INTERNATIONAL STANDARD

ISO 21219-9

First edition
2023-05

Intelligent transport systems — Traffic and travel information (TTI) via transport protocol experts group, generation 2 (TPEG2) —

Part 9: **Service and network information (TPEG2-SNI)**

Systèmes intelligents de transport — Informations sur le trafic et le tourisme via le groupe expert du protocole de transport, génération 2 (TPEG2) —

Partie 9: Information de service et de réseau (TPEG2-SNI)

Reference number
ISO 21219-9:2023(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html.

This document was prepared by Technical Committee ISO/TC 204, *Intelligent transport systems*.

This first edition cancels and replaces the first edition (ISO/TS 21219-9:2016), which has been technically revised.

The main changes are as follows:

- the document has been changed from a Technical Specification to an International Standard;
- outdated applications have been updated with current TPEG2 specifications (e.g. RTM to TEC, PTI to PTS, CTT to TFP);
- application identification numbers (AIDs) have been updated accordingly.

A list of all parts in the ISO 21219 series can be found on the ISO website.

Any feedback or questions on this document should be directed to the user’s national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

Introduction

0.1 History

TPEG technology was originally proposed by the European Broadcasting Union (EBU) Broadcast Management Committee, who established the B/TPEG project group in the autumn of 1997 with a brief to develop, as soon as possible, a new protocol for broadcasting traffic and travel-related information in the multimedia environment. TPEG technology, its applications and service features were designed to enable travel-related messages to be coded, decoded, filtered and understood by humans (visually and/or audibly in the user's language) and by agent systems. Originally, a byte-oriented data stream format, which can be carried on almost any digital bearer with an appropriate adaptation layer, was developed. Hierarchically structured TPEG messages from service providers to end-users were designed to transfer information from the service provider database to an end-user's equipment.

One year later, in December 1998, the B/TPEG group produced its first EBU specifications. Two documents were released. Part 2 (TPEG-SSF, which became ISO/TS 18234-2) described the syntax, semantics and framing structure which was used for all TPEG applications. Meanwhile, Part 4 (TPEG-RTM, which became ISO/TS 18234-4) described the first application for road traffic messages.

Subsequently, in March 1999, CEN/TC 278, in conjunction with ISO/TC 204, established a group comprising members of the former EBU B/TPEG and this working group continued development work. Further parts were developed to make the initial set of four parts, enabling the implementation of a consistent service. Part 3 (TPEG-SNI, later ISO/TS 18234-3) described the service and network information application used by all service implementations to ensure appropriate referencing from one service source to another.

Part 1 (TPEG-INV, later ISO/TS 18234-1) completed the series by describing the other parts and their relationship; it also contained the application IDs used within the other parts. Additionally, Part 5, the public transport information application (TPEG-PTI, later ISO/TS 18234-5), was developed. The so-called TPEG-LOC location referencing method, which enabled both map-based TPEG-decoders and non-map-based ones to deliver either map-based location referencing or human-readable text information, was issued as ISO/TS 18234-6 to be used in association with the other applications of parts of the ISO 18234 series to provide location referencing.

The ISO 18234 series has become known as TPEG Generation 1.

0.2 TPEG Generation 2

When the Traveller Information Services Association (TISA), derived from former forums, was inaugurated in December 2007, TPEG development was taken over by TISA and continued in the TPEG applications working group.

It was about this time that the (then) new Unified Modelling Language (UML) was seen as having major advantages for the development of new TPEG applications in communities who would not necessarily have the binary physical format skills required to extend the original TPEG TS work. It was also realized that the XML format for TPEG described within the ISO 24530 series (now superseded) had a greater significance than previously foreseen, especially in the content-generation segment, and that keeping two physical formats in synchronism, in different standards series, would be rather difficult.

As a result, TISA set about the development of a new TPEG structure that would be UML-based. This has subsequently become known as TPEG Generation 2 (TPEG2).

TPEG2 is embodied in the ISO 21219 series and it comprises many parts that cover an introduction, rules, toolkit and application components. TPEG2 is built around UML modelling and has a core of rules that contain the modelling strategy covered in ISO 21219-2, ISO 21219-3 and ISO 21219-4 and the conversion to two current physical formats: binary (see Annex A) and XML (see Annex B); others can be added in the future. TISA uses an automated tool to convert from the agreed UML model XMI file directly into an MS Word document file, to minimize drafting errors; this file forms the annex for each physical format.

TPEG2 has a three-container conceptual structure: message management (ISO 21219-6), application (several parts) and location referencing (ISO/TS 21219-7). This structure has flexible capability and can accommodate many differing use cases that have been proposed within the TTI sector and wider for hierarchical message content.

TPEG2 also has many location referencing options as required by the service provider community, any of which may be delivered by vectoring data included in the location referencing container.

The following classification provides a helpful grouping of the different TPEG2 parts according to their intended purpose. Note that the list below is potentially incomplete, as there is the possibility that new TPEG2 parts will be introduced after the publication of this document.

— Toolkit parts: TPEG2-INV (ISO 21219-1), TPEG2-UML (ISO 21219-2), TPEG2-UBCR (ISO 21219-3), TPEG2-UXCR (ISO 21219-4), TPEG2-SFW (ISO 21219-5), TPEG2-MMC (ISO 21219-6), TPEG2-LRC (ISO/TS 21219-7).

— Special applications: TPEG2-SNI (ISO 21219-9 - this document), TPEG2-CAI (ISO 21219-10), TPEG2-LTE (ISO/TS 21219-24).

— Location referencing: TPEG2-OLR (ISO/TS 21219-22), TPEG2-GLR (ISO/TS 21219-21), TPEG2-TLR (ISO 17572-2), TPEG2-DLR (ISO 17572-3).

— Applications: TPEG2-PKI (ISO 21219-14), TPEG2-TEC (ISO 21219-15), TPEG2-FPI (ISO 21219-16), TPEG2-SPI (ISO 21219-17) TPEG2-TFP (ISO 21219-18), TPEG2-WEA (ISO 21219-19), TPEG2-RMR (ISO/TS 21219-23), TPEG2-EMI (ISO/TS 21219-25), TPEG2-VLI (ISO/TS 21219-26).

TPEG2 has been developed to be broadly (but not totally) backward compatible with TPEG1 to assist in transitions from earlier implementations, while not hindering the TPEG2 innovative approach and being able to support many new features, such as dealing with applications with both long-term, unchanging content and highly dynamic content, such as parking information.

This document is based on the TISA specification technical/editorial version reference: SP20009_3.3_001.

