI definitions are contained in a physical file named "TS26510_Maf_Provisioning_ContentPreparationTemplates.yaml".

A.3.5 Maf_Provisioning_EdgeResources API

For the purpose of referencing entities specified in this clause, it shall be assumed that the OpenAPI definitions are contained in a physical file named "TS26510_Maf_Provisioning_EdgeResources.yaml".

A.3.6 Maf_Provisioning_PolicyTemplates API

For the purpose of referencing entities specified in this clause, it shall be assumed that the OpenAPI definitions are contained in a physical file named "TS26510_Maf_Provisioning_PolicyTemplates.yaml".

A.3.7 Maf_Provisioning_ContentHosting API

For the purpose of referencing entities specified in this clause, it shall be assumed that the OpenAPI definitions are contained in a physical file named "TS26510_Maf_Provisioning_ContentHosting.yaml".

A.3.8 Maf_Provisioning_ContentPublishing API

For the purpose of referencing entities specified in this clause, it shall be assumed that the OpenAPI definitions are contained in a physical file named "TS26510_Maf_Provisioning_ContentPublishing.yaml".

A.3.9 Maf_Provisioning_RealTimeCommunication API

For the purpose of referencing entities specified in this clause, it shall be assumed that the OpenAPI definitions are contained in a physical file named "TS26510_Maf_Provisioning_RealTimeCommunication.yaml".

A.3.9 Maf_Provisioning_MetricsReporting API

For the purpose of referencing entities specified in this clause, it shall be assumed that the OpenAPI definitions are contained in a physical file named "TS26510_Maf_Provisioning_MetricsReporting.yaml".

A.3.10 Maf_Provisioning_ConsumptionReporting API

For the purpose of referencing entities specified in this clause, it shall be assumed that the OpenAPI definitions are contained in a physical file named "TS26510_Maf_Provisioning_ConsumptionReporting.yaml".

A.3.11 Maf_Provisioning_EventDataProcessing API

For the purpose of referencing entities specified in this clause, it shall be assumed that the OpenAPI definitions are contained in a physical file named "TS26510_Maf_Provisioning_EventDataProcessing.yaml".

A.4 OpenAPI representation of Maf_SessionHandling APIs

A.4.1 Maf_SessionHandling_ServiceAccessInformation API

For the purpose of referencing entities specified in this clause, it shall be assumed that the OpenAPI definitions are contained in a physical file named "TS26510_Maf_SessionHandling_ServiceAccessInformation.yaml".

A.4.2 Maf_SessionHandling_DynamicPolicy API

For the purpose of referencing entities specified in this clause, it shall be assumed that the OpenAPI definitions are contained in a physical file named "TS26510_Maf_SessionHandling_DynamicPolicy.yaml".

A.4.3 Maf_SessionHandling_NetworkAssistance API

For the purpose of referencing entities specified in this clause, it shall be assumed that the OpenAPI definitions are contained in a physical file named "TS26510_Maf_SessionHandling_NetworkAssistance.yaml".

A.4.4 Maf_SessionHandling_MetricsReporting API

For the purpose of referencing entities specified in this clause, it shall be assumed that the OpenAPI definitions are contained in a physical file named "TS26510_Maf_SessionHandling_MetricsReporting.yaml".

A.4.5 Maf_SessionHandling_ConsumptionReporting API

For the purpose of referencing entities specified in this clause, it shall be assumed that the OpenAPI definitions are contained in a physical file named "TS26510_Maf_SessionHandling_ConsumptionReporting.yaml".

Annex B (normative): Controlled vocabularies

B.1 Media Delivery locator type

This controlled vocabulary is used to indicate the type of a locator in conjunction with the geofencing feature of the Content Hosting Configuration specified in clause 8.8.3.1.

Table B.1-1: Media Delivery locator type controlled vocabulary

Term identifier	Term name	Semantic
urn:3gpp:media-delivery:locator-type:iso3166	ISO 3166 administrative area	String representation of an ISO 3166-1 alpha-2 country code [41] (e.g. *US*, *CN*, *KR*, *GB*, *FR*) or an ISO 3166-2 code [42] comprising an alpha-2 country code and a country subdivision code valid for that country (e.g. *US-CA*, *CN-GD*, *KR-26*, *GB-ENG*, *GB-WSM*, *FR-IDF*, *FR-75*).
urn:3gpp:media-delivery:locatortype:trackingAreaCode	3GPP Tracking Area Code	The Fully-Qualified Domain Name representation of a Tracking Area Code, as defined in clause 19.4.2.3 of TS 23.003 [16].

