**3GPP TSG-SA4 Meeting #109  *S4-200839***

**Electronic, Online, 20th May–3rd June 2020**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *CR-Form-v12.0* | | | | | | | | |
| **PSEUDO CHANGE REQUEST** | | | | | | | | |
|  | | | | | | | | |
|  | **TS 26.512** | **CR** | **–** | **rev** | **–** | **Current version:** | **1.1.0** |  |
|  | | | | | | | | |
| *For* [***HE******LP***](http://www.3gpp.org/3G_Specs/CRs.htm#_blank)*on using this form: comprehensive instructions can be found at* [*http://www.3gpp.org/Change-Requests*](http://www.3gpp.org/Change-Requests)*.* | | | | | | | | |
|  | | | | | | | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Proposed change affects:*** | UICC apps |  | ME |  | Radio Access Network |  | Core Network | **X** |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | | |
| ***Title:*** | APIs for Server Certificates, Content Preparation Templates and Ingest Protocols | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Source to WG:*** | BBC | | | | | | | | | |
| ***Source to TSG:*** | S4 | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Work item code:*** | 5GMS3 | | | | |  | ***Date:*** | | | 2020-05-18 |
|  |  | | | |  | |  | | |  |
| ***Category:*** | **D** |  | | | | | ***Release:*** | | | Rel-16 |
|  | *Use one of the following categories:* ***F*** *(correction)* ***A*** *(mirror corresponding to a change in an earlier release)* ***B*** *(addition of feature),* ***C*** *(functional modification of feature)* ***D*** *(editorial modification)*. | | | | | | | |  | |
|  |  | | | | | | | | | |
| ***Reason for change:*** | | Restructure of API clauses requires some existing text to be moved. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Summary of change:*** | | 1. Additional references. 2. Corrections to names of APIs in Table 4.2‑1. 3. API specifications for Server Certificates, Content Preparation Templates and Protocols. 4. Remove redundant lines from the list of Content Hosting Configuration API operations. 5. Minor corrections to ContentHostingConfiguration resource structure, including addition of missing references. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Consequences if not approved:*** | | These API definitions will appear in the wrong clauses. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Clauses affected:*** | | 2, 4.2, 7.3.,7.4, 7.5, 7.6.2, 7.6.3.1. | | | | | | | | |
|  | |  | | | | | | | | |
|  | | **Y** | **N** |  | | | |  | | |
| ***Other specs*** | |  | **X** | Other core specifications | | | |  | | |
| ***affected:*** | |  | **X** | Test specifications | | | |  | | |
| ***(show related CRs)*** | |  | **X** | O&M Specifications | | | |  | | |
|  | |  | | | | | | | | |
| ***Other comments:*** | |  | | | | | | | | |
|  | |  | | | | | | | | |
| ***This CR's revision history:*** | |  | | | | | | | | |

# Discussion

## Service Certificates Provisioning API

In the absence of a technical solution to authority delegation, best practice when provisioning a server certificate within an X.509-based Public Key Infrastructure is for the entity that will ultimately present the certificate (in this case, the 5GMS System) to issue a Certificate Signing Request to the entity that will be supplying the certificate (in this case, the 5GMSd Application Provider).

Since this procedure needs to be initiated by the 5GMSd Application Provider, two phases are specified in clause 7.3, represented by steps 1 and 3 below:

**1. 5GMSd Application Provider solicits a Certificate Signing Request from the 5GMSd AF by invoking the POST method on the Server Certificates collection resource.**

a. The 5GMSd AF returns 201 Created.

b. The 5GMSd AF reserves a Server Certificate resource identifier in the collection and returns it to the caller in the Location response header.

c. The 5GMSd AF includes a Certificate Signing Request in the HTTP response body that includes the public key of the eventual Server Certificate.

d. The 5GMSd AF also creates a private key for the eventual Server Certificate *but keeps it secret*.

2. 5GMSd Application Provider creates a Server Certificate based on the Certificate Signing Request and signs it with the public key that was included in the Certificate Signing Request.

