ETSI TS 126 512 V16.1.0 (2021-01)

5G;
5G Media Streaming (5GMS);
Protocols
(3GPP TS 26.512 version 16.1.0 Release 16)

Reference
RTS/TSGS-0426512vg10

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 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is 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 IPR Policy, no investigation, 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.

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 http://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

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 the set of protocols and APIs for 5G Media Streaming (5GMS) services based on the 5G Media Streaming Architecture (5GMSA). 5GMS supports services including MNO and third-party Downlink Media Streaming Services, and MNO and third-party Uplink Media Streaming Services.

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 26.501: "5G Media Streaming (5GMS); General description and architecture".

[3] DASH Industry Forum, "Specification of Live Media Ingest", https://dashif-documents.azurewebsites.net/Ingest/master/DASH-IF-Ingest.pdf

[4] 3GPP TS 26.247: "Transparent end-to-end Packet-switched Streaming Service (PSS); Progressive Download and Dynamic Adaptive Streaming over HTTP (3GP-DASH)".

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

[6] IETF RFC 6234: "US Secure Hash Algorithms (SHA and SHA-based HMAC and HKDF)".

[7] 3GPP TS 23.003: "Numbering, addressing and identification".

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

[9] IETF RFC 7230: "Hypertext-Transfer Protocol (HTTP/1.1): Message Syntax and Routing".

[10] IETF RFC 4648: "The Base16, Base32, and Base64 Data Encodings".

[11] IEEE Standard 1003.1™, Issue 7: "The Open Group Base Specifications", 2018. https://pubs.opengroup.org/onlinepubs/9699919799/

[12] 3GPP TS 29.122: "T8 reference point for Northbound APIs".

[13] 3GPP TS 38.321: "NR; Medium Access Control (MAC) protocol specification".

[14] 3GPP TS 36.321: "Evolved Universal Terrestrial Radio Access (E-UTRA); Medium Access Control (MAC) protocol specification".

[15] 3GPP TS 27.007: "AT Command set for User Equipment (UE)".

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

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

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

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

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

[21] 3GPP TS 29.500: "5G System; Technical Realization of Service Based Architecture; Stage 3".

[22] 3GPP TS 29.501: "5G System; Principles and Guidelines for Services Definition; Stage 3".

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

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

[25] IETF RFC 7231: "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content".

[26] IETF RFC 7232: "Hypertext Transfer Protocol (HTTP/1.1): Conditional Requests".

[27] IETF RFC 7233: "Hypertext Transfer Protocol (HTTP/1.1): Range Requests".

[28] IETF RFC 7234: "Hypertext Transfer Protocol (HTTP/1.1): Caching".

[29] IETF RFC 7235: "Hypertext Transfer Protocol (HTTP/1.1): Authentication".

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

[31] IETF RFC 7540: "Hypertext Transfer Protocol Version 2 (HTTP/2)".

[32] ISO/IEC 23009-1: "Information technology; Dynamic adaptive streaming over HTTP (DASH) — Part 1: Media presentation description and segment formats".

[33] 3GPP TS 23.503: "Policy and charging control framework for the 5G System (5GS); Stage 2".

[34] 3GPP TS 29.514: "5G System; Policy Authorization Service; Stage 3".

[35] 3GPP TS 26.511: "5G Media Streaming (5GMS); Profiles, codecs and formats".

[36] Void.

[37] 3GPP TS 26.244: "Transparent end-to-end packet switched streaming service (PSS); 3GPP file format (3GP)".

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

3 Definitions of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the terms given in 3GPP TR 21.905 [1] and the following apply. A term defined in the present document takes precedence over the definition of the same term, if any, in 3GPP TR 21.905 [1].

3.2 Symbols

Void.

3.3 Abbreviations

For the purposes of the present document, the abbreviations given in 3GPP 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 3GPP TR 21.905 [1].

5GMS	5G Media Streaming
5GMSd	5GMS downlink
5GMSu	5GMS uplink
5GMSA	5GMS Architecture
ABR	Adaptive Bit Rate
AF	Application Function
ANBR	Access Network Bit rate Recommendation
AS	Application Server
CDN	Content Delivery Network / Content Distribution Network
CGI	Cell Global Identifier
CRUD	Create, Read, Update, Delete
CNAME	Canonical Name
CORS	Cross-Origin Resource Sharing
CRL	Certificate Revocation List
DASH	Dynamic Adaptive Streaming over HTTP
DER	Distinguished Encoding Rule
DNN	Domain Name News
DNS	Domain Name Server
ECGI	E-UTRAN Cell Global Identifier
ECMA	European Computer Manufacturers Association
FQDN	Fully Qualified Domain Name
HLS	HTTP Live Streaming
JSON	JavaScript Object Notation
LCID	Logical Channel IDentifier
MFBR	Maximum Flow Bit Rate
MIME	Multipurpose Internet Mail Extensions
MNO	Mobile Network Operator
MPD	Media Presentation Description
NCGI	NR Cell Global Identifier
NEF	Network Exposure Function
OAM	Operations, Administration and Maintenance
PCC	Policy Control and Charging
PCF	Policy Control Function
PEM	Privacy-Enhanced Mail
QoE	Quality of Experience
QoS	Quality of Service
SDF	Service Data Flow
SHA	Secure Hash Algorithm
TLS	Transport Layer Security
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
UTC	Coordinated Universal Time

4 Procedures for Downlink Streaming

4.1 General

This clause defines all procedures for Downlink Streaming using the different 5G Media Streaming Reference Points.

4.2 APIs relevant to Downlink Streaming

Table 4.2-1 summarises the APIs used to provision and use the various downlink streaming features specified in TS 26.501 [2].

Table 4.2-1: Summary of APIs relevant to downlink streaming features

5GMSd feature	Abstract	Relevant APIs		
		Interface	API name	Clause
Content Hosting	Content is ingested, hosted and distributed by the 5GMSd AS according to a Content Hosting Configuration associated with a Provisioning Session.	M1d	Provisioning Sessions API	7.2
			Server Certificates Provisioning API	7.3
			Content Preparation Templates Provisioning API	7.4
			Content Protocols Discovery API	7.5
			Content Hosting Provisioning API	7.6
		M2d	HTTP-pull based content ingest protocol	8.2
			DASH-IF push based content ingest protocol	8.3
		M4d	DASH [4] or 3GP [37]	10
		M5d	Service Access Information API	11.2
Metrics reporting	The 5GMSd Client uploads metrics reports to the 5GMSd AF according to a provisioned Metrics Reporting Configuration it obtains from the Service Access Information for its Provisioning Session.	M1d	Provisioning Sessions API	7.2
			Metrics Reporting Provisioning API	7.8
		M5d	Service Access Information API	11.2
			Metrics Reporting API	11.4
Consumption Reporting	The 5GMSd Client provides feedback reports on currently consumed content according to a provisioned Consumption Reporting Configuration it obtains from the Service Access Information for its Provisioning Session.	M1d	Provisioning Sessions API	7.2
			Consumption Reporting Provisioning API	7.7
		M5d	Service Access Information API	11.2
			Consumption Reporting API	11.3
Dynamic Policy invocation	The 5GMSd Client activates different traffic treatment policies selected from a set of Policy Templates configured in its Provisioning Session.	M1d	Provisioning Sessions API	7.2
			Policy Templates Provisioning API	7.9
		M5d	Service Access Information API	11.2
			Dynamic Policies API	11.5
Network Assistance	The 5GMSd client requests bitrate recommendations and delivery boosts from the 5GMSd AF.	M5d	Service Access Information API	11.2
			Network Assistance API	11.6

4.3 Procedures of the M1d (5GMS Provisioning) interface

4.3.1 General

A 5GMSd Application Provider may use the procedures in this clause to provision the network for downlink media streaming sessions that are operated by the 5GMSd Application Provider. These sessions may be DASH streaming sessions, progressive download sessions, or any other type of media streaming or distribution (e.g. HLS) sessions.

The M1d interface offers three different sets of procedures:

- Configuration of content ingest at M2d for onward distribution over M4d by the 5GMSd AS: designed as an API that is equivalent to the functionality of a public CDN. The resource types involved in content hosting

configuration are provisioning session (see clause 4.3.2), content hosting procedures (see clause 4.3.3), ingest protocols (see clause 4.3.4), content preparation template (see clause 4.3.5), and server certificates (see clause 4.3.6) .

- Configuration of dynamic policies: allows the configuration of Policy Templates at M5d that can be applied to M4d downlink sessions.
- Configuration of reporting: permits the MNO to collect at M5d QoE and consumption reports about M4d downlink sessions.

A 5GMSd Application Provider may use any of these procedures, in any combination, to support its downlink media streaming sessions.

4.3.2 Provisioning Session procedures

4.3.2.1 General

Prior to configuring content hosting, dynamic policies, or reporting, the 5GMSd Application Provider shall create a new Provisioning Session. The following CRUD operations are used to manage a provisioning session.

4.3.2.2 Create Provisioning Session

This procedure is used by the 5GMSd Application Provider to create a new Provisioning Session. The 5GMSd Application Provider shall use the HTTP `POST` method to create a new Provisioning Session. Upon successful creation, the 5GMSd AF shall respond with a *201 (Created)* response message that includes the resource identifier of the newly created Provisioning Session in the body of the reply and the URL of the resource, including its resource identifier, shall be returned as part of the HTTP `Location` header field.

4.3.2.3 Read Provisioning Session properties

This procedure is used by the 5GMSd Application Provider to obtain the properties of the Provisioning Session from the 5GMSd AF. The 5GMSd Application Provider uses the `GET` method for this purpose.

4.3.2.4 Update Provisioning Session properties

The Update operation is not allowed on Provisioning Sessions.

4.3.2.5 Delete Provisioning Session

This procedure is used by the 5GMSd Application Provider to delete a Provisioning Session. The 5GMSd AF will release any associated resources, purge any cached data, delete all QoS and reporting configurations associated with this Provisioning Session. The 5GMSd AF shall use the HTTP `DELETE` method for this purpose.

4.3.3 Content Hosting Configuration procedures

4.3.3.1 General

These procedures are used by the 5GMSd Application Provider and the 5GMSd AF on M1d to configure the content hosting feature for downlink streaming. They are further elaborated in clause 5.2.

4.3.3.2 Create Content Hosting Configuration

This procedure is used by the 5GMSd Application Provider to create a new Content Hosting Configuration. The 5GMSd Application Provider shall use the HTTP `POST` method for this purpose and the request message body shall include a *ContentHostingConfiguration* resource, as specified in clause 7.6.3.1.

If the Content Hosting Configuration uses the Push-based content ingest method, i.e. the *pull* attribute is set to False, then the *path* and *entryPoint* properties are read-only and shall not be set by the 5GMSd Application Provider. In this case, the *canonicalDomainName* property is also read-only and shall be assigned by the 5GMSd AF.

If the procedure is successful, the 5GMSd AF shall generate a resource identifier representing the new Content Hosting Configuration. In this case, the 5GMSd AF shall respond with a *201 (Created)* HTTP response message and shall provide the URL to the newly created resource in the `Location` header field. The response message body may include a *ContentHostingConfiguration* resource (see clause 7.6.3.1) that represents the current state of the Content Hosting Configuration, including any fields set by the 5GMSd AF.

If the procedure is not successful, the 5GMSd AF shall provide a response code as defined in clause 6.3.

4.3.3.3 Read Content Hosting Configuration properties

This procedure is used by the 5GMSd Application Provider to obtain the properties of an existing Content Hosting Configuration resource from the 5GMSd AF. The HTTP `GET` method shall be used for this purpose.

If the procedure is successful, the 5GMSd AF shall respond with a *200 (OK)* response message that includes the *ContentHostingConfiguration* resource in the response message body.

If the procedure is not successful, the 5GMSd AF shall provide a response code as defined in clause 6.3.

4.3.3.4 Update Content Hosting Configuration properties

The update operation is invoked by the 5GMSd Application Provider to modify the properties of an existing *ContentHostingConfiguration* resource. All writeable properties except *domainNameAlias* may be updated. The HTTP `PATCH` or HTTP `PUT` methods shall be used for the update operation.

If the procedure is successful, the 5GMSd AF shall respond with a *200 (OK)* and provide the content of the resource in the response, confirming the successful update operation.

If the procedure is not successful, the 5GMSd AF shall provide a response code as defined in clause 6.3.

4.3.3.5 Delete Content Hosting Configuration

This operation is used by the 5GMSd Application Provider to destroy a Content Hosting Configuration resource and to terminate the related distribution. The HTTP `DELETE` method shall be used for this purpose. As a result, the 5GMSd AF will release any associated network resources, purge any cached content, and delete any corresponding configurations.

If the procedure is successful, the 5GMSd AF shall respond with a *200 (OK)* response message.

If the procedure is not successful, the 5GMSd AF shall provide a response code as defined in clause 6.3.

4.3.4 Content Protocols procedures

4.3.4.1 General

The set of content ingest protocols supported by the 5GMSd AS at interface M2d is described by the *ContentProtocols* resource at M1d, as specified in clause 7.5.3.1.

4.3.4.2 Create Content Protocols

The Create operation is not permitted for the *ContentProtocols* resource.

4.3.4.3 Read Content Protocols

This procedure is used by the 5GMSd Application Provider to retrieve a list of content ingest protocols supported by the 5GMSd AS. The HTTP `GET` method shall be used for this purpose.

If the procedure is successful, the 5GMSd AF shall respond with a *200 (OK)* response that includes a *ContentProtocols* resource in the response message body, as specified in clause 7.5.3.1. If the procedure is not successful, the 5GMSd AF shall provide a response code as defined in clause 6.3.

4.3.4.4 Update Ingest Protocols

The Update operation is not permitted for the *ContentProtocols* resource.

4.3.4.5 Delete Ingest Protocols

The Delete operation is not permitted for the *ContentProtocols* resource.

4.3.5 Content Preparation Template procedures

4.3.5.1 General

The 5GMSd AS is able to process content ingested at interface M2d before serving it on interface M4d, as specified in clause 5.2.4.4. The content processing operations are specified in a Content Preparation Template resource, as specified in clause 5.2.2.3.

4.3.5.2 Create Content Preparation Template

This procedure is used by the 5GMSd Application Provider to register a new Content Preparation Template with a Provisioning Session. The 5GMSd Application Provider shall use the HTTP `POST` method to upload a new Content Preparation Template resource. The MIME content type of the Content Preparation Template shall be supplied in the `Content-Type` HTTP request header.

Upon successful creation, the 5GMSd AF shall respond with a *201 (Created)* response message and the URL of the newly created resource, including its resource identifier, shall be returned as part of the HTTP `Location` header field.

If the MIME content type indicated in `Content-Type` is not understood by the 5GMSd AF, the creation of the Content Preparation Template resource shall fail with HTTP error response status code *422 (Unprocessable entity)*.

If the 5GMSd AF is unable to provision the resources indicated in the supplied Content Preparation Template, the creation operation shall fail with an HTTP response status code of *503 (Service Unavailable)*.

If the procedure is not successful, the 5GMSd AF shall provide a response code as defined in clause 6.3.

4.3.5.3 Read Content Preparation Template

This procedure is used by the 5GMSd Application Provider to download a copy of a Content Preparation Template resource from the 5GMSd AF. The 5GMSd Application Provider shall use the `GET` method for this purpose.

If the procedure is successful, the 5GMSd AF shall respond with *200 (OK)* and shall provide 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 was created.

If the procedure is not successful, the 5GMSd AF shall provide a response code as defined in clause 6.3.

4.3.5.4 Update Content Preparation Template

The update procedure is used by the 5GMSd Application Provider to modify or replace an existing Content Preparation Template resource. The HTTP `PATCH` or HTTP `PUT` methods shall be used for the update operation.

If the procedure is successful, the 5GMSd AF shall respond with a *200 (OK)* and provide the content of the resource in the response, reflecting the successful update operation.

If the procedure is not successful, the 5GMSd AF shall provide a response code as defined in clause 6.3.

4.3.5.5 Delete Content Preparation Template

This operation is used by the 5GMSd Application Provider to destroy a Content Preparation Template resource. The HTTP `DELETE` method shall be used for this purpose.

If the procedure is successful, the 5GMSd AF shall respond with a *200 (OK)* response message.

If the procedure is not successful, the 5GMSd AF shall provide a response code as defined in clause 6.3. If the Content Preparation Template is in use as part of a Content Hosting Configuration, the procedure shall fail with HTTP error response status code *409 (Conflict)*.

4.3.6 Server Certificate procedures

4.3.6.1 General

Each X.509 server certificate [8] presented by the 5GMSd AS at interface M4d is represented by a Server Certificate resource at M1d. The Server Certificates Provisioning API as specified in clause 7.3 enables a Server Certificate resource to be created within the scope of a Provisioning Session, and subsequently referenced by a Content Hosting 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 5GMS System operator on behalf of the 5GMSd Application Provider; the other in which a certificate is generated by the 5GMSd Application Provider from a Certificate Signing Request solicited from the 5GMSd AF. Both methods shall be supported by implementations of the 5GMSd AF.

4.3.6.2 Create Server Certificate

This procedure is used by the 5GMSd Application Provider to request that the 5GMS System generates a new X.509 certificate on its behalf within the scope of a Provisioning Session. In this case, the certificate's Common Name (*CN*) is assigned in a domain under the control of the 5GMSd System operator.

The 5GMSd Application Provider shall use the HTTP `POST` method to create a new Server Certificate resource. Upon successful creation, the 5GMSd AF shall respond with a *201 (Created)* response message and the URL of the resource, including its resource identifier, shall be returned in the HTTP `Location` header. The response message body may optionally include a copy of the X.509 certificate corresponding to the newly created Server Certificate resource, as specified in clause 7.3.3.2.

NOTE: The X.509 certificate corresponding to the newly created Server Certificate resource may not be available immediately for interrogation and use. See clause 4.3.6.4 below for more details.

If the procedure is not successful, the 5GMSd AF shall provide a response code as defined in clause 6.3.

4.3.6.3 Reserve Server Certificate

This procedure is used by the 5GMSd Application Provider to solicit a Certificate Signing Request from the 5GMSd AF for the purpose of generating an X.509 certificate independently of the 5GMSd System. In this case, the certificate's Common Name (*CN*) is assigned in a domain under the control of the 5GMSd Application Provider itself, or that of a third party acting on its behalf. The 5GMSd Application Provider shall separately arrange for the FQDN carried in the Common Name of the certificate, or that of a Subject Alternative Name (*subjectAltName*) extension in the same certificate (see section 4.2.1.6 of RFC 5280 [20]), to resolve to the address of a 5GMSd AS in the target 5GMS System.

The 5GMSd Application Provider shall use the HTTP `POST` method to create a new Server Certificate. Upon successful creation of the resource, the 5GMSd AF shall respond with a *201 (Created)* response message and the URL of the resource, including its resource identifier, shall be returned in the HTTP `Location` header. The `Content-Type` response header and the body of the HTTP response message shall be as specified in clause 7.3.3.1.

If the procedure is not successful, the 5GMSd AF shall provide a response code as defined in clause 6.3.

4.3.6.4 Retrieve Server Certificate

This procedure is used by the 5GMSd Application Provider to download a Server Certificate resource for inspection. The HTTP `GET` method shall be used for this purpose. If the requested resource exists and is populated with an X.509 certificate, the 5GMSd AF shall respond with *200 (OK)* and shall return the requested Server Certificate in accordance with clause 7.3.3.2.

In the case where the X.509 certificate was provisioned by the 5GMSd System on behalf of the 5GMSd Application Provider according to clause 4.3.6.2 above, the HTTP response *503 (Service Unavailable)* 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 5GMSd Application Provider from a Certificate Signing Request obtained according to clause 4.3.6.3 above, the HTTP response *404 (Not Found)* shall be returned until such time as the X.509 certificate has been uploaded using the procedure specified in clause 4.3.6.5 below.

4.3.6.5 Upload Server Certificate

This procedure is used by a 5GMSd Application Provider to upload an X.509 certificate that it has generated in response to a Certificate Signing Request solicited using the reservation procedure specified in clause 4.3.6.3 above. The HTTP `PUT` method shall be used for this purpose. The `Content-Type` request header and the body of the HTTP request message shall be as specified in clause 7.3.3.2.

Before accepting the supplied X.509 certificate, the 5GMSd AF shall verify that the party originating the upload is the same party that reserved the Server Certificate resource using the procedure specified in clause 4.3.6.3 above. 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 that has not been reserved shall elicit a *404 (Not Found)* HTTP response.

4.3.6.6 Update Server Certificate

Updating a previously uploaded Server Certificate is not permitted for security reasons. Any attempt to do so using the `PUT` method shall result in the HTTP response *405 (Method Not Allowed)*.

To supply a replacement X.509 certificate, for example when a previously supplied certificate is shortly due to expire, the 5GMSd Application Provider should instead use one of the procedures specified in clause 4.3.6.2 or 4.3.6.3 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.

4.3.6.7 Destroy Server Certificate

This procedure is used to remove a Server Certificate from a Provisioning Session. The HTTP `DELETE` method shall be used for this purpose. On success, the HTTP response *200 (OK)* or *204 (No content)* shall be returned and afterwards the identifier of the Service Certificate resource is no longer valid.

Only the party that created (see clause 4.3.6.2) or reserved (see clause 4.3.6.3) 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 if an attempt is made to destroy a Server Certificate resource that is currently referenced by a Content Hosting Configuration.

Attempting to destroy a Server Certificate resource that has been reserved but never uploaded shall elicit a *200 (OK)* HTTP response. In this case, the 5GMSd AF should release any resources associated with the reservation.

If the procedure is not successful, the 5GMSd AF shall provide a response code as defined in clause 6.3.

4.3.7 Dynamic Policy Configuration procedures

4.3.7.1 General

These procedures are used by the 5GMS Application Provider to configure the Policy Templates for streaming sessions of a particular Provisioning Session.

Figure 4.3.7.1-1 below is a state diagram showing the life-cycle of a Policy Template.

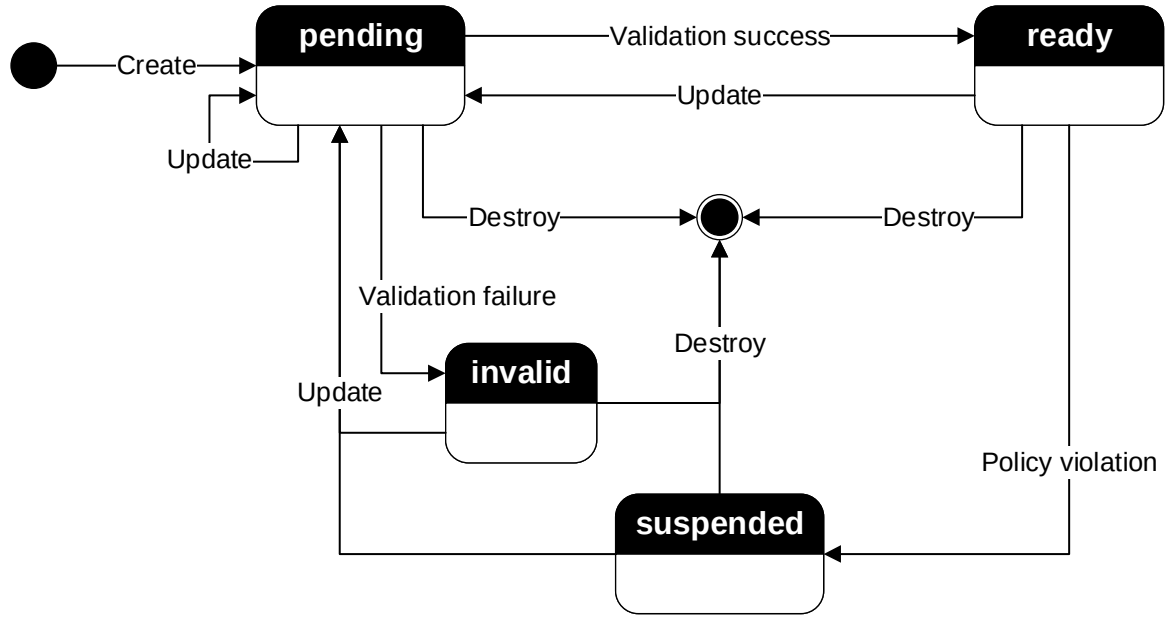

Figure 4.3.7.1-1: Policy Template State Diagram

Since Policy Templates require 5GMS System operator verification, a Policy Template that is newly created cannot be used immediately. Upon creation, a Policy Template shall be in the *pending* state. Once all mandatory properties are provided, the 5GMS AF triggers validation. If the Policy Template is not deemed to be valid by the operator of the 5GMS System, it shall move to the *invalid* state, from where it can be updated to remedy the defect. Once it has been successfully validated by the 5GMS System operator, a Policy Template shall take the *ready* state, indicating that it may be applied to streaming sessions. If it is subsequently updated by the 5GMS Application Provider, a Policy Template shall return to the *pending* state, awaiting revalidation by the operator of the 5GMS System. Finally, a Policy Template may be *suspended* by the 5GMS 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 into *pending* and triggers revalidation. A Policy Template may be destroyed when it is in any of the abovementioned states.

The 5GMSd/5GMSu AF shall verify the status of a Policy Template prior to allowing a Dynamic Policy Instance to instantiate it. Only Policy Templates in the *ready* state are eligible to be instantiated in this way.

4.3.7.2 Create Policy Template

This procedure is used by the 5GMS Application Provider to create a new Policy Template. The HTTP `POST` method shall be used for this purpose.

If the procedure is successful, the 5GMSd/5GMSu AF shall generate a resource identifier to uniquely identify the newly created Policy Template. In that case, it shall respond with a *201 (Created)* HTTP response message and provide the URL to the newly created resource in the `Location` header field.

If the procedure is not successful, the 5GMSd AF shall provide a response code as defined in clause 6.3.

The default state of a newly created Policy Template is *pending*. If all mandatory property values have been provided, the Policy Template is eligible for validation.

4.3.7.3 Read Policy Template

This procedure is used by the 5GMS Application Provider and other 5GMSd/5GMSu AFs to query the properties of an existing Policy Template resource from the 5GMSd/5GMSu AF. The HTTP `GET` method shall be used for this purpose.

If the procedure is successful, the 5GMSd/5GMSu AF shall respond with a *200 (OK)* response that includes the Policy Template in the response message body.

If the procedure is not successful, the 5GMSd AF shall provide a response code as defined in clause 6.3.