Annex C (informative): Usage of TOS/DSCP for traffic identification

C.1 General

This annex provides guidelines on the usage of the Type of Service (TOS) field of the IPv4 header or Traffic Class field of the IPv6 header for the purpose of traffic identification in the Media Delivery System as part of different features, such as Dynamic Policies and AF-based Network Assistance. The IP Packet Filter Set defined in TS 23.501 [2] allows traffic filtering based on this field within the IP header.

C.2 Differentiated Services/TOS-enabled Collaboration Scenarios

Differentiated Services (DS) as specified in RFC 2474 [43] is a scalable scheme for managing application traffic by classifying the traffic into a set of coarse-grained traffic classes. A *Differentiated Service (DS) domain* is a continuous set of DS-capable routers, which are operated with a common set of configurations. Each IP packet in a DS domain is marked and conditioned according to its traffic class. A 6-bit *DS Code Point* (DSCP) of the 8-bit differentiated services field (DS field) is used for marking. The DS field replaces the TOS field in the IPv4 packet headers and the Traffic Class field in the IPv6 header.

End host systems may mark IP packets with a specific DSCP value prior to transmission. DS-enabled routers treat the packet according to the DSCP value when performing routing operations on it. Border gateway routers typically mark packets with a DSCP value based on some traffic policy, overriding any value set by hosts.

NOTE: Usage of Differentiated Services across administrative borders is technically possible. The preservation of the DSCP field by networks between the MNO network and the external Data Networks hosting the Media Delivery functions is assumed to be governed by an SLA and by transport-level arrangements that are outside 3GPP scope. When the DSCP field is used only for traffic identification, preservation of the DSCP field could be achieved by using a tunnelling solution.

The RFCs defining Differentiated Services recommend a set of Per-Hop Behaviors (PHB), namely:

- *Default Forwarding (DF) PHB*, defined in section 4.1 of RFC 2474 [43], is used for traffic without special treatment.
- *Class Selector PHB*, defined in section 4.2.2.2 of RFC 2474 [43] is used for maintaining backwards compatibility with the IP precedence field of TOS.
- *Expedited Forwarding (EF) PHB*, defined by RFC 3246 [44], is dedicated to low loss or low latency traffic.
- *Assured Forwarding (AF) PHB*, defined by RFC 2597 [45], offers different levels of forwarding assurances.

The DS domain operator can also implement additional custom PHBs.

In the context of TOS-based traffic identification and separation, it is reasonable to assume the Data Network north of the UPF (N6) is DS-enabled. The 5G System is embedded in a larger DS domain, using same TOS values across multiple devices in order to provide Quality of Service Support like a DSCP-enabled link. However, it is not required to deploy DS capable routers for using in order to use the TOS field in the IP packet filter set for traffic identification.

According to clause 4.1 of TS 26.501 [4], the 5GMS functions may be deployed within the trusted Data Network or an external Data network. As noted above, DS Code Points are often reset at network domain borders, but not always. There may be deployments e.g., with localized Edge Computing or with direct peering realizations, where the DSCP values can be used up to the Media AF and/or Media AS in an external Data Network. In this case, the logical DS domain is extended to include those externally-deployed Media Delivery functions.

Figure C.2-1 illustrates a deployment with a DS domain between the 5G System and the Media Delivery functions deployed in the external DN. (The model is also valid for deployments in which the Media Delivery functions both reside in the trusted DN.)