**3. 5GMSd Application Provider uploads the signed Server Certificate to the 5GMSd AF by invoking the PUT method on the Server Certificate resource identifier received from the 5GMSd AF in step 1.**

4. 5GMSd AF verifies the signature of the uploaded Server Certificate (using the private key) and, if successful, stores the uploaded Server Certificate alongside the corresponding private key for later use.

5. When the 5GMSd Application Provider subsequently provisions a Content Hosting Configuration that references the Server Certificate resource identifier, both the Server Certificate and the private key are passed to the 5GMSd AS (via interface M3d) for presentation to 5GMSd Clients at interface M4d.

In this manner, the private key is never divulged to the 5GMSd Application Provider, and the risk of key compromise is thereby minimised.

In addition, because uploaded Server Certificates are signed, the 5GMSd AS has confidence that the Server Certificate uploaded in step 3 corresponds to the Certificate Signing Request that it originally issued in step 1.

***================================START OF FIRST CHANGE==============================***

# 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: "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: "Technical Specification Group Core Network and Terminals; 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] IETF RFC 8446: "The Transport Layer Security (TLS) Protocol Version 1.3", August 2018.

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

***=================================END OF FIRST CHANGE===============================***

***===============================START OF SECOND CHANGE=============================***

## 4.2 APIs relevant to Downlink Streaming

Table 4.2‑1 below 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 |
| Ingest Protocols Discovery API | 7.5 |
| Content Hosting Provisioning API | 7.6 |
| M2d |  |  |
| M4d | DASH (TS 26.247) or 3GP (TS 26.244) |  |
| 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 |  |
| M5d | Service Access Information API | 11.2 |
| Metrics Reporting API | 11.3 |
| 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 |  |
| M5d | Service Access Information API | 11.2 |
| Consumption Reporting API | 11.4 |
| 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 |  |
| M5d | Service Access Information API | 11.2 |
| Dynamic Policies API | 11.5 |
| Network Assistance | The 5GMSd Client queries the 5GMSd AF for information about the network according to a provisioned configuration it obtains from the Service Access Information for its Provisioning Session. | M5d | Service Access Information API | 11.2 |
| Network Assistance API | 11.6 |

***================================END OF SECOND CHANGE==============================***

***================================START OF THIRD CHANGE==============================***

## 7.3 Server Certificates Provisioning API

### 7.3.1 Overview

The Server Certificates Provisioning API is used to provide 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 Certificates are provisioned within the scope of a 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}/{provisioningSubresource}

Table 7.3.2‑1 below specifies the operations and the corresponding HTTP methods that are supported by this API. In each case, the Provisioning Session identifier shall be substituted into {provisioningSessionId} in the above URL template and the sub-resource path specified in the second column shall be substituted into {provisioningSubresource}.

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

|  |  |  |  |
| --- | --- | --- | --- |
| Operation | Sub‑resource path | Allowed HTTP method(s) | Description |
| Reserve Server Certificate | certificates | 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 generated by the 5GMSd AF, as specified in clause 7.3.3 below. |
| Retrieve Server Certificate | certificates/{certificateId} | GET | Used to retrieve a previously 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, or to replace an existing Server Certificate.  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. |
| 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

The Certificate Signing Request and the Server Certificate resource shall comply with the Privacy-Enhanced Mail (PEM) textual format specified in RFC 7468 [13], i.e. a Base64-encoded DER certificate request or certificate, including leading and trailing encapsulation boundary lines.

Certificate Signing Requests and Server Certificate resources shall have the MIME content type 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 Prepation Templates Provisioning API is accessible through the following URL base path:

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

Table 7.4.2‑1 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 substituted into {provisioningSubresource}.

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

Editor’s note: Missing specification.

## 7.5 Ingest Protocols Discovery API

### 7.5.1 Overview

The Ingest 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 Ingest Protocols Discovery API is accessible through the followng URL base path:

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

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 shall be substituted into {provisioningSubresource}.