Intelligent transport systems — Traffic and travel information (TTI) via transport protocol experts group, generation 2 (TPEG2) —

Part 9: Service and network information (TPEG2-SNI)

1 Scope

This document specifies the method for delivering service and network information within a TPEG service. The TPEG-SNI application is designed to allow the efficient, language-independent delivery of information about the availability of the same service on another bearer channel, or similar service data from another service provider, directly from service provider to end-users.

A number of tables of information are described in this document which provide comprehensive options for describing services, their timing, content, geographical coverage, etc. In all TPEG streams, it is mandatory to deliver the so-called guide to the service table (GST). Additionally, it is possible to signal linkage of content between different bearers and services.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 21219-1, *Intelligent transport systems — Traffic and travel information (TTI) via transport protocol experts group, generation 2 (TPEG2) — Part 1: Introduction, numbering and versions (TPEG2-INV)*

ETSI EN 300-401, *Radio broadcasting systems; Digital Audio Broadcasting (DAB) to mobile, portable and fixed receivers*

ETSI/TS 101 759, *Digital Audio Broadcasting (DAB); Data Broadcasting — Transparent Data Channel*

IETF RFC 1738, *Uniform Resource Locators (URL)*

IEC 62106:2015, *Radio data system (RDS) - VHF/FM sound broadcasting in the frequency range from 64,0 MHz to 108,0 MHz*

ETSI EN 300-468, *Digital Video Broadcasting (DVB); Specification for Service Information (SI) in DVB systems*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO 21219-1 and the following apply.

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at https://www.electropedia.org/

3.1
application identification
AID
identifier that specifies how to process TPEG content and route information to the appropriate application decoder

Note 1 to entry: Each TPEG application has a unique number, which identifies the application according to Clause 5. The application identification is part of the TPEG specification and is defined as and when new applications are developed.

3.2
application instance
actual data stream containing content as defined by an application

3.3
application and content identification
ACID
worldwide unique identifier that defines the content of a service

Note 1 to entry: The ACID is composed of the originator *service identification* (SID-A, SID-B, SID-C) (3.13), the *content identification* (3.5) and the *application identification* (3.1).

3.4
content description
textual description of a selected *service component* (3.11)

3.5
content identification
COID
identifier that is unique within a given application and used to specify its content

Note 1 to entry: The COID is defined by the originator of the content and is unique within a specific application. It is used for labelling the content of a component.

3.6
data radio channel
DARC
FM sub-carrier system for data transmission

3.7
content originator
original provider of an *application instance* (3.2)

Note 1 to entry: The content originator may distribute the application data to different service providers. In some cases, the service provider generates its own application data and is therefore also the content originator.

3.8
fast tuning GST
FT-GST
directory of the applications and content of the service that indicates in which components the relevant information can be found

Note 1 to entry: This contains the minimum set of information required for the acquisition of application data.

3.9
guide to the service table
GST
basic service information such as service structure, service timing and *content description* (3.4), etc.

3.10
Reference-English "word"
word which enables information to be transmitted as a concept, thereby letting the receiver device choose the best possible representation of the given concept in the context of the other parts of the message

Note 1 to entry: This approach means that devices can present concepts in any language or even as graphical icons, for example. For further explanation, see ISO 21219-2.

3.11
service component
virtual channel for messages of a particular application

3.12
service component identification
SCID
unique identifier that defines a *service component* (3.11) within a service

Note 1 to entry: The SCID is chosen by the carrier service provider and identifies a component, which itself has an ACID comprising originator SID, COID and AID. The same number may be used in a different service or, in the same service at a later time to identify a completely different combination of originator SID, COID and AID.

3.13
service identification
worldwide unique identifier for a service

Note 1 to entry: It consists of three elements called SID A, SID-B, SID-C. These are allocated as described in ISO/TS 18234-2.

3.14
service table
table containing basic service information, such as service structure, service timing and *content description* (3.4), etc.

3.15
time schedule GST
T-GST
table indicating the operation times of selected *service components* (3.11)

4 Abbreviated terms

For the purposes of this document, the terms defined in ISO 21219-1 and the following apply.

DAB	digital audio broadcasting
DVB	digital video broadcasting
ECC	extended country code
EID	ensemble identification
ETSI	European Telecommunications Standards Institute
FM	frequency modulation
LHW	local hazard warning
PI	programme identification
RDS	radio data system

SCR	service component reset
SIS	station information service
SIT	service information table
STI	status and travel-time information (proposed TPEG application)
tba	to be announced
URL	uniform resource locator
UTC	coordinated universal zime

5 Application specific constraints

5.1 Application identification

The word “application” is used in the TPEG specifications to describe specific subsets of the TPEG structure. An application defines a limited vocabulary for a certain type of messages, for example, parking information or road traffic information. Each TPEG application is assigned a unique number, called the application identity (AID). An AID number is defined in ISO 21219-1 whenever a new application is developed.

The AID number is used within the TPEG2-SNI application (this document) to indicate how to process TPEG content. It facilitates the routing of information to the appropriate application decoder.

5.2 Version number signalling

Version numbering is used to track the separate versions of an application through its development and deployment. The differences between these versions could have an impact on client devices.

The version numbering principle is defined in ISO 21219-1.

Table 1 shows the current version numbers for signalling SNI within the SNI application.

Table 1 — Current version numbers for signalling of SNI

Major version number	3
Minor version number	3

5.3 TPEG 1 binary compatibility of SNI

The UML model for this application has been modelled according to TPEG2-UBR. The XML physical format conforms with the UXCR Specification ISO 21219-4 and is hence fully TPEG2 conformant. For the binary physical format, the TPEG1 conformance was mandatory to allow the coexistence of TPEG1 and TPEG2 level applications within a single service. So, it was not possible to completely follow the binary conversion rules specified in ISO 21219-3. Details are stated in Annex A.

5.4 TPEG service component frame

SNI makes use of the “service component frame with dataCRC and messageCount” according to this document.

Each SNI component shall appear only once at most in the SNI component frame.

5.5 Conceptual model — Multiplexed applications and services

Figure 1 illustrates the conceptual model of the SNI application.