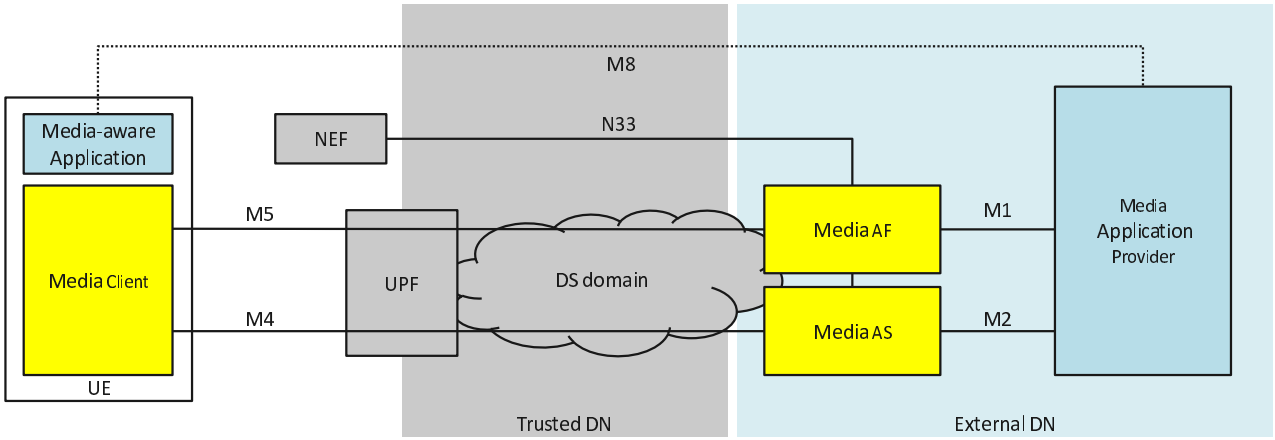

Figure C.2-1: Media Delivery System deployment within a DS domain

Figure C.2-2 illustrates a deployment with a DS domain between the 5G System and an externally deployed Media AS. The Media AF is deployed in the trusted DN.

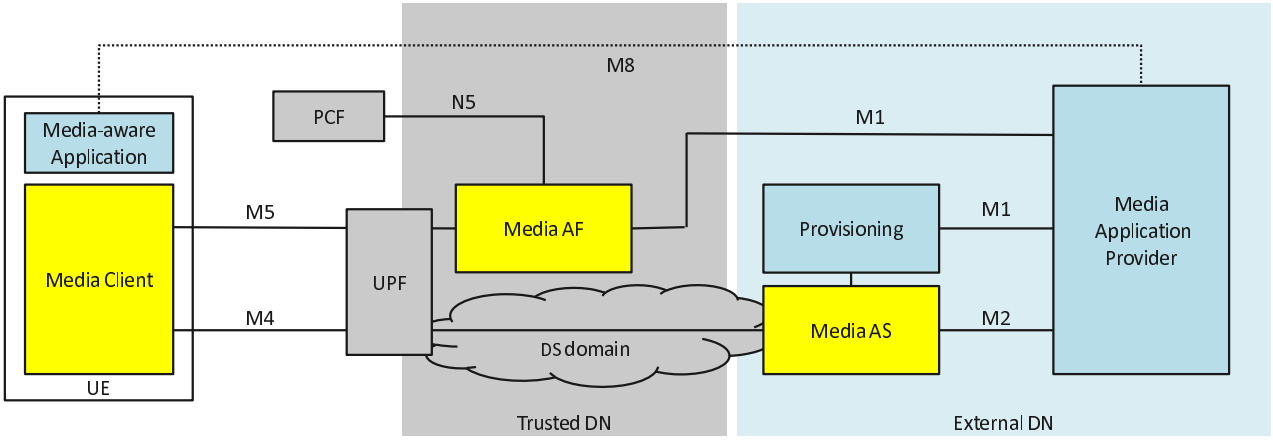

Figure C.2-2: Media Delivery System deployment within a DS domain

C.3 Procedure for using TOS Traffic Class for traffic identification

This call flow focuses on a scenario where both downlink and uplink traffic for a particular application flow within a PDU Session shared by several application flows needs to be mapped to a specific QoS Flow and handled separately by the 5G System. In this call flow, the Media Client initiates the QoS Flow establishment by using specific TOS values in the uplink traffic. A TOS-based QoS rule is already provisioned, so that the Uplink Traffic is mapped to the correct QoS Flow.

It is assumed here that the QoS flow should be used (e.g. for Premium QoS) as described in annex A of TS 26.501 [4].

