International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

X.786

(01/2022)

SERIES X: DATA NETWORKS, OPEN SYSTEM COMMUNICATIONS AND SECURITY

OSI management – Management functions and ODMA functions

Guidelines for implementation conformance statement proformas associated with REST-based management systems

Recommendation ITU-T X.786

ITU-T X-SERIES RECOMMENDATIONS

DATA NETWORKS, OPEN SYSTEM COMMUNICATIONS AND SECURITY

PUBLIC DATA NETWORKS	
Services and facilities	X.1–X.19
Interfaces	X.20–X.49
Transmission, signalling and switching	X.50–X.89
Network aspects	X.90–X.149
Maintenance	X.150–X.179
Administrative arrangements	X.180–X.199
OPEN SYSTEMS INTERCONNECTION	
Model and notation	X.200–X.209
Service definitions	X.210–X.219
Connection-mode protocol specifications	X.220–X.229
Connectionless-mode protocol specifications	X.230–X.239
PICS proformas	X.240–X.259
Protocol Identification	X.260–X.269
Security Protocols	X.270–X.279
Layer Managed Objects	X.280–X.289
Conformance testing	X.290–X.299
INTERWORKING BETWEEN NETWORKS	
General	X.300–X.349
Satellite data transmission systems	X.350–X.369
IP-based networks	X.370–X.379
MESSAGE HANDLING SYSTEMS	X.400–X.499
DIRECTORY	X.500–X.599
OSI NETWORKING AND SYSTEM ASPECTS	
Networking	X.600–X.629
Efficiency	X.630–X.639
Quality of service	X.640–X.649
Naming, Addressing and Registration	X.650–X.679
Abstract Syntax Notation One (ASN.1)	X.680–X.699
OSI MANAGEMENT	
Systems management framework and architecture	X.700–X.709
Management communication service and protocol	X.710–X.719
Structure of management information	X.720–X.729
Management functions and ODMA functions	**X.730–X.799**
SECURITY	X.800–X.849
OSI APPLICATIONS	
Commitment, concurrency and recovery	X.850–X.859
Transaction processing	X.860–X.879
Remote operations	X.880–X.889
Generic applications of ASN.1	X.890–X.899
OPEN DISTRIBUTED PROCESSING	X.900–X.999
INFORMATION AND NETWORK SECURITY	X.1000–X.1099
SECURE APPLICATIONS AND SERVICES (1)	X.1100–X.1199
CYBERSPACE SECURITY	X.1200–X.1299
SECURE APPLICATIONS AND SERVICES (2)	X.1300–X.1499
CYBERSECURITY INFORMATION EXCHANGE	X.1500–X.1599
CLOUD COMPUTING SECURITY	X.1600–X.1699
QUANTUM COMMUNICATION	X.1700–X.1729
DATA SECURITY	X.1750–X.1799
IMT-2020 SECURITY	X.1800–X.1819

For further details, please refer to the list of ITU-T Recommendations.

Recommendation ITU-T X.786

Guidelines for implementation conformance statement proformas associated with REST-based management systems

Summary

Recommendation ITU-T X.786 establishes guidelines for implementation conformance statement (ICS) proformas for interface systems based on representational state transfer (REST). Recommendation ITU-T X.786 provides an overview and constructions for the open application programming interface (OpenAPI) specification (OAS), and provides several proformas (tables) for each OAS syntax component to be used in REST-based interfaces. Instructions on how to complete the columns in the conformance tables are also provided. Examples of REST-based interface ICSs are provided in appendices.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T X.786	2022-01-13	2	11.1002/1000/14836

Keywords

Conformance testing, implementation conformance statement (ICS), representational state transfer (REST).

* To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents/software copyrights, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the appropriate ITU-T databases available via the ITU-T website at http://www.itu.int/ITU-T/ipr/.

Table of Contents

Recommendation ITU-T X.786

Guidelines for implementation conformance statement proformas associated with REST-based management systems

1 Scope

This Recommendation establishes guidelines for implementation conformance statement (ICS) proformas for interface systems based on representational state transfer (REST) and the specification of these proformas. The REST-based interface implementation conformance statement (RIICS) is made by an implementer to claim conformance to a REST-based interface specification. This Recommendation provides the concepts of conformance testing of REST-based interfaces, describes the containment relationship of the YAML (Yet Another Markup Language) Ain't a Markup Language (YAML) elements, designs the support tables needed for the ICSs, and provides guidelines for completing ICS proformas.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T Q.819] Recommendation ITU-T Q.819 (2022): *REST-based management services*.

[ITU-T X.724] Recommendation ITU-T X.724 (1996), *Information technology – Open Systems Interconnection – Structure of management information: Requirements and guidelines for implementation conformance statement proformas associated with OSI management*.

[ITU-T X.785] Recommendation ITU-T X.785 (2021), *Guidelines for defining REST-based management objects and management interfaces*.

[IETF RFC 3339] IETF RFC 3339 (2002), *Date and time on the Internet: Timestamps.*

[IETF RFC 3986] IETF RFC 3986 (2005), *Uniform resource identifier (URI): Generic syntax.*

[IETF RFC 6838] IETF RFC 6838 (2013), *Media type specifications and registration procedures.*

[IETF RFC 7231] IETF RFC 7231 (2014), *Hypertext transfer protocol (HTTP/1.1): Semantics and content*.

[IETF RFC 8259] IETF RFC 8259 (2017), *The JavaScript object notation (JSON) data interchange format*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 abstract test case [b-ITU-T X.290]: A complete and independent specification of the actions required to achieve a specific test purpose, defined at the level of abstraction of a particular Abstract Test Method, starting in a stable testing state and ending in a stable testing state. This specification may involve one or more consecutive or concurrent connections.

NOTE 1 – The specification should be complete in the sense that it is sufficient to enable a test verdict to be assigned unambiguously to each potentially observable test outcome (i.e., sequence of test events).

