

# ETSI TS 132 158 V18.4.0 (2026-02)



TECHNICAL SPECIFICATION

**LTE;**  
**5G;**  
**Management and orchestration;**  
**Design rules for REpresentational State Transfer (REST)**  
**Solution Sets (SS)**  
**(3GPP TS 32.158 version 18.4.0 Release 18)**



---

**Reference**RTS/TSGS-0532158vi40

---

**Keywords**5G, LTE

---

**ETSI**

650 Route des Lucioles  
F-06921 Sophia Antipolis Cedex - FRANCE

---

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - APE 7112B  
Association à but non lucratif enregistrée à la  
Sous-Préfecture de Grasse (06) N° w061004871

---

**Important notice**

---

The present document can be downloaded from the  
[ETSI Search & Browse Standards](#) application.

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the prevailing version of an ETSI deliverable is the one made publicly available in PDF format on [ETSI deliver](#) repository.

Users should be aware that the present document may be revised or have its status changed, this information is available in the [Milestones listing](#).

If you find errors in the present document, please send your comments to the relevant service listed under [Committee Support Staff](#).

If you find a security vulnerability in the present document, please report it through our [Coordinated Vulnerability Disclosure \(CVD\)](#) program.

---

**Notice of disclaimer & limitation of liability**

---

The information provided in the present deliverable is directed solely to professionals who have the appropriate degree of experience to understand and interpret its content in accordance with generally accepted engineering or other professional standard and applicable regulations.

No recommendation as to products and services or vendors is made or should be implied.

No representation or warranty is made that this deliverable is technically accurate or sufficient or conforms to any law and/or governmental rule and/or regulation and further, no representation or warranty is made of merchantability or fitness for any particular purpose or against infringement of intellectual property rights.

In no event shall ETSI be held liable for loss of profits or any other incidental or consequential damages.

Any software contained in this deliverable is provided "AS IS" with no warranties, express or implied, including but not limited to, the warranties of merchantability, fitness for a particular purpose and non-infringement of intellectual property rights and ETSI shall not be held liable in any event for any damages whatsoever (including, without limitation, damages for loss of profits, business interruption, loss of information, or any other pecuniary loss) arising out of or related to the use of or inability to use the software.

---

**Copyright Notification**

---

No part may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm except as authorized by written permission of ETSI.

The content of the PDF version shall not be modified without the written authorization of ETSI.

The copyright and the foregoing restriction extend to reproduction in all media.

© ETSI 2026.  
All rights reserved.

---

# Intellectual Property Rights

## Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: "*Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards*", which is available from the ETSI Secretariat. Latest updates are available on the [ETSI IPR online database](#).

Pursuant to the ETSI Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

## Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

**DECT™**, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™**, **LTE™** and **5G™** logo are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM®** and the GSM logo are trademarks registered and owned by the GSM Association.

---

# Legal Notice

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

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

The cross reference between 3GPP and ETSI identities can be found at [3GPP to ETSI numbering cross-referencing](#).

---

# Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the [ETSI Drafting Rules](#) (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