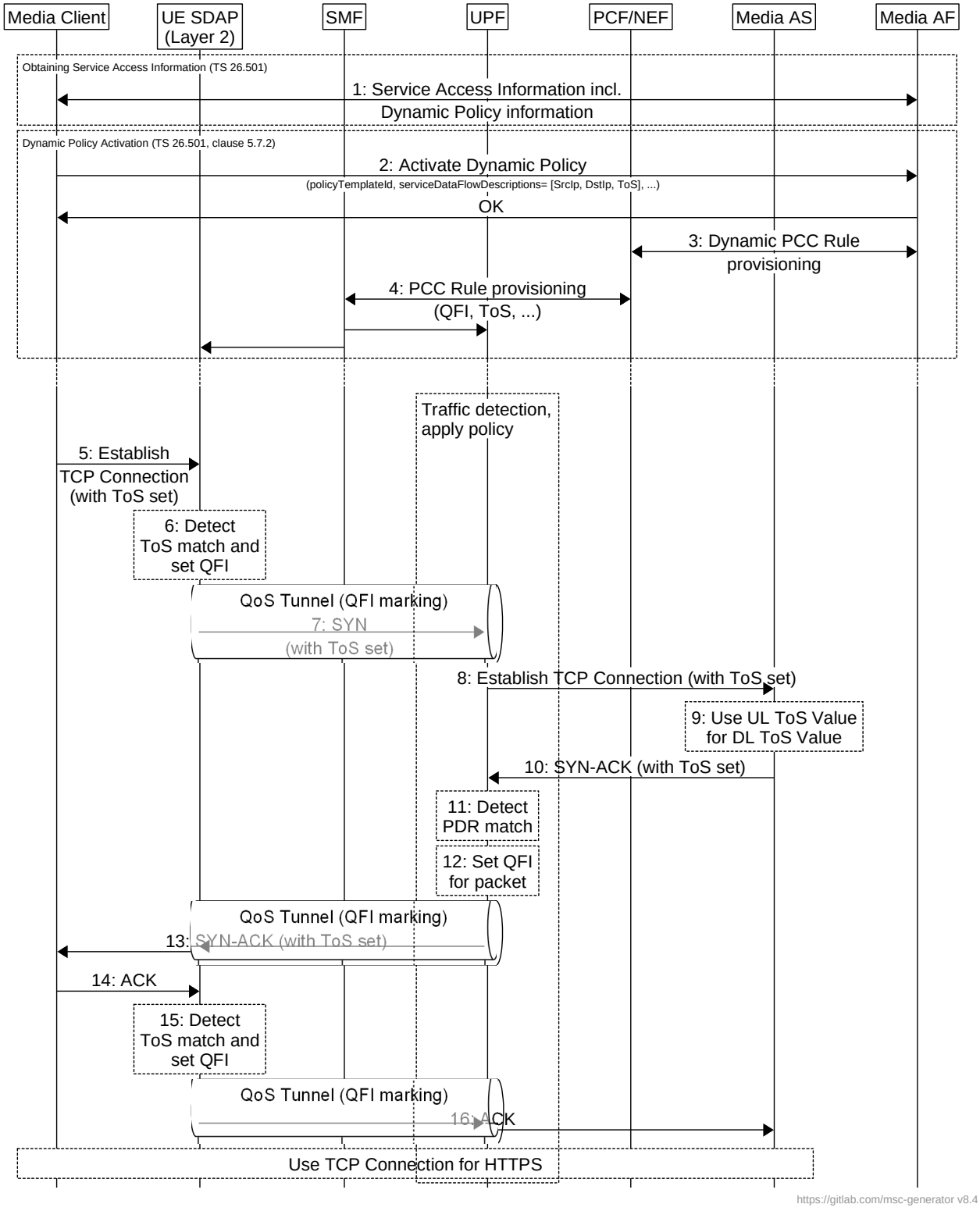

Figure C.3-1: High-level call flow for using TOS Traffic Class for traffic identification

Prerequisites:

- It is assumed that the Media Delivery System is already provisioned for Dynamic Policy usage as specified in clause 5.2.7 of the present document. As result, various functions of the 5G System are provisioned for QoS usage.

The steps are as follows:

1. The Media Delivery Client acquires Service Access Information (through reference points M8+M6 and/or M5 according to clause 5.3.2.3 of the present document), providing the information needed to use the Dynamic Policy API (see clause 5.3.3). Here, the *sdfMethod* indicates the usage of TOS.