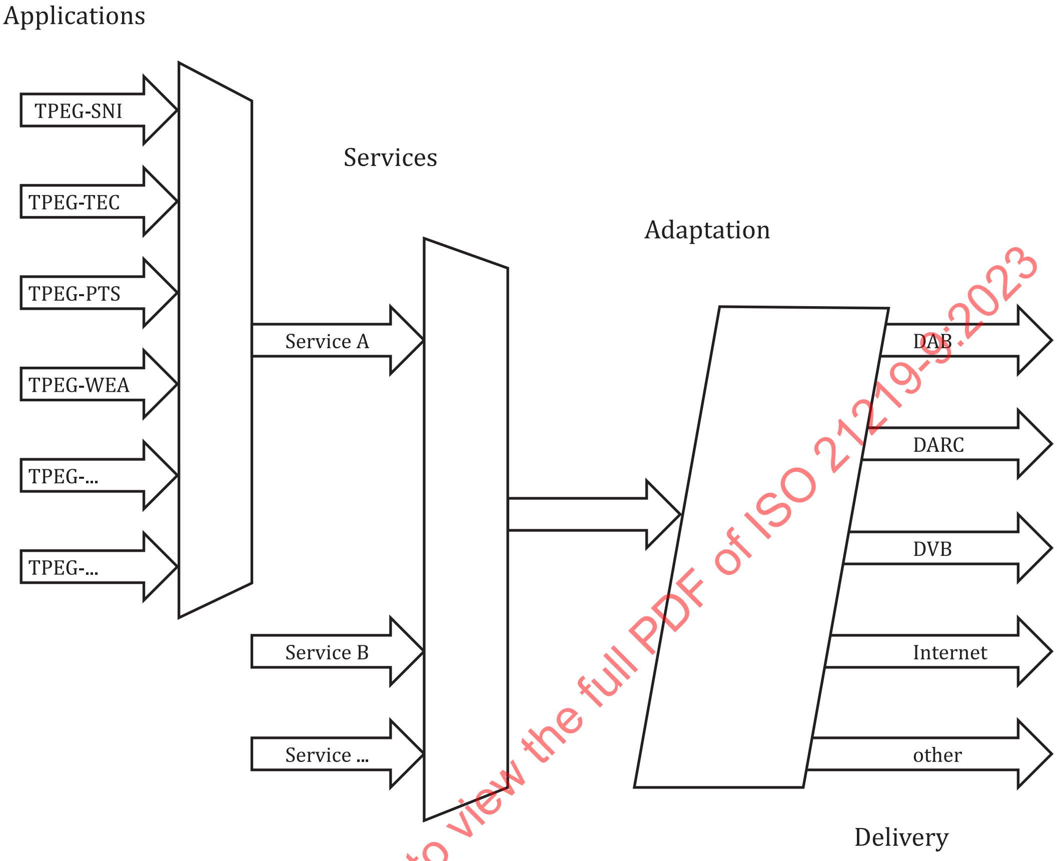
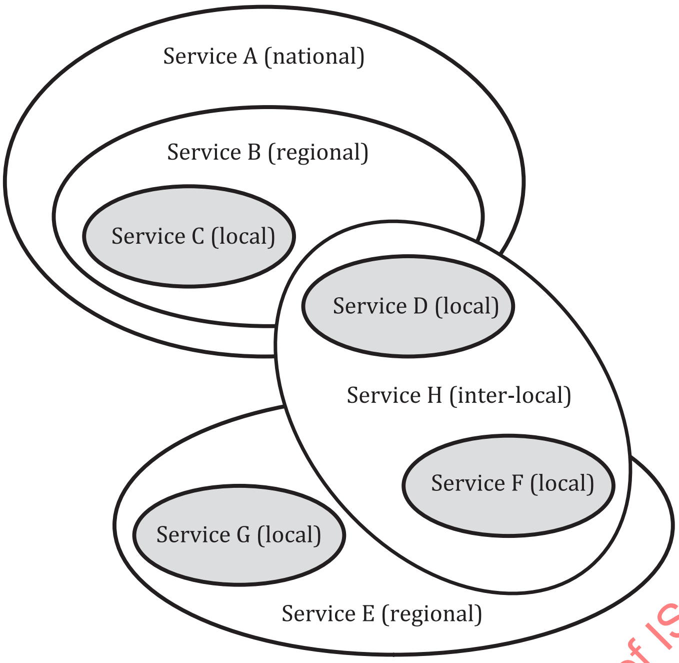


Figure 1 — Multiplexed applications and services

6 Design principle

6.1 Variable content referencing

Figure 2 contains a diagrammatic representation of the use of SCIDs in related services.



Key

Service A (national):	SCID: 02, 03, 04, 05	Bearer: ii and iii
Service B (regional):	SCID: 02, 03, 04	Bearer: iii
Service C (local):	SCID: 02	Bearer: i
Service D (local):	SCID: 03	Bearer: i
Service E (regional):	SCID: 06, 07, 08	Bearer: ii
Service F (local):	SCID: 06	Bearer: i
Service G (local):	SCID: 07	Bearer: i
Service H (inter-local):	SCID: 03, 06	Bearer: ii

Figure 2 — Diagrammatic representation of the use of SCIDs in related services

6.2 Example of the TPEG-SNI application in a TPEG data-stream

Figure 3 gives an example of the TPEG-SNI application in a TPEG data-stream.

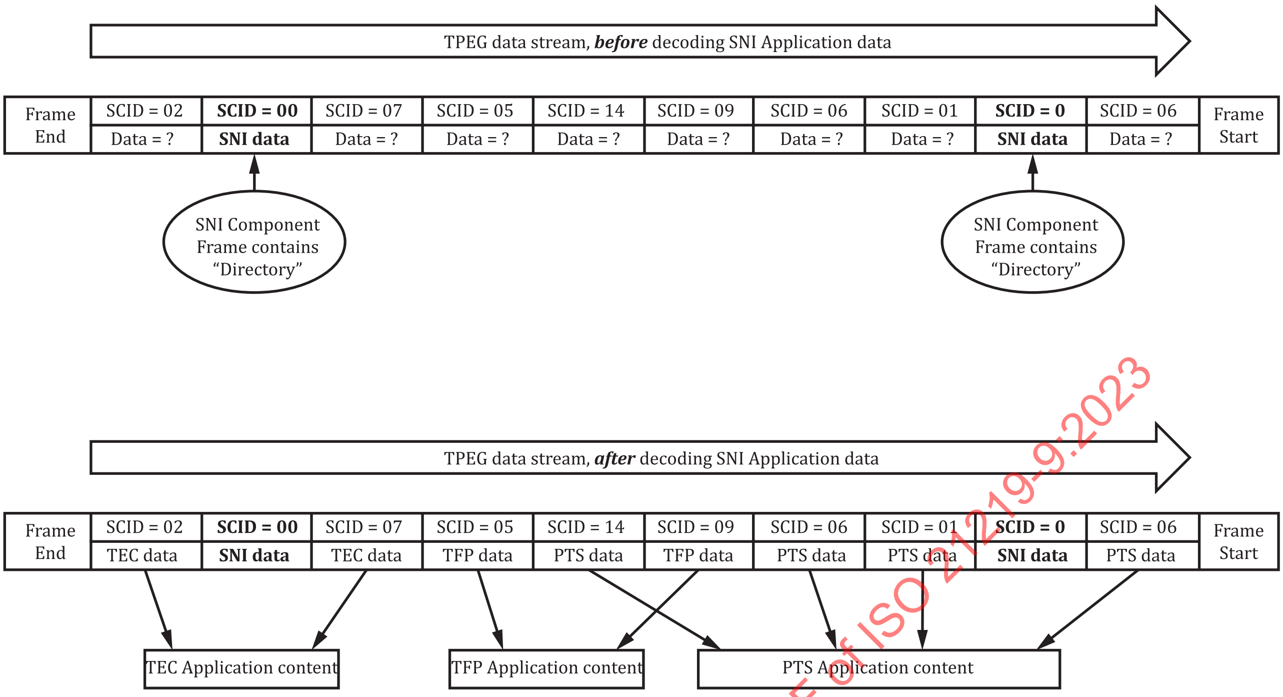


Figure 3 — Example of the TPEG-SNI application in a TPEG data-streamConcept of allocating services

Figure 4 shows the use of TPEG application names and AIDs.