# Contents

Intellectual Property Rights .....	2
Legal Notice .....	2
Modal verbs terminology.....	2
Foreword.....	6
1 Scope .....	7
2 References .....	7
3 Definitions and abbreviations.....	8
3.1 Definitions .....	8
3.2 Abbreviations .....	8
4 General rules .....	8
4.1 Information models and resources.....	8
4.1.1 Information models.....	8
4.1.2 Resources .....	9
4.1.3 Resource archetypes .....	9
4.1.4 Mapping of information models to resources .....	9
4.1.5 Usage of information models.....	9
4.2 Managed object naming and resource identification .....	10
4.2.1 Managed object naming.....	10
4.2.1.0 Distinguished Name (DN).....	10
4.2.1.1 Global and local namespaces .....	10
4.2.2 Resource identification .....	10
4.2.3 Mapping of DN to URIs.....	10
4.2.4 Canonical URI .....	11
4.3 Message content formats .....	12
4.3.1 Media types.....	12
4.3.2 Response content format negotiation.....	12
4.4 URI structure .....	12
4.4.1 Introduction.....	12
4.4.2 URI structure for resources representing managed object instances.....	13
4.4.3 URI structure for resources not representing managed object instances.....	14
4.4.4 Resource "{MnSName}/{MnSVersion}" .....	14
4.5 Response status codes .....	15
5 Basic design patterns .....	15
5.1 Design pattern for creating a resource .....	15
5.1.1 Creating a resource with identifier creation by the MnS Producer .....	15
5.1.2 Creating a resource with identifier creation by the MnS Consumer .....	16
5.2 Design pattern for reading a resource.....	17
5.3 Design pattern for updating a resource.....	17
5.4 Design pattern for deleting a resource.....	18
5.5 Design pattern for subscribe/notify .....	19
5.5.1 Concept.....	19
5.5.2 Subscription creation .....	19
5.5.3 Subscription deletion .....	19
5.5.4 Notification emission.....	20
5.5.5 Subscription retrieval.....	20
6 Advanced design patterns.....	21
6.1 Design pattern for scoping and filtering .....	21
6.1.1 Introduction.....	21
6.1.2 Query parameters for scoping.....	21
6.1.3 Query parameters for filtering .....	21
6.1.4 Construction rules for the response message body .....	22
6.2 Design patterns for attribute and attribute field selection.....	23

6.2.1	Introduction.....	23
6.2.2	Query parameters for attribute and attribute field selection.....	23
6.2.3	Construction rules for the response message body .....	23
6.3	Design pattern for partially updating a resource.....	23
6.3.1	Introduction.....	23
6.3.2	JSON Merge Patch.....	24
6.3.3	JSON Patch.....	25
6.4	Design patterns for patching multiple resources .....	29
6.4.1	Introduction.....	29
6.4.2	3GPP JSON Merge Patch .....	29
6.4.3	3GPP JSON Patch.....	29
6.5	Design pattern for large queries .....	32
6.6	Design pattern for error responses.....	32
6.6.1	Introduction.....	32
6.6.2	HTTP error codes .....	33
6.6.3	Error response body .....	34
6.6.3.1	Overview .....	34
6.6.3.2	Error response format for GET requests .....	35
6.6.3.3	Error response format for PUT, POST, DELETE, JSON Merge Patch and 3GPP JSON Merge Patch requests.....	35
6.6.3.4	Error response format for JSON Patch and 3GPP JSON Patch requests .....	36
6.6.4	The "type" property .....	36
6.6.5	The "reason" property.....	37
6.6.5.1	Overview.....	37
6.6.5.2	Error reasons for GET .....	38
6.6.5.3	Error reasons for attribute manipulations .....	39
6.6.5.3.1	JSON Patch and 3GPP JSON Patch .....	39
6.6.5.3.2	JSON Merge Patch, 3GPP JSON Merge Patch and PUT .....	41
6.6.5.4	Error reasons for object manipulations .....	42
6.6.6	Error reasons for application layer errors .....	46
6.6.7	Security considerations .....	47
6.7	Design pattern for conditional data node selection.....	47
7	Resource representation formats .....	47
7.1	Introduction .....	47
7.2	Top-level object.....	47
7.3	Data objects .....	48
7.4	Data arrays.....	48
7.5	Error objects .....	48
7.6	Resource objects.....	49
7.7	Resource objects carried in data objects and arrays .....	49
8	REST SS specification template.....	50
<b>Annex A (informative): Examples.....</b>		<b>55</b>
A.1	Example data model.....	55
A.2	Retrieval of resources.....	60
A.2.1	Retrieval of a single complete resource with HTTP GET .....	60
A.2.2	Attribute and attribute field selection on a single resource .....	61
A.2.3	Retrieval of multiple complete resources using scoping and filtering.....	62
A.2.4	Large queries .....	72
A.3	Creation of resources.....	72
A.3.1	Creation of a resource with HTTP PUT .....	72
A.3.2	Creation of a resource with HTTP POST .....	73
A.3.3	Creation of multiple resources with 3GPP JSON Merge Patch.....	74
A.3.4	Creation of multiple resources with 3GPP JSON Patch.....	76
A.4	Deletion of resources.....	78
A.4.1	Deletion of a resource with HTTP DELETE.....	78
A.4.2	Deletion of multiple resources with HTTP DELETE.....	78
A.4.3	Deletion of multiple resources with 3GPP JSON Merge Patch.....	78