NOTE 2 – The specification should be independent in the sense that it should be possible to execute the derived executable test case in isolation from other such test cases (i.e., the specification should always include the possibility of starting and finishing in the "idle" state).

3.1.2 executable test case [b-ITU-T X.290]: A realization of an abstract test case.

3.1.3 ICS proforma specification [b-ITU-T X.296]: The specification which provides a complete ICS proforma.

3.1.4 (ICS (proforma)) item [b-ITU-T X.290]: A row in an ICS (proforma) table.

3.1.5 (ICS (proforma)) question [b-ITU-T X.290]: The question to be answered in the intersection of an ICS item and either a support column (i.e., "Is this item supported in the context applying to this table and column?") or supported values column (i.e., "What values are supported for this item in the context applying to this table and column?") in an ICS proforma table.

3.1.6 ICS template [b-ITU-T X.296]: A template which is to be used as the basis for developing an ICS proforma.

3.1.7 status (value) [b-ITU-T X.290]: An allowed entry in the status column for an item in an ICS proforma table.

3.1.8 (support) answer [b-ITU-T X.290]: An allowed entry in the support or supported values columns for an item in an ICS, in answer to an ICS question.

3.1.9 test case [b-ITU-T X.290]: An abstract or executable test case.

NOTE – In general the use of the word "test" in [b-ITU-T X.290] to [b-ITU-T X.296] will imply its normal English meaning. Sometimes it may be used as an abbreviation for abstract test case or executable test case. The context should make the meaning clear.

3.2 Terms defined in this Recommendation

None.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

API Application Programming Interface

HTTP Hypertext Transfer Protocol

ICS Implementation Conformance Statement

IT Information Technology

IUT Implementation Under Test

JSON JavaScript Object Notation

OAS Open API Specification

REST Representational State Transfer

RIICS REST-based Interface Implementation Conformance Statement

URI Uniform Resource Identifier

YAML YAML (Yet Another Markup Language) Ain't a Markup Language

5 Conventions

A few conventions are followed in this Recommendation to make the reader aware of the purpose of the text.

Examples of JavaScript object notation (JSON) and YAML schema are included in this Recommendation. The JSON/YAML schemas are presented in `10 point Courier New` typeface as follows.

A JSON schema example

```
{
    "title": "root",
    "items": {
        "title": "array item"
    }
}
```

A YAML schema example

```
SomeType:
  type: object
  properties:
    attr1:
      type: string
    attr2:
      type: string
      enum:
        - e1
        - e2
```

6 Overview of conformance testing for REST-based management system interface

REST-based technologies have been widely used in the information technology (IT) industry. [ITU-T X.785] and [ITU-T Q.819] establish a REST-based management paradigm that provides guidelines for using REST technology in network management interfaces.

Conformance relates an implementation to a standard. It states in which way systems, implemented with respect to a standard, can vary without errors occurring in their cooperation. If an implementation fulfils these requirements, then it conforms to the standard. The check of the statements is the conformance test. The starting point is the determination of conformance requirements in implementation independent interface specifications on the bases of an identification of reference points. A management interface specification should list conformance reference points on which an object must be tested to check whether it fulfils a set of conformance criteria. During the test, a number of stimuli and events are observed and evaluated on these conformance points. Management interface specifications should include conformance statements that identify conformance reference points at every interface of the specified objects. Because in general the information flow between two system components is realized through several reference points, the conformance test has to take into consideration:

a) the test of such information flow at each reference point; and

b) the test of consistency between the combinations.

A coordinated test at all identified reference points is therefore necessary.

7 REST-based management interface ICS proformas

The open application programming interface (OpenAPI) Specification (OAS) is used to determine REST interfaces for REST-based systems (in JSON/YAML schema language, see [IETF RFC 8259] and [b-OAI-OAS3] for more details). The RIICS proforma shall express OAS features such as server, paths, operations, requestBodies, responses and data types. This clause first introduces the OAS features associated with the interface implementation conformance test requirements, and then specifies the RIICS proforma according to these features.

7.1 OAS overview

Systems that interact through a REST interface mainly exchange data through hypertext transfer protocol (HTTP) messages. As long as the transmitted HTTP message conforms to the REST interface specification, the implementation of the interface conforms to the agreement of both parties, which also meets the implementation conformance requirements of the REST interface.

Figure 1 illustrates an example of an OpenAPI map, which shows the relationship among some OpenAPI syntax elements.