TPEG application	AID	Comment
SNI	0000	
TEC	0005	
PTS	0013	Notional future application code.
TFP	0007	
WEA	0010	

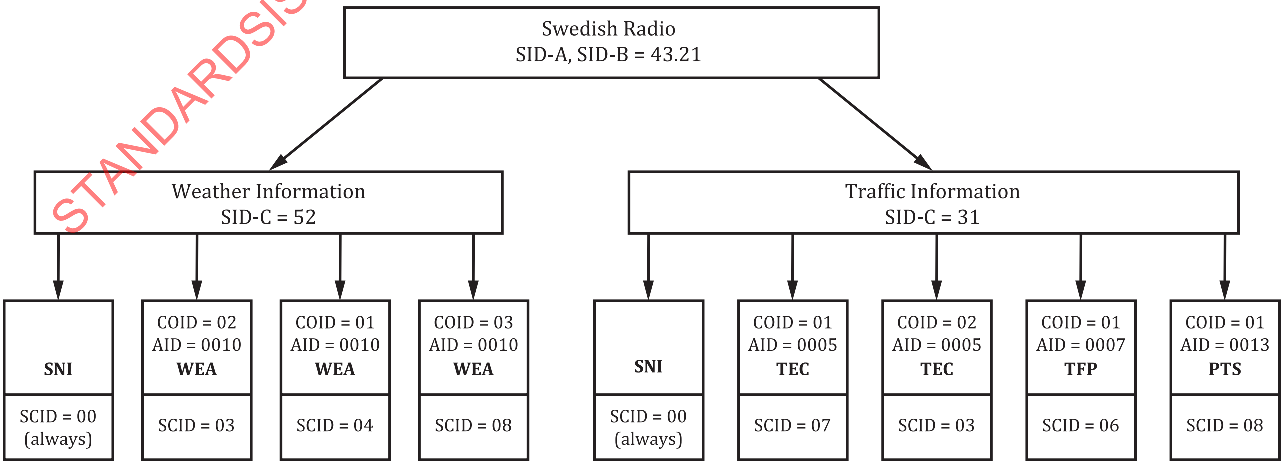
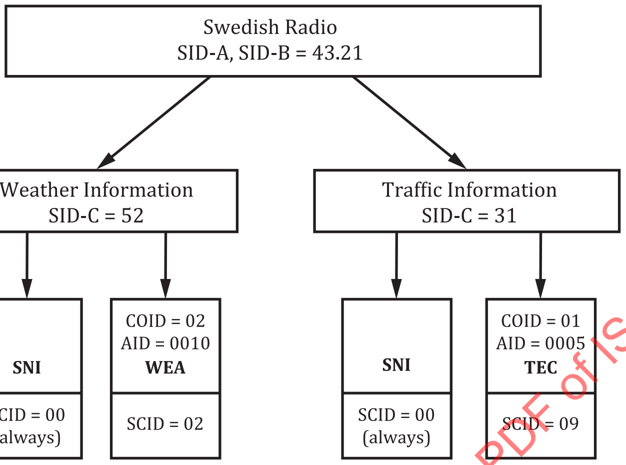


Figure 4 — Example of service allocation on a wideband bearer

There are two instances where SID is used.

- Originator SID (SID-A, SID-B, SID-C): This is the SID of the service provider who generates the content.
- Carrier SID (SID-A, SID-B, SID-C): This is the SID of the service provider who is delivering the service at the service frame level (see ISO 21219-5).

Figure 5 shows an example for the use of the SID and AID.



TPEG application	AID	Comment
SNI	0000	
TEC	0005	
WEA	0010	

Figure 5 — Example of service allocation on a narrowband bearer

6.3 General rules for the TPEG-SNI application

The following rules for the allocation of services by the service provider on one single bearer apply:

- for every service, the service and network information is mandatory;
- the SNI application shall only occur once within a service and has the reserved SCID of 00;
- the fast tuning GST is mandatory within the SNI;
- the SCID identifies the combination of an application and its content within a service.
- the AIDs are standardized by ISO 21219-1;
- the COID is used for specifying the content of a component within a service;
- the originator service identification (SID-A, SID-B, SID-C), COID and AID together form the ACID which uniquely identifies the same content worldwide;
- the ACID is associated with the SCID within a service;
- some instances of a service are equivalent, but not necessarily identical. For example, the same service may be distributed on different bearers with different SCIDs. In this case, the services do not have an identical “byte-stream”, but carry equivalent data content;
- each SNI component (e.g. GST time schedule, linkage table, etc.), shall not occur more than once in each SNI component frame.

7 SNI Structure

Figure 6 shows the structure of SNI components, while GST and SIT components are displayed as summary boxes. The binary format and XML format of the TPEG2-SNI application for use in transmission shall be in accordance with Annexes A and B, respectively.