A.4.4	Deletion of multiple resources with 3GPP JSON Patch .....	79
A.5	Complete update of a resource .....	79
A.6	Partial update of a resource .....	80
A.6.1	Partial update of a resource with JSON Merge Patch.....	80
A.6.2	Partial update of a resource with 3GPP JSON Merge Patch .....	81
A.6.3	Partial update of a resource with JSON Patch .....	81
A.6.4	Partial update of a resource with 3GPP JSON Patch.....	83
A.7	Manipulating multiple resources .....	84
A.7.1	Manipulating multiple resources with 3GPP JSON Merge Patch .....	84
A.7.2	Manipulating multiple resources with 3GPP JSON PATCH .....	85
A.8	Partitioning a data model.....	87
<b>Annex B (informative):</b>	<b>Change history .....</b>	<b>88</b>
History .....		91

# Sample Document

get full document from [standards.iteh.ai](https://standards.iteh.ai)

---

# 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.

# Sample Document

get full document from [standards.iteh.ai](https://standards.iteh.ai)

---

# 1 Scope

The present document defines design rules for REpresentational State Transfer (REST) Solution Sets (SS). These rules are applied when specifying REST Solution Sets.

---

# 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] IETF RFC 7231: "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content".
- [3] 3GPP TS 32.300: "Telecommunication management; Configuration Management (CM); Name convention for Managed Objects".
- [4] IETF RFC 3986: "Uniform Resource Identifier (URI): Generic Syntax".
- [5] IETF RFC 7230: "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing".
- [6] IETF RFC 7159: "The JavaScript Object Notation (JSON) Data Interchange Format".
- [7] draft-bhutton-json-schema-01 (June 2022): "JSON Schema: A Media Type for Describing JSON Documents".

Note: The above document is an individual draft from IETF. It cannot be formally referenced until it is published as an RFC. It is available from the following link:

<https://datatracker.ietf.org/doc/html/draft-bhutton-json-schema-01>.

- [8] draft-bhutton-json-schema-validation-01 (June 2022): "JSON Schema Validation: A Vocabulary for Structural Validation of JSON".

Note: The above document is an individual draft from IETF. It cannot be formally referenced until it is published as an RFC. It is available from the following link:

<https://datatracker.ietf.org/doc/html/draft-bhutton-json-schema-validation-01>.

- [9] draft-handrews-json-schema-hyperschema-02 (September 2019): "JSON Hyper-Schema: A Vocabulary for Hypermedia Annotation of JSON".

Note: The above document is an individual draft from IETF. It cannot be formally referenced until it is published as an RFC. It is available from the following link:

<https://datatracker.ietf.org/doc/html/draft-handrews-json-schema-hyperschema-02>.

- [10] OpenAPI Specification (<https://github.com/OAI/OpenAPI-Specification>)
- [11] IETF RFC 5789: "PATCH Method for HTTP".
- [12] IETF RFC 7396: "JSON Merge Patch".
- [13] IETF RFC 6902: "JavaScript Object Notation (JSON) Patch".
- [14] IETF RFC 6901: "JavaScript Object Notation (JSON) Pointer".
- [15] XML Path Language (XPath) Version 1.0, W3C Recommendation 16 November 1999 (<https://www.w3.org/TR/xpath-10/>)

- [16] 3GPP TS 32.160: "Management and orchestration; Management service template".
- [17] IETF RFC 4918: "HTTP Extensions for Web Distributed Authoring and Versioning (WebDAV)"
- [18] IETF RFC 6585: "Additional HTTP Status Codes"
- [19] IETF RFC 7807: "Problem Details for HTTP APIs"
- [20] IETF RFC 7725: "An HTTP Status Code to Report Legal Obstacles"
- [21] 3GPP TS 32.161: "JSON expressions (Jex)"

---

## 3 Definitions and abbreviations

### 3.1 Definitions

For the purposes of the present document, the terms and definitions given in 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 TR 21.905 [1].