4.3.7.4 Update Policy Template

The update operation is invoked by the 5GMS Application Provider to modify the properties of an existing Policy Template. All available properties except *state* may be updated. The HTTP `PATCH` or HTTP `PUT` methods shall be used for the update operation.

Any update to the Policy Template resource will change its state back to *pending*, which makes it temporarily unusable. If all mandatory property values have been provided, the Policy Template is eligible for revalidation.

If the procedure is successful, the 5GMSd/5GMSu AF shall respond with a *200 (OK)* response message that includes the Policy Template in the response message body. Modifications to read-only properties, such as changes to the state of a Policy Template, shall be rejected with a *403 (Forbidden)* HTTP response.

If the procedure is not successful, the 5GMSd AF shall provide a response code as defined in clause 6.3.

4.3.7.5 Delete Policy Template

This operation is used by the 5GMS Application Provider to destroy a Policy Template resource. The HTTP `DELETE` method shall be used for this purpose. As a result, the 5GMSd/5GMSu AF will remove the Policy Template from any Provisioning Sessions that reference it.

Currently active streaming sessions using the destroyed Policy Template, if any exist, shall be stopped by the removal of the Policy Template.

If the procedure is successful, the 5GMSd/5GMSu AF shall respond with a *200 (OK)* response message.

If the procedure is not successful, the 5GMSd AF shall provide a response code as defined in clause 6.3.

4.3.8 Consumption Reporting Configuration procedures

4.3.8.1 General

These procedures are used by the 5GMSd Application Provider to activate and to configure consumption reporting. This clause defines the basic procedures. More details are provided in clause 7.7.

4.3.8.2 Create Consumption Reporting Configuration

This procedure is used by the 5GMSd Application Provider to activate consumption reporting for a particular Provisioning Session. The 5GMSd Application Provider shall use the HTTP `POST` method to activate the consumption reporting procedure and to transmit the Consumption Reporting Configuration to the 5GMSd AF. Upon successful operation, the 5GMSd AF shall respond with a *201 (Created)* response message and the same resource URL shall be returned in the `Location` header field.

If the procedure is not successful, the 5GMSd AF shall provide a response code as defined in clause 6.3.

4.3.8.3 Read Provisioning Session properties

This procedure is used by the 5GMSd Application Provider to obtain the current Consumption Reporting Configuration from the 5GMSd AF. The 5GMSd Application Provider uses the `GET` method for this purpose.

4.3.8.4 Update Provisioning Session properties

The update operation is invoked by the 5GMSd 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 the update operation.

If the procedure is successful, the 5GMSd AF shall respond with a *200 (OK)* reflecting the successful update operation.

If the procedure is not successful, the 5GMSd AF shall provide a response code as defined in clause 6.3.

4.3.8.5 Delete Provisioning Session

This operation is used by the 5GMSd Application Provider to terminate the related consumption reporting procedure. The HTTP `DELETE` method shall be used for this purpose. As a result, the 5GMSd AF will release any associated resources, purge any cached data, and delete any corresponding configurations.

If the procedure is successful, the 5GMSd AF shall respond with a *200 (OK)* response message. If the procedure is not successful, the 5GMSd AF shall provide a response code as defined in clause 6.3.

4.3.9 Metrics Reporting Configuration procedures

4.3.9.1 General

These procedures are used by the 5GMSd Application Provider to configure QoE metrics reporting functionality associated with downlink streaming. This clause defines the basic procedures. More details are provided in clause 7.8.

A given instance of a Metrics Reporting Configuration is identified by the *metricsReportingConfigurationId* property of the *MetricsReportingConfiguration* resource, The properties of that resource, as described in clause 7.3.8.1, pertain to metrics collection and reporting by the Media Session Handler to the 5GMS AF.

4.3.9.2 Create Metrics Reporting Configuration

This procedure is used by the 5GMSd Application Provider to create a Metrics Reporting Configuration for a particular Provisioning Session. The 5GMSd Application Provider shall use the HTTP `POST` method for this purpose and the request message body may include a *MetricsReportingConfiguration* resource, as specified in clause 7.8.3.1. Upon successful operation, the 5GMSd AF shall respond with a *201 (Created)* response message and the resource URL for the newly-created Metrics Reporting Configuration shall be returned in the `Location` header field. If the procedure is unsuccessful, the 5GMSd AF shall provide a response code as defined in clause 6.3.

This procedure may be perfomed multiple times to provision different Metrics Reporting Configurations in the scope of a particular Provisioning Session. Each such configuration is represented by a different value of *metricsReportingConfigurationId*.

4.3.9.3 Read Metrics Reporting Configuration

This procedure is used by the 5GMSd Application Provider to obtain the properties of an existing Metrics Reporting Configuration resource from the 5GMSd AF. The 5GMSd Application Provider shall use the `GET` method for this purpose. If successful, the 5GMSd AF shall respond with a *200 (OK)* and the requested *MetricsReportingConfiguration* resource (see clause 7.8.3.1) shall be returned in the body of the HTTP response message. If the procedure is unsuccessful, the 5GMSd AF shall provide a response code as defined in clause 6.3.

4.3.9.4 Update Metrics Reporting Configuration

The update operation is invoked by the 5GMSd Application Provider to initially upload the Metrics Reporting Configuration resource, or in the case of an existing Metrics Reporting Configuration resource, to entirely replace or modify certain properties of that resource. All available properties may be updated. The HTTP `PATCH` or HTTP `PUT` methods shall be used for the update operation.

If the procedure is successful, the 5GMSd AF shall respond with a *200 (OK)* reflecting the successful update operation. If the procedure is unsuccessful, the 5GMSd AF shall provide a response code as defined in clause 6.3.

4.3.9.5 Delete Metrics Reporting Configuration

This operation is used by the 5GMSd Application Provider to destroy a Metrics Reporting Configuration resource and to terminate the related metrics reporting procedure. The HTTP `DELETE` method shall be used for this purpose. As a result, the 5GMSd AF should release any associated resources, discard any pending metrics reports, and delete any corresponding configurations.

If the procedure is successful, the 5GMSd AF shall respond with a *200 (OK)* response message. If the procedure is unsuccessful, the 5GMSd AF shall provide a response code as defined in clause 6.3.

4.4 Procedures of the M2d (5GMS Ingest) interface

Editor's Note: This clause should contain content ingestion procedures between Network External Media Application Servers and the 5GMSd AS. This Clause may be removed, in case only external referenceable content ingest procedures are used.

4.5 Procedures of the M3d interface

Interface M3d is internal and no procedures on this interface are specified.

4.6 Procedures of the M4d (Media Streaming) interface

4.6.1 Procedures for DASH Session

This procedure is used by a 5GMSd Client to establish a DASH session via the M4d interface. In order to establish such a session, the 5GMSd AS shall host an MPD as defined in ISO/IEC 23009-1 [32] or TS 26.247 [4] and the MPD URL is known to the 5GMSd Client typically using M8d.

The Media Player receives an MPD URL from the 5GMSd-Aware Application through M7d by methods defined in clause 13. The Media Player shall send an HTTP `GET` message to the 5GMSd AS including the URL of the MPD resource. On success, the 5GMSd AS shall respond with a *200 (OK)* message that includes the requested MPD resource.

Additional procedures for reactions to different HTTP status codes are provided in TS 26.247 [4], clause A.7 and ISO/IEC 23009-1 [32] clause A.7.

Additional procedures for handling partial file responses are provided in TS 26.247 [4], clause A.9.

This information is provided through M7d to the application for selection. In addition, the currently used service description parameters are provided as status information through M7d in order for the Media Session Handler to make use of this information, for example for Dynamic Policy and Network Assistance.

The detailed handling of service description information is documented in clause 13.2 of the present document.

4.6.2 Procedures for Progressive Download Session

This procedure is used by a 5GMSd client to establish a Progressive Download session via the M4d interface. In order to establish such a session, the 5GMSd AS shall host an 3GP/MP4 file as defined in TS 26.247 [4]. The 3GP/MP4 URL is known to the Media Player (in this case a progressive download player), typically by using M8d.

The Media Player receives a URL from the 5GMSd-Aware Application through M7d by methods defined in clause 13. The Media Player shall send an HTTP `GET` message to the 5GMSd AS including the URL of the 3GP/MP4 resource. On success, the 5GMSd AS shall respond with a *200 (OK)* message that includes the requested 3GP/MP4 resource.

Additional procedures for reactions to different HTTP status codes are provided in TS 26.247 [4].

4.7 Procedures of the M5d (Media Session Handling) interface

4.7.1 Introduction

The M5d APIs are used by a Media Session Handler within a 5GMSd Client to invoke services at the 5GMSd AF.

4.7.2 Procedures for Service Access Information

4.7.2.1 General

Service Access Information is the set of parameters and addresses needed by the 5GMSd Client to activate reception of a downlink streaming session. Typically, through M8d the 5GMSd Client receives a media entry point (e.g. a URL to a DASH MPD or a URL to a progressive download file) that can be consumed by the Media Player and is handed to the Media Player through M7d. In addition, the media entry point URL may trigger the Media Session Handler to fetch the Service Access information from the 5GMSd AF for this streaming session.

This clause specifies the procedures where the 5GMSd Client fetches the Service Access Information from the 5GMSd AF.

4.7.2.2 Create Service Access Information

The Create operation is not allowed on Service Access Information.

4.7.2.3 Read Service Access Information properties

This procedure shall be used by the Media Session Handler to acquire Service Access Information from the 5GMSd AF. The Media Session Handler uses the `GET` method for this purpose.

The downlink streaming session for which the Media Session Handler is requesting data is identified by a unique reference contained in the path of the URL, as specified in clause 11.2.2.

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 containing either:

- an `If-None-Match` request header with the value of the entity tag (`ETag`) that was returned with the most recently acquired ServiceAccessInformation resource; or else
- an `If-Modified-Since` request header with the `Last-Modified` value of that most recently acquired resource.

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 procedure.

4.7.2.4 Update Service Access Information properties.

The Update operation is not allowed on Service Access Information.

4.7.2.5 Delete Service Access Information properties

The Delete operation is not allowed on Service Access Information.

4.7.3 Procedures for dynamic policy invocation

This procedure is used by a Media Session Handler to manage Dynamic Policy Instance resources via the M5d interface. A dynamic policy invocation consists of a Policy Template Id, flow description(s), a 5GMSd Application Service Configuration Id and potentially other parameters, according to TS 26.501 clause 5.7.

A Policy Template Id identifies the desired Policy Template to be applied to an application flow. A Policy Template includes properties such as specific QoS (e.g. background data) or different charging treatments. The 5GMSd AF combines the information from the Policy Template with dynamic information from the Media Session Handler to

gather a complete set of parameters to invoke the N33 or N5 API call. The Policy Template may contain for example the AF identifier.

The flow description allows the identification and classification of the media traffic, such as the packet filter sets given in clause 5.7.6 of [2].

In order to instantiate a new dynamic policy, the Media Session Handler shall first create a resource for the Dynamic Policy Instance on the 5GMSd AF. When the Media Session Handler needs several dynamic policies, it repeats the step as often as needed.

The Media Session Handler creates a new Dynamic Policy Instance by sending an HTTP `POST` message to the 5GMSd AF. The body of the HTTP `POST` message shall include a Provisioning Session Id, the Policy Template Id and the traffic descriptor. The traffic descriptor identifies the actual application flow(s) to be policed according to the Policy Template. If the operation is successful, the 5GMSd AF creates a new resource URL representing the Dynamic Policy Instance. In this case, the 5GMSd AF shall respond to the Media Session Handler with a *201 Created* HTTP response message, including the URL for the newly created Dynamic Policy Instance resource as the value of the `Location` header field.

Editor's Note: At minimum, the N5 and N33 API requires the UE IP Address at time of API invocation. The full Flow Description is an optional element, when more fine-grained traffic flow identification is required. It needs to be studied, how to enable usage of other traffic filtering parameters, such as an application id.

The Media Session Handler can modify the parameters of an existing Dynamic Policy Instance resource using either the HTTP `PUT` or `PATCH` methods, as appropriate to the desired update. The 5GMSd AF shall trigger the appropriate actions towards other Network Functions like PCF or NEF when all information is set.

Editor's Note: It is not clear what triggers the 5GMSd AF to start the PCF/NEF interactions.

The Media Session Handler can destroy a Dynamic Policy Instance resource using the HTTP `DELETE` method. As a result, the 5GMSd AF shall trigger the appropriate actions towards other Network Functions like PCF or NEF to remove the associated PCC rule.

Editor's Note: Notification subscription will be added in the next version of the pCR.

4.7.4 Procedures for consumption reporting

These procedures are used by the Media Session Handler and the Consumption Reporting functions of the 5GMSd Client to submit a consumption report via the M5d interface if Consumption Reporting is applied for a downlink streaming session.

The Service Access Information indicating whether Consumption Reporting is provisioned for downlink streaming sessions is described in clause 11.2.3. When the *ClientConsumptionReportingConfiguration.samplePercentage* value is 100, the Media Session Handler shall activate the consumption reporting procedure. If the *samplePercentage* is less than 100, 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 report procedure when the generated random number is of a lower value than the *samplePercentage* value.

Editors'note: -Missing text that will describe M6d/M7d APIs

If the consumption reporting procedure is activated, the Media Session Handler shall submit a consumption report to the 5GMSd AF when any of the following conditions occur:

- Start of consumption of a downlink streaming session;
- Stop of consumption of a downlink streaming session;
- Upon determining the need to report ongoing 5GMS consumption at periodic intervals determined by the *ClientConsumptionReportingConfiguration.reportingInterval* property.
- Upon determining a location change, if the *ClientConsumptionReportingConfiguration.locationReporting* property is set to *True*.

Whenever a consumption report is sent, the Media Session Handler shall reset its reporting interval timer to the value of the *reportingInterval* property and it shall begin countdown of the timer again. Whenever the Media Session Handler stops the consumption of a downlink streaming session, it shall disable its reporting interval timer.

In order to submit a consumption report, the Media Session Handler shall send an HTTP `POST` message to the 5GMSd AF. If several 5GMSd AF addresses are listed in the *ClientConsumptionReportingConfiguration.serverAddresses* array (see table 11.2.3.1-1), the Media Session Handler shall choose one and send the message to the selected. The request body shall be a *ConsumptionReport* structure, as specified in clause 11.3.3.1.The server shall respond with a *200 (OK)* message to acknowledge successful processing of the consumption report.

The Consumption Reporting API, defining the data formats and structures and related procedures for consumption reporting, is described in clause 11.3.

4.7.5 Procedures for metrics reporting

The M5d procedures for QoE metrics reporting pertain to the combination of the provisioning of metrics collection and reporting in the Media Session Handler using relevant Service Access Information, and the sending of collected metrics by the Media Session Handler to the 5GMSd AF in accordance with the configured metrics scheme(s). A metrics scheme may be 3GPP-defined or non-3GPP-defined.

When the metrics collection and reporting feature is activated for a downlink streaming session, one or more metrics configuration sets, each associated with a metrics scheme, may be provided to the 5GMSd client. A given metrics configuration set contains information such as the 5GMSd AF address(es) to which metrics are to be sent by the Media Session Handler, metrics reporting interval, target percentage of streaming sessions for which reports should be sent, and the set of metrics to be collected and reported. See TS 26.501 [2] for additional details.

For progressive download and DASH streaming services, the listed metrics in a given metrics configuration set are associated with the 3GPP metrics scheme and shall correspond to one or more of the metrics as specified in clauses 10.3 and 10.4, respectively, of TS 26.247 [4].

Details of the metrics reporting API are provided in clause 11.4, and for 3GP-DASH based downlink streaming services, the 3GPP-defined metrics reporting scheme and metrics report format are defined in clause 11.4.3.

4.7.6 Procedures for network assistance

This procedure is used by the 5GMSd Client to request Network Assistance from the 5GMSd AF.

The 5GMSd Client first creates a Network Assistance Session. It provides information that will be used by the Network Assistance function to request QoS from the PCF and to recommend a bit rate to the 5GMSd Client.

The 5GMSd client may also request a delivery boost to be provided.

After the Network Assistance Session resource is provisioned, the 5GMSd Client uses the Network Assistance Session identifier when requesting a bit rate recommendation.

In order to terminate a Network Assistance Session, the 5GMSd Client deletes the Network Assistance session resource.

4.8 Procedures of the M6d (UE Media Session Handling) interface

4.8.1 General

This clause contains the procedures for the interaction between the 5GMSd-Aware application or the Media Player and the Media Session Handler through the M6d API. Details are provided in clause 12.

4.8.2 Consumption reporting procedures

Before a streaming session is started, the Media Session Handler shall check if the Service Access Information contains any Consumption reporting configuration, as specified in clause 4.7.3. If such a configuration is present, the Media Session Handler shall initiate consumption reporting based on this configuration for the current streaming session.

The Media Session Handler shall first determine whether consumption reporting is active for the session. The determination shall be based on the *samplePercentage* attribute specified in the consumption reporting configuration. When the *samplePercentage* is not present or its value is 100, consumption reporting is active for the session. If the *samplePercentage* is less than 100, the Media Session Handler generates a random number which is uniformly distributed in the range 0 to100; consumption reporting is active for the session when the generated random number is of a lower value than the *samplePercentage* value.

If consumption reporting for this session is active, the Media Session Handler shall regularly determine the consumption reporting parameters defined in clause 11.3.2.4 from the Media Player through the M7d interface and shall report these values according to the *reportingInterval* specified in the Client Consumption Reporting Configuration.

4.9 Procedures of the M7d (UE Media Player) interface

4.9.1 General

This clause contains the procedures for the interaction between the 5GMSd-Aware Application or the Media Session Handler and the Media Player through the M7d API. Details are provided in clause 13.

4.9.2 Metrics reporting procedures

These procedures shall be used by the Media Session Handler function to control metrics reporting when such reporting is configured via metadata sent in-band via the media manifest.

When a streaming session is started, the Media Session Handler shall check if the manifest contains any metrics configuration, as specified in TS 26.247 clauses 10.4 and 10.5. If such a configuration is found, the Media Session Handler shall use it for the current streaming session.

The Media Session Handler shall first determine whether metrics from this session shall be reported. The determination shall be based on the *sample percentage* attribute specified in the metrics configuration, according to TS 26.247 clause 10.5.

If metrics are reported for the session, the Media Session Handler shall request the Media Player to create a metrics collection job. The Media Player 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 collector job with the set of metrics which shall be collected during the session. The format of the configuration shall be according to TS 26.247 Annex L.2, but note that 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 Player according to the *reporting interval* specified in the metrics configuration. The metrics returned by the Media Player shall use the format as described in TS 26.247 clause 10.6, and the Media Session Handler shall forward these to the *server address* using the specified *DNN* according to the procedures described in TS 26.247 clause 10.6.

When the session is finished the Media Session Handler shall delete the metrics collection job.

4.10 Procedures of the M8d interface

This clause defines basic procedures for M8d.

No specific procedures are defined but it is expected that the 5GMSd Application Provider can provide media session entry points to a 5GMSd-Aware Application through M8d. The 5GMSd-Aware Application would then initiate the media session by providing such an entry point to the 5GMSd Client through M7d.

5 Procedures for Uplink Streaming

5.1 General

Uplink streaming functional entities in the 5GMS System include the 5GMSu Application Provider, 5GMSu AF, 5GMSu AS and the UE. To make use of these other entities, the UE includes a 5GMSu-Aware Application that is provided by the 5GMS Application Provider and a 5GMSu Client comprising the Media Session Handler and the Media Streamer.

The M1u Provisioning API enables the 5GMSu Application Provider to establish and manage the uplink media session handling and streaming options of the 5GMSu system.

The M2u Egest interface enables uplink streaming content sent by the 5GMSu Client to the 5GMSu AS over interface M4u to be subsequently delivered to the 5GMSu Application Provider. Uplink streaming media transfer from the 5GMSu AS to the 5GMSu Application Provider may be either pull-based and initiated by the 5GMSu Application Provider using the HTTP `GET` method, or push-based and initiated by the 5GMSu AS using the HTTP `PUT` method. The resource identifier of the 5GMSu Application Provider for push-based streaming content delivery is provided to the 5GMSu AS by the 5GMSu AF over the M3u interface, as part of the M1u Provisioning Session.

The 5GMSu AF, having acquired M1u Provisioning information, sets up the M5u interface that the 5GMSu Client can use for uplink streaming session management, remote control, metrics reporting, network assistance and request for policy and/or charging treatment. Certain types of configuration and policy information accessed over M5u by the Media Session Handler, such as uplink metrics reporting, QoS policy, or support for AF-based network assistance are further passed to the Media Streamer via the M7u API.

Based on the configuration information received on M5u and a request from the Media Streamer received over the M6u interface, the Media Session Handler sets up an uplink streaming session with the 5GMSu AF. Upon successful session establishment, the Media Session Handler triggers the Media Streamer to begin uplink streaming of media content to the 5GMSu AS over the M4u interface.

Subscription to status and other event notification services are offered by the Media Session Handler to the 5GMSu-Aware Application and to the Media Streamer via the M6u APIs exposed by the Media Session Handler.

Subscription to status and other event notification services are also offered by the Media Streamer to the 5GMSu-Aware Application and to the Media Session Handler via the M7u APIs exposed by the Media Player.

5.2 APIs relevant to Uplink Streaming

Table 5.2-1 summarises the APIs used to provision and use the various uplink streaming features specified in TS 26.501 [2].

Table 5.2-1: Summary of APIs relevant to uplink streaming features

5GMSu feature	Abstract	Relevant APIs		
		Interface	API name	Clause
Dynamic Policy invocation	The 5GMSd Client activates different traffic treatment policies selected from a set of Policy Templates configured in its Provisioning Session.	M1u	Provisioning Sessions API	7.2
			Policy Templates Provisioning API	7.9
		M5u	Service Access Information API	11.2
			Dynamic Policies API	11.5

6 General aspects of APIs for 5G Media Streaming

6.1 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 [22], i.e.:

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

with the following components:

- *{apiRoot}* shall be set as described in TS 29.501 [22].
- *{apiName}* shall be set as defined by the following clauses.
- *{apiVersion}* shall be set to "v1".
- *{apiSpecificResourceUriPart}* shall be set as described in the following clauses.

6.2 Usage of HTTP

6.2.1 HTTP protocol version

6.2.1.1 5GMS AF

Implementations of the 5GMS AF shall expose both HTTP/1.1 [24] and HTTP/2 [31] endpoints at interfaces M1 and M5, including support for the HTTP/2 starting mechanisms specified in section 3 of RFC 7540 [31]. In both protocol versions, TLS [29] shall be supported and HTTPS interactions should be used on these interfaces in preference to cleartext HTTP.

The 5GMS Application Provider may use any supported HTTP protocol version at interface M1.

The Media Session Handler may use any supported HTTP protocol version at interface M5.

All responses from the 5GMS AF that carry a message body shall include 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 shall support the conditional HTTP requests `If-none-Match` and `If-Modified-Since`.

6.2.1.2 5GMS AS

Implementations of the 5GMS AS shall expose HTTP/1.1 [24] endpoints at interfaces M2 and M4 and may additionally expose HTTP/2 [31] endpoints at these interfaces. In both protocol versions, TLS [30] shall be supported and HTTPS interactions should be used on these interfaces in preference to cleartext HTTP.

For pull-based content ingest, the 5GMS Application Provider shall expose an HTTP/1.1-based origin endpoint to the 5GMSd AS at interface M2 and may additionally expose an HTTP/2-based origin endpoint.

For push-based content ingest, the 5GMS Application Provider may use any supported HTTP protocol version at interface M2.

The Media Stream Handler may use any supported HTTP protocol version at interface M4.

6.2.2 HTTP message bodies for API resources

The OpenAPI [23] specification of HTTP messages and their content bodies is contained in Annex C.

6.2.3 Usage of HTTP headers

6.2.3.1 General

Standard HTTP headers shall be used in accordance with clause 5.2.2 of TS 29.500 [21] for both HTTP/1.1 and HTTP/2 messages.

6.2.3.2 User Agent identification

6.2.3.2.1 Media Stream Handler identification

The Media Stream Handler in the 5GMSd Client shall identify itself to the 5GMS AS at interface M4 using a User-Agent request header (see section 5.3.3 of RFC 7231 [25]) that should include the *product* token *5GMSdMediaPlayer* optionally suffixed with a *product-version*.

The Media Stream Handler may additionally supply a *comment* element in the `User-Agent` request header containing a vendor-specific identification string.

6.2.3.2.2 Media Session Handler identification

The Media Session Handler in the 5GMS Client shall identify itself to the 5GMSd AF at interface M5d using a User-Agent request header (see section 5.3.3 of RFC 7231 [25]) in which the first element shall be a *product* identified by the token *5GMSdMediaSessionHandler* (or *5GMSuMediaSessionHandler*) and optionally suffixed with a *product-version*.

The Media Session Handler may additionally supply a *comment* element in the `User-Agent` request header containing a vendor-specific identification string.

6.2.3.3 Server identification

6.2.3.3.1 5GMSd AF identification

The 5GMSd AF shall identify itself using a `Server` response header (see section 7.4.2 of RFC 7231 [25]) of the following form:

5GMSdAF-{FQDN}/{implementationSpecificSuffix}

where *{FQDN}* shall be the Fully-Qualified Domain Name of the 5GMSd AF exposed to the requesting client, and *{implementationSpecificSuffix}* shall be determined by the implementation.

6.2.3.4 Support for conditional HTTP GET requests

All responses from the 5GMS AF that carry a resource message body shall include:

- a strong entity tag for the resource, conveyed in an `ETag` response header,
- a resource modification timestamp, conveyed in a `Last-Modified` response header, and
- a predicted time-to-live period for the resource, conveyed in a `Cache-Control: max-age` response header.

All API endpoints on the 5GMS AF that expose the HTTP `GET` method shall support conditional requests using the `If-None-Match` and `If-Modified-Since` request headers. 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.

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

All API endpoints on the 5GMS AF that expose the HTTP `POST`, `PUT`, `PATCH` or `DELETE` methods shall support conditional requests using the `If-Match` request header. The API client should supply a strong entity tag in an `ETag` request header when invoking any of these HTTP methods.

6.3 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 [22]. API specific error responses are specified in the respective technical specifications.

6.4 Common API data types

6.4.1 General

The data types defined in this clause are intended to be used by more than one of the 5GMS APIs.

6.4.2 Simple data types

Table 6.4.2-1 below specifies common simple data types used within the 5GMS APIs, including a short description of each. In cases where types from other specifications are reused, a reference is provided.