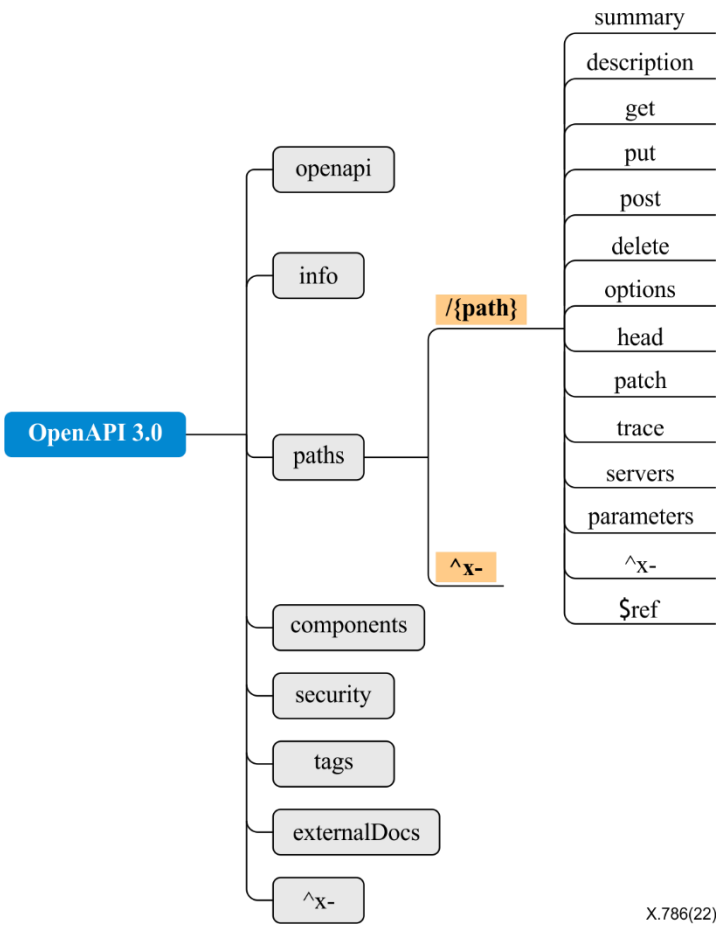

Figure 1 – An OpenAPI map example

OAS is the guiding standard for REST interface specifications. Some OAS syntax elements are closely related to RIICSs, while others are not. With reference to HTTP message structure, some OAS syntax elements can be included in a subset in which OAS syntax elements are closely related to RIICSs (called a subset for ICS). According to OAS, syntax elements can be organized into a tree structure. As a result, OAS syntax elements in a subset for ICS can also be organized into a tree structure. OAS syntax elements in a subset for ICS and their containment and association relationship are illustrated in Figure 2.

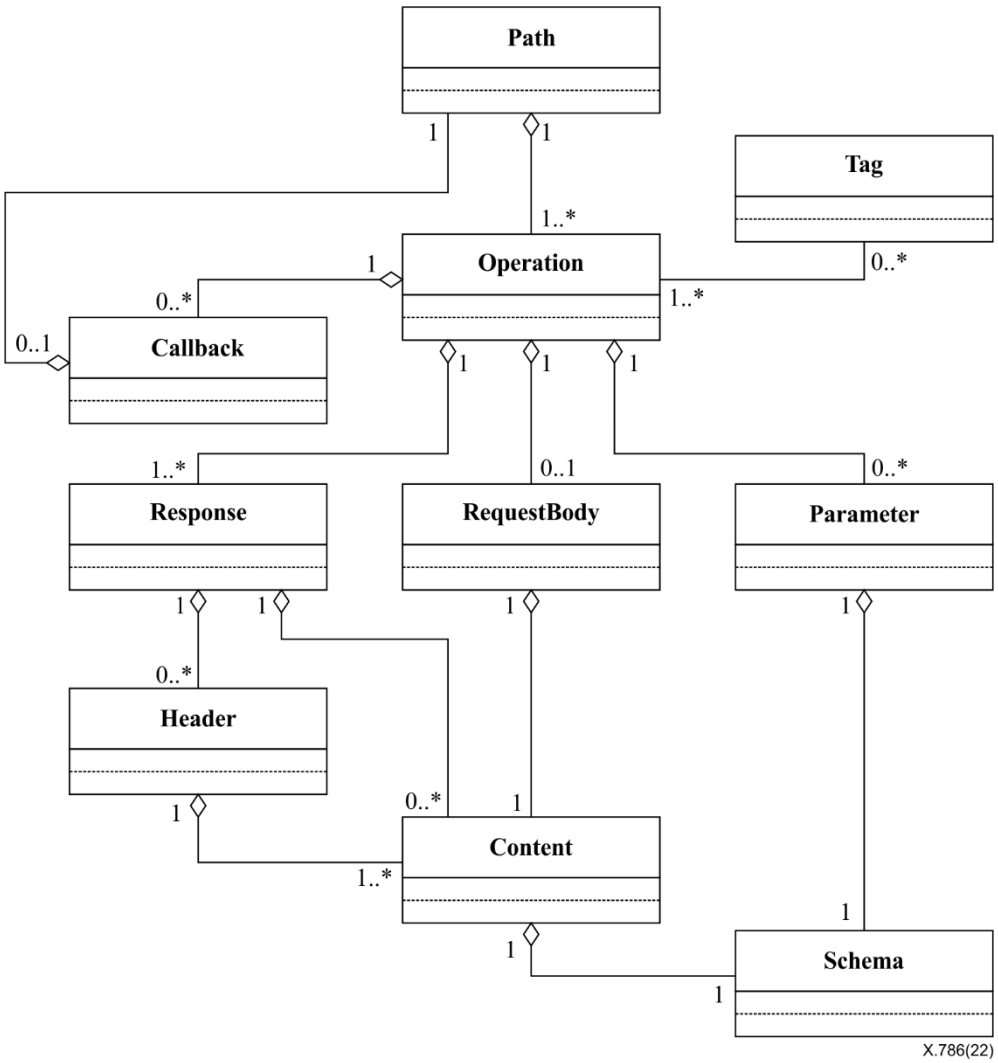

Figure 2 – Containment relationship of OAS syntax elements in a subset for ICS

7.2 Guidelines for specification of RIICS proformas

Proforma specifications shall follow the style specified in clauses 7.2.1 to 7.2.11. Proforma specifications shall provide the information required by this Recommendation.

There are three levels of documentation pertaining to RIICS, namely:

1) guidelines or recommendation tools for the production of RIICS proformas;
2) the RIICS proforma, which is associated with a standard related to REST-based network management and is to be completed by a supplier of the implementation;
3) the completed RIICS prepared by a supplier of the implementation as part of a conformance claim to a standard related to REST-based network management.

7.2.1 General instructions for RIICS proforma specification

This Recommendation provides instructions to construct REST-based management system ICS proforma specifications. RIICS proformas include 10 sub-proformas: path support proforma; operation support proforma; parameter support proforma; requestBody support proforma; response support proforma; header support proforma; content support proforma; dataType support proforma; callback support proforma; and tag profile. All of these proformas, described in clauses 7.2.2 to 7.2.11, are in a tabular form and are similar to those specified in [ITU-T X.724]. Appendices I and II provide examples of RIICS proforma specifications that are to be completed by a supplier of an implementation.

The following common notations, specified in [b-ITU-T X.291] and [b-ITU-T X.296], are used for the column headed "Status" in Tables 7-1 to 7-9:

m mandatory

o optional

– not applicable or outside the scope.