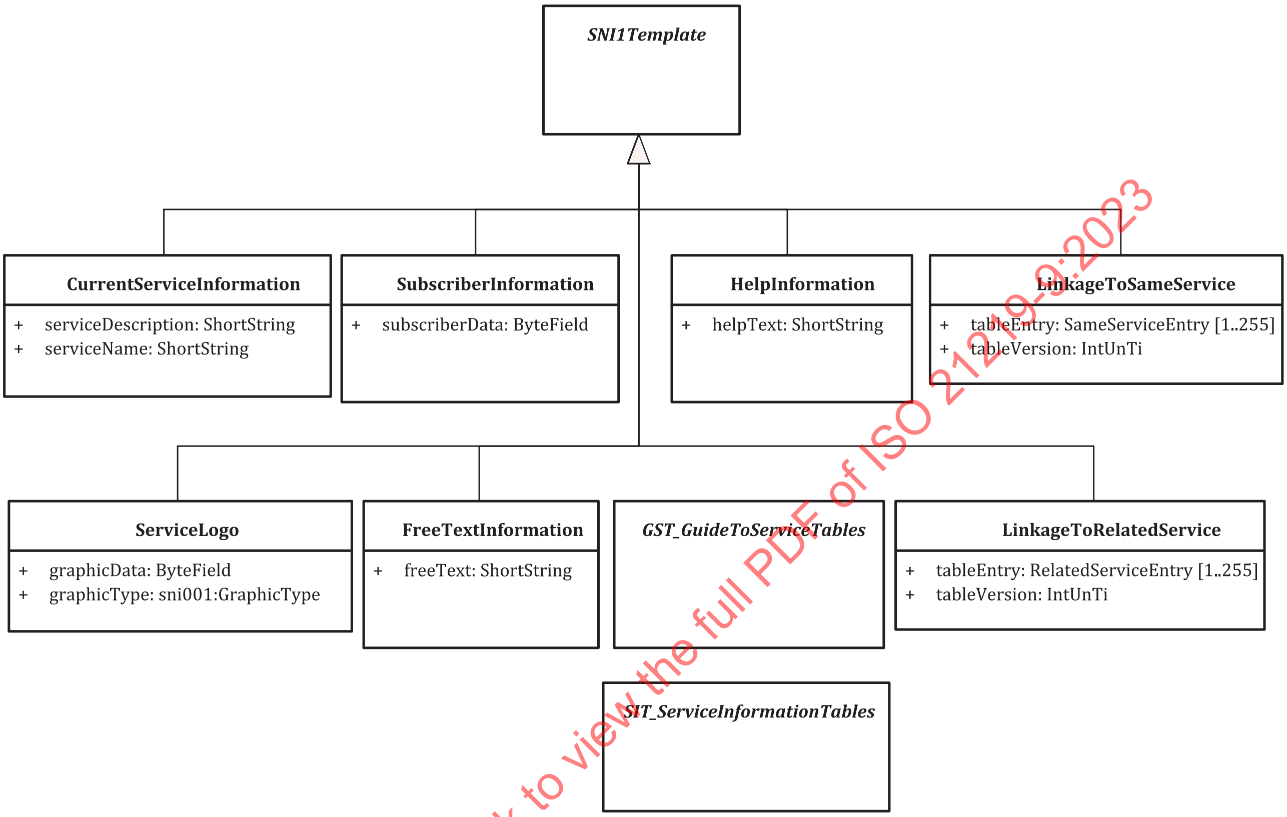


Figure 6 — SNI message structure

8 SNI Message components

8.1 SNI1Template

8.1.1 General

The SNI does not use the typical TPEG message management, but instead provides various components necessary to decode a TPEG service. For conformance with the UML modelling rules (ISO 21219-2), an abstract “SNI1Template” has been introduced to provide the anchor point for the framing.

The components can be grouped as service information, component information and linkage information:

a) **service information**

SNIMessages, that describe the service, like CurrentServiceInformation or ServiceLogo.

b) **component information**

The GST consists of several parts that carry the basic service information of different importance to the system and the user. Taking this into account, the repetition rate of these basic tables can be adjusted to the available channel capacity.

The SIT delivers dynamic information about the current service, for example, the number of messages available for reception in a service component.

c) **linkage information**

TPEG services can be linked independently of the bearer system. These linking features are provided for the same service or other services.

8.1.2 Usage of the version number

There is only one version number within the SNI application. It is used to synchronize the various tables. If in any table, within the SNI, a version number exists, then the version number shall always be the same in all of the tables. If any of the tables change, the version number in all tables shall be incremented.

Exception for SITs: due to the dynamic nature of SITs, the version number in the SIT shall only be changed if the version number of the GST1 changes.

8.2 CurrentServiceInformation

CurrentServiceInformation provides the TPEG service name and description to the receiver. This information is mandatory for TPEG services. Exactly one instance of CurrentServiceInformation is allowed in the SNI. The encoding of the CurrentServiceInformation is shown in Table 2.

Table 2 — Current service information

Name	Type	Multiplicity	Description
serviceName	ShortString	1	Identifies the service to a human being. Identifies the service by a label, comparable to PS in RDS. EXAMPLE “BBC 2 - TPEG Service”.
serviceDescription	ShortString	1	Identifies the applications and scope thereof within a service. Describes in more detail the content of a service. EXAMPLE “Local and interurban road traffic information combined with public transport information for South-East England”.

8.3 ServiceLogo

Service Logo provides a graphical identification of the service or the service provider. It promotes the service or provider. The ServiceLogo is optional. Multiple logos can be referenced. The encoding of the ServiceLogo is shown in Table 3.

Table 3 — Service logo

Name	Type	Multiplicity	Description
graphicType	sni001:GraphicType	1	n.a.
graphicData	ByteField	1	n.a.

8.4 SubscriberInformation

This subclause describes additional payment and encryption information delivered to the end-user. This information is optional for the SNI application, but enhances information provided to the end-user. This mechanism will allow for tariffs to be announced to the end-user. The encoding of the SubscriberInformation is shown in Table 4. This information optionally occurs in any encrypted or potentially encrypted service.

Table 4 — SubscriberInformation

Name	Type	Multiplicity	Description
subscriberData	ByteField	1	Contents defined by the service provider. Gives information about payment and tariffs for restricted service components.

8.5 FreeTextInformation

In this subclause, more textual information for the end-user is defined. This information is not mandatory and the occurrence in the stream is selected by the service provider. The encoding of the FreeTextInformation is shown in Table 5.

EXAMPLE Announcement of service disruption, disclaim information, legal advice.

Table 5 — FreeTextInformation

Name	Type	Multiplicity	Description
freeText	ShortString	1	Additional information that is not coded and is therefore language-dependent.

8.6 HelpInformation

In this subclause, more textual information for the end-user is defined. This information is not mandatory and the occurence in the stream is selected by the service provider. The encoding of the HelpInformation is shown in Table 6.

EXAMPLE Internet address, hotline number, helpdesk.

Table 6 — HelpInformation

Name	Type	Multiplicity	Description
helpText	ShortString	1	Additional information that gives addresses to which the user can apply. A link between the user and the service provider for feedback.

8.7 GST_GuideToServiceTables

The GST consists of seven parts that carry the basic service information to the system and the user. Taking this into account, the repetition rate of these basic tables can be adjusted to the available channel capacity. The encoding of the GST_GuideToServiceTables is shown in Figure 7.