Table 6.4.2-1: Simple data types

Type name	Type definition	Description	Reference
Percentage	`number`	A percentage expressed as a floating point value between 0.0 and 100.0 (inclusive).	
DurationSec	`integer`	An unsigned integer identifying a period of time expressed in units of seconds.	TS 29.122 [12] table 5.2.1.3.2-2
DateTime	`string`	An absolute date and time expressed using the OpenAPI *date-time* string format.	TS 29.122 [12] table 5.2.1.3.2-2

6.4.3 Structured data types

6.4.3.1 IpPacketFilterSet type

Table 6.4.3.1-1: Definition of type IpPacketFilterSet

Property name	Data type	Cardinality	Usage	Description
srcIp	`String`	0..1		Source IP address or IPv6 prefix.
dstIp	`String`	0..1		Destination IP address or IPv6 prefix.
protocol	`Integer`	0..1		Protocol.
srcPort	`Integer`	0..1		Source port.
dstPort	`Integer`	0..1		Destination Port.
toSTc	`String`	0..1		Type of Service (TOS) (IPv4) / Traffic class (IPv6) and Mask.
flowLabel	`Integer`	0..1		Flow Label (IPv6).
spi	`Integer`	0..1		Security Parameter Index.
direction	`String`	1..1		Packet Filter Set Direction.

6.4.3.2 ServiceDataFlowDescription type

Table 6.4.3.2-1: Definition of type ServiceDataFlowDescription

Property name	Data type	Cardinality	Usage	Description
flowDescription	`IpPacketFilterSet`	0..1		Service Data Flow Description.
domainName	`String`	0..1		FQDN of the 5GMS AS.

An object of type ServiceDataFlowDescription shall contain at least one property.

6.4.3.3 M5QoSSpecification type

Table 6.4.3.2-1: Definition of type ServiceDataFlowDescription

Property name	Data type	Cardinality	Usage	Description
marBwDlBitRate	`BitRate`	1..1		Maximum requested bit rate for the Downlink.
marBwUlBitRate	`BitRate`	1..1		Maximum requested bit rate for the Uplink.
minDesBwDlBitRate	`BitRate`	0..1		Minimum desired bit rate for the Downlink.
minDesBwUlBitRate	`BitRate`	0..1		Minimum desired bit rate for the Uplink.
mirBwDlBitRate	`BitRate`	1..1		Minimum requested bit rate for the Downlink.
mirBwUlBitRate	`BitRate`	1..1		Minimum requested bandwidth for the Uplink.
desLatency	`Integer`	0..1		Desire Latency.
desLoss	`Integer`	0..1		Desired Loss Rate.

6.4.3.4 M1QoSSpecification type

Table 6.4.3.2-1: Definition of type ServiceDataFlowDescription

Property name	Data type	Cardinality	Usage	Description
qosReference	`String`	0..1		As defined in clause 5.6.2.7 of TS 29.514 [34].
maxBtrUl	`BitRate`	0..1	RO	Maximum Bitrate Uplink.
maxBtrDl	`BitRate`	0..1	RO	Maximum Bitrate Downlink.
maxAuthBtrUl	`BitRate`	0..1	RW	Maximum Authorized Bitrate Uplink by 5GMS Application Provider.
maxAuthBtrDl	`BitRate`	0..1	RW	Maximum Authorized Bitrate Downlink by 5GMS Application Provider.
defPacketLossRateDl	`Integer`	0..1		Default packet loss rate for Downlink.
defPacketLossRateUl	`Integer`	0..1		Default packet loss rate for Uplink.

6.4.3.5 ChargingSpecification type

Table 6.5.3.2-1: Definition of type ChargingSpecification

Property name	Data type	Cardinality	Usage	Description
sponId	`SponId`	0..1		As defined in clause 5.6.2.3 of TS 29.514 [34].
sponStatus	`SponsoringStatus`	0..1		
gpsi	`Array(Gpsi)`	0..1		List of UEs permitted to instantiate this Policy Template.

6.4.3.6 TypedLocation type

Table 6.4.3.6-1: 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.

6.4.3.7 Operation Success Response type

The data model for the *OperationSuccessResponse* type is specified in table 6.4.3.7-1 below:

Table 6.4.3.7-1: 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.

6.4.4 Enumerated data types

6.4.4.1 CellIdentifierType enumeration

Indicates the type of a cell identifier, as defined in TS 23.003 [7].

Table 6.4.4.1-1: Definition of CellIdentifierType enumeration

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

6.4.4.2 SdfMethod enumeration

The data model for the *SdfMethod* enumeration is specified in table 6.4.4.2-1 below:

Table 6.4.4.2-1: Definition of SdfMethod enumeration

Enumeration value	Description
5Tuple	The Media Session Handler shall use 5-Tuples for Service Data Flow descriptions. The 5-Tuple shall not contain a wildcard.
2Tuple	The Media Session Handler shall use a 2-Tuple of UE IP and Server IP as Service Data Flow Description.
typeOfServiceMarking	The Media Session Handler shall apply Type of Service (ToS) marking to the Service Data Flow.
flowLabel	The Media Session Handler shall apply IPv6 flow label marking and provide the IPv6 flow label of the Service Data Flow.
domainName	The Media Session Handler shall provide the domain name of the 5GMSd AS.

6.5 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 [Z] 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 notation [38].
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 propery is optional in the data structure. An array with cardinality 1 indicates that the propery 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” (Read) and “U” (Update) refers to the CRUD procedure during which the property is present in the resource type. (The Delete operation never takes any input data type.)

   - “RO” signifies a read-only property. Only the API provider function is permitted to modify the property value. The API invoker can only read the value.

   - “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 Provisioning (M1) APIs

7.1 General

This clause defines the provisioning API used by a 5GMS Application Provider to configure 5G Media Streaming services.

7.2 Provisioning Sessions API

7.2.1 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 4.3.2. Having created a Provisioning Session, the 5GMS Application Provider can then go on to provision other 5GMS features in the context of that Provisioning Session, using the APIs specified in clause 7.3 *et seq*.

7.2.2 Resource structure

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

`{apiRoot}/3gpp-m1d/v1/provisioning-sessions/`

Table 7.4.2-1 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 7.2.2-1: Operations supported by the Provisioning Sessions API

Operation	Sub-resource path	Allowed HTTP method(s)	Description
Create Provisioning Session		`POST`	Used to create a new Provisioning Session resource. If the operation succeeds, the URL of the created Provisioning Session resource shall be returned in the `Location` header of the response.
Retrieve Provisioning Session	*{provisioningSessionId}*	`GET`	Used to retrieve a Provisioning Session resource for inspection.
Destroy Provisioning Session		`DELETE`	Used to destroy an existing Provisioning Session resource.

7.2.3 Data model

7.2.3.1 ProvisioningSession resource

The data model for the *ProvisioningSession* resource is specified in table 7.2.3.1-1 below:

Table 7.2.3.1-1: Definition of ProvisioningSession resource

Property name	Type	Cardinality	Usage	Description
provisioningSessionId	`String`	1..1	C: R R: RO	A unique identifier for this Provisioning Session.
aspId	`AspId`	0..1	C: W R: RO	The identity of the Application Service Provider responsible for this Provisioning Session, as specified in clause 5.6.2.3 of TS 29.514 [34].
serverCertificateIds	`Array(String)`	0..1	C: – R: RO	A (possibly empty) array of Server Certificate identifiers currently associated with this Provisioning Session.
contentPreparation TemplateIds	`Array(String)`	0..1	C: – R: RO	A (possibly empty) array of Content Preparation Template identifiers currently associated with this Provisioning Session.
contentHosting ConfigurationId	`String`	0..1	C: – R: RO	The Content Hosting Configuration identifier currently associated with this Provisioning Session, if any.
consumptionReporting ConfigurationId	`String`	0..1	C: – R: RO	The Consumption Reporting Configuration identifier currently associated with this Provisioning Session, if any.
metricsReporting ConfigurationIds	`Array(String)`	0..1	C: – R: RO	A (possibly empty) array of Metrics Reporting Configuration identifiers currently associated with this Provisioning Session.
policyTemplateIds	`Array(String)`	0..1	C: – R: RO	A (possibly empty) array of Policy Template identifiers currently associated with this Provisioning Session.

7.3 Server Certificates Provisioning API

7.3.1 Overview

The Server Certificates Provisioning API is used to provision X.509 [8] server certificates that can be referenced by a Content Hosting Configuration and subsequently presented by the 5GMSd AS when it distributes content to 5GMSd Clients at interface M4d using Transport Layer Security [12]. Server Certificate resources are provisioned within the scope of an enclosing Provisioning Session.

7.3.2 Resource structure

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

{apiRoot}/3gpp-m1d/v1/provisioning-sessions/*{provisioningSessionId}*/

Table 7.3.2-1 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 7.3.2-1: Operations supported by the Server Certificates Provisioning API

Operation	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 5GMS System creates a new Server Certificate on behalf of the 5GMSd Application Provider. The request message body shall be empty. If the operation succeeds, the URL of the created Server Certificate resource shall be returned in the Location header of the response and this shall comply with the sub-resource path specified below for manipulating Server Certificate resources in the collection. The body of the response message may include a copy of the created X.509 certificate, as specified in clause 7.3.3.2 below.
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. The request message body shall be empty. If the operation succeeds, the URL of the reserved Server Certificate resource shall be returned in the Location header of the response and this shall comply with the sub-resource path specified below for manipulating Server Certificate resources in the collection. The body of the response shall be a PEM-encoded X.509 Certificate Signing Request, as specified in clause 7.3.3.1 below.
Retrieve Server Certificate	certificates/*{certificateId}*	GET	Used to retrieve a previously created or uploaded Server Certificate. If a Server Certificate resource has been reserved but not yet uploaded, this operation shall return *404 (Not Found)*.
Upload Server Certificate		PUT	Used by the 5GMSd Application Provider to supply a new Server Certificate in response to a solicited Certificate Signing Request. The body of the request message shall be a PEM-encoded X.509 certificate signed with the public key of the Certificate Signing Request, as specified in clause 7.3.3 below. The 5GMSd AF shall associate the Server Certificate with the private key it generated alongside the Certificate Signing Request. Attempting to update a previously uploaded Server Certificate is an error.
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}* differs from the serial number of the X.509 certificate.			

7.3.3 Data model

7.3.3.1 Certificate Signing Request

The Certificate Signing Request shall comply with the Privacy-Enhanced Mail (PEM) textual format specified in RFC 7468 [17], 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*.

7.3.3.2 Server Certificate resource

The Server Certificate resource shall comply with the Privacy-Enhanced Mail (PEM) textual format specified in RFC 7468 [17], i.e. a Base64-encoded DER certificate request or certificate, including leading and trailing encapsulation boundary lines. The resource shall include only the public parts of the X.509 certificate. In particular, the private key shall not be included.

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

7.3.4 Operations

Under no circumstances shall the 5GMSd AF reveal the private key associated with the Certificate Signing Request to the 5GMSd Application Provider.

7.4 Content Preparation Templates Provisioning API

7.4.1 Overview

Content Preparation Templates are used to specify manipulations applied by a 5GMSd AS to media resources ingested at interface M2d for distribution at interface M4d. The Content Preparation Templates 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.

7.4.2 Resource structure

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

{apiRoot}`/3gpp-m1d/v1/provisioning-sessions/`*{provisioningSessionId}*`/`

Table 7.4.2-1 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 7.4.2-1: Operations supported by the Content Preparation Templates Provisioning API

Operation	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. If the operation succeeds, the URL of the newly created Content Preparation Template resource shall be returned in the `Location` header of the response and this shall comply with the sub-resource path specified below for manipulating Content Preparation Templates.
Retrieve Content Preparation Template	`content-preparation-templates/` *{contentPreparationTemplateId}*	`GET`	Used to retrieve a Content Preparation Template resource.
Update Content Preparation Template		`PUT`, `PATCH`	Used to modify an existing Content Preparation Template resource.
Destroy Content Preparation Template		`DELETE`	Used to destroy an existing Content Preparation Template resource.

7.4.3 Data model

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

7.4.4 Operations

The operations shall be determined by the MIME content type of the Content Preparation Template resource.

7.5 Content Protocols Discovery API

7.5.1 Overview

The Content Protocols Discovery API is used by a 5GMSd Application Provider to find out which content ingest protocols are supported by the 5GMSd AS(s) associated with a 5GMSd AF. One of the supported ingest protocols is subsequently indicated in a Content Hosting Configuration for downlink streaming.

7.5.2 Resource structure

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

{apiRoot}`/3gpp-m1d/v1/provisioning-sessions/`*{provisioningSessionId}*`/`

Table 7.5.2-1 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 7.5.2-1: Operations supported by the Ingest Protocols Discovery API

Operation	Sub-resource path	Allowed HTTP method(s)	Description
Fetch list of supported content protocols	`protocols`	`GET`	This operation is used to retrieve a list of supported content protocols.

7.5.3 Data model

7.5.3.1 ContentProtocols resource

The data model for the *ContentProtocols* resource is specified in table 7.5.3.1-1 below:

Table 7.5.3.1-1: Definition of ContentProtocols resource

Property name	Data Type	Cardinality	Description
downlinkIngestProtocols	`array(ContentProtocolDescriptor)`	0..1	An array of *ContentProtocolDescriptor* objects, as specified in clause 7.5.3.2, each one uniquely identifying a content ingest protocol supported at interface M2d by the 5GMSd AS(s) associated with the corresponding 5GMSd AF.
geoFencingLocatorTypes	`array(URI String)`	0..1	An array of fully-qualified term identifiers, each one indicating a content geo-fencing locator type supported by the 5GMS System. Every 5GMS System shall support at least the locator type *urn:3gpp:5gms:locatortype:iso3166*.

7.5.3.2 ContentProtocolDescriptor type

The data model for the *ContentProtocolDescriptor* type is specified in table 7.5.3.2-1 below:

Table 7.5.3.2-1: Definition of ContentProtocolDescriptor type

Property name	Data Type	Cardinality	Description
termIdentifier	`URI String`	1..1	A fully-qualified term identifier from the controlled vocabulary *urn:3gpp:5gms:content-protocol*, as specified in clause 7.5.4.
descriptionLocator	`URL String`	0..1	The location of a description of the content protocol, for example the public web URL of its specification.

7.6 Content Hosting Configuration API

7.6.1 Overview

This clause specifies the API that a 5GMSd Application Provider uses at interface M1d to provision and manage 5GMSd AS Content Hosting Configurations by interacting with a 5GMSd AF. Each such configuration is represented by a *ContentHostingConfiguration*, the data model for which is specified in clause 7.6.3 below. The RESTful resources for managing Content Hosting Configurations are specified in clause 7.6.2 and the operations on these resources are further elaborated in clause 7.6.4.

7.6.2 Resource structure

The Content Hosting Configuration API is accessible through this URL base path:

{apiRoot}`/3gpp-m1d/v1/provisioning-sessions/`*{provisioningSessionId}*`/`

Table 7.6.2-1 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 7.6.2-1: Operations supported by the Content Hosting Configuration API

Operation	Sub-resource path	Allowed HTTP method(s)	Description
Create Content Hosting Configuration	`content-hosting-configuration`	`POST`	Used to create a Content Hosting Configuration resource.
Retrieve Content Hosting Configuration		`GET`	Used to retrieve an existing Content Hosting Configuration.
Update Content Hosting Configuration		`PUT`, `PATCH`	Used to modify an existing Content Hosting Configuration.
Delete Content Hosting Configuration		`DELETE`	Used to delete an existing Content Hosting Configuration.
Purge Content Hosting Configuration cache	`content-hosting-configuration/purge`	`POST`	This operation is used to invalidate some or all cached media resources associated with this Content Hosting Configuration.

7.6.3 Data model

7.6.3.1 ContentHostingConfiguration resource

The data model for the *ContentHostingConfiguration* resource is specified in table 7.6.3.1-1 below:

Table 7.6.3.1-1: Definition of ContentHostingConfiguration resource

Property name	Data Type	Cardinality	Description
name	`String`	1..1	A name for this Content Hosting Configuration.
IngestConfiguration	`Object`	1..1	Describes the 5GMSd Application Provider's origin server from which media resources will be ingested via interface M2d.
path	`String`	1..1	The relative path which will be used to address the media resources at interface M2d. This path is provided by the 5GMSd AF in the case of Push-based ingest.
pull	`Boolean`	1..1	Indicates whether to the 5GMSd AS shall use Pull or Push for ingesting the content.
protocol	`URI` `String`	1..1	A fully-qualified term identifier allocated in the name space *urn:3gpp:5gms:content-protocol* that identifies the content ingest protocol. The set of supported protocols is defined in clause 8.
entryPoint	`String`	1..1	An entry point to ingest the content. The semantics of the entry point are dependent on the selected ingest protocol. In the case of Push ingest (*pull* flag is set to False), this parameter is returned by the 5GMSd AF to the 5GMSd Application Provider and indicates the entry point for pushing the content. In case of Pull (*pull* flag is set to *True*), the *entryPoint* shall be provided to the 5GMSd AF to indicate the location from which content is to be pulled. In this case, the *entryPoint* shall be used as the base URL. A request received by the 5GMSd AS is mapped to a URL using the provided base URL to fetch the content from the origin server.

Property name	Data Type	Cardinality	Description
DistributionConfigurations	`Array(Object)`	1..1	Specifies the distribution method and configuration for the ingested content. More than one distribution may be configured for the ingested content, e.g. to offer different distribution configurations such as DASH and HLS.
contentPreparationTemplateId	`String`	0..1	Indicates that content preparation prior to distribution is requested by the 5GMSd Application Provider. It identifies the Content Preparation Template that shall be used as defined in clause 7.4
canonicalDomainName	`String`	1..1	All resources of the current distribution shall be accessible through this default FQDN assigned by the 5GMSd AF.
domainNameAlias	`String`	1..1	The 5GMSd Application Provider may assign another FQDN through which media resources are additionally accessible at M4d. This domain name is used by the 5GMSd AS to select an appropriate Server Certificate to present at M4d, and to set appropriate CORS HTTP response headers at M4d. If this property is present, the 5GMSd Application Provider is responsible for providing in the DNS a CNAME record that resolves *domainNameAlias* to *canonicalDomainName*.
PathRewriteRules	`Array(Object)`	0..1	An ordered list of rules for rewriting the request URL paths of media resource requests handled by the 5GMSd AS. 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 [5] against which the path part of each 5GMSd 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 ingest, the M4d download request path is used in the comparison. In the case of Push-based ingest, the M2d 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 5GMSd AS request path that matches *requestPathPattern*. In the case of Pull-based ingest, *IngestConfiguration.entryPoint* is concatenated with the mapped path and any leaf path element from the original M4d download request to form the M2d origin request URL. In the case of Push-based ingest, *canonicalDomainName* (and, optionally, *domainNameAlias*) are concatenated with the mapped path and any leaf path element from the original M2d upload request to form the distribution URL(s) exposed over M4d.

Property name	Data Type	Cardinality	Description
CachingConfigurations	`Array(Object)`	0..1	Defines a configuration of the 5GMSd AS cache for a matching subset of media resources ingested in relation to this Content Hosting Configuration.
urlPatternFilter	`String`	1..1	A pattern that will be used to match media resource URLs to determine whether a given media resource is eligible for caching by the 5GMSd AS. The format of the pattern shall be a regular expression as specified in [5].
CachingDirectives	`Object`	1..1	If a *urlPatternFilter* applies to a resource, then the provided *CachingDirectives* shall be applied by the 5GMSd AS at M4d, potentially overwriting any origin caching directives ingested at M2d.
statusCodeFilters	`Array(Integer)`	0..1	The set of HTTP origin response status codes to which these *CachingDirectives* apply. The filter shall be provided as a regular expression as specified in [5]. If the list is empty, the *CachingDirectives* shall apply to all HTTP origin response status codes at M2d.
noCache	`Boolean`	1..1	If set to *True*, this indicates that the media resources matching the filters shall not be cached by the 5GMSd AS and shall be marked as not to be cached when served by the 5GMSd AS at M4d.
maxAge	`Integer`	0..1	The caching time-to-live period that shall be set on ingested media resources matching the filters. This determines the minimum period for which the 5GMSd AS shall cache matching media resources as well as the time-to-live period signalled by the 5GMSd AS at interface M4d 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.
GeoFencing	`Object`	0..N	Limit access to the content to the indicated geographic areas.
locatorType	`URI String`	1..1	The type of the locators shall be indicated using a fully-qualified term identifier URI from the controlled vocabulary *urn:3gpp:5gms:locator-type*, as specified in clause 7.6.4.6, 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 value of *locatorType*, as specified in clause 7.6.4.6.
UrlSignature	`Object`	0..1	Defines the URL signing scheme. Only correctly signed and valid URLs will be allowed to access the content resource at M4d.
urlPattern	`String`	1..1	A pattern that shall be used to match M4d media resource URLs. The 5GMSd AS shall not serve a matching media resource at M4d unless it includes a valid authentication token. The format of the pattern shall be a regular expression as specified in [5].
tokenName	`String`	1..1	The name of the M4d request query parameter that the Media Player should use to present the authentication token when required to do so.
passphraseName	`String`	1..1	The name of the query parameter that is used to refer to the passphrase when constructing the authentication token. Note that the token is not included in the cleartext part of the M4d URL query component.

Property name	Data Type	Cardinality	Description
passphrase	`String`	1..1	The shared secret between the 5GMSd Application Provider and the 5GMSd AS for this *DistributionConfiguration*. The passphrase is used in the computation and verification of the M4d authentication token but is never sent in-the-clear over that interface.
tokenExpiryName	`String`	1..1	The name of the M4d request query parameter that the Media Player should use to present the token expiry field.
useIPAddress	`Boolean`	1..1	If set to *True*, the IP address of the UE 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 5GMSd AF only when the M4d request is made from a UE with this IP address.
ipAddressName	`String`	0..1	The name of the M4d request query parameter that is encoded as part of the authentication token if the *useIPAddress* flag is set to *True*. Note that the IP address is not passed in the cleartext part of the M4d URL query component.
certificateId	`String`	0..1	When content is distributed using TLS [16], the X.509 [8] certificate for the origin domain is shared with the 5GMSd AF so that it can be presented by the 5GMSd AS in the TLS handshake at M4d. This attribute indicates the identifier of the certificate to use.

7.6.4 Operations

7.6.4.1 Overview

This clause defines the behaviour that is expected from the 5GMSd AS when the Content Hosting Configuration has been successfully provisioned. The main operations that are performed affect the caching and purging of cached content as well as the processing for media preparation and at the edge.

7.6.4.2 Content caching

A Content Hosting Configuration may specify caching rules to be applied to media resources when they are distributed by the 5GMSd AS over interface M4d. The distribution shall use the *urlPatternFilter* in the *CachingConfiguration* object to determine which caching directives apply to that object. In case a media resource's URL matches the pattern filter of more than one *CachingConfiguration*, the first match shall apply. In case no *CachingConfiguration* is identified as a match, the 5GMSd AS shall apply the caching directives that were received from the origin. In the case where no match is found and the origin server does not supply caching directives at M2d, then default caching directives based on the media resource type shall be applied.

A caching directive shall either indicate that a matching media resource is not to be cached by the 5GMSd AS, nor by downstream M4d clients (*noCache* set to *True*), or that the 5GMSd AS and downstream M4d clients are to cache it for *maxAge* seconds. The *maxAge* value applies relative to the time when a media resource was ingested, *t_ingest*. For an HTTP-based ingest, this corresponds to the `Date` header field in the HTTP request/response that carries the media resource at M2d. At the time *t_ingest* + *maxAge*, the object is considered stale and should not be served at M4d from the 5GMSd AS cache. The 5GMSd AS shall compensate for any synchronization skew between the origin and its own clock. This can be for instance done by including the `max-stale` HTTP cache directive in its M4d responses.

The *maxAge* value may be signalled at M4d by the 5GMSd AS using the `Expires` HTTP response header or the HTTP `Cache-Control` directives `max-age` or `s-maxage`.

When distributing a media resource using HTTP, a *no-cache* request may be translated into a `no-cache` and `no-store` HTTP `Cache-Control` directive and/or a `max-age=0` HTTP `Cache-Control` directive.

By default, all origin HTTP header fields shall be assumed as not forwarded by the 5GMSd AS, unless specified otherwise by setting the flag *originCacheHeaders* to *True*.

7.6.4.3 Cache purging

The 5GMSd Application Provider may perform a purge operation to invalidate some or all cached media resources of a particular Content Hosting Configuration. A regular expression describing the set of media resource URLs to be purged from the 5GMSd AS cache for the Content Hosting Configuration in question shall be supplied in the body of the request. The body shall be encoded using the *application/x-www-form-urlencoded* MIME 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 5GMSd AF shall immediately invalidate all media resources in the 5GMSd AS cache matching the regular expression by declaring them as stale. Any request at interface M4d for a purged media resource will trigger the fetching (and possible caching) of the current version from the origin via M2d in case of a Pull-based ingest. For Push-based ingest, the request shall be responded to with a 404 (Not Found) HTTP response, until a new version of the object is pushed by the origin to the 5GMSd AS via M2d.

7.6.4.4 Content processing

The 5GMSd AF can perform various content processing tasks (such as repackaging, encryption, ABR transcoding) on media resources ingested at M2d prior to serving them at M4d. These processing tasks shall be specified in a Content Preparation Template resource referenced from the Content Hosting Configuration object.

7.6.4.5 URL signing

The URL signing procedure allows the 5GMSd Application Provider to prevent deep linking and unauthorized access to M4d media resources. It works by cryptographically signing some elements of the M4d request URL and then appending this authentication token to the URL as an additional query parameter. The token is generated by the 5GMSd Application Provider and supplied to the player, for example as part of an initial URL. When it receives a request that requires URL signing, the 5GMSd AS verifies the presence and validity of the token in the M4d request URL before allowing access to the requested media resource. The 5GMSd AS(s) and the origin share a secret that is encoded as part of the query parameter hash, but not shared with the 5GMSd Media Player.

The validity of the authentication token can also be limited to a single UE. If *useIPAddress* is set to True, then the public IP address of the UE as viewed by the 5GMSd AS, *ue_public_ip_address*, shall be incorporated into the token calculation. The parameter name shall be indicated by *ipAddressName*.

The shared secret shall be provided in *UrlSignature[passphrase]* as a string of length between 6 and 50 characters. The parameter name for the passphrase shall be provided by *passphraseName*.