NOTE – The notations "m" or "o" are prefixed by a "c:" when nested under an optional item in the same table.

The following common notations, specified in [b-ITU-T X.291] and [b-ITU-T X.296] are used for the column headed "Support" in Tables 7-1 to 7-9:

Y implemented

N not implemented

– no answer required

Ig the item is ignored (i.e., processed syntactically but not semantically).

The RIICS proforma specification is formed by copying clauses 7.2.2 to 7.2.11, completing Tables 7-1 to 7-10 except for the columns headed "Support" and "AdditionalInfo". An RIICS proforma shall provide tables for OAS syntax elements that are not only defined in the REST interface specification, but also contained in the subset for ICS. To form an RIICS from an RIICS proforma, the supplier of the implementation shall fill in the "Support" and, if appropriate, "AdditionalInfo" columns of all the tables in the RIICS proforma.

7.2.2 Proforma for path support

The purpose of the proforma for path support is to provide a mechanism for a supplier of an implementation who claims conformance to a path specification to provide conformance information in a standard form.

The path support proforma is given in Table 7-1.

Table 7-1 – Table for path support

Index	SubIndex	PathIdentifier	AdditionalInfo	Status	Support

Descriptions follow of the column headings in Table 7-1.

– The "Index" field contains consecutive numbers for readers of the RIICS to refer to each support table. Each support table is given a unique number as table index, which is also a part of the table name, according to the containment relationship of OAS syntax elements defined in the REST interface specification. All path items in a path support table are given the same index as that of the table.

– The "SubIndex" field contains consecutive numbers for readers of the RIICS to refer to each path item. Each path defined in the REST interface specification is given a unique number as its sub-index.

– The "PathIdentifier" field is the absolute name of the path, which is actually part of the uniform resource identifier (URI).

– The "AdditionalInfo" field is used to provide a space for the implementer to add more specific information about this path.

– For each path, in the "Status" field, if it is mandatory, "m" should be entered; if it is optional, "o" should be entered.

– The "Support" field is used by suppliers to claim whether this path is supported.

7.2.3 Proforma for operation support

The purpose of the proforma for operation support is to provide a mechanism for a supplier of an implementation who claims conformance to an operation specification to provide conformance information in a standard form.

The operation support proforma is given in Table 7-2.

Table 7-2 – Table for operation support

Index	SubIndex	OperationID	HTTPMethod	AdditionalInfo	Status	Support

Descriptions follow of the column headings in Table 7-2.

– The "Index" field specifies the index of the parent path to which this operation belongs. All operation items in an operation support table are given the same index as that of the table.

– The "SubIndex" field is the unique reference of the operation.

– The "OperationID" field indicates the operation name.

– The "HTTPMethod" field indicates the request method of the operation. Its value complies with the provisions of the HTTP.

– The "AdditionalInfo" field is used to provide a space for the implementer to add more specific information about this operation.

– For each operation, in the "Status" field, if it is mandatory, "m" should be entered; if it is optional, "o" should be entered.

– The "Support" field is used by suppliers to claim whether an operation is supported.

7.2.4 Proforma for parameter support

The purpose of the proforma for parameter support is to provide a mechanism for a supplier of an implementation who claims conformance to a parameter specification to provide conformance information in a standard form.

The parameter support proforma is given in Table 7-3.

Table 7-3 – Table for parameter support

Index	SubIndex	ParameterName	ParameterPosition	AllowEmpty	AdditionalInfo	Status	Support

Descriptions follow of the column headings in Table 7-3.

– The "Index" field specifies the index of the parent operation to which this parameter belongs.

– The "SubIndex" field is the unique reference of a parameter.

– The "ParameterName" field is the absolute name of the parameter.

– The "ParameterPosition" field specifies the position of the parameter. Possible values are "query", "header", "path" or "cookie".

– The "AllowEmpty" field sets the ability to pass empty-valued parameters. This is valid only for query parameters and allows sending a parameter with an empty value.

– The "AdditionalInfo" field is used to provide a space for the implementer to add more specific information about this parameter.

In addition, attributes related to parameter serialization rules can also be entered in this field.

– For each parameter, in the "Status" field, if it is mandatory, "m" should be entered; if it is optional, "o" should be entered.

– The "Support" field is used by suppliers to claim whether this parameter is supported.

7.2.5 Proforma for requestBody support

The purpose of the proforma for requestBody support is to provide a mechanism for a supplier of an implementation who claims conformance to a request body specification to provide conformance information in a standard form.

The requestBody support proforma is given in Table 7-4.

Table 7-4 – Table for requestBody support

Index	SubIndex	RequestBodyName	AdditionalInfo	Status	Support

Descriptions follow of the column headings in Table 7-4.

– The "Index" field specifies the index of the parent operation of this request body.

– The "SubIndex" field contains consecutive numbers for readers of the RIICS to refer to each request body.

– The "RequestBodyName" field is the absolute name of the request body.

– The "AdditionalInfo" field is used to provide a space for the implementer to add more specific information about this request body.

– For each request body, in the "Status" field, if it is mandatory, "m" should be entered; if it is optional, "o" should be entered.

– The "Support" field is used by suppliers to claim whether this requestBody is supported.

7.2.6 Proforma for response support

The purpose of the proforma for response support is to provide a mechanism for a supplier of an implementation who claims conformance to a response specification to provide conformance information in a standard form.

The response support proforma is given in Table 7-5.

Table 7-5 – Table for response support

Index	SubIndex	HTTPStatusCode	AdditionalInfo	Status	Support

Descriptions follow of the column headings in Table 7-5.

– The "Index" field specifies the index of the parent operation of this response.

– The "SubIndex" field contains consecutive numbers for readers of the RIICS to refer to each response.

– The "HTTPStatusCode" field shows the HTTP status code indicating the status of the executed operation. The available status codes are specified in [IETF RFC 7231].

– The "AdditionalInfo" field is used to provide a space for the implementer to add more specific information about this response.