### 3.2 Abbreviations

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

CRUD	Create, Retrieve, Update, Delete
DC	Domain Component
DN	Distinguished Name
DNS	Domain Name Service
FQDN	Fully Qualified Domain Name
HTTP	Hypertext Transfer Protocol
JSON	JavaScript Object Notation
LDN	Local Distinguished Name
MnS	Management Service
REST	REpresentational State Transfer
RPC	Remote Procedure Call
TCP	Transmission Control Protocol
URI	Uniform Resource Identifier

---

## 4 General rules

### 4.1 Information models and resources

#### 4.1.1 Information models

An information model is a representation of a system. Typical models do not reflect all facets of the system, but only certain aspects required to solve the management problem the model is designed for. 3GPP follows an object-oriented modelling approach. Models are built from managed object classes. Each object class contains information elements called attributes. Relationships between classes represent the logical connections. Models are specified formally with class diagrams produced using the Unified Modelling Language (UML).

The instantiation of a managed object class is called managed object instance, or concisely just managed object or object. All managed object instances together with the relationships between them constitute an object tree. An object tree is also called containment tree.

## 4.1.2 Resources

HTTP uses a different terminology based on the notion of resources, as defined in clause 2 of RFC 7231 [2]. Each resource is represented by one or more resource representations as defined in clause 3 of RFC 7231 [2]. Valid resource representations are e.g. XML instance documents or JSON instance documents.

Besides this primary resource, RFC 3986 [4], clause 3.5 introduces the concept of secondary resources. Secondary resources are specific portions or subsets of primary resources, that are identifiable.

## 4.1.3 Resource archetypes

Resources can be classified according to their structure and behaviour into resource archetypes. This helps specifying clear and understandable interfaces. The following three archetypes are defined:

- **Document resource:** This is the standard resource containing data in form of name value pairs and links to related resources. This kind of resource typically represents a real-world object or a logical concept.
- **Collection resource:** A collection resource is grouping resources of the same kind. The resources below the collection resource are called items of the collection. An item of a collection is normally a document resource. Collection resources typically contain links to the items of the collection and information about the collection like the total number of items in the collection. Collection resources can be further distinguished into server-managed and client-managed resources. Collection resources are also known as container resources.
- **Operation resource:** Operation resources represent executable functions. They may have input and output parameters. Operation resources allow some sort of fall back to an RPC style design in case application specific actions cannot be mapped easily to CRUD style operations.

## 4.1.4 Mapping of information models to resources

RESTful SS shall be specified in a way that managed object instances are described by (primary) document resources. Collection resources have no equivalent in an information model unless some dedicated collection class is introduced.

Attributes are mapped to secondary resources.

## 4.1.5 Usage of information models

Information models are used for two purposes when specifying interfaces to observe and act upon information models:

- They provide a means to identify information in request messages.
- They provide a format to transfer information in request and response messages.
- They provide constraints on the structure of information on the MnS Producer.
- They provide constraints on the possibilities to update information on the MnS Producer.

Identification of information is necessary when retrieving information from a MnS Producer; the MnS Consumer needs to be able to specify in his retrieval request the information the MnS Producer shall return. But also when information needs to be updated or deleted the MnS Consumer needs to identify the information to be updated or deleted in his request. When information is added, the location of the new information is specified relative to the location of existing information.

Request and response message bodies carrying (some parts of) the information model are also constructed based on the information model supported by the MnS Producer. The message format is either identical to the information model format or identical to some transformation of the information model format.

## 4.2 Managed object naming and resource identification

### 4.2.1 Managed object naming

#### 4.2.1.0 Distinguished Name (DN)

The Distinguished Name (DN) is used in 3GPP to uniquely identify a managed object instance within a specific name space. The DN is a comma (",") separated list of Relative Distinguished Names (RDNs). Each managed object instance has an associated RDN. The sequence of RDNs is governed by name containment relationships in the UML class diagram describing the modelled network. The RDN consists of a naming attribute name separated by an equal sign ("=") from the naming attribute value. The naming attribute name is equal to the class name of the MOI.

In addition to the RDNs associated to a managed object instance the DN may have as leftmost RDN whose naming attribute name is "DC" (Domain Component) and whose value is a domain name. A DN with DC is globally unique.