The expiry time of the signed URL, *tokenExpiry*, shall be included as an additional query parameter in the URL exposed at M4d with the name indicated in *tokenExpiryName*. The expiry time shall be the string representation of the number of seconds from 1970-01-01T00:00:00Z UTC until the specified UTC date/time, ignoring leap seconds, as defined in section 4.16 of POSIX.1 [11].

Given the above, the authentication token shall be calculated as:

token = SHA512(*url&UrlSignature[tokenExpiryName]=token_expiry&UrlSignature[ipAddressName]=ue_public_ip_address&UrlSignature[passphraseName]=passphrase*)

where the SHA512 function shall be the SHA-512 hash [6] of the enclosed string. The *url* parameter shall be the original M4d media resource request URL, including the scheme, authority and path components but excluding any query and fragment components.

The resulting token value shall be "base64url" encoded, as specified in section 5 of RFC 4648 [10], prior to inclusion in the M4d URL.

The query part of the signed URL presented by the 5GMSd Media Player at M4d as proof of authenticity shall be composed as follows:

query= UrlSignature[tokenExpiryName]=token_expiry &*UrlSignature[tokenName]*=base64url(*token*)

For all media resources requested at reference point M4d that match the regular expression specified in *UrlSignature[urlPattern]*, the 5GMSd AS shall validate the *query* presented in the request URL according to the following steps:

1) If the parameter indicated by *UrlSignature.tokenName* is absent from *query*, or if the supplied *token* value is malformed, the 5GMSd AS shall respond with a *403 (Forbidden)* error response message and terminate further processing of the M4d request.

2) If the parameter indicated by *UrlSignature.tokenExpiryName* is absent from *query*, or if the supplied *token_expiry* value has expired, or if the supplied *token_expiry* is malformed, the 5GMSd AS shall respond with a *403 (Forbidden)* error response message and terminate further processing of the M4d request.

3) The 5GMSd AS shall compute the authentication token according to the *token* production specified above using the requesting UE's public IP address as the value of *ue_public_ip_address* if required by *UrlSignature.useIPAddress* being set to *True*. After applying "base64url" encoding, the 5GMSd AS shall compare this with the value supplied in the URL *query* parameter whose name is *UrlSignature.tokenName*. If the two values differ, the 5GMSd AS shall respond with a *403 (Forbidden)* error response message and terminate further processing of the M4d request.

4) Otherwise, the presented authentication token is valid. The 5GMSd AS shall either return the media resource in a *200 (OK)* response message (if it is able to serve that media resource), or else return an appropriate error response, such as *404 (Not Found)* or *503 (Service Unavailable*).

7.6.4.6 Geofencing

The 5GMSd Application Provider may wish to limit access to its media content at interface M2d to UEs located in certain geographical zones. Geofencing is used to configure the zone from which content is accessible.

Two different types of locator are specified here:

- **Administrative area locator:** the value of *GeoFencing.locatorType* shall be *urn:3gpp:5gms:locator-type:iso3166* and each member of the *GeoFencing.locators* array shall be either a string representation of an ISO 3166-1 alpha-2 country code [18] (e.g. *US*, *CN*, *KR*, *GB*, *FR*) or an ISO 3166-2 code [19] 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*).

[- **Tracking Area locator:** the value of *GeoFencing.locatorType* shall be *urn:3gpp:5gms:locatortype:trackingAreaCode* and each member of the *GeoFencing.locators* array shall be the Fully-Qualified Domain Name representation of a Tracking Area Code, as defined in clause 19.4.2.3 of TS 23.003 [7].]

7.7 Consumption Reporting Provisioning API

7.7.1 Overview

The Consumption Reporting Provisioning API is a RESTful API that allows a 5GMSd Application Provider to configure the Consumption Reporting Procedure for a particular Provisioning Session at interface M1d. The different procedures are described in clause 4.2.5. The Consumption Reporting Configuration is represented by a *ConsumptionReportingConfiguration*, the data model for which is specified in clause 7.7.3 below. The RESTful resources for managing the Consumption Reporting Configuration is specified in clause 7.7.2.

7.7.2 Resource structure

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

{apiRoot}/3gpp-m1d/v1/provisioning-sessions/{provisioningSessionId}/

Table 7.7.2-1 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 7.7.2-1: Operations supported by the Consumption Reporting Provisioning API

Operation	Sub-resource path	Allowed HTTP method(s)	Description
Activate Consumption Reporting procedure with a Consumption Reporting Configuration	`consumption-reporting-configuration`	`POST`	Activate the consumption reporting procedure and to set the Consumption Reporting Configuration.
Fetch Consumption Reporting Configuration	`consumption-reporting-configuration`	`GET`	Retrieve an existing Consumption Reporting Configuration.
Update Consumption Reporting Configuration	`consumption-reporting-configuration`	`PUT`, `PATCH`	Modify an existing Consumption Reporting Configuration.
Delete Consumption Reporting Configuration	`consumption-reporting-configuration`	`DELETE`	Deactivate the consumption reporting procedure for that particular session.

7.7.3 Data model

7.7.3.1 ConsumptionReportingConfiguration resource

The data model for the *ConsumptionReportingConfiguration* resource is specified in table 7.7.3.1-1.

Table 7.7.3.1-1: 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 streaming session has ended.
samplePercentage	`Percentage`	0..1	The proportion of 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.
locationReporting	`boolean`	0..1	Stipulates whether the Media Session Handler is required to provide location data to the 5GMSd AF in consumption reporting messages (in case of MNO or trusted third parties).

7.8 Metrics Reporting Configuration API

7.8.1 Overview

The Metrics Reporting Configuration API allows an 5GMS System operator or a 5GMSd Application Provider to configure the Metrics Collection and Reporting procedure for a particular Provisioning Session at interface M1d.

7.8.2 Resource structure

The Metrics Reporting Configuration API is accessible through the following URL base path:

{apiRoot}/3gpp-m1d/v1/provisioning-sessions/{provisioningSessionId}/

Table 7.8.2-1 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 of the table shall be appended to the URL base path.

Table 7.8.2-1: Metrics Reporting Configuration resource

Operation	Sub-resource path	Allowed HTTP method(s)	Description
Create Metrics Reporting Configuration	`metrics-reporting-configuration`	`POST`	Create and optionally provide a configuration. If the operation succeeds, the URL of the created Metrics Reporting Configuration resource shall be returned in the `Location` header of the response.
Read Metrics Reporting Configuration	`metrics-reporting-configuration/` *{metricsReportingConfigurationId}*	`GET`	Retrieve the values of an existing Metrics Reporting Configuration.
Update Metrics Reporting Configuration		`PUT`, `PATCH`	Provide initial upload of a new configuration, or either the modification of, or replacement to an existing configuration.
Delete Metrics Reporting Configuration		`DELETE`	Delete a configuration, disables reporting.

7.8.3 Data model

7.8.3.1 MetricsReportingConfiguration resource

The data model for the *MetricsReportingConfiguration* resource is specified in table 7.8.3-1 below:

Table 7.8.3-1: Definition of MetricsReportingConfiguration resource

Property name	Type	Cardinality	Description
metricsReportingConfigurationId	`String`	1..1	An identifier for this Metrics Reporting Configuration that is unique within the scope of the enclosing Provisioning Session.
scheme	`Array(URI String)`	0..1	The scheme associated with this Metrics Reporting Configuration. A scheme may be associated with 3GPP or with a non-3GPP entity. If not specified, the 3GPP metrics scheme *urn:3GPP:ns:PSS:DASH:QM10* from TS 26.247 shall apply.
dataNetworkName	`String`	0..1	The Data Network Name (DNN) which shall be used when sending metrics reports. If not specified, the default DNN shall be used.
reportingInterval	`DurationSec`	0..1	The time interval between successive metrics reports. If not specified, a single final report shall be sent after the streaming session has ended.
samplePercentage	`Percentage`	0..1	The proportion of streaming sessions for which metrics shall be reported. If not specified, reports shall be sent for all sessions.
urlFilters	`Array(String)`	0..1	A list of content URL patterns for which metrics shall be reported. If not specified, reporting shall be done for all URLs.
metrics	`Array(String)`	0..1	A non-empty list of metrics which shall be collected and reported. For the 3GPP scheme *urn:3GPP:ns:PSS:DASH:QM10* the listed metrics shall correspond to one or more of the metrics as specified in clauses 10.3 and 10.4, respectively, of TS 26.247 [7], and the quality reporting scheme and quality reporting protocol as defined in clauses 10.5 and 10.6, respectively, of [7] shall be used. If not specified, the complete (or default if applicable) set of metrics associated with the specified scheme shall be collected and reported.

7.9 Policy Templates Provisioning API

7.9.1 Overview

The Policy Templates Provisioning API allows a 5GMS Application Provider to configure a set of Policy Templates within the scope of a Provisioning Session that can subsequently be applied to media streaming sessions belonging to that Application Provider using the Dynamic Policies API specified in clause 11.5. A Policy Template is used to specify the traffic shaping and charging policies to be applied to these media streaming sessions.

A Policy Template, identified by its *policyTemplateId*, represents a set of PCF/NEF API parameters which defines the service quality and associated charging for the media streaming sessions. The Policy Template is configured as part of the provisioning procedures with the 5GMS AF and is then used by the 5GMS AF to request specific QoS and charging policies for that session from the PCF or NEF.

The state of a Policy Template can be:

- *pending*: The Policy Template is awaiting validation, 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 5GMS AF.
- *ready*: After successful validation by the 5GMS AF the Policy Template moves into this state.
- *suspended*: The 5GMS AF may move a Policy Template into this state under certain conditions defined within the Service Level Agreement.

When the Policy Template is used for QoS Flows, the *qoSSpecification* object (of type *M1QoSSpecification*) shall be present:

- The *qosReference* value is obtained with the Service Level Agreement. See TS 23.502 for detailed usage.
- The *maxBtrUl* and *maxBtrDl* properties define the maximal bit rate which can be used for QoS Flows. This value is defined by the 5G System.
- The *maxAuthBtrUl* and *MaxAuthBtrDl* properties define the maximal authorized bit rate values which can be requested by a Media Session Handler. Higher bit rate values are not authorized for use by the 5GMS Application Provider.
- The *minPacketLossRateDl* and *minPacketLossRateUl* properties define the minimal authorized packet loss rate, which can be requested by a Media Session Handler.

When the Policy Template is used for differential changing the *chargingSpecification* property shall be present.

The *ApplicationSessionContext* Object is a mandatory object, which contains at least the *aspId* property.

- The *aspId* identifies the API invoker.
- The *dnn* property contains the Data Network Name of the data network, in which the 5GMS AF is hosted.
- When Network Slicing is used, the *sliceInfo* property contains information about the network slice, which is serving the UE.

7.9.2 Resource structure

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

{apiRoot}/3gpp-m1d/v1/provisioning-sessions/*{provisioningSessionId}*/

Table 7.9.2-1 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 7.9.2-1: Operations supported by the Policy Template Provisioning API

Operation	Sub-resource path	Allowed HTTP method(s)	Description
Create a new Policy Template	policy-templates	POST	Used to create a new Policy Template resource. If the operation succeeds, the URL of the created Policy Template resource shall be returned in the Location header of the response.
Fetch a Policy Template	policy-templates/ *{policyTemplateId}*	GET	Used to retrieve an existing Policy Template resource.
Update a Policy Template		PUT, PATCH	Used to modify the configuration of an existing Policy Template.
Delete a Policy Template		DELETE	Used to delete an existing Policy Template resource.

7.9.3 Data model

7.9.3.1 PolicyTemplate resource

The data model for the *PolicyTemplate* resource is specified in table 7.9.3-1 below:

Table 7.9.3-1: Definition of PolicyTemplate resource

Property	Type	Cardinality	Usage	Visibility	Description
policyTemplateId	String	1..1	C: RO R: RO U: RO		Unique identifier of this Policy Template within the scope of the Provisioning Session.
state	Enumeration of Strings	1..1	C: RO R: RO U: RO		A Policy Template may be in the *pending*, *ready*, or *suspended* state. Only a Policy Template in the *ready* state may be instantiated as a Dynamic Policy Instance and applied to streaming sessions.
apiEndPoint	String	1..1	C: RW R: RO U: RW	MNO Admin	The API endpoint that should be invoked when activating a Dynamic Policy Instance based on this Policy Template.
apiType	Enumeration of Strings	1..1	C: RW R: RO U: RW	MNO Admin	*N5*: Npcf Policy Authorization Service. *N33*: AsSessionWithQoS or CHargableParty.
externalReference	String	1..1	C: RW R: RO U: RW		Additional identifier for this Policy Template, unique within the scope of its Provisioning Session, that can be cross-referenced with external metadata about the streaming session.

qoSSpecification	`M1QoSSpecification`	0..1	C: RW R: RO U: RW		Specifies the network quality of service to be applied to streaming sessions at this Policy Template.
ApplicationSessionContext	`Object`	1..1			Specifies information about the application session context to which this Policy Template can be applied.
afAppId	`AfAppId`	0..1		Read-Only	As defined in clause 5.6.2.3 of TS 29.514 [34].
sliceInfo	`Snssai`	0..1			
dnn	`Dnn`	0..1			
aspId	`AspId`	0..1			
chargingSpecification	`ChargingSpecification`	0..1			Provides information about the charging policy to be used for this Policy Template.

Editor's Note: The parameter *externalReference* is for further study. It may be a provisioning parameter of the Media Player and/or a Media Session Handler to assist mapping of external references to a *policyTemplateId*.

Editor's Note: The *ChargingSpecification object may contain any charging related information, such as sponId or afChargeId.*

8 Media Ingest and Publish (M2) protocols

8.1 General

The set of content protocols supported by the 5GMS AS is listed in table 8.1-1 below:

Table 8.1-1: Supported content protocols

Description	Term identifier	Clause
Content ingest protocols at interface M2d		
HTTP pull-based content ingest protocol	*urn:3gpp:5gms:content-protocol:http-pull-ingest*	8.2
DASH-IF push-based content ingest protocol	*urn:3gpp:5gms:content-protocol:dash-if-ingest*	8.3
Content egest protocols at interface M2u		

8.2 HTTP pull-based content ingest protocol

If *IngestConfiguration.protocol* is set to *urn:3gpp:5gms:content-protocol:http-pull-ingest* in the Content Hosting Configuration, media resources shall be ingested by the 5GMSd AS using HTTP [9]. The *IngestConfiguration.pull* property shall be set to *True*, indicating that a Pull-based protocol is used. The *IngestConfiguration.entryPoint* property shall point at the 5GMSd Application Provider's origin server, as specified in table 7.6.3.1-1 and may indicate the use of HTTPS [16]. The *IngestConfiguration.entryPoint* shall not contain a path part.

When the 5GMSd AS receives a request for a media resource at interface M4d that cannot be satisfied from its content cache, the request shall be transformed into a corresponding HTTP `GET` request directed to the 5GMSd Application Provider's origin server via interface M2d, using the abovementioned *entryPoint* property concatenated with the *mappedPath* from the applicable path rewrite rule (if any) selected from *DistributionConfiguration.PathRewriteRules* and the leaf path element from the original M4d request URL to construct the M2d request URL.

8.3 DASH-IF push-based content ingest protocol

If *IngestConfiguration.protocol* is set to *urn:3gpp:5gms:content-protocol:dash-if-ingest* in the Content Hosting Configuration, media resources shall be ingested by the 5GMSd AS as specified by the DASH-IF Live Media Ingest specification [3]. The *IngestConfiguration.pull* property shall be set to False, indicating that a Push-based protocol is used. The *IngestConfiguration.entryPoint* property shall be set to the URL that will be used to upload the DASH segments and MPD to the 5GMSd AS at interface M2d. This entry point URL shall not contain a path: the path for the URL shall instead be specified by the *IngestConfiguration.path* property.

9 Internal (M3) APIs

APIs of this reference point are not specified within this release.

10 Media Streaming (M4) APIs

10.1 General

This clause deals with the interface and APIs for media streaming for different distribution formats and protocols.

10.2 DASH Distribution

In the case of DASH distribution, M4d is relevant for the distribution as shown in Figure 10.2-1.

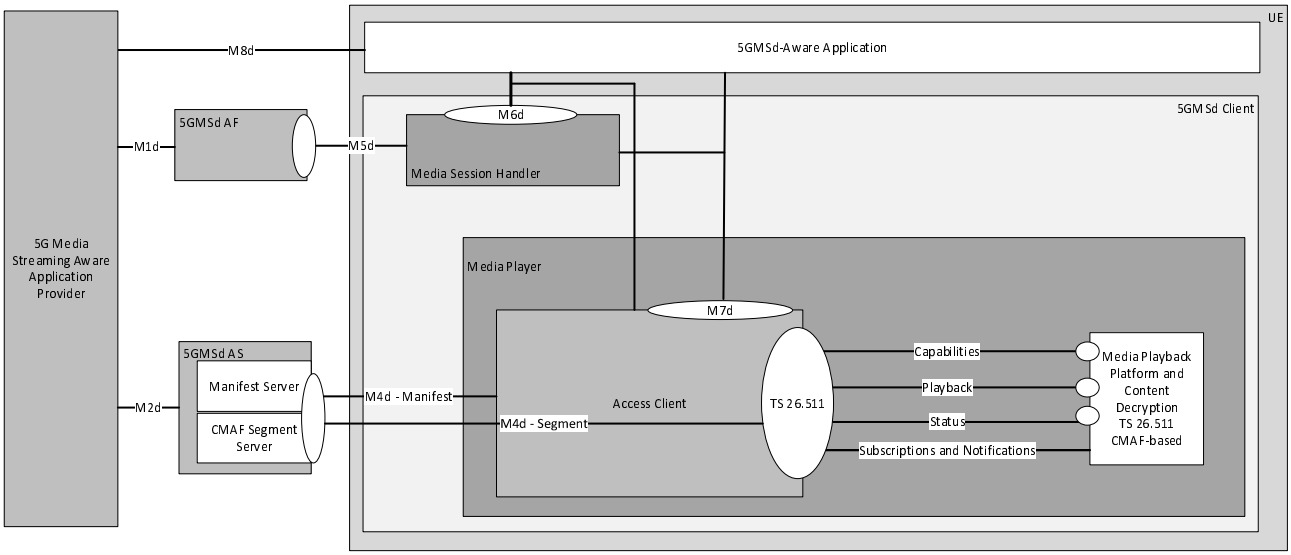

Figure 10.2-1: M4d usage for DASH distribution

For DASH-based distribution according to TS 26.247 [4] and ISO/IEC 23009-1 [32], two main formats are of relevance:

1) The Media Presentation Description (MPD) that is processed in the DASH Access Client.

2) The Segment formats that are passed through the DASH Access Client and processed in the Media Playback and Content Decryption Platform. Note that the DASH Access Client may parse Segments to extract for example Inband Events or producer reference times.

Other resources may be referenced in the MPD, for example DRM related information.

The Segment formats for DASH Streaming in the context of 5G Media Streaming are defined in TS 26.511 [35] based on the CMAF encapsulation. The DASH Access Client downloads the Segments from the 5GMSd AS based on the instructions in the MPD and the instructions from the 5GMSd-Aware Application through M7d (see clause 13 for details).

The interface between the DASH Access Client and the Media Playback and Content Decryption Platform as well as the 5GMSd Client requirements for media codecs are documented in TS 26.511 [12].

The following requirements apply for M4d:

1) The Media Presentation Description (MPD) and Segments shall conform to an MPD according to ISO/IEC 23009-1 [32] or TS 26.247 [4].

2) The Segment formats should conform to CMAF addressable resources as well as to the requirements in TS 26.511 [35].

3) The Media Presentation should conform to the 5G Media Streaming DASH Interoperability Point as defined in clause 7.3.11 of TS 26.247 [4].

A 5GMSd Client shall support the 5G Media Streaming DASH Interoperability Point as defined in TS 26.247 [4], clause 7.3.11. A 5GMSd Client may support additional DASH profiles and interoperability points.

The MPD may contain a one or several **`ServiceDescription`** elements that include operational parameters. The MPD may also include multiple configurations for the media (different codecs, different content protection, different resolutions, etc.), for example for playback under different operating policies. The handling of this information is documented in clause 13.2.

11 Media Session Handling (M5) APIs

11.1 General

This clause defines the Media Session Handling APIs used by the Media Session Handler to access resources exposed by the 5GMS AF at interface M5.

11.2 Service Access Information API

11.2.1 General

The Service Access Information API is used by the Media Session Handler to obtain configuration information from the 5GMS AF that enables it to use the other Media Session Handling APIs specified in clause 11.3 *et seq.*

11.2.2 Resource structure

The Service Access Information API is accessible through the following URL base path:

{apiRoot}/3gpp-m5d/v1/service-access-information/

The operations and the corresponding HTTP methods in table 11.2.2-1 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	Sub-resource path	Allowed HTTP method(s)	Description
Fetch Service Access Information	*{provisioningSessionId}*	GET	Used to acquire the Service Access Information resource for the specified Provisioning Session. The *{provisioningSessionId}* uniquely identifies the Service Access Information Resource and is allocated by the 5GMSd AF during creation of a Provisioning Session.

11.2.3 Data model

11.2.3.1 ServiceAccessInformation resource type

The data model for the *ServiceAccessInformtion* resource is specified in table 11.2.3.1-1 below:

Table 11.2.3.1-1: Definition of ServiceAccessInformation resource

Property name	Type	Cardinality	Usage	Description
provisioningSessionId	String	1..1	RO	Unique identification of the M1d Provisioning Session.
StreamingAccess	Object	0..1	RO	
mediaPlayerEntry	URL String	0..1	RO	A document or a pointer to a document that defines a media presentation e.g. MPD for DASH content or URL to a video clip file.

ClientConsumptionReportingConfiguration	`Object`	0..1	RO	
reportingInterval	`DurationSec`	0..1	RO	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 streaming session has ended.
serverAddresses	`Array(URLString)`	1..1	RO	A list of 5GMSd AF addresses (URLs) where the consumption reporting messages are sent by the Media Session Handler. (Opaque URL, following the 5GMS URL format.)
locationReporting	`Boolean`	1..1	RO	Stipulates whether the Media Session Handler is required to provide location data to the 5GMSd AF in consumption reporting messages (in case of MNO or trusted third parties).
samplePercentage	`Percentage`	1..1	RO	The percentage of streaming sessions that shall send consumption reports, expressed as a floating point value between 0.0 and 100.0.
DynamicPolicyInvocationConfiguration	`Object`	0..1	RO	
serverAddresses	`Array(URLString)`	1..1	RO	A list of 5GMSd AF addresses (URLs) which offer the APIs for dynamic policy invocation sent by the Media Session Handler. (Opaque URL, following the 5GMS URL format.)
validPolicyTemplateIds	`Array(String)`	1..1	RO	A list of Policy Template identifiers which the 5GMSd Client is authorized to use.
sdfMethods	`Array(SdfMethod)`	1..1	RO	A list of recommended service data flow description methods (descriptors), e.g. 5-Tuple, ToS, 2-Tuple, etc, which should be used by the Media Session Handler to describe the service data flows for the traffic to be policed.
externalReferences	`Array(String)`	0..1	RO	Additional identifier for this Policy Template, unique within the scope of its Provisioning Session, that can be cross-referenced with external metadata about the streaming session. Example: "HD_Premium".

ClientMetricsReportingConfigurations	`Array(Object)`	0..1	RO	
serverAddresses	`Array(URL String)`	1..1	RO	A list of 5GMSd AF addresses to which metrics reports shall be sent. (Opaque URL, following the 5GMS URL format.)
dataNetworkName	`String`	0..1	RO	The DNN which shall be used when sending metrics reports. If not specified, the name of the default DN shall be used.
reportingInterval	`DurationSec`	0..1	RO	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 streaming session has ended.
samplePercentage	`Percentage`	1..1	RO	The percentage of streaming sessions that shall report metrics, expressed as a floating point value between 0.0 and 100.0.
urlFilters	`Array(String)`	1..1	RO	A list of URL patterns for which metrics reporting shall be done. The format of each pattern shall be a regular expression as specified in [5]. If not specified, reporting shall be done for all sessions.
metrics	`Array(String)`	1..1	RO	A list of metrics which shall be reported.
NetworkAssistanceConfiguration	`Object`	0..1	RO	
serverAddress	`URL String`	1..1	RO	Address of the 5GMSd AF that offers the APIs for 5GMSd AF-based Network Assistance, for access by the 5GMSd Media Session Handler. This address shall be an opaque URL, following the 5GMS URL format.

11.2.4 Operations

This clause defines the behaviour that is expected from the 5GMSd AF when a Service Access Information resource is acquired by the Media Session Handler. The main operation that is performed is to look up or generate the Service Access Information.

11.3 Consumption Reporting API

11.3.1 General

The Consumption Reporting API allows the Media Session Handler to report media consumption to the 5GMSd AF. The API defines data models, resources and the related procedures for the creation and management of the consumption reporting procedures. This procedure is configured by the *ServiceAccessInformation* resource, as defined in clause 11.2.3.

11.3.2 Reporting procedure

Consumption reports shall be submitted to one of the URLs selected from the *ClientConsumptionReporting Configuration.serverAddresses* array of the *ServiceAccessInformation* resource (see clause 11.2.3). The path of the URL should conform to the following general format:

{apiRoot}`/3gpp-m5d/v1/consumption-reporting/`*{aspId}*

where *{aspId}* shall be substituted by the 5GMS Client with the relevant Application Service Provider identifier.

The only HTTP method supported by this endpoint is `POST`.

11.3.3 Report format

11.3.3.1 ConsumptionReport format

This type represents a consumption report data. This structure is used by the Media Session Handler to report the consumption.

Table 11.3.3.1-1: Definition of ConsumptionReport format

Attribute name	Data type	Cardinality	Description
mediaPlayerEntry	`string`	1..1	Identifies the Media player entry. In the case of DASH, the media player entry pointer shall be the URL of the MPD.
reportingClientId	`string`	1..1	Identifies the identifier of the UE that consumes data. The client ID can be an MSISDN.
consumptionReportingUnits	`Array(ConsumptionReportingUnit)`	1..1	An array of consumption reporting units.

11.3.3.2 ConsumptionReportingUnit type

This type represents a single consumption reporting unit.