2. The Media Delivery Client activates a Dynamic Policy (see clause 9.3). Each *applicationFlowBindings. applicationFlowDescription* contains a *packetFilter* object of data type *IpPacketFilterSet*, where the *srcIp*, *toSTc*, *dstIP* properties are present. The filter for a bi-directional application flow requires two such application flow bindings, one with *IpPacketFilterSet.direction* set to *in* and one with *IpPacketFilterSet.direction* set to *out*.

3. As a result of the previous step, the Media AF provisions the information for a Dynamic PCC rule with either the PCF or the NEF.

   - When using the NEF AF Session with required QoS API, the TOS value is provided within the *tosTC* attribute within the *FlowInfo* data type.

   - When using the PCF Policy Authorization Service API, the TOS value is provided within the *tosTrCl* attribution of the *MediaSubComponent* data type.

4. The PCF authorizes the request and creates a PCC rule. The PCF compiles and notifies the SMF about the PCC rule (containing the flow descriptions and providing parameters for policy control and/or charging control), has and the SMF forwards the QoS rule to the UE and (in the form of a PDR) to the UPF. The information contains, among other things, the QFI value and the TOS value.

During a media delivery session:

5. The Media Client initiates connection establishment by sending a TCP *SYN* packet. The TOS value in the TCP *SYN* packet is set by the Media Client to the same value as provided to the Media AF in earlier step 2.

6. The UE SDAP entity detects a matching TOS value in the uplink traffic.

7. The UE SDAP entity (Layer 2) encapsulates the IP packet into the according radio protocols, including the QFI marking.

8. The Media AS reads the TOS value from the uplink packet. The Media AS uses the uplink TOS value to mark all downlink packets in that TCP connection.

NOTE: When the 5G System employs an N6 NAT, the N6 NAT may set the downlink TOS value to the same value as the uplink TOS value.

9. The Media AS marks its acknowledgement IP packet (conveying the TCP *SYN–ACK*) with the same TOS value as the incoming packet.

10. The Media AS sends the TCP *SYN–ACK* packet back to the UE. The packet reaches the UPF on its path to the UE.

11. The UPF detects a match for the PDR rule configured in step 4 above containing the UE's IP address and TOS value.

12. The UPF encapsulates the downlink IP packet into an GTP-U packet, and sets the QFI value in the GTP-U packet header.

13. The UPF sends the GTP-U-encapsulated packet to the RAN via reference point N3 and the RAN marks the QFI value in the SDAP layer, sending the packet to the UE. The UE SDAP entity (Layer 2) forwards the TCP *SYN–ACK* to the 5GMS Client.

14. The Media Client sends the TCP *ACK* (again with the TOS field set in the IP header) to complete the TCP connection handshake.

15. The UE SDAP entity (Layer 2) detects a TOS match for the UE.

16. The UE SDAP entity (Layer 2) encapsulates the IP packet into the according radio protocols, including the QFI marking.

The Media Client continues to use the established TCP connection.