The DN concept is described in detail in TS 32.300 [3]. The following example DN has a DC.

```
DN = "DC=operatorA.com,SubNetwork=south,ManagedElement=a,ENBFunction=1,Cell=1"
```

#### 4.2.1.1 Global and local namespaces

A DN in the global name space is globally unique and starts with the RDN of the global root. A DN in a local name space starts with the RDN of the local root and is unique only within this name space. A DN in a local namespace is also referred to as Local Distinguished Name (LDN). The DN of the local root relative to the global root is called DN prefix. The concatenation of DN prefix and LDN is equal to the globally unique DN of a managed object.

The local root is typically the root of the network resource model representing the managed network.

### 4.2.2 Resource identification

HTTP uses a subset of the generic Uniform Resource Identifier (URI) scheme (RFC 3986 [4]) defined in RFC 7230 [5] for target resource identification.

```
http-URI = "http:" "://" authority path-abempty [ "?" query ] [ "#" fragment ]
```

The path component is an absolute path (one that starts with a single slash character) or empty.

The origin server is identified by the authority component, which includes a host identifier and an optional TCP port. The hierarchical path component and optional query component serve as an identifier for a potential target resource within that origin server's name space. The optional fragment component allows for indirect identification of a secondary resource. The host identifier is either an IP address or an indirect identifier such as a FQDN to be resolved with DNS.

URIs are used by HTTP for routing and addressing of target resources.

### 4.2.3 Mapping of DN to URIs

URIs are globally unique. For this reason only a globally unique DN with DC is mappable into a URI. The mapping rules are as follow:

- The DN prefix is mapped semantically to the authority component of the URI. The syntax of the DN prefix is modified to match the syntax of the authority component.
- The LDN is mapped semantically to the path component of the URI. The syntax of the LDN is modified to match the syntax of the path component.

When mapping a LDN the equal sign "=" shall be used as delineator between the naming attribute name and naming attribute value when constructing a RDN.

```
URI-RDN = {namingAttributeName} "=" {namingAttributeValue}
```

The URI-LDN is the concatenation of URI-RDNs separated by a slash "/".

URI-LDN = \*( "/" RDN )

For example, the LDN

LDN = "SubNetwork=south,ManagedElement=a,ENBFunction=1,Cell=1"  
maps to

URI-LDN = "/SubNetwork=south/ManagedElement=a/ENBFunction=1/Cell=1"

and the LDN

LDN = "ManagedElement=a,ENBFunction=1,Cell=1"  
to

URI-LDN = "/ManagedElement=a/ENBFunction=1/Cell=1"

When constructing the authority part from the DN prefix, it shall be reformatted according to the name conventions applying to FQDNs. For example, the DN prefix

DN-prefix = "DC=operatorA.com"

maps to

URI-DN-prefix = "operatorA.com"

and the DN prefix

DN-prefix = "DC=operatorA.com,SubNetwork=south"

to

URI-DN-prefix = "south.subNetwork.operatorA.com"

The complete URIs for the examples are

<http://operatorA.com/SubNetwork=south/ManagedElement=a/ENBFunction=1/Cell=1>  
<http://south.subNetwork.operatorA.com/ManagedElement=a/ENBFunction=1/cell=1>

The constructed URI-DN-prefix is a FQDN that can be registered into a name resolution service such as DNS. The sole presence of a constructed FQDN does not mean it can be resolved to an IP address and there is a server listening at that address.

Using the mapping rule, a DN is mapped predictably into the URI authority component and path component.

The character set allowed in DNs is much bigger than the character set allowed in the path component and authority component of a URI. Care needs to be taken when selecting the naming attribute names and values that the mapping from a DN to a URI does not become impossible as a consequence of not mappable characters.

When no registered name can be used, the IP address shall be specified directly in the host component, for example:

<http://168.212.226.204/SubNetwork=south/.../Cell=1>

This might be required in multiple situations. For example, when a DN prefix is used but the corresponding URI-DN-prefix cannot be resolved, the MnS Consumer needs to specify an IP address in the target URI of HTTP request messages. The same is true when no DN prefix is used at all. Another example is when no DN prefix is configured into MnS Producers and the MnS Producer wants to report events, that occurred related to resources, using notifications sent to MnS Consumers. The MnS Producer has no other option than to put its own IP address into the host component of the URI identifying the resource where the event occurred.