Table 11.3.3.2-1: Definition of type ConsumptionReportingUnit

Attribute name	Data type	Cardinality	Description
mediaConsumed	`string`	1..1	Identifies the media consumed. In the case of DASH, the value of the **`Representation`**`@id` attribute shall be quoted.
startTime	`DateTime`	1..1	The time when this consumption reporting unit started.
duration	`DurationSec`	1..1	The duration of this consumption reporting unit.

11.4 Metrics Reporting API

11.4.1 General

The Metrics Reporting API allows the Media Session Handler to send metrics reports to the 5GMSd AF. This procedure is configured by the *ServiceAccessInformation* resource, as defined in clause 11.2.3. Note that multiple metrics configurations can be active at the same time, each identified by a unique *metricsReportingConfigurationId*.

11.4.2 Reporting procedure

Metrics reports related to a specific *metricsReportingConfigurationId* shall be submitted to one of the URLs selected from the *ClientMetricsReportingConfiguration.serverAddresses* array of the *ServiceAccessInformation* resource (see clause 11.2.3). The path of the URL should conform to the following general format:

{apiRoot}`/3gpp-m5d/v1/metrics-reporting/`*{provisioningSessionId}/{metricsReportingConfigurationId}*

where *{provisioningSessionId}* shall be substituted by the 5GMS Client with the relevant Provisioning Session identifier and *{metricsReportingConfigurationId}* shall be substituted with the relevant Metrics Reporting Configuration identifier.

The only HTTP method supported by this endpoint is `POST`.

11.4.3 Report format

Metrics reports shall be submitted by the Media Session Handler in a format specified by the metrics reporting scheme in question. The `Content-Type` HTTP request header shall be set in accordance with the relevant metrics reporting scheme specification.

NOTE: TS 26.247 [7] clauses 10.6.1 and 10.6.2 specify the required MIME content type and metrics report format for the 3GPP *urn:3GPP:ns:PSS:DASH:QM10* metrics reporting scheme.

11.5 Dynamic Policies API

11.5.1 Overview

The Dynamic Policies API allows the Media Session Handler to request a specific policy and charging treatment to be applied to a particular application data flow by invoking RESTful operations on the 5GMSd AF at interface M5d. The API defines a set of data models, resources and the related procedures for the creation and management of the dynamic policy request.

11.5.2 Resource structure

The Dynamic Policies API is accessible through the following URL base path:

{apiRoot}/3gpp-m5/v1/dynamicpolicies/

Table 11.5.2-1 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 11.5.2-1: Operations supported by the Dynamic Policies API

Resource name	Sub-resource path	Allowed HTTP methods	Description
Dynamic Policies	policies	POST	Create a new Dynamic Policy resource. If the operation succeeds, the URL of the created Dynamic Policy Instance resource shall be returned in the Location header of the response.
Dynamic Policy	policies/*{dynamicPolicyId}*	GET	Read a Dynamic Policy resource.
		PUT	Replace an existing Dynamic Policy resource.
		PATCH	Modify an existing Dynamic Policy resource.
		DELETE	Delete an existing Dynamic Policy resource.

11.5.3 Data model

11.5.3.1 DynamicPolicy resource

Table 11.5.3.1-1: Definition of Dynamic Policy resource

Property name	Data type	Cardinality	Usage	Description
dynamicPolicyId	String	1..1	RO	Unique identifier for this Dynamic Policy.
policyTemplateId	String	1..1	C: RW R: RO U: RW	Identifies the Policy Template which should be applied to the application flow(s).
serviceDataFlowDescriptions	Array(ServiceDataFlowDescription)	1..1	C: RW R: RO U: RW	Describes the service data flows managed by this Dynamic Policy.
provisioningSessionId	String	1..1	C: RW R: RO U: RW	Uniquely identifies Provisioning Session, which is linked to the Application Service Provider.
qosSpecification	M5QoSSpecification	0..1	C: RW R: RO U: RW	Describes the network Quality of Service properties of this Dynamic Policy.
enforcementMethod	String	0..1	C: RO R: RO U: RO	Description of the Policy Enforcement Method. The parameter is set by the 5GMSd AF.
enforcementBitRate	Integer	0..1	C: RO R: RO U: RO	Description of the enforcement bit rate.

11.5.4 Operations

This clause defines the behaviour that is expected when activating a Dynamic Policy Instance. The *policyTemplateId* uniquely identifies the Policy Template, to which the Dynamic Policy Instance is associated. The *provisioningSessionId* associates the Dynamic Policy Instance to a Provisioning Session.

The Dynamic Policy resource contains a *serviceDataFlowDescription* property which contains the service data flow template according to TS 23.503. The ServiceDataFlowDescription shall contain one of:

- a *flowDescription* Object (incl. 5-Tuples, Type of Service, Security Parameter Index, etc.).
- a *domainName.*

When the Media Session Handler activate a QoS-related Dynamic Policy Template, then the *qosSpecifcation* property shall be present and it shall contain the following properties:

- *marBwDlBitRate* and *marBwUlBitRate*, indicating the actual requested bit rate by the Media Session Handler.
- *mirDwDlBitRate* and *mirBwUlBitRate*, indicating the absolute minimal usable bit rate.
- *minDesBwDlBitRate* and *minDesBwUlBitrate*, indicating the desired lower bit rate.

When the 5G System employs a traffic enforcement function to ensure that the traffic is complying a certain traffic policy, the Dynamic Policy resource may contain the following two properties

- an *enforcementMethod*, indicating the type of enforcement method (like leaky bucket).
- an *enforcementBitrate* property, indicating the maximal bit rate.

11.6 Network Assistance API

11.6.1 Overview

If AF-based Network Assistance is supported, then the Network Assistance API component of interface M5d, as defined in the present sub-clause, is first used to provision a Network Assistance Session resource. The Network Assistance Resource can then be used to obtain bit rate recommendations and to issue delivery boost requests during the ongoing media streaming session.

11.6.2 Resource structure

The Network Assistance API is accessible via the following URL base path:

`{apiRoot}/3gpp-m5d/v1/network-assistance/`

Table 11.6.2-1 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 11.6.2-1: Operations supported by the Network Assistance API

Operation	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.
Fetch a Network Assistance Session resource	*{naSessionId}*	`GET`	Fetch the properties of an existing Network Assistance Session.
Update a Network Assistance Session resource	*{naSessionId}*	`PUT`, `PATCH`	Update the properties of an existing Network Assistance Session.
Request a bit rate recommendation	*{naSessionId}*`/recommendation`	`GET`	Obtain a bit rate recommendation for the next recommendation window.
Request a delivery boost	*{naSessionId}*`/boostRequest`	`POST`	Request a delivery boost for the next recommendation window.
Terminate Network Assistance Session	*{naSessionId}*	`DELETE`	Terminate a Network Assistance session.

11.6.3 Data model

11.6.3.1 NetworkAssistanceSession resource

The *NetworkAssistanceSession* resource is specified in table 11.6.3.1-1 below.

Table 11.6.3.1-1: Definition of NetworkAssistanceSession resource

Property name	Type	Cardinality	Usage	Description
naSessionId	`String`	1..1	C: RO R: RO U: RO	Unique identifier for this Network Assistance Session.
serviceDataFlowInformation	`Array(ServiceDataFlowDescription)`	0..1	C: RW R: RO U: RW	Identification of the application flows for the streaming session for which Network Assistance is to be used, e.g. 2-tuple (IP addresses) or 5-tuple (IP Addresses, protocol and ports).
policyTemplateId	`String`	0..1	C: RW R: RO U: RW	Identification of the policy that is in force for the streaming session.
requestedQoS	`M5QoSSpecification`	0..1	C: RW R: RO U: RW	The requested QoS parameters.
recommendedQoS	`M5QoSSpecification`	0..1	C: RO R: RO U: RO	The QoS parameters currently recommended by the 5GMS AF.
notficationURL	`String`	0..1	C: RO R: RO U: RO	A URL to the MQTT channel over which notifications are to be sent by the 5GMS AF for this session. When set, the Media Session Handler shall subscribe to this channel. The notification messages shall be in the form of the *M5QoSSpecification* data type.

11.6.4 Operations

The 5GMSd client uses the `POST` method to create a Network Assistance session with the 5GMS AF. The AF returns the Network Assistance session identifier if session setup was successful, otherwise an error code is returned without a Network Assistance session identifier.

The 5GMSd Client uses the Network Assistance session resource identifier (naSessionId) provided by the AF to refer all subsequent API calls to the AF applicable to that Network Assistance session.

The 5GMSd populates the Network Assistance session resource with the service data flow information and optionally the policy template id that are valid for the streaming session for which Network Assistance operations are to be performed. The AF uses this information to execute Network Assistance operations in the 5GC.

The 5GMSd Client uses the `GET` method with the Network Assistance Session resource identifier to retrieve a Network Assistance Session resource from the 5GMS AF. The AF returns the Network Assistance Session resource if retrieval was successful, otherwise an appropriate error code is returned without the session resource in case of failure.

The 5GMSd Client uses the `GET` method with the sub-resource path specified in table 11.6.2-1 to request a bit rate recommendation from the 5GMS AF. The 5GMSd AF shall return the recommended bit rate in an HTTP response body of type *M5QoSSpecification* if a bit rate recommendation could be obtained, otherwise an appropriate HTTP error code shall be returned with no response body. The recommended minimum and maximum bit rates shall be indicated in the properties *mirBwDlBitRate* and *marBwDlBitRate* respectively. If a unique recommendation is given by the 5GMSd AF then this recommended bit rate shall be set in both of these properties. The optional properties *minDesBwDlBitRate*, *minDesBwUlBitRate*, *desLatency* and *desLoss* shall not be included in the response. The 5GMSd Client shall ignore the mandatory properties related to uplink streaming, i.e. *marBwUlBitRate* and *mirBwUlBitRate*.

The 5GMSd Client uses the `POST` method with the sub-resource path specified in table 11.6.2-1 to request a delivery boost from the 5GMSd AF. The 5GMSd AF shall respond with the *OperationSuccessResponse* data type indicating whether or not the delivery boost will be attempted by the network within an upcoming nominal time period.

The 5GMSd Client uses the `PUT` or `PATCH` methods to replace the existing steaming session parameters with new settings. The AF returns the *NetworkAssistanceSession* resource with settings resulting from the `PUT` or `PATCH` update operation.

The 5GMSd Client uses the `DELETE` method to terminate the indicated Network Assistance session. The 5GMS AF returns an appropriate response code. If the termination was successful then any subsequent calls referring to the terminated session will result in the error *404 (Not Found)*.

12 UE Media Session Handling (M6) APIs for uplink and downlink

12.1 General

This clause defines the client APIs for Media Session Handling to be used by other 5G System components such as a Media Player in a 5GMSd client or the Media Streamer in a 5GMSu client.

12.2 Media Session Handling for Downlink Streaming – APIs and Functions

12.2.1 Overview

In the following, it is assumed that the Media Session Handler for downlink streaming adheres to a basic set of functionalities as shown in Figure 12.2.1-1.

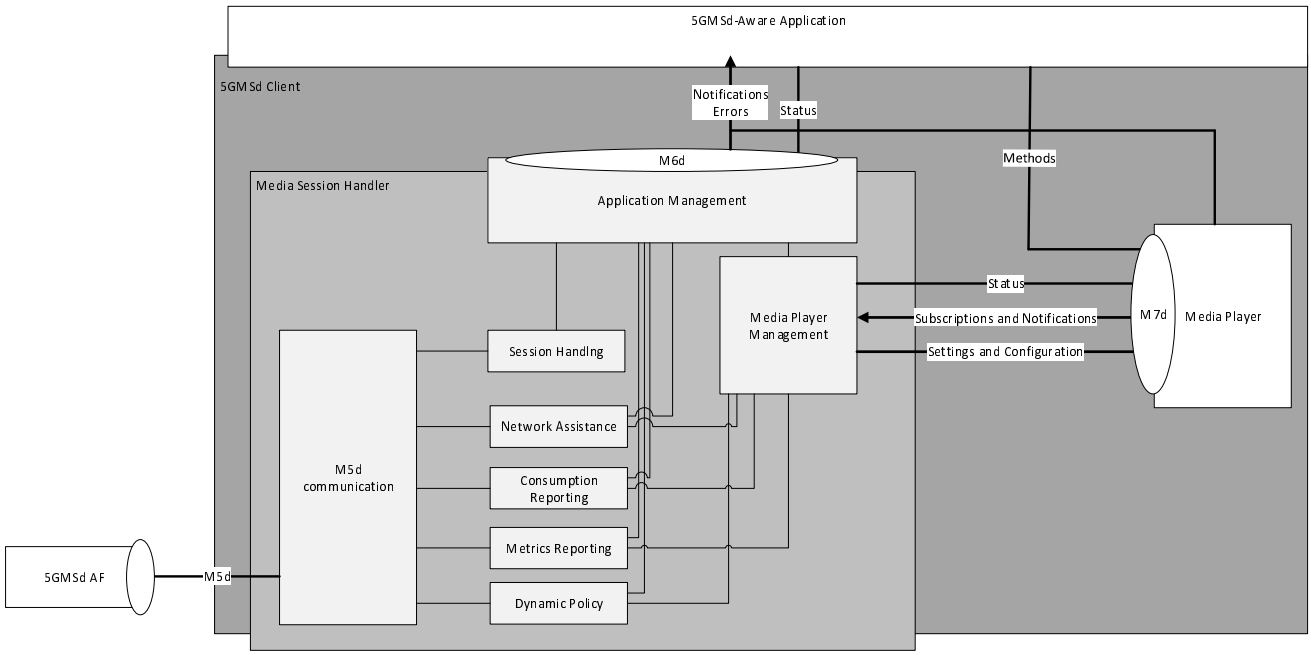

Figure 12.2.1-1: Usage of M6d in Media Downlink Streaming

The Media Session Handler is considered to run as a service in the background, and is invoked for a media session once a media player in the 5GMSd streaming client is activated with an MPD URL of media MIME type `"application/dash+xml"`. Based on the MPD URL, the Media Session Handler may initiate communication with the 5GMSd AF through M5d.

NOTE: The initiation of the Media Session Handler for other media types than DASH is for further study.

For an ongoing 5G Media Streaming session, the Media Session Handler is given the following authorities:

1) The ability to do status query on M7d. For details see clause 13.
2) The ability to process notifications and error on M7d. For details see clause 13.
3) The ability to configure certain parameters on the media player based on M7d. For details again see clause 13.

In addition, the MSH can provide information on M6d to the application and possibly delegated to Media Player using M6d for each of the Media Session Handler functionalities, namely providing:

1) Notification and Error Events;

2) Status Information.

12.2.2 Media Session Handler model

12.2.2.1 State model

An informative state model for the Media Session Handler is for further study.

12.2.2.2 Media Session Handler internal properties

The Media Session Handler maintains internal properties as defined Table 12.2.2.2-1. Note that the parameters are conceptual and internal and only serve for the purpose to describe message generation on the API calls.

Table 12.2.2.2-1: Parameters of Media Session Handler

States and Parameters	Definition
_Configuration	
_networkAssistance	Network Assistance configuration.
_policyTemplate	Policy Template configuration.
_consumptionReporting	Consumption reporting configuration.
_metricsReporting	Metrics reporting configuration.
_status[]	The Media Session Handler maintains a status record.

12.2.2.3 Media Session Handler internal operations

This aspect is for further study.

12.2.2.4 Starting and Stopping a Media Session Handler

There are different ways to start a Media Session Handler. The most typical one is that the start is bound to the call of a Media Player with an MPD URL. That start method offers a client–server like interface realized by M6d. The service is bound such that the Media Session Handler communicates back to the Media Player.

12.2.3 General

Table 12.2.3-1 provides a list status information that can be obtained from the Media Session Handler through M6d.

Table 12.2.3-1: Status Information

Status	Type	Parameter	Definition

Table 12.2.3-2 provides a list of general notification events exposed on M6d.

Table 12.2.3-2: General Notification Events

Event	Definition	Payload
SESSION_HANDLING_ACTIVATED	Triggered when media session handling was activated for a specific MPD URL.	
SESSION_HANDLING_STOPPED	Triggered when media session handling stopped for a specific MPD URL.	

Table 12.2.3-3 provides a list of general error events through M6d.

Table 12.2.3-3: General Error Events

Status	Definition	Payload
ERROR_SESSION_HANDLING	Triggered when there is an error in the media session handling.	Not applicable.

12.2.4 Dynamic Policy Information

Details are for further study.

12.2.5 Network Assistance Information

Details are for further study.

12.2.6 Consumption Reporting Information

Table 12.2.6-1 provides a list status information that can be obtained from the MSH through M6d.

Table 12.2.6-1: Status Information related to Consumption Reporting

Status	Type	Parameter	Definition
consumptionReport	`Object`		The latest sent consumption report.

Table 12.2.6-2 provides a list of general notification events exposed on M6d.

Table 12.2.6-2: Notification Events related to Consumption Reporting

Status	Definition	Payload
CONSUMPTION_REPORTING_ACTIVATED	Informs that consumption reporting has been activated.	Not applicable.
CONSUMPTION_REPORTING_STOPPED	Informs that consumption reporting has been stopped.	Not applicable.
NEW_CONSUMPTION_REPORT	Informs that a new consumption report is available and has been sent.	

Table 12.2.6-3 provides a list of general error events through M6d.

Table 12.2.6-3: Error Events to Consumption Reporting

Status	Definition	Payload
ERROR_CONSUMPTION_REPORTING	Error in consumption reporting occurred.	Not applicable.

12.2.7 Metrics Reporting Information

Details are for further study.

12.3 Media Session Handling for Uplink Streaming – APIs and Functions

Details are for further study.

13 UE Media Stream Handler (M7) APIs for uplink and downlink

13.1 General

This clause defines a set of APIs and methods that permit an application or other UE functions to communicate with a Media Player or Media Streamer. The main focus of this clause is to formalize and harmonize commonly available proprietary APIs in order to support the usage of a Media Player or a Media Streamer in a 5G Media Streaming context.

The APIs specified in this clause are language- and runtime-independent. Implementations are expected to provide language bindings appropriate to the UE runtime environment.

13.2 DASH Media Player – APIs and Functions

13.2.1 Overview

In the following, it is assumed that the Media Player (in this case a DASH client) adheres to a basic set of functionalities as shown in Figure 13.2-1. The DASH client downloads, processes and presents a DASH Media Presentation by instruction of a 5GMSd-Aware Application using the M7d interface.

The 5GMSd-Aware Application can, in addition, configure the presentation of the media, can receive notifications on events, or can query the internal status of the DASH Player, also supported through M7d. Different functions of the DASH Access Client that are typically necessary to process a DASH Media Presentation, are show in Figure 13.2-1. Additional functions may be available as well.

The key functionalities of each of the functions as shown in Figure 13.2-1 are summarized in the following:

- *5GMSd-Aware Application:* Application that makes use of the DASH/Media Player to playback a DASH Media Presentation using the APIs defined in this clause.
- *Media Player:* A complete player for the playback of a Media Presentation, including the Media Playback and Content Decryption Platform as defined in TS 26.511.
- *Access Client:* A part of the DASH Player that accesses and downloads of the resources and provides the downloaded resources to the Media Playback Platform and Content Decryption for the playback of DASH content.

- *Management:* Controls all internal processes and the communication with the 5GMSd-aware application. In particular this includes the handling of service descriptions and operation points.
- *MPD Processing:* parses and processes the MPD and extracts the relevant information.
- *Adaptation Set Selection:* selects the Adaptation Set based on user, application and/or device capability information. Information provided through M7d may be used.
- *ABR Controller and Dynamic Switching:* runs adaptive bit rate logic and triggers adaptive switching of Representations. Information provided to the DASH client through M7d may be used.
- *Throughput Estimation:* estimates the throughput from the 5GMSd Application Server.
- *Metrics Logging:* logs relevant low-level metrics and provides those to the metrics aggregation and reporting functions in the Media Session Handler.
- *Media Playback Management and Protection Controller:* manages the media playback by moving downloaded information into media playback platform and also addresses handling of protection and DRM related information.
- *Media Playback and Content Decryption Platform:* plays back CMAF-based media content according to the playback requirements in TS 26.511. It also provides status information as well as events that maybe be provided through M7d.
- *Event Processing:* Processes DASH events and provides information to application as defined in TS 26.247 [4].

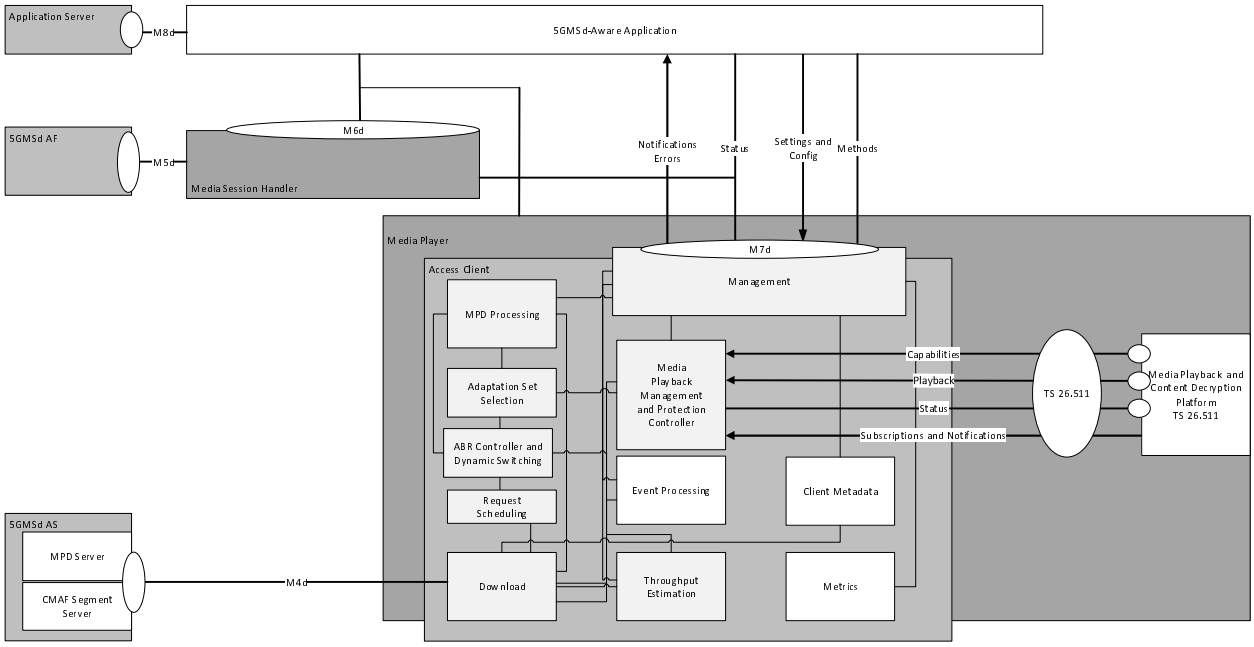

Figure 13.2.1-1: DASH Client Architecture

This clause focuses on Media Player related communication through M7d. In particular, the following aspects of M7d are defined:

1) Methods to interact with the Media Player are defined in clause 13.2.3.
2) Notification and Error Events are defined in clause 13.2.4.
3) Configuration and Settings APIs are defined in clause 13.2.5.
4) Status Information API is defined in clause 13.2.6.

The communication to the media playback platform is defined through the details in TS 26.511 [35].

A 5GMSd client for DASH distribution shall support the APIs defined in this clause 13.

NOTE: The initial APIs have largely been designed based on the dash.js APIs documented here: http://cdn.dashjs.org/latest/jsdoc.

13.2.2 Media Player model

Figure 13.2.2-1 provides an informative client state model in order to appropriately describe the messages on the Media streaming service API. Six different states are defined.

State changes may happen based on:

- Calls from application.
- Information provided in the Media Presentation Description (MPD).

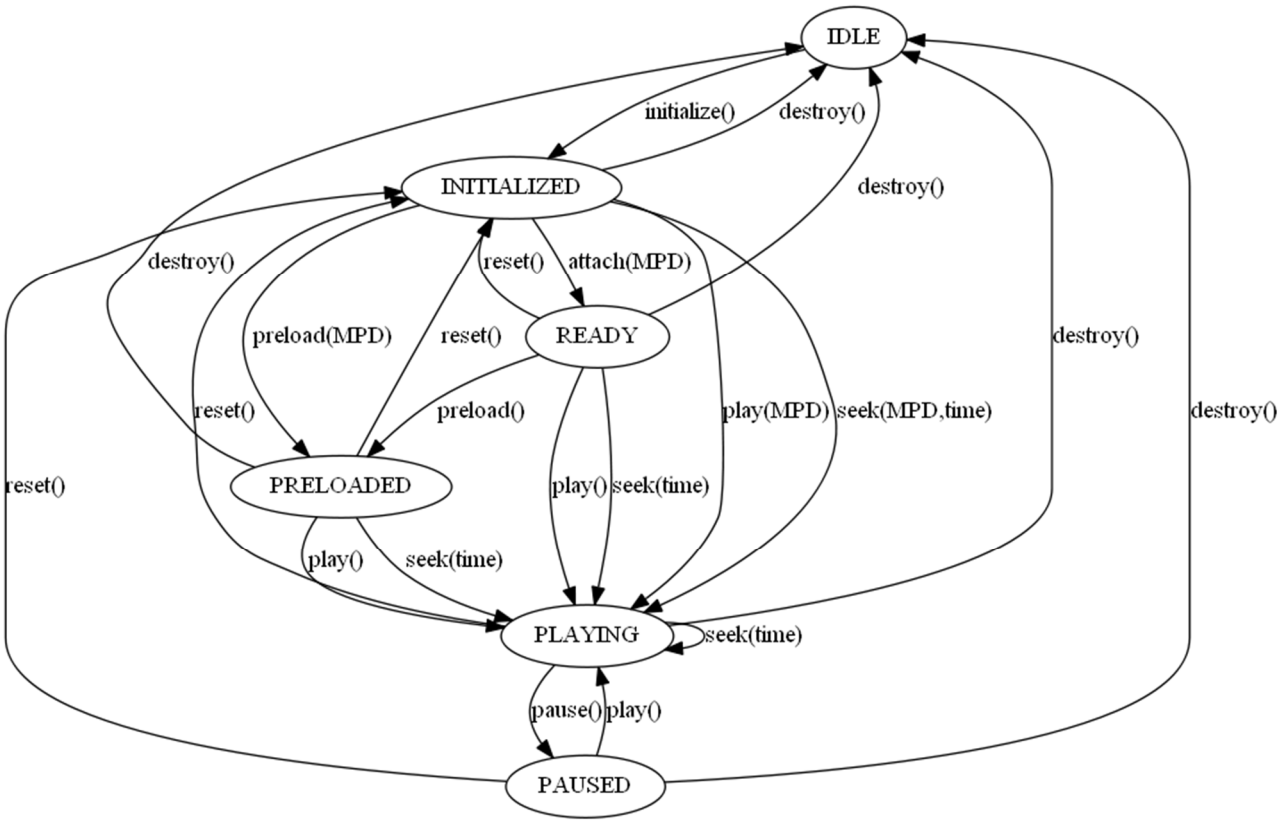

Figure 13.2.2-1: State Diagram for Media Player

Table 13.2.2-1 defines states for the Media Player. Detailed descriptions are provided in the following subclauses.