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 ingest protocols | protocols | GET | This operation is used to retrieve a list of supported ingest protocols. |

### 7.5.3 Data model

#### 7.5.3.1 IngestProtocols resource

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

Table 7.6.3.1-1: Definition of IngestProtocols resource

| Property name | Type | Cardinality | Description |
| --- | --- | --- | --- |
| downlinkIngestProtocols | [URI String] | 1..1 | An array of URI strings, each one uniquely identifying a media ingest protocol supporting at interface M2d by the 5GMSd AS(s), as specified in clause 7.5.4. |

### 7.5.4 Media ingest protocols

#### 7.5.4.1 Introduction

Every media ingest protocol shall be uniquely identified by a URI string allocated within the scope of the name space urn:3gpp:5gms:ingest-protocol.

This specification defines a single DASH-based media ingest protocol.

#### 7.5.4.2 DASH-based media ingest protocol

If IngestConfiguration[protocol] in the Content Hosting Configuration is set to urn:3gpp:5gms:ingest-protocol:dash-if-ingest, media resources shall be ingested by the 5GMSd AS as specified by [3]. The IngestConfiguration[pull] shall be set to False, indicating that a Push-based protocol is used. The IngestConfiguration[entryPoint] parameter 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] parameter.

***=================================END OF THIRD CHANGE==============================***

***===============================START OF FOURTH CHANGE=============================***

### 7.6.2 Resource structure

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

{apiRoot}/3gpp-m1d/v1/provisioning-sesions/{provisioningSessionId}/{provisioning-subresource}

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 substituted into {provisioning-subresource}.

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 | Used to invalidate some or all cached media resources associated with this Content Hosting Configuration. |

***================================END OF FOURTH CHANGE=============================***

***================================START OF FIFTH CHANGE==============================***

#### 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 | Type | Cardinality | Description |
| --- | --- | --- | --- |
| IngestConfiguration | Object | 1..1 | Describes the 5GMSd Application Provider’s origin server from which media resources will be ingested via interface M2d. |
| name | String | 1..1 | A name associated with this origin server. |
| 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. |
| pull | Boolean | 1..1 | Indicates whether to the 5GMSd AS shall use Pull or Push for ingesting the content. |
| protocol | String | 1..1 | Identifies the media ingest protocol. The set of supported protocols is defined in clause 7.5.4. |
| 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 by the origin 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. |
| DistributionConfiguration | [Object] | 1..n | 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. |
| targetDomain | String | 1 | All resources of the current distribution shall be accessible through this targetDomain FQDN. This FQDN is assigned by the 5GMS AF for the current distribution.  There is a one-to-one association between the originDomain and the targetDomain for this distribution. |
| originDomain | String | 1 | The 5GMSd Application Provider assigns an originDomain for the distribution, which is for example used by the 5GMSd AS to select an appropriate certificate to present at M4d, and to set appropriate CORS HTTP response headers at M4d.  The originDomain is associated with a targetDomain that is assigned by the 5GMSd AF.  The 5GMSd Application Provider may choose to publish the content using its own originDomain (e.g. to use its own certificates or to avoid CORS issues). In such case, the 5GMSd Application Provider creates a CNAME DNS record to point from the originDomain to the targetDomain. |
| PathRewriteRules | [Object] | 0..n | An ordered list of rules for rewriting the ingest URL path to the distribution URL path.  If multiple rules match a particular resource’s path, only the first rule, in order of appearance, shall be applied. |
| originPathPattern | String | 1..1 | A pattern that shall be used to match the path part of the M2d ingest URL for the resource. If the pattern matches, then the path mapping rule is applied. The path part of the ingest URL used for the matching shall include the leading “/”. The originPathPattern shall be provided as a regular expression as specified in [5]. |
| distributionPath | String | 1..1 | If the pathPattern is a match for the current resource, then the ingest path, starting with the “/” and ending with the last “/” in the path, excluding the last segment of the path, shall be replaced by this distributionPath in the URL exposed by the 5GMSd AS at M4d. |
| CachingConfiguration | [Object] | 0..n | 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 | [Integer] | 0..\* | 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..1 | Limit access to the content to the indicated geographic areas. |
| *locationType* | String | 1 | The type of the location information. |
| *locations* | [String] | 1..n | Array of locations from which access to the resources is to be allowed. |
| 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 must 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. |
| 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 must 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 [12], 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. |

***=================================END OF FIFTH CHANGE==============================***