## 4.2.4 Canonical URI

The URI defined in clause 4.2.3 is called canonical URI. It is the main or official URI of a resource. It shall be used whenever the resource as such shall be identified. The URI for sending HTTP requests to a resource may be different as described in clause 4.4. Special kinds of requests may have all their own URI. Therefore, a resource has typically one

canonical URI and one or more other URIs. The canonical URI may be looked at as a protocol specific version of the protocol neutral DN.

A canonical URI may or may not yield further information if dereferenced.

An example usage of a canonical URI is in event notifications such as alarm notifications for identifying the resource where the event occurred.

## 4.3 Message content formats

### 4.3.1 Media types

The format of HTTP request and response message content is indicated with media types consisting of a type, a subtype and optional parameters, as defined in clause 3.1.1.1 of RFC 7231 [2]. The "Content-Type" header field of a message contains the media type of the message content (clause 3.1.1.5 of RFC 7231 [2]).

If not otherwise stated, the media type of request and response message bodies in the REST SS is

- application/json (RFC 7159 [6]).

Exceptions are when JSON patch documents are contained in request bodies. They are identified with the media types

- application/merge-patch+json (RFC 7396 [12], and clause 6.3.2 of the present document),
- application/json-patch+json (RFC 6902 [13], and clause 6.3.3 of the present document).

Furthermore, this specification defines four new formats. Their media types are

- application/vnd.3gpp.merge-patch+json (clause 6.4.2 of the present document),
- application/vnd.3gpp.json-patch+json (clause 6.4.3 of the present document),
- application/vnd.3gpp.object-tree-hierarchical+json (clause 6.1.4 of the present document),
- application/vnd.3gpp.object-tree-flat+json (clause 6.1.4 of the present document).

JSON documents shall conform to JSON Schema ([7], [8], [9]).

### 4.3.2 Response content format negotiation

The MnS Consumer shall engage in proactive content negotiation as defined in clause 3.4.1 of RFC 7231 [2] by including the "Accept" request header field in HTTP requests that expect a message body in the response. The "Accept" header field indicates to the MnS Producer the media types acceptable to the MnS Consumer.

If the MnS Producer cannot provide any of the acceptable resource representations, it shall respond either with a "406 Not Acceptable" error code or provide a representation for the resource that is not specified in the "Accept" header field.

## 4.4 URI structure

### 4.4.1 Introduction

MnS producers can be divided into two categories. The first category exposes MnS(s) to manipulate resources representing managed object instances. In this case the URI structure is governed by the mapping rules defined in clause 4.2.3. The second category exposes MnS(s) to manipulate resources not representing managed object instances. In this case the DN concept is not relevant. The URI structure for both categories is different.

## 4.4.2 URI structure for resources representing managed object instances

URIs identifying resources representing managed object instances shall follow, when being used as a target URI in HTTP requests, the structure given by

```
{scheme}://{URI-DN-prefix}/{root}/{MnSName}/{MnSVersion}/{URI-LDN}
```

with:

{scheme}	Scheme component "http" or "https"
{URI-DN-prefix}	Authority component (host identifier and optional TCP port), the host name is constructed from the DN prefix as defined in clause 4.2.3.
{root}	Part of the path component, allows specifying one or more optional path segments for structuring the resource hierarchy on a HTTP server. The DN or parts thereof shall not be mapped to this path component.
{MnSName}	Part of the path component, allows specifying an optional MnS name in a single path segment.
{MnSVersion}	Part of the path component, allows specifying an optional MnS version in a single path segment.
{URI-LDN}	Part of the path component, constructed from the LDN as defined in clause 4.2.3, containing zero, one or more path segments.

As seen above, to construct the URI from a DN, it is necessary to map the "DNPrefixPlusRDNSeparator" as defined in clause 7.3 of TS 32.300 [3], the "LocalDN" as defined in clause 7.3 of TS 32.300 [3], and to add the additional optional path segments "{root}/{MnSName}/{MnSVersion}".

To allow for a predictive mapping from an URI to the original DN it is necessary to specify "{MnSName}/{MnSVersion}" in such a way that the beginning of the "LocalDN" can be unambiguously identified.