– For each response, in the "Status" field, if it is mandatory, "m" should be entered; if it is optional, "o" should be entered with.
– The "Support" field is used by suppliers to claim whether this response is supported.

7.2.7 Proforma for header support

The purpose of the proforma for header support is to provide a mechanism for a supplier of an implementation who claims conformance to a response header specification to provide conformance information in a standard form.

The header support proforma is given in Table 7-6.

Table 7-6 – Table for header support

Index	SubIndex	AttributeName	AdditionalInfo	Status	Support

Descriptions follow of the column headings in Table 7-6.

– The "Index" field specifies the index of the parent response of this header.
– The "SubIndex" field contains consecutive numbers for readers of the RIICS to refer to each header attribute.
– The "AttributeName" field is the name of an attribute in the response header.
– The "AdditionalInfo" field is used to provide a space for the implementer to add more specific information about this header attribute.
– For each header attribute, in the "Status" field, if it is mandatory, "m" should be entered; if it is optional, "o" should be entered.
– The "Support" field is used by suppliers to claim whether this header attribute is supported.

7.2.8 Proforma for content support

The purpose of the proforma for content support is to provide a mechanism for a supplier of an implementation who claims conformance to a content specification to provide conformance information in a standard form.

The content support proforma is given in Table 7-7.

Table 7-7 – Table for content support

Index	SubIndex	MediaType	AdditionalInfo	Status	Support

Descriptions follow of the column headings in Table 7-7.

– The "Index" field specifies the index of the parent syntax element of this content, and according to Figure 2, the parent syntax element can be either a requestBody, a response, or a header.
– The "SubIndex" field contains consecutive numbers for readers of the RIICS to refer to each content type.
– The "MediaType" field specifies the media type of transmitted data. Values of this field should comply with [IETF RFC 6838].
– The "AdditionalInfo" field is used to provide a space for the implementer to add more specific information about this content.

– For each content type, in the "Status" field, if it is mandatory, "m" should be entered; if it is optional, "o" should be entered.
– The "Support" field is used by suppliers to claim whether this content is supported.

7.2.9 Proforma for dataType support

The purpose of the proforma for dataType support is to provide a mechanism for a supplier of an implementation who claims conformance to a data type specification to provide conformance information in a standard form.

The dataType support proforma is given in Table 7-8.

Table 7-8 – Table for dataType support

Index	SubIndex	Name	SchemaType	DataType	Format	Constraints	AdditionalInfo	Status	Support

Descriptions follow of the column headings in Table 7-8.

– The "Index" field specifies the index of the parent syntax element of this data type (in the schema).
– The "SubIndex" field contains consecutive numbers for readers of the RIICS to refer to each data type.
– The "Name" field shows the name of the data type.
– The "SchemaType" field is used to describe the association between the data type and other data types. There are three optional values for this column: object.property, array.item, or null.
– The "DataType" field represents the basic type of a data type. This attribute is primitively taken from the specification of data type in a JSON/YAML schema and adjusted to the OAS. The values that can be entered are "null", "string", "boolean", "number", "integer", "object" and "array".
– The "Format" field is used to indicate the detailed data type being used.
– The "Constraints" field records the constraints on the data type, such as default value, maximum value, minimum value, maximum length, minimum length and so on.
– The "AdditionalInfo" field is used to provide a space for the implementer to add more specific information about this data type.
– For each data type, in the "Status" field, if it is mandatory, "m" should be entered; if it is optional, "o" should be entered.
– The "Support" field is used by suppliers to claim whether this data type is supported.

7.2.10 Callback support proforma

The purpose of the proforma for callback support is to provide a mechanism for a supplier of an implementation who claims conformance to a callback specification to provide conformance information in a standard form.

The callback support proforma is given in Table 7-9.

Table 7-9 – Table for callback support

Index	SubIndex	EventName	AdditionalInfo	Status	Support

Descriptions follow of the column headings in Table 7-9.

– The "Index" field specifies the index of the parent operation of this callback.
– The "SubIndex" field contains consecutive numbers for readers of the RIICS to refer to each callback.
– The "EventName" field shows the name of the callback event.
– The "AdditionalInfo" field is used to provide a space for the implementer to add more specific information about this callback.
– For each callback, in the "Status" field, if it is mandatory, "m" should be entered; if it is optional, "o" should be entered.
– The "Support" field is used by suppliers to claim whether this content is supported.

7.2.11 Tag profile

In a REST interface definition specification, the Tag object is used to classify operations. In a RIICS proforma, a tag profile makes REST interface elements easier to be grouped and navigated.

The tag profile is given in Table 7-10.

Table 7-10 – Tag profile

Index	TagIdentifier	PathIndex	PathIdentifier	OperationIndex	OperationID

Descriptions follow of the column headings in Table 7-10.

– The "Index" field in the tag profile specifies the index of the tag, which is a number.
– The "TagIdentifier" field shows the name of the tag.
– The "PathIndex" field indicates the index of the path whose operation is marked by the tag.
– The "PathIdentifier" field indicates the name of the path whose operation is marked by the tag.
– The "OperationIndex" field indicates the index of the operation marked by the tag, which should have the same value as the "Subindex" field in the corresponding operation support table of the same operationID.
– "OperationID" field indicates the name of the operation marked by the tag.

8 Instructions for completing RIICS proformas

This clause gives the instructions for completing each column type described in Tables 7-1 to 7-10.

8.1 Description of "supported"

A capability is supported if the implementation under test (IUT) is able to realize the specified functionality.

8.2 The "Index" column

The "Index" column, located in the leftmost column of each row, is used to uniquely identify all support tables in the RIICS proforma. The column is made up of consecutive numbers separated by the character ".", which shows the containment relationship between OAS syntax elements of a REST interface specification. For the containment relationship, see clause 7.1.

As the basis of index number recursion, "–" is entered in the "Index" column when the item is one of the topmost path syntax elements in an OAS syntax tree. Otherwise, for each item in an RIICS proforma, the "Index" column specifies the super-clause item (containing or parent item with a superior clause number in index) of the item by copying the value of its "SubIndex" column.