Annex D (informative): Change history

Change history							
Date	**Meeting**	**TDoc**	**CR**	**Rev**	**Cat**	**Subject/Comment**	**New version**
2023-08-24	SA4#125	S4-231500				Initial document skeleton proposal.	0.0.1
2023-10-24	ad hoc post SA4#125	S4aI230151				Added assigned TS number to cover page. Ported provisioning and session handling operations to clause 5.	0.1.0
2023-11-03	SA4#126	S4-231638				Resubmitted for discussion.	0.2.0
2023-11-14	SA4#126	S4-231918				Implemented pCRs: S4-231637: Consumption reporting clarifications (tracking changes to TS 26.512 Rel-17). S4-231834: Content Publishing provisioning operations.	0.3.0
2023-11-15	SA4#126	S4-231942				Editorial fixes to porting from TS 26.512.	0.3.1
2023-11-16	SA4#126	S4-232002				Clean version for presentation to WG Closing Plenary.	0.4.0
2024-01-08	ad hoc post SA4#126	S4aI230188				Comprehensive overhaul of HTTP error responses in clauses 5.2 and 5.3 to achieve uniformity across APIs. 5G Core interactions ported to clause 5.5. ANBR interactions ported to clause 5.6. Common data types ported to clause 7.3. Added missing *sdfMethod* property to *ServiceDataFlow Description* data type. Provisioning APIs ported to clause 8. Network media session handling APIs ported to clause 9. UE media session handling APIs ported to clause 10.	1.0.1
2024-01-18	ad hoc post SA4#126	S4aI230209				Clean version agreed as the basis of further work.	1.0.2
2024-02-02	SA4#127	S4-240103				S4-240099: API changes to support 3GPP Service URL. S4-240100: Media delivery session identification and life-cycle. S4-240101: Add Provisioning Sessions enumeration operation. S4-240102: *Maf_SessionHandling* endpoint rationalisation. S4-240373: Content Publishing Configuration. S4-240374: Relax name space restriction on metrics reporting scheme identifier URIs. S4-240380: Corrections on RAN-based metrics reporting. S4-240382: Supplementary media distribution over MBS. S4-240386: Traffic identification annex. S4-240414: QoE metrics reporting range. Alignment of clause 5.2.7.1 text with *M1QoSSpecification*. Alignment of packet latency and loss data types with TS 29.514.	1.1.0
2024-03-03	ad hoc post SA4#127	S4aI240018				Updates to *ContentPulshingConfiguration* data type.	1.1.1
2024-03-07	ad hoc post SA4#127	S4aI240025				Online typo corrections.	1.1.2
2024-03-22	ad hoc post SA4#127	S4aI240028				Added scope in clause 1. Added RTC applicability to tables 5.1-1 and 8.2.3.1-1. Clarified criteria for successful creation of Content Publishing Configuration resource in clauses 5.2.9.2 and 5.2.9.4. Prefixed enumerated *ProvisoningSessionType* values in clause 7.3.4.3 in preparation for addition of RTC. Uplink Media Entry Point in clauses 8.9.3.1 and 9.2.3.1 now allows bare paths as well as documents. Specified minimum values for durations in clauses 8.10.3.1 and 9.6.3.2. Refactored client APIs in clause 10 and added methods, information and notifications for Dynamic Policies and Network Assistance.	1.1.3
2024-03-28	ad hoc post SA4#127	S4aI240041				Changes in response to online review comments: Switched to using TS 26.571 common data type for expressing packet loss rate as integer tenths of a percent.	1.1.4

2024-04-12	SA4#127-bis-e	S4-240546				Provided term definitions for media delivery session and media delivery session identifier. S4-240766: Media Entry point protocol identifier. S4-240787: Corrected description of *eligibilityCriteria* in *EdgeResourcesConfiguration*.	1.2.0
2024-05-03	ad hoc post SA4#127-bis-e	S4aI240045				S4-240812: Added Background Data Transfer support to Dynamic Policies feature. S4aI240049: Added missing implicit data reporting parameters to *DynamicPolicy*, *NetworkAssistanceSession* and *ConsumptionReportingUnit* resources. Moved *locationReporting* from *ConsumptionReporting Configuration* to *ProvisioningSession* so that it controls location reporting for Dynamic Policies and Network Assistance Sessions as well as consumption reporting. Also made corresponding change in Service Access Information resource. Resolved Editor's Note in clause 5.3.2.1 concerning Media Entry Point for uplink media streaming and RTC. Editorial corrections to PCF interactions in clause 5.5.3 and 5.5.4. Reworked table NOTEs with normative requirements in clauses 7.3.3 and 8.7.3.1. Respecified metrics as fully-qualified URIs in clauses 8.10.3.1 and 9.2.3.1.	1.2.1
2024-05-07	ad hoc post SA4#127-bis-e	S4aI240062				Typo correction.	1.2.2
2024-05-10	SA4#128	S4-240882				-Inserted empty clauses 5.2.10, 8.10 and A.3.9 to accommodate RTC provisioning interactions and APIs. -Renumbered following clauses 5.2.x, 8.x and A.3.x. -Correction of M5 QoS property names in clause 5.3.4.4.	1.2.3.
2024-05-24	SA#128	S4-241306				Numbered references. S4-241196: UE media session handling client API (M6). S4-241270: OAuth 2.0 security. S4-241264: Service URL. S4-241147: MQTT notification channel. S4-241347: RTC additions to operations and APIs.	1.3.0
2024-06						Version 2.0.0 created by MCC	2.0.0
2024-06						Version 18.0.0 created by MCC	18.0.0
2024-06						Attachments added to the zip file	18.0.1
2024-06						Formatting fix	18.0.2

History

Document history		
V18.0.2	August 2024	Publication