Table 13.2.2-1: States of Media Player

States	Definition
IDLE	The Media Player is not associated with any application.
INITIALIZED	The Media Player is associated with an application and the M7d API communication is established.
READY	The Media Player has loaded an MPD and is able to playback the media in this Media Presentation. It also updates the MPD according to the MPD update mechanism.
PRELOADED	The Media Player has pre-loaded all media information in order to start playback instantaneously. It also updates the MPD according to the MPD update mechanism.
PLAYING	The Media Player is playing the Media Presentation. It also updates the MPD according to the MPD update mechanism.

States	Definition
PAUSED	The playback of the Media Presentation is paused. It also updates the MPD according to the MPD update mechanism.

It is assumed that the DASH Access Client manages the playback of at most one CMAF track for each media type, namely one for video, one for audio and one for subtitles as defined in TS 26.511 [35]. Playback of multiple CMAF tracks of the same media type is not excluded for 5GMS, but details is for further study.

13.2.3 Methods

13.2.3.1 General

Based on the state model in clause 13.2.2, this clause introduces relevant procedures and API calls.

Table 13.2.3.1-1 provides an overview over the methods defined for the DASH-based streaming API. Note that in implementations, additional methods may be supported.

Table 13.2.3.1-1: Methods defined for DASH Streaming API

Method	State after success	Brief description	Clause
`initialize()`	*INITIALIZED*	The Media Player is created.	13.2.3.2
`attach(MPD)`	*READY*	sets a source URL to an MPD file or a previously downloaded and parsed MPD.	13.2.3.3
`preload(MPD)`	*PRELOADED*	Streaming the media is initiated.	13.2.3.4
`play(MPD)`	*PLAYING*	Playback of the media is initiated.	13.2.3.5
`pause()`	*PAUSED*	Playback of the media is paused.	13.2.3.6
`seek(MPD, time)`	*PLAYING*	The playback time of the media is altered.	13.2.3.7
`reset()`	*INITIALIZED*	All media related information is reset.	13.2.3.8
`destroy()`	*IDLE*	All media player related information is reset and API communication is stopped.	13.2.3.9

13.2.3.2 Initialize

This clause defines the `initialize()` method.

The Media Player is created by initializing using the `initialize()` method. The following functions are initialized:

- Media Playback Management in order to enable API-based communication through M7d. In particular, the *M7d Notifications and Errors API* (see clause 13.2.4) and the *Status Query* (see clause 13.2.5) are established.

13.2.3.3 Attach

This clause defines the `attach()` method.

The following pre-conditions apply:

- The MediaPlayer is be in *INITIALIZED* state.

An 5GMSd-Aware Application calls `attachMPD()` to set a source URL to an MPD file or a previously downloaded and parsed MPD.

The parameters of the method are defined in Table 13.2.3.3-1.

Table 13.2.3.3-1: Parameters for `attachMPD()`

Name	Type	Description
urlOrMPD	`string | Object`	A URL to a valid MPD or a valid MPD as defined in ISO/IEC 23009-1 [32] or TS 26.247 [4]. The URL may be augmented by MPD Anchors as defined in ISO/IEC 23009-1 [32], Annex C.4.

The following Media Player Actions are expected:

- The *Request Scheduling* and *Download* functions are established.
- If the input is a URL, the Media Player requests the MPD at the corresponding URL through M4d.
- If the MPD is not found after multiple retries, an error *ERROR_MPD_NOT_FOUND* is returned and the process is terminated.
- The *MPD Processing* function is established and the MPD parsed.
- If the MPD is not valid, an error *ERROR_MPD_NOT_VALID* is returned and the process is terminated.
- If the DASH Player does not support the profiles as indicated in the MPD, an error *ERROR_PROFILE_NOT_SUPPORTED* is returned and the process is terminated.
- Depending on the type of the MPD, possibly present anchors as well as the wall-clock time, the Media Player selects the Period in the content that is expected to be played next.
- The *Media Playback Management and Protection Controller* is established.
- The MPD is parsed for available Service Descriptions (including Media Subsets and Adaptation Sets). By using capability mechanisms defined in TS 26.511 [35] as well as using other information (language settings, output capabilities, accessibility settings), the Media Player identifies a set of permissible Service Descriptions including Media Subsets and Adaptation Sets. If no Adaptation Sets are capable to be played, an error *ERROR_MEDIA_NOT_SUPPORTED* is returned and the process is terminated.
- The available Service Descriptions including included Adaptation Sets are provided to the application through M7d.
- The application may select a Service Description instance as well as Adaptation Sets. Additional Service Descriptions parameters may be configured through M7d.
- Based on the service description parameters and selected Adaptation Sets:
  - the Operation Point parameters are set.
  - the *Media Playback Platform and Content Decryption* is established using the methods defined in TS 26.511.
  - The selected Adaptation Sets are initialized by downloading the relevant Initialization Segments/CMAF Headers through M4d in the Media Playback Platform as in TS 26.511 [35] establishing a track buffer for each selected media type.
- Depending on the MPD information and/or M7d configuration, one or more of the following functions may be established:
  - Metrics Logging and Collection
  - Event Processing and Notification
  - Client Metadata handling
- The Media Player is left in the *READY* state.

An application may use this method to load an MPD and in order to prepare playback. In case of errors notifications, it is up to the application to initiate appropriate actions.

13.2.3.4 Pre-load

This clause defines the `preload()` method.

The following pre-conditions apply:

- The MediaPlayer is in *INITIALIZED* or *READY* state.

An 5GMSd-Aware Application calls `preload()` to cause the player to begin streaming the media as set by the `attach()` method in preparation for playing.

The parameters of the method are defined in Table 13.2.3.4-1.

Table 13.2.3.4-1: Parameters for `attachSource()`

Name	Type	Description
urlOrMPD	`string | Object`	A URL to a valid MPD or a valid MPD as defined in ISO/IEC 23009-1 [32] or TS 26.247 [4]. The URL may be augmented by MPD Anchors as defined in ISO/IEC 23009-1 [32], Annex C.4.

The following Media Player Actions are expected:

- If in *INITIALIZED* state, the `attach()` method is invoked.
- Depending on the type of the MPD, possibly present anchors as well as the wall-clock time, and other MPD information, the earliest media time span for pre-loading is identified.
- The Access Client schedules and generates requests for the relevant media segments based on the ABR Controller information, as well as the throughput estimation and downloads this media.
- The Segments are downloaded from the corresponding URLs through M4d earliest at the segment availability start time of the Segments.
- The Segments ate appropriately appended to the track buffers as established according to *Media Playback Platform and Content Decryption* APIs, following the description in TS 26.511 [35] for playback requirements.
- Configuration and service description parameters are taking into account, for example the content is continuously loaded to remain at the live edge following the latency requirements provided in the service description setting. Content not at the live edge is removed. For static services, the content is loaded from the beginning up to a suitable buffer duration, possibly as configured, and then downloading is stopped.
- Appropriate notifications and error messages are generated. For details refer to clause 13.2.5.
- Appropriate Status Information is generated. For details refer to clause 13.2.6.
- The Media Player is in *PRELOADED* state.

An application may use this method to preload media into the player in order minimize the start-up time.

13.2.3.5 Play

This clause defines the `play()` method.

The following pre-conditions apply:

- The MediaPlayer is in *INITIALIZED* or *READY* or *PRELOADED* or *PAUSED* state.

An 5GMSd-Aware Application calls `play()` to cause the player to begin playback of the media as set by the `attach()` method.

The parameters of the method are defined in Table 13.2.3.5-1.

Table 13.2.3.5-1: Parameters for `play()`

Name	Type	Description
urlOrMPD	`string | Object`	A URL to a valid MPD or a valid MPD as defined in ISO/IEC 23009-1 [32] or TS 26.247 [4]. The URL may be augmented by MPD Anchors as defined in ISO/IEC 23009-1 [32], Annex C.4.

The following Media Player Actions are expected:

- If in *INITIALIZED* state, the `attach()` method is invoked.
- If in *PAUSED* state, the earliest media time is *MEDIA_TIME* (for details see clause 13.2.3.6), else, depending on the type of the MPD, possibly present anchors as well as the wall-clock time, and other MPD information, the earliest media time for start-up is identified.
- The Access Client checks the available buffer state of media in the Media Playback Platform. Based on this, the Access Client schedules and generates requests for the relevant media segments based on the ABR Controller information, as well as the throughput estimation and downloads this media.
- The Segments are downloaded from the corresponding URLs through M4d earliest at the segment availability start times.
- The media is appropriately appended to the *Media Playback Platform and Content Decryption* APIs, following the description in TS 26.511 [35] for playback requirements.
- Once a threshold for sufficient buffering is reached, the Media Playback platform is initiated to be started, i.e. a playback is initiated, following the description in TS26.511 [35] for playback requirements.
- The content is continuously streamed, downloaded and played back.
- Appropriate notifications and error messages are generated. For details refer to clause 13.2.4.
- Appropriate Status Information is generated. For details refer to clause 13.2.5.
- The Media Player is in *PLAYING* state.

An application may use this method to initiate playback of media.

13.2.3.6 Pause

This clause defines `pause()` method.

The following pre-conditions apply:

- The Media Player is in *PLAYING* state.

An 5GMSd-Aware Application calls `pause()` to cause the Media Playback Platform to pause playback.

No parameters are attached.

The following Media Player Actions are expected:

- The playback on the playback platform is paused and the media time is maintained as *MEDIA_TIME*.
- The Access Client checks the available buffer state of media in the Media Playback Platform. Based on this, the Access Client schedules and generates requests for the relevant media segments based on the ABR Controller information, as well as the throughput estimation and downloads this media.
- The media is downloaded from the corresponding URL through M4d earliest at the segment availability start time of the media.
- The media is appropriately appended to the *Media Playback Platform and Content Decryption* APIs, following the description in TS 26.511 [35] for playback requirements.
- Once the buffers are sufficiently filled, the client stops downloading.

- Appropriate notifications and error messages are generated. For details refer to clause 13.2.4.
- Appropriate Status Information is generated. For details refer to clause 13.2.5.
- The Media Player is in *PAUSED* state.

An application may use this method to play back media.

13.2.3.7 Seek

This clause defines `seek()` method.

The following pre-conditions apply:

- The MediaPlayer is in *INITIALIZED*, *READY*, *PRELOADED* or *PAUSED* state.

An 5GMSd-Aware Application calls `seek()` to cause the player to go a specific media time.

The parameters of the method are defined in Table 13.2.3.7-1.

Table 13.2.3.7-1: Parameters for `seek()`

Name	Type	Description
urlOrMPD	`string | Object`	A URL to a valid MPD or a valid MPD. The URL may be augmented by MPD Anchors as defined in ISO/IEC 23009-1 [32], Annex C.4.
mediaTime	`Unsigned integer`	The media time in milliseconds for playback.

The following Media Player Actions are expected:

- If in *INITIALIZED* state, the `attach()` method is carried out.
- If the `mediaTime` is not accessible return an error *ERROR_MEDIA_TIME_NOT_ACCESSIBLE* and terminate the process.
- The earliest media time is set to the `mediaTime`.
- The state is set to *PAUSED*.
- The `play()` command is issued.

An application may use this method to initiate playback of media.

13.2.3.8 Reset

This clause defines the `reset()` method.

The following pre-conditions apply:

- The Media Player may be in any state.

An 5GMSd-Aware Application calls `reset()`resets all information related to the media and the Media Presentation described by the MPD is destroyed.

No parameters are attached.

The following Media Player Actions are expected:

- The playback on the playback platform terminated.
- All open requests are cancelled.
- All scheduled requests are deleted.
- The current MPD is removed.

- The Media Player is left in the *INITIALIZED* state.

An application may use this method to terminate the playback of any media.

13.2.3.9 Destroy

This clause defines `destroy()` method.

The following pre-conditions apply:

- The Media Player may be in any state.

An 5GMSd-Aware Application calls `destroy()`resets all information related to the media and the network.

No parameters are attached.

The following Media Player Actions are expected:

- The playback on the playback platform terminated.
- All open requests are cancelled.
- All scheduled requests are deleted.
- The current MPD is removed.
- All network information is history is cleared.
- The Media Player is left in the *IDLE* state.

An application may use this method to terminate the playback of any media clear and download related information.

13.2.4 Configurations and settings API

DASH streaming may be configured with the parameters provided in Table 13.2.4-1. Note that these parameters may be set and they may also be observed.

Table 13.2.4-1: Configuration API

Status	Type	Definition
source	`Object`	Provides the MPD and all contained information.
consumptionMode	`Enum`	Defines two modes: *live*: in this case the target latency is maintained, if specified in the service description, according to the parameters *vod*: in this case the latency is set by the application and the latency settings are ignored.
maxBufferTime	`Integer`	Maximum buffer time in milliseconds for the service.
serviceDescriptionId	`id`	Selects a service description by selecting an identifier.
serviceDescriptions[]	`Service description parameters`	Configures a service description as defined in ISO/IEC 23009-1 [32], Annex K. This allows the application to define additional service descriptions beyond those defined in the MPD.
id	`id`	Sets a service description identifier different from the ones available in the service descriptions in the MPD or modifies existing service descriptions.
serviceLatency	`Object`	Sets service description parameters for the service latency, as defined in ISO/IEC 23009-1 [32], Table K.1.
playBackRate	`Object`	Sets service description parameters for the playback rate, as defined in ISO/IEC 23009-1 [32], Table K.2 when the service is consumed in live mode.
operatingQuality	`Object`	Sets service description parameters for the operating quality, as defined in ISO/IEC 23009-1 [32], Table K.3.
operatingBandwidth	`Object`	Sets service description parameters for the operating bandwidth, as defined in ISO/IEC 23009-1 [32], Table K.4.
mediaSettings[]	Media type `audio`, `video`, `subtitle`	Sets the selected Adaptation Set based on the available Adaptation Sets for each media type.
metricsConfiguration[]	`Object`	Defines the setting for collecting metrics.

13.2.5 Notifications and error events

Table 13.2.5-1 provides a list of notification events that are provided by the Media Player.

Table 13.2.5-1: Notification events

Status	Definition	Payload
AST_IN_FUTURE	Triggered when playback will not start yet as the MPD's availabilityStartTime is in the future.	Time before playback will start.
AVAILABLE_MEDIA_CHANGED	The list of available media has changed.	Media type: video, audio, subtitle, all
BUFFER_EMPTY	Triggered when the media playback platform's buffer state changes to stalled.	Media Type
BUFFER_LOADED	Triggered when the media playback platform's buffer state changes to loaded.	Media Type
CAN_PLAY	Sent when enough data is available that the media can be played.	Not applicable.
MANIFEST_LOADED	Triggered when the manifest load is complete	Not applicable.
METRIC_ADDED	Triggered every time a new metric is added.	
METRIC_CHANGED	The minimum bit rate that the ABR algorithms will choose. Use NaN for no limit.	
METRIC_UPDATED	Set to true if you would like DASH Client to keep downloading fragments in the background when the video element is paused.	
METRICS_CHANGED	Triggered whenever there is a change to the overall metrics.	
OPERATION_POINT_CHANGED	Triggered whenever there is a change of an operation point parameter.	
PLAYBACK_ENDED	Sent when playback completes.	
PLAYBACK_ERROR	Sent when an error occurs. The element's error attribute contains more information.	Error attribute.
PLAYBACK_PAUSED	Sent when playback is paused.	
PLAYBACK_PLAYING	Sent when the media begins to play (either for the first time, after having been paused, or after ending and then restarting).	
PLAYBACK_SEEKED	Sent when a seek operation completes.	
PLAYBACK_SEEKING	Sent when a seek operation begins.	
PLAYBACK_STALLED	Sent when the media playback platform reports stalled	
PLAYBACK_STARTED	Sent when playback of the media starts after having been paused; that is, when playback is resumed after a prior pause event.	
PLAYBACK_WAITING	Sent when the media playback has stopped because of a temporary lack of data.	
SERVICE_DESCRIPTION_SELECTED	sent when the DASH client has selected a service description.	
SERVICE_DESCRIPTION_CHANGED	Sent when the DASH client has changed a service description.	
SERVICE_DESCRIPTION_VIOLATED	Provides notification that the service description parameters are currently not met.	Parameters of service description that are not met.
SOURCE_INITIALIZED	Triggered when the source is setup and ready.	

Table 13.2.5-2 provides a list of error events.

Table 13.2.5-2: Error events

Status	Definition	Payload
ERROR_MPD_NOT_FOUND	Triggered when the MPD is not found.	
ERROR_MEDIA_PLAYBACK	Triggered when there is an error from the media playback platform buffer.	
ERROR_MPD_NOT_VALID	The provided MPD is not valid according to the XML schema and schematron rules.	Detailed error information.
ERROR_MEDIA_TIME_NOT_ACCESSIBLE	After seek operation, the media time is not accessible.	
ERROR_PROFILE_NOT_SUPPORTED	The profile of the Media Presentation is not supported.	

13.2.6 Status Information

Table 13.2.6-1 provides a list of dynamically changing status information that can be obtained from the client.

Table 13.2.6-1: Dynamic Status information

Status	Type	Parameter	Definition
AverageThroughput	`float`	none	Current average throughput computed in the ABR logic in bit/s.
BufferLength	`float`	`MediaType` "video", "audio" and "subtitle"	Current length of the buffer for a given media type, in seconds. If no type is passed in, then the minimum of video, audio and subtitle buffer length is returned. NaN is returned if an invalid type is requested, the presentation does not contain that type, or if no arguments are passed and the presentation does not include any adaption sets of valid media type.
liveLatency	`float`	none	Current live stream latency in seconds based on the latency measurement.
MediaSetting[]	`MPDAdaptationSet`	`MediaType` "video", "audio" and "subtitle"	Current media settings for each media type based on the CMAF Header and the MPD information based on the selected Adaptation Set for this media type.
MediaTime	`float`	None	Current media playback time from media playback platform. The media time is in seconds and is relative to the start of the playback and provides the media that is actually rendered.
PlaybackRate	`float`	None	The current rate of playback. For a video that is playing twice as fast as the default playback, the *playbackRate* value should be 2.00.
availableServiceDescriptions[]	Provides the available service descriptions		Provides the list of available selectable service descriptions with an id to select from. Those are either configured ones or the ones in the MPD.
availableMediaOptions[]	List of Adaptation Set or Preselection ids	`MediaType` "video", "audio" "subtitle" "all"	Provides the list of available media options that can be selected by the application based on the capability discovery and the subset information.
Metrics[][]	`Metrics`		A data blob of metrics for each defined metrics collecting scheme.

Table 13.2.6-2 provides a list of configured operation point information that can be obtained from the client. Any change to a parameter below shall be announced with a notification *OPERATION_POINT_CHANGED*.

Table 13.2.6-2: Operation Point Information

OperationPoint	Operation Point Parameters	The currently configured operation point parameters according to which the DASH client is operating.
mode	`Enum`	The following operation modes are defined: *live*: The DASH client operates to maintain configured target latencies using playback rate adjustments and possibly resync. *vod*: The DASH client operates without latency requirements and rebuffering may result in additional latencies
maxBufferTime	`Integer`	maximum buffer time in milliseconds for the service.
switchBufferTime	`Integer`	buffer time threshold below which the DASH clients attempts to switch Representations.
Latency		Defines the latency parameters used by the DASH client when operating in live mode.
target	`Integer`	The target latency for the service in milliseconds.
max	`Integer`	The maximum latency for the service in milliseconds.
min	`Integer`	The maximum latency for the service in milliseconds.
PlaybackRate	`MediaType` audio, video, all	Defines the playback rate parameters used by the DASH client for catchup mode and deceleration to avoid buffer underruns and maintaining target latencies.
max	`Real`	The maximum playback rate for the purposes of automatically adjusting playback latency and buffer occupancy during normal playback, where 1.0 is normal playback speed.
min	`Real`	The minimum playback rate for the purposes of automatically adjusting playback latency and buffer occupancy during normal playback, where 1.0 is normal playback speed.
Bandwidth		Defines the operating bandwidth parameters used by the DASH client used for a specific media type or aggregated. The values are on IP level.
target	`Integer`	The target bandwidth for the service in bit/s that the client is configured to consume.
max	`Integer`	The maximum bandwidth for the service in bit/s that the client is configured to consume.
min	`Integer`	The minimum bandwidth for the service in bit/s that the client is configured to consume.
PlayerSpecificParameters		Player specific parameters may be provided, for example about the used algorithm, etc.

13.2.7 Usage of M7d Information by Media Session Handler

The media session handler may use the notifications, errors and status information provided through M7d to execute relevant tasks.

14 Application (M8) APIs for uplink and downlink

APIs of this reference point are not specified within this release.

15 Miscellaneous UE-internal APIs

15.1 General

While the core functionality of 5GMS is specified in terms of the dedicated system interfaces and APIs that impact the UE, specified in clauses 10 to 14 (M4 to M8 respectively), certain features of 5GMS rely on interfaces and APIs that are essentially UE-internal.

Each usage of a UE-internal interface is specified in subsequent sub-clauses of the present clause.

15.2 RAN Signaling-based Network Assistance API

If RAN Signaling-based Network Assistance is supported, the Media Session Handler uses an interface to the RAN Modem (specifically, the UE MAC entity in the modem) to send and receive bit rate recommendation messages. The interface to the modem may be based on the AT commands *+CGBRRREQ* and *+CGBRRREP* as defined in [15].

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 [13] for 5G NR and [14] 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 *+C5GQOSRDP=?* command may be used to get a list of CID values that are associated with QoS flows (both network and MT/TE initiated).When used for requesting 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 dropping.

The *Bit Rate Recommendation Query* shall indicate the bit rate desired by the application, as described by [13] and [14]. This request may be used by the 5GMSd Media Session Handler to request for a temporary increase in bit rate for the corresponding flow (bit rate 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 RAN Modem, corresponding to the logical translation of the *Bit Rate Recommendation* message for the associated RAN uplink or downlink, it shall indicate the associated bit rate recommendation to either the Media Player (via M7d, in the case of downlink streaming) or Media Streamer (via M7u, in the case of uplink streaming) function of an affiliated PDU session. Furthermore, whenever the Media Session Handler receives a request for a bit rate boost from either the Media Player (via M6d in the case of downlink streaming) or the Media Streamer (via M6u, in the case of uplink streaming) function of an affiliated PDU session, it may send a bit rate boost message to the RAN Modem. That bit rate 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 player to decide how to best use the bit rate recommendation and the bit rate recommendation query information for the media streaming sessions.

15.3 RAN-based Metrics Reporting API

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 channel.

The Media Session Handler shall subscribe to metrics configurations from the OAM according to TS 26.247 Annex L.1. When a metrics configuration is received, the Media Session Handler shall store this configuration and use it for all subsequent streaming sessions.

When a streaming 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* and *streaming source filter* specified in the stored metrics configuration, according to TS 26.247 Annex F.

If metrics are to be reported for the session, the Media Session Handler shall request the Media Player to create a metrics collection job. The Media Player 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 that shall be collected during the session. The format of the configuration shall be according to TS 26.247 Annex L.2, but note that 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 Player according to the *reporting interval* specified in the metrics configuration. The metrics returned by the Media Player shall use the format

as described in TS 26.247 clause 10.6, and the Media Session Handler shall forward these to the OAM according to TS 26.247 Annex L.1.

When the session is finished the Media Session Handler shall delete the metrics collection job.

16 Usage of 5GC interfaces and APIs

16.1 General

While the core functionality of 5GMS is specified in terms of the dedicated system interfaces and APIs specified in clauses 7 to 14 (for M1 to M8 respectively), certain features of 5GMS rely on interfaces and APIs defined within the scope of the 5GC.

Each such case of usage of a 5GC interface and API is documented in subsequent sub-clauses of the present clause.

NOTE: The 5GMS architecture may be applied to an EPS although such an application is not specified in the present document and is left to the discretion of deployments and implementations.

16.2 Usage of N5/N33 for AF-based Network Assistance

The feature of AF-based Network Assistance operates within interface M5 between the UE and an AF that provides Network Assistance capabilities, as defined in clause 11.6. The Network Assistance protocol and API within M5 is defined in a generic way so that the associated Network Assistance functionality in the 5GC may be realised by various means.

In the present specification the 5GMS AF converts the Network Assistance API calls and responses carried in interface M5 into API calls to the Session Management Policy Control Service, as specified in TS 29.514 [34].

If the Network Assistance feature is supported, then the 5GMS AF shall offer the bitrate recommendation and delivery boost request API based on existing policy templates that match the filtering criteria for a media streaming session, through the usage of either the `Npcf_PolicyAuthorization` API over N5, or the `Nnef_AFSessionWithQoS` over N33 interface to the PCF.

When serving a media streaming session that belongs to the AF application session context, the AF shall subscribe to the following PCF notifications with the PCF:

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

If no corresponding AF application session context already exists, the AF shall use the `Npcf_PolicyAuthorization_Create` method with the appropriate service information to create and provision an application session context. The information in the *AppSessionContextReqData* shall be derived from the policy template.

When requesting QoS provisioning for a media streaming session, the 5GMS AF shall use the configured policy templates of the Provisioning Session to determine the list of the QoS references within the "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 requested operation point by the UE (if the UE is allowed to use that operation point).

Media streaming sessions shall use exactly one component per session. It is assumed that a single sub-component is used, unless otherwise indicated.

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

Annex A (informative): 5GMS Parameter propagation for DASH Streaming

A.1 End-to-end model

Figure A.1-1 below depicts an end-to-end model for the 5GMS parameter propagation for DASH streaming with dynamic policy. The arrows indicate the main information flow. The interfaces specified in TS 26.501 [2] are used throughout. However, there are additional interfaces (i.e. P1 or U1), which are not in the 5GMS Architecture.