8.3 The "SubIndex" column

The "SubIndex" column is used to uniquely identify all possible implementation details within the RIICS proforma. It has a similar meaning and the same format as the "Index" column.

The subindex number is constructed with a sequence as follows:

a) a reference to the super-clause of the item;

b) the separating character ".";

c) a unique number.

For each row, the "Index" column is a reference to its super-clause item, while the "SubIndex" column identifies the item itself.

The rules in the preceding paragraphs also apply for the "PathIndex" and "OperationIndex" columns.

8.4 The "Support" column

The "Support" column shall be completed by the supplier or implementer to indicate the level of support provided by the implementation for each item. The available selections for this field are listed in clause 7.2.1.

Guidelines for completing this field follow.

a) If an item is claimed as "supported", all mandatory items it contains must also be supported. Otherwise, "N" is entered into the "Support" column.

b) If "–" is entered in the "Status" column for an item, the only selection for the corresponding "Support" column is "–".

c) In the RIICS proforma tables, every item marked with "m" should be supported by the IUT.

8.5 The "Additional information" column

The "Additional information" column may contain information provided by suppliers that is not contained anywhere else. To complete an RIICS for each proforma, there may be some important and necessary information that is not or cannot be contained in any column of the row. In such cases, suppliers can enter such information in this field.

8.6 The "PathIdentifier" column

The value is entered in the "PathIdentifier" column of the "path support proforma" as part of an URI in string format, which should comply with [IETF RFC 3986].

8.7 The "OperationID" column

The "OperationID" column in the "operation support proforma" can be used to identify and distinguish interface requests in a REST interface specification. The OperationID must be unique among all operations described in an RIICS proforma.

8.8 The "HTTPMethod" column

The "HTTPMethod" column in the "operation support proforma" records the name of the request method in string format, such as "GET", "POST", "UPDATE" or "DELETE". The value to be entered in this column must conform to the specifications of [IETF RFC 7231].

8.9 The "ParameterPosition" column

The "ParameterPosition" column in the "parameter support proforma" shows the location of parameters. Possible values are "query", "header", "path" or "cookie", which correspond to parameters for queries, request headers, paths and cookies, respectively.

8.10 The "AllowEmpty" column

The "AllowEmpty" column in the "parameter support proforma" sets the ability to pass empty-valued parameters. This column is valid only for query parameters and allows sending a parameter with an empty value. The default value is "false".

8.11 The "HTTPStatusCode" column

The "HTTPStatusCode" column in the "response support proforma" indicates the name of the response. Each HTTP status code corresponds to a response message, so it can be entered in this column with the HTTP response status code as the name of the HTTP response. The possible status code values can be found in [IETF RFC 7231].

8.12 The "AttributeName" column

The "AttributeName" column in the "header support proforma" records the name of the attributes in the response header. The values of this column are not case sensitive. The media type of the transmitted data is determined in the "Content support proforma" in the RIICS proforma, so when the attribute name in the response header is "Content-Type", the attribute is ignored.

8.13 The "MediaType" column

The "MediaType" column in the "content support proforma" specifies the media type of the transmitted data. The value of the media type must conform to the definition in [IETF RFC 6838].

8.14 The "SchemaType" column

The "SchemaType" column in the "dataType support proforma" is used to describe the association between the data type represented by this row and other data types.

There are three optional values for this column:

a) when the row of data is an attribute of an object, "object.property" is entered in this column;

b) when the row of data is an item in an array, "array.item" is entered in this column;

c) this column is left empty when the data in this row is independent of other data types.

8.15 The "DataType" column

The "DataType" column in the "dataType support proforma" represents the basic type of a data type. This column is taken from the definition of data type in a JSON/YAML schema, and adjusted according to the OAS. The values that can be entered are "null", "string", "boolean", "number", "integer", "object" and "array".

8.16 The "Format" column

The "Format" column in the "dataType support proforma" indicates the data format of a data type. The values that can be entered in this column are listed in Table 8-1.

Table 8-1 – Basic data types

type	format	Descriptions
integer	int32	signed 32 bits
integer	int64	signed 64 bits (also known as "long")
number	float	single precision floating point real number
number	double	double precision floating point real number
string		an unbounded list of normal characters
string	byte	base64 encoded characters
string	binary	any sequence of octets
boolean		the value can be true (1) or false (0)
string	date	as defined by full-date – [IETF RFC 3339]
string	date-time	as defined by date-time – [IETF RFC 3339]
string	password	a hint to UIs to obscure input

8.17 The "Constraints" column

The "Constraints" column in the "dataType support proforma" is used to identify other constraints on a data type, such as default, maximum or minimum. Zero or more key-value pairs in string format and separated with semicolons ";" are entered in this column. The format is "key1=value1; key2=value2; …".

Appendix I

Examples of REST-based interface ICS proforma specifications

(This appendix does not form an integral part of this Recommendation.)

I.1 YAML definition example

This clause provides a YAML interface definition example, based on which the example RIICS proformas are illustrated. Table I.1-1 is an example of a REST interface definition:

Table I.1-1 – An example of a REST interface definition

```
openapi: 3.0.3
info:
  title: TestCase
  version: 1.0.0
tags:
  - name: PhysicalNetwork
  - name: VirtualNetwork
paths:
  /pn:
    get:
      tags:
        - PhysicalNetwork
      operationId: getPN
      responses:
        '200':
          description: success
          content:
            application/json:
              schema:
                  type: object
                  properties:
                    pnName:
                      type: string
        '404':
          description: PN not found
          content: {}
        '405':
          description: Validation exception
          content: {}
  /vn:
    get:
      tags:
      - VirtualNetwork
      operationId: getVN
      parameters:
        - name: vnID
          in: query
          required: false
          schema:
            type: string
      responses:
        '200':
          description: success
          content:
            application/json:
              schema:
                type: object
                properties:
                  vnName:
```

```
                    type: string
            headers:
              'Content-Length':
                schema:
                  type: string
      post:
        tags:
        - VirtualNetwork
        operationId: postVN
        requestBody:
          content:
            application/json:
              schema:
                type: object
                properties:
                  vnName:
                    type: string
        responses:
          '200':
            description: success
        callbacks:
          vnCallback:
              '{$vnCallback}':
                post:
                  requestBody:
                    description: Callback payload
                    content:
                      application/json:
                        schema:
                          type: integer
                          format: int32
                  responses:
                    '200':
                      description: success
```

I.2 RIICS proforma example

RIICS proforma examples are provided in this clause, based on the YAML interface definition provided in clause I.1.