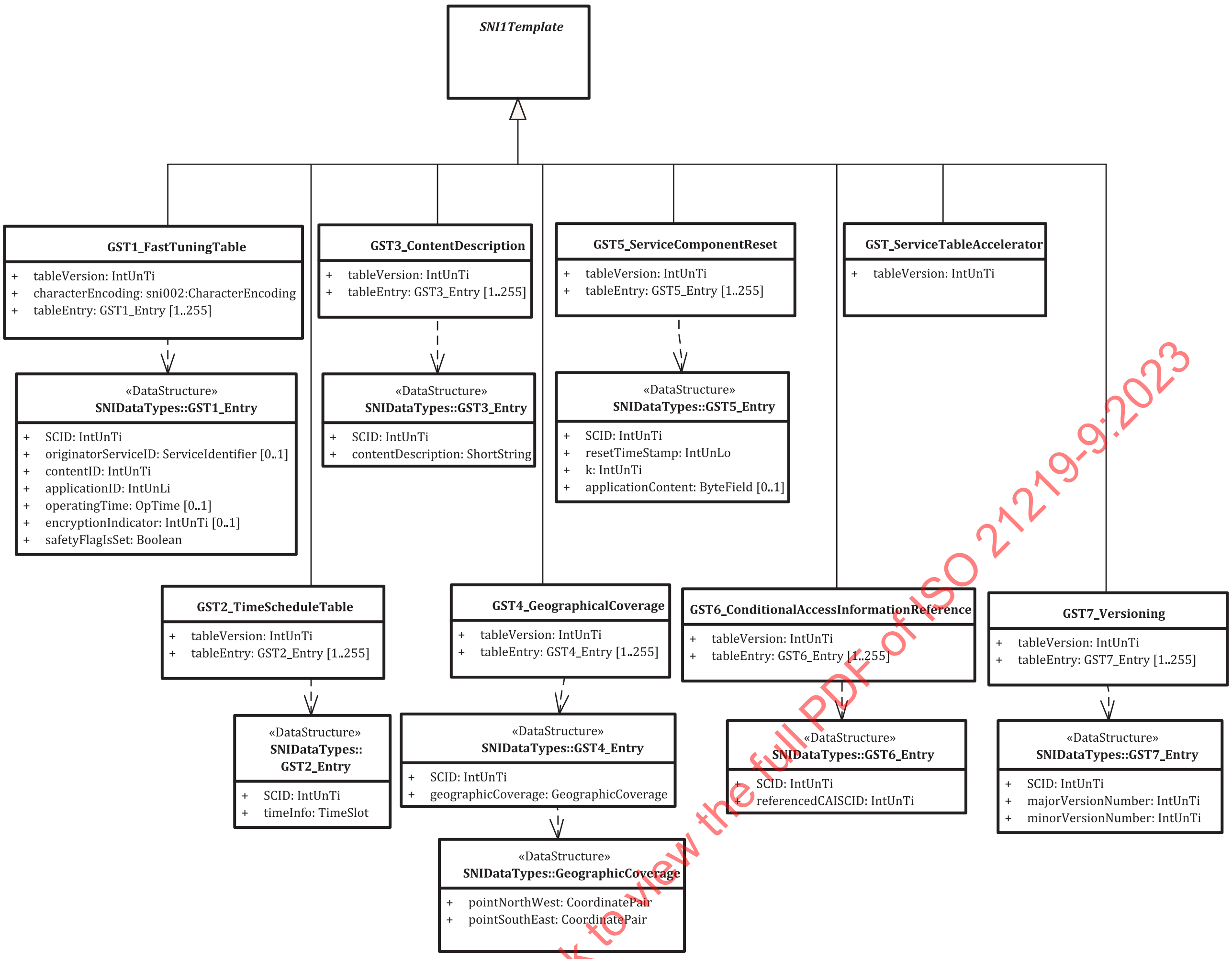

SNI1Template
GST1_FastTuningTable
+ tableVersion: IntUnTi
+ characterEncoding: sni002:CharacterEncoding
+ tableEntry: GST1_Entry [1..255]
GST3_ContentDescription
+ tableVersion: IntUnTi
+ tableEntry: GST3_Entry [1..255]
GST5_ServiceComponentReset
+ tableVersion: IntUnTi
+ tableEntry: GST5_Entry [1..255]
GST_ServiceTableAccelerator
+ tableVersion: IntUnTi
«DataStructure»
SNIDataTypes::GST1_Entry
+ SCID: IntUnTi
+ originatorServiceID: ServiceIdentifier [0..1]
+ contentID: IntUnTi
+ applicationID: IntUnLi
+ operatingTime: OpTime [0..1]
+ encryptionIndicator: IntUnTi [0..1]
+ safetyFlagIsSet: Boolean
«DataStructure»
SNIDataTypes::GST3_Entry
+ SCID: IntUnTi
+ contentDescription: ShortString
«DataStructure»
SNIDataTypes::GST5_Entry
+ SCID: IntUnTi
+ resetTimeStamp: IntUnLo
+ k: IntUnTi
+ applicationContent: ByteField [0..1]
GST2_TimeScheduleTable
+ tableVersion: IntUnTi
+ tableEntry: GST2_Entry [1..255]
GST4_GeographicalCoverage
+ tableVersion: IntUnTi
+ tableEntry: GST4_Entry [1..255]
GST6_ConditionalAccessInformationReference
+ tableVersion: IntUnTi
+ tableEntry: GST6_Entry [1..255]
GST7_Versioning
+ tableVersion: IntUnTi
+ tableEntry: GST7_Entry [1..255]
«DataStructure»
SNIDataTypes::
GST2_Entry
+ SCID: IntUnTi
+ timeInfo: TimeSlot
«DataStructure»
SNIDataTypes::GST4_Entry
+ SCID: IntUnTi
+ geographicCoverage: GeographicCoverage
«DataStructure»
SNIDataTypes::GST6_Entry
+ SCID: IntUnTi
+ referencedCAISCID: IntUnTi
«DataStructure»
SNIDataTypes::GST7_Entry
+ SCID: IntUnTi
+ majorVersionNumber: IntUnTi
+ minorVersionNumber: IntUnTi
«DataStructure»
SNIDataTypes::GeographicCoverage
+ pointNorthWest: CoordinatePair
+ pointSouthEast: CoordinatePair

Figure 7 — SNI service tables

8.8 GST1_FastTuningTable

GST1 (fast tuning GST) is mandatory. All service components shall be defined in this table. The same SCID may never occur more than once within this table.

EXAMPLE Table 7 shows a very simple example of a fast tuning GST, with only one entry (05) of one application (0001). This application has only one content identification (03). This information itself is carried in a component frame identified by 00, which is the default value for the SNI application.

Table 7 — Simple example of a GST tuning table

Version number	Character table identifier	SCID:	COID:	AID:
Mandatory	Mandatory	Mandatory	Mandatory	Mandatory
<IntUnTi>1 Byte	<sni002>1 Byte	<IntUnTi>1 Byte	<IntUnTi>1 Byte	<IntUnLi>2 Bytes
7B	7D	05	03	0001

The encoding of the GST1_FastTuningTable is shown in Table 8.