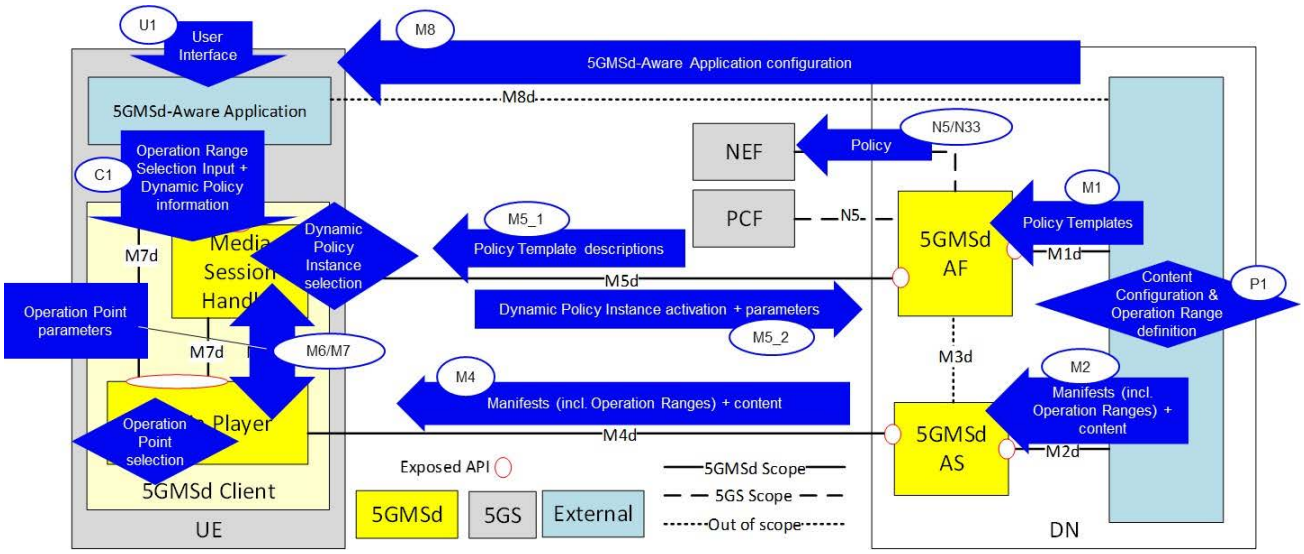

Figure A.1-1: End-to-end model for dynamic policy parameter propagation

The interfaces involved and their roles in this feature are as follows:

- M1: Provisioning interface between the 5GMS Application Provider and the 5GMS AF.
- P1: The 5GMS Application Provider provisions the DASH MPD generator, e.g. by annotating the MPD with Service Descriptions.
- U1: User Interface to the 5GMS-Aware Application.

NOTE: The 5GMS Application Provider controls the application, i.e. controls the GUI choices.

- M8: Non-standardized input from the 5GMS Application Provider to the 5GMS-Aware Application, such as country-specific application behaviours (languages, on-demand catalogue, etc).
  - Input on subscriptions (e.g. 4K subscription versus SD subscription).
  - Device-specific content selection rules (e.g. SmartPhone versus Smart TV).
  - Additional service offering features (e.g. background download possible).
- C1 (one of M6 or M7): Information from the 5GMS-Aware Application to the 5GMS Client, e.g. user content selections.
- M6: Information flow from the DASH Player to the Media Session Handler.
- M7: Information flow from the Media Session Handler to the DASH Player.
- M5_1: Information flow into the Media Session Handler for parameter provisioning (Policy Descriptions, which originate from 5GMS AF and 5GMS Application Provider). The Policy Descriptions contain or reference the detailed Service Access Information, i.e. URLs to activate a certain policy.

- M5_2: Information flow from the Media Session Handler to the 5GMS AF. This includes:
  - input to create the Service Data Flow Templates (see TS 23.503 [33]) for identifying the application data flows within a PDU Session,
  - an identifier for the Dynamic Policy instance (e.g. QoS, Conditional Zero-rating, charging, etc) and
  - optionally, Network Assistance information, e.g. bit rate recommendations.

In its Annex K, the DASH standard [32] specifies so-called "Service Descriptions". The purpose of Service Descriptions is to provide additional information to a DASH player to influence its "Selection Logic", e.g. a DASH player should prefer a certain set of representations within an adaptation set. It is assumed in the following that the DASH MPD can be annotated using Service Descriptions to give hints for subscription models and different device types.

The 5G System specifies a number of different means to detect application flows. When activating a Dynamic Policy, the Media Session Handler provides a Service Data Flow Template to the 5GMS System, which identifies the application flow(s) of interest. It is assumed here that multiple applications are executing simultaneously on a given UE and that each application may independently access the network. Therefore, the Media Session Handler needs to provide (and update) these Service Data Flow Templates in order that the application traffic can be treated according to the corresponding Dynamic Policy.

In the following clauses, the parameter propagation for a number of different use cases is described.

A.2 Premium QoS dynamic policy

A.2.1 General

To realise a Premium QoS service offering, the 5GMS Client should activate a QoS Flow with characteristics matching the service needs. It is assumed that the DASH content is prepared for different subscription levels, e.g. 4K, HDR or SD, and for different target device types, e.g. SmartPhone or SmartTV. When commencing playback of a DASH presentation according to a particular subscription level (e.g. 4K), the 5GMS Client needs to activate a QoS Flow with a matching bit rate setting.

NOTE: The 5GMS Client may choose to activate a QoS Flow with a lower bit rate than the maximum supported by the 5G System, e.g. a small screen SmartPhone may select different QoS settings from a large screen device.

The per-title quality and the subscription levels of an example on-demand catalogue are illustrated in the figure below. The subscription levels in this example are 4K, FullHD, HD, SD and 480p. Only devices entitled to activate a 4K quality should actually select the according representations from the MPDs. In this example, all titles are available in SD and HD quality. Often, not all titles are available in 4K quality. Thus, a device with a 4K subscription can only activate reception of the HD or SD representations.

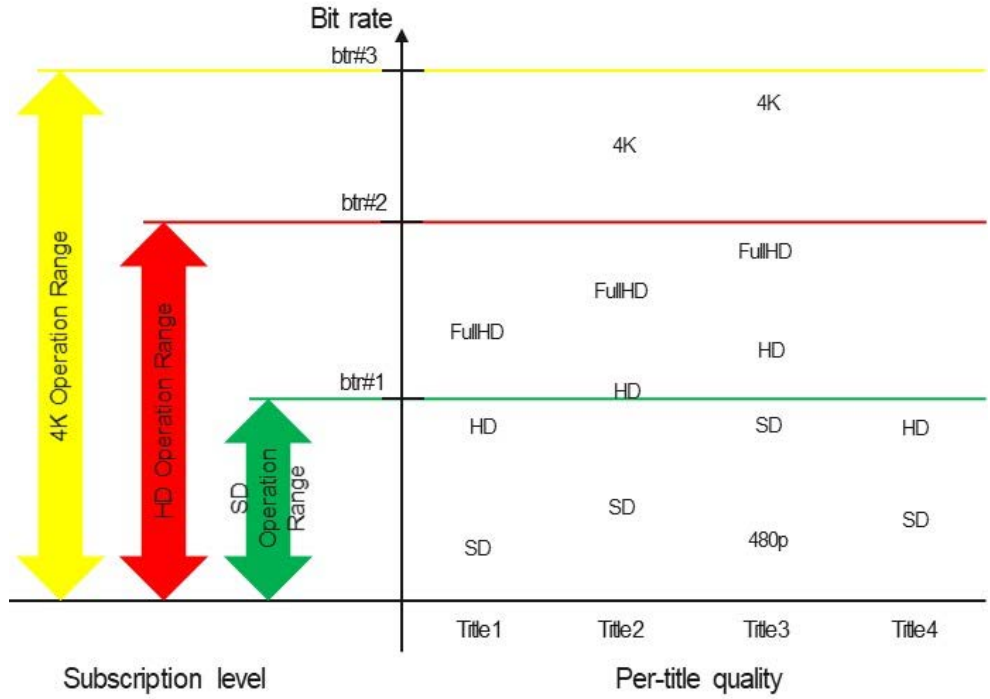

Figure A.2.1-1: Subscription Levels for Premium QoS

The bit rate required to sustain a certain quality varies from title to title. In the figure, the bit rate needed for *Title4* in HD is in the same range as SD quality of *Title3*.

The various consumer-facing Network Subscription Levels define a set of bounded Operation Ranges, as illustrated on the right side of the figure. Each such Operation Range is conveniently modelled in the 5GMS architecture as a Policy Template. The Policy Template for SD subscription level (*SD Operation Range*) is authorized to activate a maximal bit rate of *btr#1*. The Policy Template for 4K subscription level is authorized to activate between any low bit rate and a maximal bit rate of *btr#3*.

When activating a Dynamic Policy instance, the 5GMSd Client provides a desired bit rate for the selected title. The desired bit rate can be smaller than the maximal bit rate allowed by the Policy Template. The 5GMSd Client always activates a Dynamic Policy instance from its assigned Network Subscription Level, even when the desired bit rate justifies a different Policy Template.

When activating a QoS Flow for a certain subscription level and title, the 5GMSd Client should preferably select a desired bit rate matching the quality needed. For example, a device with an *HD Operation Range* subscription needs a higher desired bit rate when consuming *Title3* in HD quality and a lower desired bit rate when consuming *Title4* in HD quality.

In some cases, the system rejects a requested QoS Flow or drops an established QoS Flow due to insufficient available network resource. The 5GMSd Client can then try to activate a different QoS Flow with a lower desired bit rate.

A.2.2 Procedure

The procedure for activating a Premium Qos dynamic policy is illustrated in figure A.2.2-1 below.

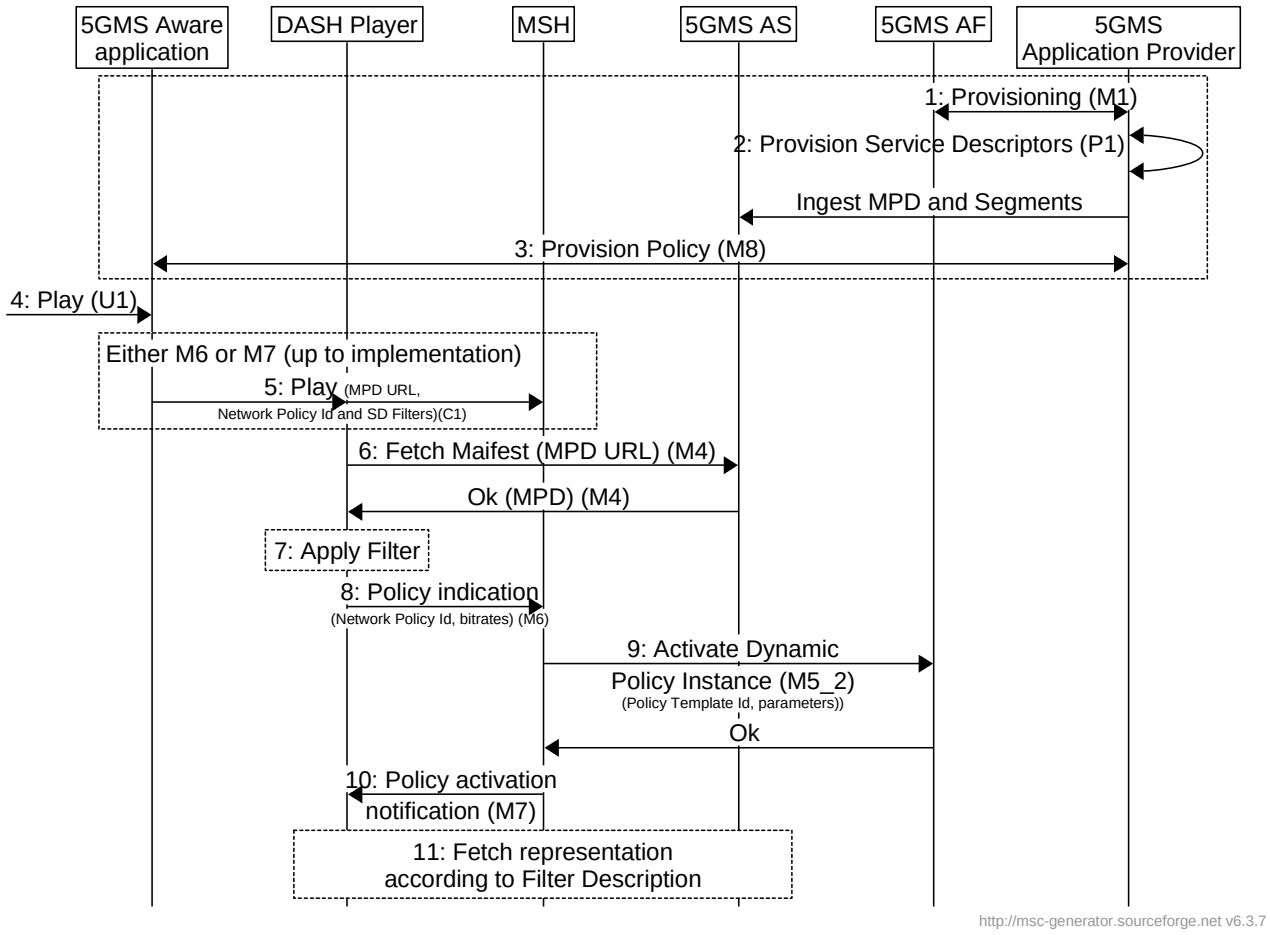

Figure A.2.2-1: Procedure for activating Premium QoS dynamic policy

Steps:

1. The 5GMS Application Provider interacts with the 5GMS AF to set up one or more Policy Templates (using M1). Each Policy Template is identified by a Policy Template identifier and contains information about how to activate the corresponding policy within the 5G System (e.g. N5 URLs and parameters).

2. The 5GMS Application Provider interacts with its DASH content generation function (e.g. an MPD provider) to annotate the DASH MPD with Service Descriptions (using P1). The Service Descriptions define the Operational Ranges within the Media Player should operate. The DASH MPD and the DASH Media Segments are then ingested by the 5GMS AS.

3. The 5GMS-Aware Application is configured via M8 (step 3) with information about the available content catalogue (e.g. resolving MPD URLs), the available subscription identifiers (e.g. the user has a 4K subscription or the user has an SD subscription), device type identifiers and network policy identifiers.

   The subscription identifiers and the device type identifiers are collectively referred to as Service Description Filters in the following.

NOTE 1: It is for further study whether network policy identifiers are embedded in the MPD Service Descriptions or derived from the Service Descriptions.

NOTE 2: The network policy identifier can be equal to a Policy Template identifier when the 5GMS-Aware Application is aware about its usage (e.g. for QoS streaming or background download). It is assumed here, that a unique Network Policy identifier is assigned to each subscription level.

4. When the user selects an item via the User Interface (U1), the 5GMS-Aware Application translates the input to the needed 5GMSd API calls.

5. The 5GMS-Aware Application provides input (via C1) on the selected presentation entry (i.e. MPD URL) together with a Network Policy Identifier (the value indicates here a "HD Premium QoS" policy (alternative Network Policy Identifiers can refer to e.g. 4K quality), i.e. make the Media Session Handler request a QoS Flow) and Service Description Filters. The Service Description Filter is used by the Media Player to identify the usable Service Descriptions from the MPD. The Network Policy Identifier is used by the Media Session Handler to find the according Policy Description containing information on the Dynamic Policy instantiation method (i.e. procedure and parameters such as Policy Template identifier).

6. The DASH player fetches the MPD.

7. The Media Player selects the Service Description and applies the Service Description Filter.

8. The DASH player indicates to the Media Session Handler (M6) that a "HD Premium QoS" network service should be activated (value of the Network Policy Identifier). The DASH player provides input on bit rate ranges (which may depend on the device type and the title quality). The Media Session Handler has received one or more Policy Descriptions together with matching Service Access Information (via M5_1). When the Media Session Handler has received the policy indication, the Media Session Handler uses the Network Policy Identifier to find the procedure and the parameters to activate the Dynamic Policy instance (i.e. find the matching Policy Description). The Media Session Handler activates a Dynamic Policy instance in the 5GMS AF, providing Service Data Flow Templates identifying the DASH media flows (audio, video, etc) and to provide the desired bit rate of the video.

9. The Media Session Handler activates a Dynamic Policy instance with the 5GMS AF. The 5GMS AF uses the Policy Template identifier to look up the matching Policy Template in order to create the PCF or NEF API invocation. As result, the Media Session Handler receives the enforcement bit rate in the 5GMS AF response. The 5GMS Client should not exceed this bit rate threshold.

   The Service Access Information (via M5_1) includes a list of recommended traffic detection methods. The Media Session Handler selects a Service Data Flow description method (e.g. 5-Tuples). When the Media Session Handler selects:

   - 5-Tuples: For each new TCP connection, the Media Session Handler updates the Dynamic Policy instances and adds a new 5-Tuple. For each closed TCP connection, the Media Session Handler updates the Dynamic Policy instances and removes the 5-Tuple of the closed TCP connection.

   - TOS or Traffic Class: The Media Session Handler sets the TOS or Traffic Class for each new TCP connection.

   - Domain name: The Media Session Handler provides the domain name with the Dynamic Policy Instance.

A.2.3 Example parameters

Table A.2.3-1: M5_1 parameters for Policy Descriptions (used by the Media Session Handler)

Parameter	Type	Purpose	Example Values
Policy Description	Object		
Network Policy Identifier	String	Identifies the Policy Description.	"4K Premium QoS", "HD Premium QoS".
Service Access Information URL	URL	References the associated Service Access Information.	

Table A.2.3-2: M5_1 parameters for Service Access Information

Parameter	Type	Purpose	
Service Access Information	Object		
Policy Template identifier	String	Identifies the Policy Template.	"HD QoS".
5GMS AF URL	URL	Used to invoke the 5GMS AF.	
Mandatory Request M5 information	List	Desired bit rate, which should be provided by the network for the application.	Policy Template identifier, Desired Bit Rate, Packet Detection Filters.
M5 Response information	List	Information to the Media Session Handler on the response parameters.	OK (requested bit rate is accepted), Proposed Lower Bit rate (requested bit rate cannot be provided).
sdfMethod	[String]	Indicates which Service Data Flow Description methods are recommended to be used by the Media Session Handler.	"5-Tuple", "domainName", "TOS=xx", etc.

A.3 (Conditional) Zero Rating dynamic policy

A.3.1 General

In the case of (Conditional) Zero Rating, the quality of a video streaming service should not exceed a certain bit rate threshold (called the policy threshold). This can be realized by deploying a traffic shaper in the network (e.g. a policing function in the UPF) or by instructing the DASH Player not to exceed a certain policy threshold bit rate. The policy threshold may be network-specific, i.e. depending on the 5G System. The following realization assumes the latter, i.e. the DASH Player is not exceeding the bit rate policy and the UPF is just monitoring the compliance of the application flows (one or more TCP and/or UDP flows). The MPD is annotated using DASH Service Descriptions in such a way that the DASH Player can identify which maximal representation bit rates still comply with the policy threshold.

Figure A.3.11 below illustrates the per-title quality and the policy threshold. For *Titles1* and *Title2*, the 5GMSd Client can activate the SD and HD representations. For *Title3*, the 5GMSd Client can activate the 480p and the SD representations. For *Title4*, the 5GMSd Client can activate all available representations (i.e. SD and HD).

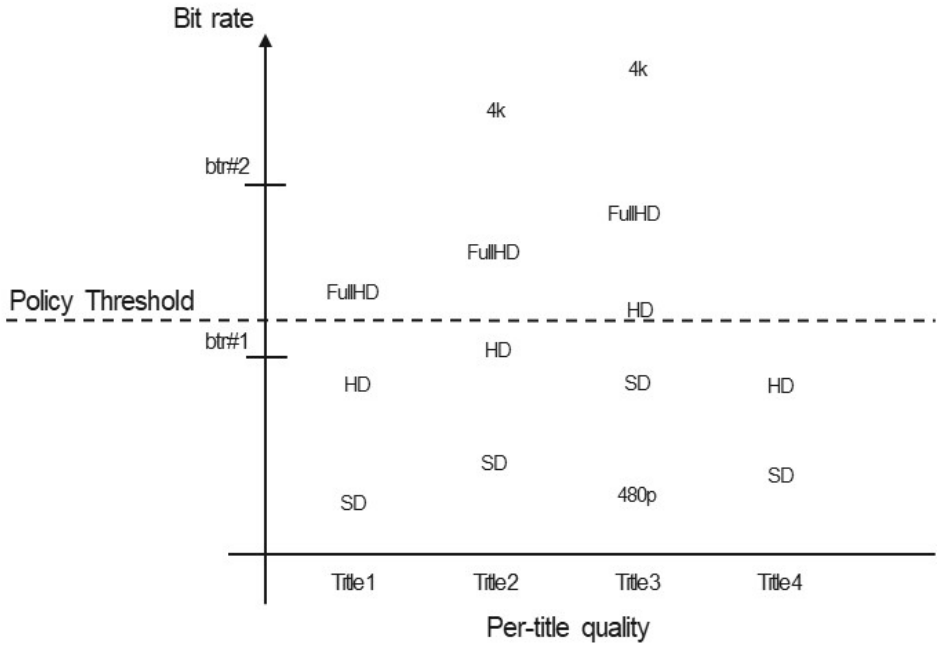

Figure A.3.1-1: Policy threshold versus quality

When the 5GMSd Client receives the bit rate of the policy threshold from the network, the 5GMSd Client filters the MPD for policy-compliant representations (i.e. those that lie at or below the policy threshold).

A.3.2 Procedure

The procedure for activating a (Conditional) Zero Rating dynamic policy is illustrated in figure A.3.2-1 below.

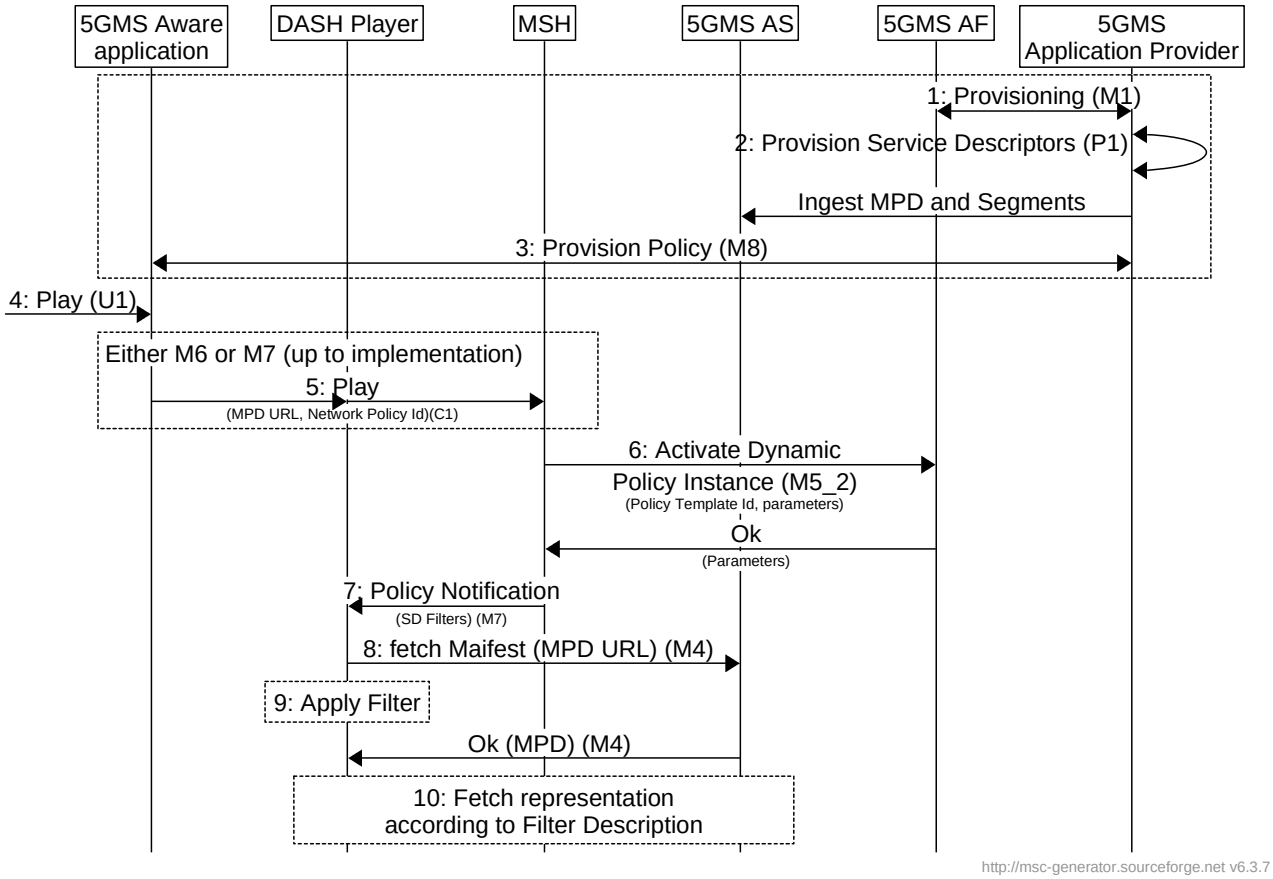

Figure A.3.2-1: Procedure for activating (Conditional) Zero Rating dynamic policy

Steps:

1. The 5GMS Application Provider interacts with the 5GMS AF to set up one or more Policy Templates. Each Policy Template is identified by a Policy Template identifier and contains information about how to activate the corresponding policy within the 5G System (e.g. N5 URLs and parameters).

2. The 5GMS Application Provider interacts with its DASH content generation function (e.g. an MPD provider) to annotate the DASH MPD with Service Descriptions (step 2). The intention of the Service Descriptions here is that the DASH Player can identify those representation combinations which do not exceed the bit rate requirement.

3. The 5GMSAware Application is configured via M8 with information about the available content catalogue (e.g. resolving MPD URLs), the available subscription identifiers (e.g. the user has a 4K content subscription or the user has an SD subscription), device type identifiers.

   The 5GMSd-Aware Application is configured via M8 about the available (Conditional) Zero Rating policy. This includes the Network Policy Ids.

4. When a user selects an item via the User Interface (U1), the 5GMS-Aware Application translates the input to the needed 5GMSd API calls.