1 Tag profile

Index	TagIdentifier	PathIndex	PathIdentifier	OperationIndex	OperationID
1.	"PhysicalNetwork"	2.1	"/pn"	2.1.1	"getPN"
2.	"VirtualNetwork"	2.2	"/vn"	2.2.1	"getVN"
			"/vn"	2.2.2	"postVN"

2 Table for path support

Index	SubIndex	PathIdentifier	AdditionalInfo	Status	Support
	2.1	"/pn"		m	
	2.2	"/vn"		m	

2.1 Table for operation support

Index	SubIndex	OperationID	HTTPMethod	AdditionalInfo	Status	Support
2.1	2.1.1	"getPN"	"get"		m	

2.1.1 Table for response support

Index	SubIndex	HTTPStatusCode	AdditionalInfo	Status	Support
2.1.1	2.1.1.3.1	"200"		m	
2.1.1	2.1.1.3.2	"404"		m	
2.1.1	2.1.1.3.3	"405"		m	

2.1.1.3.1 Table for content support

Index	SubIndex	MediaType	AdditionalInfo	Status	Support
2.1.1.3.1	2.1.1.3.1.2.1	"application/json"		m	

2.1.1.3.1.2.1 Table for dataType support

Index	SubIndex	Name	SchemaType	DataType	Format	Constraints	AdditionalInfo	Status	Support
2.1.1.3.1.2.1	2.1.1.3.1.2.1.1			"object"				m	
2.1.1.3.1.2.1	2.1.1.3.1.2.1.1.1	"pnName"	object.property	"string"				m	

2.1.1.3.2 Table for content support

Index	SubIndex	MediaType	AdditionalInfo	Status	Support
2.1.1.3.2	2.1.1.3.2.2.1				

2.1.1.3.3 Table for content support

Index	SubIndex	MediaType	AdditionalInfo	Status	Support
2.1.1.3.3	2.1.1.3.3.2.1				

2.2 Table for operation support

Index	SubIndex	OperationID	HTTPMethod	AdditionalInfo	Status	Support
2.2	2.2.1	"getVN"	"get"		m	
2.2	2.2.2	"postVN"	"post"		m	

2.2.1 Table for parameter support

Index	SubIndex	ParameterName	ParameterPosition	AllowEmpty	AdditionalInfo	Status	Support
2.2.1	2.2.1.1.1	"vnID"	"query"	true		o	

2.2.1 Table for response support

Index	SubIndex	HTTPStatusCode	AdditionalInfo	Status	Support
2.2.1	2.2.1.3.1	"200"		m	

2.2.1.3.1 Table for content support

Index	SubIndex	MediaType	AdditionalInfo	Status	Support
2.2.1.3.1	2.2.1.3.1.2.1	"application/json"		m	

2.2.1.3.1.2.1 Table for dataType support

Index	SubIndex	Name	SchemaType	DataType	Format	Constraints	AdditionalInfo	Status	Support
2.2.1.3.1.2.1	2.2.1.3.1.2.1.1			"object"				m	
2.2.1.3.1.2.1	2.2.1.3.1.2.1.1.1	"vnName"	object.property	"string"				m	

2.2.1.3.1 Table for header support

Index	SubIndex	AttributeName	AdditionalInfo	Status	Support
2.2.1.3.1	2.2.1.3.1.1.1	"Content-Length"		m	

2.2.2 Table for requestBody support

Index	SubIndex	RequestBodyName	AdditionalInfo	Status	Support
2.2.2	2.2.2.2.1			m	

2.2.2.2.1 Table for content support

Index	SubIndex	MediaType	AdditionalInfo	Status	Support
2.2.2.2.1	2.2.2.2.1.1	"application/json"		m	

2.2.2.2.1.1 Table for dataType support

Index	SubIndex	Name	SchemaType	DataType	Format	Constraints	AdditionalInfo	Status	Support
2.2.2.2.1.1	2.2.2.2.1.1.1			"object"				m	
2.2.2.2.1.1	2.2.2.2.1.1.1.1	"vnName"	object.property	"string"				m	

2.2.2 Table for response support

Index	SubIndex	HTTPStatusCode	AdditionalInfo	Status	Support
2.2.2	2.2.2.3.1	"200"		m	

2.2.2 Table for callback support

Index	SubIndex	EventName	AdditionalInfo	Status	Support
2.2.2	2.2.2.1	"vnCallback"		m	

2.2.2.1 Table for path support

Index	SubIndex	PathIdentifier	AdditionalInfo	Status	Support
2.2.2.1	2.2.2.1.1	"{$vnCallback}"		m	

2.2.2.1.1 Table for operation support

Index	SubIndex	OperationID	HTTPMethod	AdditionalInfo	Status	Support
2.2.2.1.1	2.2.2.1.1.1		"post"		m	

2.2.2.1.1.1 Table for requestBody support

Index	SubIndex	RequestBodyName	AdditionalInfo	Status	Support
2.2.2.1.1.1	2.2.2.1.1.1.2.1			m	

2.2.2.1.1.1.2.1 Table for content support

Index	SubIndex	MediaType	AdditionalInfo	Status	Support
2.2.2.1.1.1.2.1	2.2.2.1.1.1.2.1.1	"application/json"		m	

2.2.2.1.1.1.2.1.1 Table for dataType support

Index	SubIndex	Name	SchemaType	DataType	Format	Constraints	AdditionalInfo	Status	Support
2.2.2.1.1.1.2.1.1	2.2.2.1.1.1.2.1.1.1			"integer"	"int32"			m	

2.2.2.1.1.1 Table for response support

Index	SubIndex	HTTPStatusCode	AdditionalInfo	Status	Support
2.2.2.1.1.1	2.2.2.1.1.1.3.1	"200"		m	

Appendix II

An example showing the specification of a types support proforma

(This appendix does not form an integral part of this Recommendation.)