Table 8 — GST1_FastTuningTable

Name	Type	Multiplicity	Description
tableVersion	IntUnTi	1	Incremented, if any of the entries are changed.
characterEncoding	sni002:CharacterEncoding	1	Default character table for the current service. The one and only encoding to be used in TPEG is UTF-8. All other possible values are deprecated. The character table identifier is valid for the whole service including the SNI application itself. The character table identifier belongs to the basic service features and is therefore integrated into the guide to *the Service Table.* If the <characterEncoding> is invalid or unknown to the receiver, a <characterEncoding> equals 125 (UTF-8) should be assumed.
tableEntry	GST1_Entry	1..255	Lines of GST1.

8.9 GST2_TimeScheduleTable

GST2 (time schedule GST) is optional unless the GST1 OpTime feature is used, in which case it is mandatory. The same SCID shall not occur more than once within this table. The encoding of the GST2_TimeScheduleTable is shown in Table 9.

The time information field is present in both tables. If there is a contradiction between the time information in the fast tuning GST and the time schedule GST, the fast tuning table gives the master time.

If the SCID of a specific service component is not present in the time schedule GST, then the specific service is operating permanently or will start in the future, as specified in the fast tuning table.

Table 9 — GST2_TimeScheduleTable

Name	Type	Multiplicity	Description
tableVersion	IntUnTi	1	Incremented if any of the entries are changed.
tableEntry	GST2_Entry	1..255	Lines of GST2 (time schedule).

8.10 GST3_ContentDescription

GST3 (content description GST) is optional. The same SCID shall not occur more than once within this table. The encoding of the GST3_ContentDescription is shown in Table 10.

NOTE If there is no additional content in GST2 and/or GST3 related to a specific service component, it is not necessary to put this SCID into the table(s). Therefore, the total number of SCID entries in this table is less than or equal to the number of SCID entries in the fast tuning table (GST1).

Table 10 — GST3_ContentDescription

Name	Type	Multiplicity	Description
tableVersion	IntUnTi	1	Incremented, if any of the entries are changed.
tableEntry	GST3_Entry	1..255	Lines of GST3.

8.11 GST4_GeographicalCoverage

GST4 (geographical coverage GST) is optional. The same SCID shall not occur more than once within this table. The encoding of the GST4_GeographicalCoverage is shown in Table 11.

NOTE Although the geographical coordinates are optional, it is not necessary to use a bit switch function. If no geographical coordinates exist, no line is present.

Table 11 — GST4_GeographicalCoverage

Name	Type	Multiplicity	Description
tableVersion	IntUnTi	1	Incremented, if any of the entries are changed.
tableEntry	GST4_Entry	1..255	Lines of GST4.

8.12 GST5_ServiceComponentReset

GST5 (service component reset GST) is optional. The same SCID shall not occur more than once within this table. The encoding of the GST5_ServiceComponentReset is shown in Table 12.

Table 12 — GST5_ServiceComponentReset

Name	Type	Multiplicity	Description
tableVersion	IntUnTi	1	Incremented, if any of the entries are changed.
tableEntry	GST5_Entry	1..255	Lines of GST5.

8.13 GST6_ConditionalAccessInformationReference

GST6 (conditional access information reference GST) is optional. The same SCID shall not occur more than once within column 1 of this table. The encoding of the GST6_ConditionalAccessInformationReference is shown in Table 13.

NOTE 1 If a CAI component is needed by all encrypted service components within a service, the SNI service component with an SCID of 0 is used in column one with the common CAI SCID in the second column.

NOTE 2 If there is no additional content in Table 6 related to a specific service component, it is not necessary to put this SCID into the table. Therefore, the total number of SCID entries in this table is less than or equal to the number of SCID entries in the fast tuning GST.

Table 13 — GST6_ConditionalAccessInformationReference

Name	Type	Multiplicity	Description
tableVersion	IntUnTi	1	Incremented, if any of the entries are changed.
tableEntry	GST6_Entry	1..255	Lines of GST6.

8.14 GST7_Versioning

GST7 (versioning of TPEG applications GST) is mandatory. The same SCID shall not occur more than once within this table.

This table lists IDs of all components which are contained in the TPEG service. For each component the version number of the used TPEG application is given in 2 bytes. The first byte shall signal the major version number whereas the second byte describes the minor version number. All backward compatible changes to the protocol shall result in the increment of the minor version number. All other changes will lead to an increment of the major version number. The encoding of the GST7_Versioning is shown in Table 14.

Table 14 — GST7_Versioning

Name	Type	Multiplicity	Description
tableVersion	IntUnTi	1	Incremented, if any of the entries are changed.
tableEntry	GST7_Entry	1..255	Lines of GST7.

8.15 GST_ServiceTableAccelerator

This component (service table accelerator) for the shortcut version number is repeated more often than the normal GTSs. But this shall not reduce the repetition rate of all the other GSTs. The benefit is that the receiver saves time while checking the change of the GST. The service table accelerator is an optional feature.

The repetition rate of the guide to the service tables is defined as follows.

- A receiver cannot decode a service component until it has received the current fast tuning GST. It is therefore important that the fast tuning GST is repeated at a high repetition rate, such that a receiver has a low set-up time for identification of a service component.
- The scanning time of the receiver for finding a service component will be greatly affected by the repetition rate of the fast tuning GST.
- The repetition rate for the GST should by fixed according the needs of the service provider and the end-user. It also should take into account the bandwidth and capacity of the bearer system.
- All other tables are dependent on the fast tuning GST; therefore, their repetition rate should be set in relation to the repetition rate of the GST.
- The repetition rate of the GST is dependent on the types of application within the service. Some applications can require a higher rate of repetition than others.

EXAMPLE For a TEC application on a wide band bearer, the fast tuning GST can be repeated every second to allow for optimal receiver scanning functionality.

The coding for the service table accelerator is illustrated in Table 15.

Table 15 — GST_ServiceTableAccelerator

Name	Type	Multiplicity	Description
tableVersion	IntUnTi	1	Table incremental version number. Incremented each time a change to the versioning of the GST tables occurs.

8.16 LinkageToSameService

This subclause defines the linkage information to the components of the same service. The encoding of the LinkageToSameService is shown in Figure 8.