5. The 5GMS Aware Application provides input (via C1) on the selected presentation entry (i.e. MPD URL) and also on the Network Policy Id (the value in this case indicates a (Conditional) Zero-Rating policy, i.e. make the Media Session Handler request the policy threshold parameter from the network).

NOTE: C1 is an abstract interface and indicates that the 5GMS-Aware Application may either first use M6 or M7 for the interactions with the 5GMS Client.

6. The Media Session Handler uses the Network Policy Identifier to find the procedure and the parameters to activate the Dynamic Policy Instance (here a (Conditional) Zero Rating policy). The Media Session Handler has received one or more Policy Descriptions together with matching Service Access Information (via M5_1). The Media Session Handler uses the Network Policy Identifier as a key to find the correct Policy Description. Here, the Network Policy Identifier indicates a (Conditional) Zero Rating policy. The Media Session Handler should activate a dynamic policy in the 5GMS AF, providing Service Data Flow Template information about the DASH media flows (audio, video, etc.) and retrieving the bit rate threshold, which cannot be exceeded to comply with the policy. The Media Session Handler receives (as result of the Dynamic Policy activation) some information on the policy enforcement (*enforcementMethod* and/or *enforcementBitrate*), so that the representation selection logic (bit rate adaptation function) in the DASH Player can consider the effects of the enforcement scheme.

7. The Media Session Handler activates the Dynamic Policy instance on M5, providing a Policy Template identifier. Upon positive response, the Media Session Handler notifies the DASH Player, providing Service Descriptor Filters. The Media Session Handler may receive these Service Descriptor Filters with the response, or it may look up the Service Descriptor Filter values by a response value. Alternatively, the Media Session Handler receives a maximum bit rate with the response and the Media Session Handler derives the Service Descriptor Filter. The Media Session Handler may also receive information about Policy Enforcement, e.g. what type of traffic shaper will throttle the bit rate.

   The Media Session Handler may need to update the Dynamic Policy instance, depending on the selected traffic detection method. For example, when the Media Session Handler uses 5-Tuples, the Media Session Handler needs to update the Dynamic Policy instance with every newly opened and every closed TCP connection.

8. The DASH Player fetches the MPD of the selected content.

9. The Service Descriptor Filter is used by the DASH Player to filter policy-compliant Service Descriptions from the MPD. The DASH Access Engine or Selection Logic (see ISO/IEC 23009-1 [32] figure K.1) selects only adaptation sets and representations according to the filter. Here, the DASH Player fetches the MPD after the notification from the Media Session Handler.

A.3.3 Example parameters

Table A.3.3-1: M5_1 parameters for Policy Descriptions (used by the Media Session Handler)

Parameter	Type	Purpose	Example Values
Policy Description	Object		
Network Policy Id	String	Identifies the Policy Description.	"(Conditional) Zero Rating".
Service Access Information URL	URL	References the associated Service Access Information.	

Table A.3.3.-2: M5_1 parameters for Service Access Information

Parameter	Type	Purpose	
Service Access Information	Object		
Policy Template Id	String	Identifies the Policy Template.	"not exceed bit rate"
5GMS AF URL	URL	Used to invoke the 5GMS AF.	
sdfMethods	[String]	Indicates which Service Data Flow Description methods are recommended for use by the Media Session Handler.	"5-Tuple", "domainName", "TOS=xx", etc.
Mandatory M5 Request information	List		Policy Template identifier, Service Data Flow Template.
M5 Response information	List	Information to the Media Session Handler on the response parameters.	Bit rate Policy Threshold (upper bit rate bound, which should not be exceeded).

A.4 Background Download

A.4.1 General

In the case of Background Download, the asset is acquired in the background, prior to viewing. Many application services offer the capability of acquiring a VoD item for later consumption. The 5GMS-Aware Application triggers the Media Session Handler to acquire the item, providing a background download network policy id.

NOTE: Here, the DASH Player is handling the acquisition, since the DASH Player contains the MPD processing and the DASH Access engine parts. Other realizations would use a separate background download agent, which is not even try to decode and render the video.

Figure A.4.1-1 below illustrates the representation marking for background download. The MPD may be annotated with Service Descriptions clearly identifying representations intended for download. Here, *Title1* should be downloaded in Full HD quality and all other titles in regular HD quality.

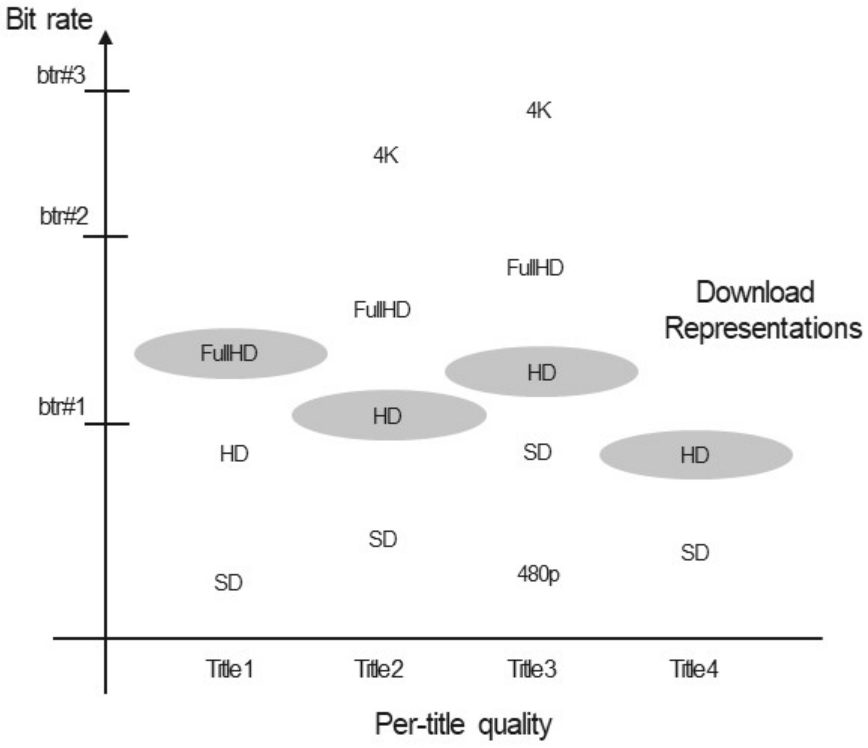

Figure A.4.1-1: Background Download Representations

A.4.2 Procedure

The procedure for activating a Background Download dynamic policy is illustrated in figure A.3.2-1 below.

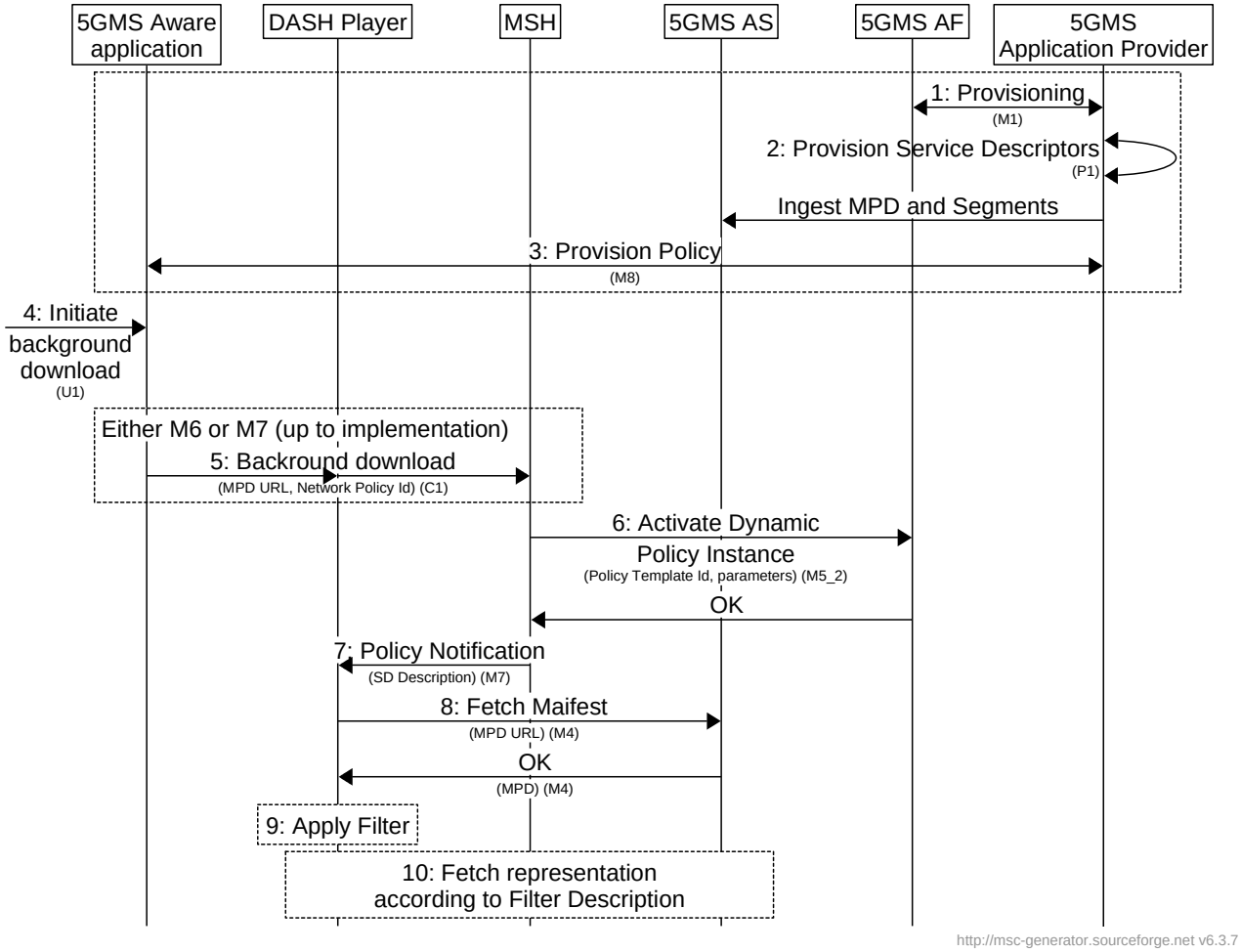

Figure A.3.2-1: Procedure for activating Background Download dynamic policy

Steps:

1. The 5GMS Application Provider interacts with the 5GMS AF to set up one or more Policy Templates (M1). Each Policy Template is identified by a Policy Template identifier and contains information about how to activate the according policy within the 5G System (e.g. N5 URLs and parameters).

2. The 5GMS Application Provider also interacts with its DASH content generation function (e.g. an MPD provider) to annotate the DASH MPD with Service Descriptions, e.g. to identify, which representation is intended for background download.

3. The 5GMS-Aware Application is configured via M8 with information about the available content catalogue (e.g. resolving MPD URLs), the available subscription identifiers (e.g. the user has a 4K subscription or the user has an SD subscription), device type identifiers.

   The 5GMSd-Aware Application is configured via M8 about the available background download policy. This includes the Network Policy Id which hints a background download policy.

4. When a user selects an item via the User Interface (U1) for Background Download the 5GMS-Aware Application translates the input to the needed 5GMSd API calls.

5. The 5GMS-Aware Application provides input (via C1) on the selected presentation entry (i.e. MPD URL) and also on the Network Policy Identifier (indicating a background download policy, i.e. make the Media Session Handler request a bearer suitable for Background Download).

NOTE: C1 is an abstract interface and indicates that the 5GMS-Aware Application may either first use M6 or M7 for the interactions with the 5GMS Client.

6. The Media Session Handler uses the Network Policy Identifier to find the procedure and the parameters to activate the Dynamic Policy Instance (here a Background Download policy). The Media Session Handler has received one or more Policy Descriptions together with matching Service Access Information (via M5_1). The Media Session Handler uses the Network Policy Identifier as a key to find the correct Policy Description. The Media Session Handler should activate a Dynamic Policy in the 5GMS AF, providing Service Data Flow Template information of the media flows (audio, video, etc). The Media Session Handler can also receive information on a bit rate policing (*enforcementMethod* and/or *enforcementBitrate*), e.g. that the bit rate is actively limited.

7. The Media Session Handler activates the Dynamic Policy instance on M5, providing the Policy Template identifier and additional parameters. Upon positive response, the Media Session handler notifies the DASH Player to start the Background Download. The notification contains a Service Descriptor Filters, which is used by the DASH Player to filter policy-compliant Service Descriptions from the MPD. The Media Session Handler may receive the Service Descriptor Filters with the response or may look up the Service Descriptor Filter values by a response value (e.g. derived from a maximum bit rate indication).

   The Media Session Handler may need to update the Dynamic Policy instance, depending on the selected traffic detection method. For example, when the Media Session Handler uses 5-Tuples, it needs to update the Dynamic Policy instance with every newly opened and every closed TCP connection.

8. The DASH Player fetches the MPD of the selected content.

9. The DASH Access Engine / Selection Logic (see ISO 23009-1 [32] figure K.1) selects only adaptation sets and representations according to the filter (i.e. suitable for Background Download). Here, the DASH Player fetches the MPD after the notification from the Media Session Handler.

A.4.3 Example parameters

Table A.4.3-1: M5_1 Parameters for Policy Descriptions (used by the Media Session Handler)

Parameter	Type	Purpose	Example Values
Policy Description	Object		
Network Policy Id	String	Identifies the Policy Description.	"Background Download".
Service Access Information URL	URL	References the associated Service Access Information.	

Table A.4.3-2: M5_1 Parameters for Service Access Information

Parameter	Type		
Service Access Information	Object		
Policy Template Id	String	Identifies the Policy Template.	"backgrounddata".
5GMS AF URL	URL	Used to invoke the 5GMS AF.	
sdfMethods	[String]	Indication, which Service Data Flow Description methods are recommended to use by the media session handler.	"5-Tuple", "domainName", "TOS=xx", etc.
Mandatory M5 Request information	List	Desired bit rate, to be provided by the network for the application.	Policy Template Id, Average Bit rate, Service Data Flow Template.
M5 Response information	List	Information to the Media Session Handler on the response parameters.	

Annex B (informative): Content Hosting Configuration examples

B.1 Pull-based content ingest example

B.1.1 Overview

1. The 5GMSd Client on the UE requests a media resource via M4d.
2. The 5GMSd AS determines that it does not have a cached copy of the requested media resource.
3. The 5GMSd AS transforms the M4d request URL into a request to the 5GMSd Application Provider's origin server via M2d.

B.1.2 Desired URL mapping

In the example shown in table B.1.2-1 below, media resources are exposed at M4d from a default canonical domain *5gmsd-as.mno.net* determined by the 5GMSd System operator, and also from a custom domain name alias *mno-cdn.5gmsd-ap.com* that has been configured by the 5GMSd Application Provider.

Table B.1.2-1: Example URL mapping for pull-based ingest

M4d request from 5GMSd Client	Mapped M2d request to origin server on 5GMSd AS cache miss
https://**5gmsd-as.mno.net**/m4d/provisioning-session9876/**asset123456/video1**/segment1000.mp4	https://origin.5gmsd-ap.com/media/**asset123456**/**video1**/segment1000.mp4
https://**mno-cdn.5gmsd-ap.com**/m4d/provisioning-session9876/**asset123456/video1**/segment1000.mp4	
https://**5gmsd-as.mno.net**/m4d/provisioning-session9876/**asset123456/video2**/segment1000.mp4	https://origin.5gmsd-ap.com/media/**asset123456**/**video2**/segment1000.mp4
https://**mno-cdn.5gmsd-ap.com**/m4d/provisioning-session9876/**asset123456/video2**/segment1000.mp4	
https://**5gmsd-as.mno.net**/m4d/provisioning-session9876/**asset123456/audio1**/segment1000.mp4	https://origin.5gmsd-ap.com/media/**asset123456**/**audio1**/segment1000.mp4
https://mno-cdn.5gmsd-ap.com/m4d/provisioning-session9876/**asset123456/audio1**/segment1000.mp4	

B.1.3 Content Hosting Configuration

Table B.1.3-1 below shows the relevant Content Hosting Configuration parameters needed to achieve the example mapping described in table B.1.2-1 above.

Table B.1.3-1: Content Hosting Configuration properties relevant to pull-based ingest

Property	Example value	Set by
IngestConfiguration		
protocol	urn:3gpp:5gms:content-protocol:**http-pull-ingest**	5GMSd Application Provider
pull	true	
entryPoint	https://origin.5gmsd-ap.com/	
path	*(Not used)*	*(Not applicable)*
DistributionConfiguration		
canonicalDomainName	5gmsd-as.mno.net	5GMSd AF
domainNameAlias	mno-cdn.5gmsd-ap.com	5GMSd Application Provider
PathRewriteRules[0].requestPathPattern	^/m4d/provisioning-session[^/]+/	
PathRewriteRules[0].mappedPath	/media/	
NOTE: The 5GMSd Application Provider needs prior knowledge of the path structure exposed at M4d in order to supply the *requestPathPattern* regular expression. In this example, the Provisioning Session identifier is included in the M4d distribution path as a discriminator (*c.f.* "Content Provider code" concept in a commercial CDN).		

B.2 Push-based content ingest example

B.2.0 Overview

1. The 5GMSd Application Provider uploads content to the 5GMSd AS via M2d.
2. The 5GMSd AS rewrites the M2d upload URL to an M4d downlink URL that is exposed to the 5GMSd Client on the UE.

B.2.1 Desired URL mapping

In the example shown in table B.2.1-1, media resources are pushed into the 5GMSd AS at M2d by the 5GMSd Application Provider and exposed to the 5GMSd Client at M4d using the canonical name of the 5GMSd AF *5gmsd-*

as.mno.net and an additional domain name alias *mno-cdn.5gmsd-ap.com* configured by the 5GMSd Application Provider.

Table B.2.1-1: Example URL mapping for pull-based ingest

M2d ingest URL pushed to 5GMSd AS	M4d URL exposed to 5GMSd Client
https://5gmsd-as.mno.net/m2d/provisioning-session9876/**asset123456**/**video1**/segment1000.mp4	https://**5gmsd-as.mno.net**/m4d/provisioning-session9876/**asset123456**/**video1**/segment1000.mp4
	https://**mno-cdn.5gmsd-ap.com**/m4d/provisioning-session9876/**asset123456**/**video1**/segment1000.mp4
https://5gmsd-as.mno.net/m2d/provisioning-session9876/**asset123456**/**video2**/segment1000.mp4	https://**5gmsd-as.mno.net**/m4d/provisioning-session9876/**asset123456**/**video2**/segment1000.mp4
	https://**mno-cdn.5gmsd-ap.com**/m4d/provisioning-session9876/**asset123456**/**video2**/segment1000.mp4
https://5gmsd-as.mno.net/m2d/provisioning-session9876/**asset123456**/**audio1**/segment1000.mp4	https://**5gmsd-as.mno.net**/m4d/provisioning-session9876/**asset123456**/**audio1**/segment1000.mp4
	https://**mno-cdn.5gmsd-ap.com**/m4d/provisioning-session9876/**asset123456**/**audio1**/segment1000.mp4

B.2.2 Content Hosting Configuration

Table B.2.2-1 below shows the relevant Content Hosting Configuration parameters needed to achieve the example mapping described in table B.2.1-1 above.

Table B.2.2-1: Content Hosting Configuration properties relevant to push-based ingest

Property	Example value	Set by
IngestConfiguration		
protocol	urn:3gpp:5gms:content-protocol:**dash-if-ingest**	5GMSd Application Provider *(first M1d request)*
pull	false	
entryPoint	https://5gmsd-as.mno.net/	5GMSd AF *(first M1d response)*
path	/m2d/provisioning-session9876/	
DistributionConfiguration		
canonicalDomainName	5gmsd-as.mno.net	
domainNameAlias	mno-cdn.5gmsd-ap.com	5GMSd Application Provider *(second M1d request)*
PathRewriteRules[0].requestPathPattern	^/m2d/provisioning-session9876/	
PathRewriteRules[0].mappedPath	/m4d/provisioning-session9876/	

NOTE 1: The 5GMSd Application Provider needs knowledge of the M2d ingest path in order to set *requestPathPattern*. This requires a two-phase transaction when provisioning the Content Hosting Configuration at M1d. In the first request to create a Content Hosting Configuration at M1d, the 5GMSd Application Provider specifies the *protocol* and *pull* properties. In response, the 5GMSd AF sets the *entryPoint* and *path*. Then, in a second request at M1d, the 5GMSd Application Provider modifies the Content Hosting Configuration to add the necessary path rewrite rule.

NOTE 2: The 5GMSd Application Provider needs knowledge of the path structure exposed at M4d in order to supply the *mappedPath*. In this example, the Provisioning Session identifier is included in the M4d path as a discriminator (*c.f.* "Content Provider code" concept in a commercial CDN).

Annex C (normative)
OpenAPI representation of the 5GMSA HTTP REST APIs

C.1 General

This Annex is based on the OpenAPI 3.0.0 specification [23] and provides corresponding representations of all APIs defined in the present specification.

NOTE 1: An OpenAPIs representation embeds JSON Schema representations of HTTP message bodies.

This Annex shall take precedence when being discrepant to other parts of the specification with respect to the encoding of information elements and methods within the API(s).

NOTE 2: The semantics and procedures, as well as conditions, e.g. for the applicability and allowed combinations of attributes or values, not expressed in the OpenAPI definitions but defined in other parts of the specification also apply.

C.2 Data Types applicable to several APIs

C.3 OpenAPI representation of the M1 APIs

C.3.1 Provisioning Sessions API

C.3.2 Server Certificates Provisioning API

C.3.3 Content Preparation Templates Provisioning API

C.3.4 Content Protocols Discovery API

C.3.5 Content Hosting Configuration API

C.3.6 Consumption Reporting Provisioning API

C.3.7 Metrics Reporting Provisioning API

C.3.8 Policy Templates Provisioning API

C.4 OpenAPI representation of the M5 APIs

C.4.1 Service Access Information API

C.4.2 Consumption Reporting API

C.4.3 Metric Reporting API

C.4.4 Dynamic Policies API

C.4.5 Network Assistance API

Annex C (informative): Change history

Change history							
Date	**Meeting**	**TDoc**	**CR**	**Rev**	**Cat**	**Subject/Comment**	**New version**
25.6.2019	SA4#104	S4-190649				Initial Version	0.0.1
23.1.2020	SA4#107	S4-200077, S4-200238, S4-200239, S4-200318				Updates during SA4#107	0.3.0
07.02.2020	ConfCall	S4-AHI931, S4-AHI932				Scope, editorial improvements and online edits from Conf Call (6th Feb 2020)	0.3.1
11.02.2020	offline					Editorial updates according to offline email discussions	0.3.2
2020-02	ConfCall	S4-AHI950				Editorial updates from Conf Call (Online, 13th Feb 2020)	0.4.0
2020-03	-	SP-200237				Specification to TSG: 5G Media Streaming (5GMS); Protocols TS 26.512, Version 1.0.0	1.0.0
		S4-AHI953					1.0.1
2020-04	SA4#108e	S4-200513, S4-200514, S4-200633				Renaming entities in the 5GMS Provisioning API, Additional clauses to specify procedures for manipulating Ingest Protocols, Content Preparation Templates and Server Certificates, Consumption Reporting Procedure API- M1d and M5d	1.0.2
2020-05	Conf Call	S4-AHI989				New Structure	1.1.0
2020-06	SA4#109e	S4-200920, S4-200886, S4-200889, S4-200883				920: Consumption reporting in M7d interface, 886: RAN Signaling-based Network Assistance, 889: API for Service Access information acquisition, 883: APIs for Server Certificates, Content Preparation Templates and Ingest Protocols	1.2.0
2020-08	SA4#110e	S4-AHI996 S4-AHI998 S4-AHIA33				996: Completion of Content Preparation Template procedures, 998: Completion of content distribution geofencing feature, A33: Completion of Server Certificates Provisioning API	1.3.0
2020-08	SA4#110e	Cor of S4-AHI998				Correction of S4-AHI998 implementation, Editorial Correction in Clause 11.2.4	1.3.1
2020-08	SA4#110e	S4-201092, S4-201114, S4-201210, S4-201208, S4-201213, S4-201230, S4-201004, S4-201229, S4-201221, S4-201231, S4-201225, S4-201271, S4-201266, S4-201282, S4-201281				1092: Editorial Improvements, 1114: Specification structure – interfaces and APIs, 1210: Completion of Ingest Protocols API, 1208: Informative Annex on Parameter Population, 1213: Addition of General Sections, 1230: M6d APIs for 5GMS, 1004: Informative annex on Content Hosting Configuration examples, 1229: Correction of the Policy Template resource state transitions, 1221: DASH/CMAF in 5GMSd, 1231: M7d APIs, 1225: Update on consumption reporting, 1271: Update on Metrics Reporting, 1266: Updated on M5 Dynamic Policy activation API and M1 Policy Template Provisioning API, 1282: 5GMS3: AF-based Network Assistance, 1281: Provisioning Sessions API	1.4.0
2020-09	SA#89-e	SP-200666				5G Media Streaming (5GMS); Protocols (This was the presentation of Specification to TSG: 5G Media Streaming (5GMS); Protocols TS 26.512, Version 2.0.0 to bring UCC)	16.0.0

2020-12	SA#90-e	SP-200935	0004	3	F	Cumulative corrections of 5GMS3 APIs [CRs implemented: S4-201432: Cumulative corrections of 5GMS3 APIs, Ericsson S4-201305: Editorial corrections, BBC S4-201363: Additions and Modifications to M1 API on Metrics Reporting Configuration, Qualcomm S4-201622: Text on Procedures for Uplink Streaming, Qualcomm, Ericsson S4-201580: Correction of the missing SdfMethod type definition, Ericsson S4-201593: Correction of the missing CRUD operation notation, Ericsson S4-201594: Correction of the MediaPlayerEntry and ClientMetricsReportingConfiguration cardinality in the Service Access Information resource, Ericsson S4-201596: Correction of the Service Access Information subresource (URL), Ericsson S4-201597: Annex for OpenAPI Implementation, Ericsson S4-201595, Update Consumption reporting, Enensys Technology, BBC S4-201590: Bug Fixes on Metrics Reporting Functionality, Ericsson LM, Qualcomm Incorporated S4-201486: AF-based Network Assistance, Sony Europe B.V., Ericsson LM S4-201608: CR on AT Commands for RAN-based Assistance, Qualcomm Inc.]	16.1.0

History

Document history		
V16.0.0	November 2020	Publication
V16.1.0	January 2021	Publication