II.1 YAML definition example

This clause provides a YAML interface definition example, based on which the example RIICS proformas are illustrated. This example focuses on showing the content of a proforma for dataType support. The omitted parts are the same as those already shown in clause I.1.

Table II.1-1 – An example of a REST interface definition

```
openapi: 3.0.3
info:
  title: TestCase
  version: 1.0.1
tags:
  - name: PhysicalNetwork
  - name: VirtualNetwork
paths:
  /pn:
    get:
      tags:
        - PhysicalNetwork
      operationId: getPN
      responses:
        '200':
          description: success
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/physicalNetwork'
      ……
  /vn:
    get:
      ……
    post:
      tags:
        - VirtualNetwork
      operationId: postVN
      requestBody:
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/virtualNetwork'
      responses:
        ……
      callbacks:
        ……
components:
  schemas:
    device:
      type: object
      properties:
        deviceID:
          type: string
        deviceName:
          type: string
    physicalNetwork:
      type: object
```

```
          properties:
            deviceList:
              type: array
              items:
                $ref: '#/components/schemas/device'
        virtualNetwork:
          type: object
          properties:
            vnID:
              type: integer
            vnName:
              type: string
              format: byte
            pnCovered:
              $ref: '#/components/schemas/physicalNetwork'
```

II.2 RIICS proforma example

RIICS proformas are provided as an example in this clause, based on the example YAML interface definition provided in clause II.1. Only two dataType support proformas are shown in this clause; other proformas are all the same as those in clause I.2.

2.1.1.3.1.2.1 Table for dataType support

Index	SubIndex	Name	SchemaType	DataType	Format	Constraints	AdditionalInfo	Status	Support
2.1.1.3.1.2.1	2.1.1.3.1.2.1.1	"physicalNetwork"		"object"				m	
2.1.1.3.1.2.1	2.1.1.3.1.2.1.1.1	"deviceList"	object.property	"array"				m	
2.1.1.3.1.2.1	2.1.1.3.1.2.1.1.1.1	"device"	array.item	"object"				m	
2.1.1.3.1.2.1	2.1.1.3.1.2.1.1.1.1.1	"deviceID"	object.property	"string"				m	
2.1.1.3.1.2.1	2.1.1.3.1.2.1.1.1.1.2	"deviceName"	object.property	"string"				m	

2.2.2.2.1.1 Table for dataType support

Index	SubIndex	Name	SchemaType	DataType	Format	Constraints	AdditionalInfo	Status	Support
2.2.2.2.1.1	2.2.2.2.1.1.1	"virtualNetwork"		"object"				m	
2.2.2.2.1.1	2.2.2.2.1.1.1.1	"vnID"	object.property	"integer"				m	
2.2.2.2.1.1	2.2.2.2.1.1.1.2	"vnName"	object.property	"string"	"byte"			m	
2.2.2.2.1.1	2.2.2.2.1.1.1.3	"pnCovered"	object.property					m	
2.2.2.2.1.1	2.2.2.2.1.1.1.3.1	"physicalNetwork"	object.property	"object"				m	
2.2.2.2.1.1	2.2.2.2.1.1.1.3.1.1	"deviceList"	object.property	"array"				m	
2.2.2.2.1.1	2.2.2.2.1.1.1.3.1.1.1	"device"	array.item	"object"				m	
2.2.2.2.1.1	2.2.2.2.1.1.1.3.1.1.1.1	"deviceID"	object.property	"string"				m	
2.2.2.2.1.1	2.2.2.2.1.1.1.3.1.1.1.2	"deviceName"	object.property	"string"				m	

Bibliography

[b-ITU-T X.290] Recommendation ITU-T X.290 (1995), *OSI conformance testing methodology and framework for protocol Recommendations for ITU-T applications – General concepts*.

[b-ITU-T X.291] Recommendation ITU-T X.291 (1995), *OSI conformance testing methodology and framework for protocol Recommendations for ITU-T applications – Abstract test suite specification*.

[b-ITU-T X.292] Recommendation ITU-T X.292 (2002), *OSI conformance testing methodology and framework for protocol Recommendations for ITU-T applications – The tree and tabular combined notation (TTCN).*

[b-ITU-T X.293] Recommendation ITU-T X.293 (1995), *OSI conformance testing methodology and framework for protocol Recommendations for ITU-T applications – Test realization.*

[b-ITU-T X.294] Recommendation ITU-T X.294 (1995), *OSI conformance testing methodology and framework for protocol Recommendations for ITU-T applications – Requirements on test laboratories and clients for the conformance assessment process*.

[b-ITU-T X.295] Recommendation ITU-T X.295 (1995), *OSI conformance testing methodology and framework for protocol Recommendations for ITU-T applications – Protocol profile test specification*.

[b-ITU-T X.296] Recommendation ITU-T X.296 (1995), *OSI conformance testing methodology and framework for protocol Recommendations for ITU-T applications – Implementation conformance statements*.

[b-OAI-OAS3] OpenAPI Initiative, *OpenAPI Specification*, Version 3.0.0 (2017). Available [viewed 2022-01-09] at: http://spec.openapis.org/oas/v3.0.0

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	Tariff and accounting principles and international telecommunication/ICT economic and policy issues
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Environment and ICTs, climate change, e-waste, energy efficiency; construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling, and associated measurements and tests
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	**Data networks, open system communications and security**
Series Y	Global information infrastructure, Internet protocol aspects, next-generation networks, Internet of Things and smart cities
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2022