Note it may be required when specifying a MnS to clearly identify the last RDN of "{URI-LDN}" and to use the following instead of "{URI-LDN}"

```
{URI-LDN-first-part}/{RDN}
```

or

```
{URI-LDN-first-part}/{className}={id}.
```

For the sake of brevity, "MnSRoot" is introduced that includes the "{scheme}" part, the colon (":"), the two slash characters ("/"), the "{authority}" part, a single slash character ("/") and the "{root}" part.

```
{MnSRoot} := {scheme}://{URI-DN-prefix}/{root}
```

When using "{MnSRoot}" the abbreviated URI structure is given by

```
{MnSRoot}/{MnSName}/{MnSVersion}/{URI-LDN}
```

or

```
{MnSRoot}/{MnSName}/{MnSVersion}/{URI-LDN-first-part}/{className}={id}
```

It is recommended to use this abbreviated form of the URI structure when defining Management Services.

The path segment "MnSVersion" allows access to resources with different MnS versions, for example:

```
http://operatorA.com/ProvMnS/v1500/SubNetwork=south/.../Cell=1
http://operatorA.com/ProvMnS/v1600/SubNetwork=south/.../Cell=1
```

Note that both URIs, though different as to the path segment indicating the version number of the ProvMnS, identify the same resource that is identified by the canonical URI:

```
http://operatorA.com/SubNetwork=south/.../Cell=1
```

and whose DN is:

DC=operatorA.com,SubNetwork=south,...,Cell=1

The optional path component "{root}" may be used to separate the name space for 3GPP management from the name space for other domains:

http://operatorA.com/3gppManagement/ProvMnS/v1600/SubNetwork=south/.../Cell=1

or to provide dedicated URIs on the same host for different tasks:

http://operatorA.com/3gppManagement/cm/ProvMnS/v1600/SubNetwork=south/.../Cell=1

http://operatorA.com/3gppManagement/fm/ProvMnS/v1600/SubNetwork=south/.../Cell=1

Note that when different hosts are used for different management tasks, like in

http://cm.operatorA.com/3gppManagement/ProvMnS/v1600/SubNetwork=south/.../Cell=1

http://fm.operatorA.com/3gppManagement/ProvMnS/v1600/SubNetwork=south/.../Cell=1

then also the resources are different and identified by the canonical URIs

http://cm.operatorA.com/SubNetwork=south/.../Cell=1

http://fm.operatorA.com/SubNetwork=south/.../Cell=1

or the DNSs

DC=cm.operatorA.com,SubNetwork=south,...,Cell=1

DC=fm.operatorA.com,SubNetwork=south,...,Cell=1

In the example above, it is assumed that both resources represent the same cell in the network. This information cannot be derived from the DN or canonical URI, though.

### 4.4.3 URI structure for resources not representing managed object instances

URIs identifying other resources shall follow, when being used as a target URI in HTTP requests, the structure given by

{scheme}://{authority}/{root}/{MnSName}/{MnSVersion}/{MnSResourcePath}

with:

{scheme}	Scheme component "http" or "https"
{authority}	Authority component (host identifier and optional TCP port)
{root}	Part of the path component, allows specifying optional path segments for structuring the resource hierarchy on a HTTP server.
{MnSName}	Part of the path component, specifies the mandatory MnS name in a single path segment.
{MnSVersion}	Part of the path component, specifies the mandatory MnS version in a single path segment.
{MnSResourcePath}	Part of the path component, one or more path segments, specifies a resource of the MnS

For the sake of brevity, {MnSRoot} is introduced that includes the "{scheme}" part, the two slash characters ("//"), the "{authority}" part, a single slash character ("/") and the "{root}" part. When using "{MnSRoot}" the abbreviated URI structure is given by

{MnSRoot}/{MnSName}/{MnSVersion}/{MnSResourcePath}

It is recommended to use this abbreviated form of the URI structure when defining Management Services.

### 4.4.4 Resource "{MnSName}/{MnSVersion}"

The resource identified by "{MnSName}/{MnSVersion}" is called NRM root. It represents the conceptual parent of the top-level managed object instances. It is created by the MnS Producer. A MnS Consumer cannot create or delete this resource.