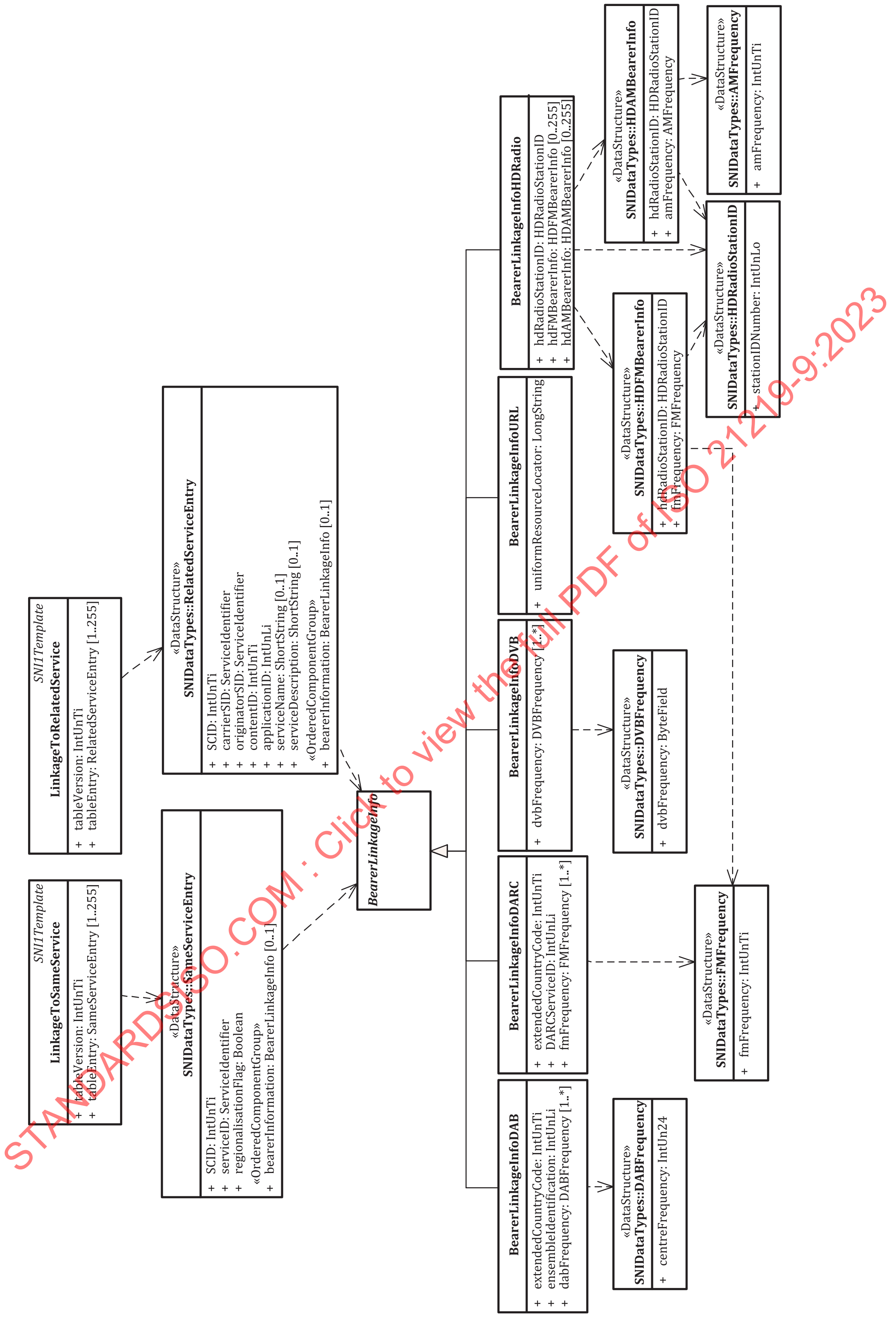

SNI1Template
LinkageToSameService
+ tableVersion: IntUnTi
+ tableEntry: SameServiceEntry [1..255]
SNI1Template
LinkageToRelatedService
+ tableVersion: IntUnTi
+ tableEntry: RelatedServiceEntry [1..255]
«DataStructure»
SNIDataTypes::SameServiceEntry
+ SCID: IntUnTi
+ serviceID: ServiceIdentifier
+ regionalisationFlag: Boolean
«OrderedComponentGroup»
+ bearerInformation: BearerLinkageInfo [0..1]
«DataStructure»
SNIDataTypes::RelatedServiceEntry
+ SCID: IntUnTi
+ carrierSID: ServiceIdentifier
+ originatorSID: ServiceIdentifier
+ contentID: IntUnTi
+ applicationID: IntUnLi
+ serviceName: ShortString [0..1]
+ serviceDescription: ShortString [0..1]
«OrderedComponentGroup»
+ bearerInformation: BearerLinkageInfo [0..1]
BearerLinkageInfo
BearerLinkageInfoDAB
+ extendedCountryCode: IntUnTi
+ ensembleIdentification: IntUnLi
+ dabFrequency: DABFrequency [1..*]
BearerLinkageInfoDARC
+ extendedCountryCode: IntUnTi
+ DARCServiceID: IntUnLi
+ fmFrequency: FMFrequency [1..*]
BearerLinkageInfoDVB
+ dvbFrequency: DVBFrequency [1..*]
BearerLinkageInfoURL
+ uniformResourceLocator: LongString
BearerLinkageInfoHDRadio
+ hdRadioStationID: HDRadioStationID
+ hdFMBearerInfo: HDFMBearerInfo [0..255]
+ hdAMBearerInfo: HDAMBearerInfo [0..255]
«DataStructure»
SNIDataTypes::DABFrequency
+ centreFrequency: IntUn24
«DataStructure»
SNIDataTypes::DVBFrequency
+ dvbFrequency: ByteField
«DataStructure»
SNIDataTypes::FMFrequency
+ fmFrequency: IntUnTi
«DataStructure»
SNIDataTypes::HDFMBearerInfo
+ hdRadioStationID: HDRadioStationID
+ fmFrequency: FMFrequency
«DataStructure»
SNIDataTypes::HDAMBearerInfo
+ hdRadioStationID: HDRadioStationID
+ amFrequency: AMFrequency
«DataStructure»
SNIDataTypes::HDRadioStationID
+ stationIDNumber: IntUnLo
«DataStructure»
SNIDataTypes::AMFrequency
+ amFrequency: IntUnTi

Figure 8 — SNI linkage

The SNI linkage helps the receiver to find the same service components on the same bearer in the same network or on other bearers on other networks. It provides information on how to find the same service components on the same or on another bearer. The SNI linkage is optional and depends on the service provider.

8.17 Same service definition

TPEG, unlike other systems, allows the splitting up of a service in smaller packets, called service components. So, the same service can be transmitted simultaneously on narrow band and on broadband bearers. As a result, a user can get more, exactly the same or fewer service components when switching from one channel to another. This increases the flexibility of the service provider, but at the same time makes the linkage procedure more complicated. The encoding of the LinkageToSameService is shown in Table 16.

Table 16 — LinkageToSameService

Name	Type	Multiplicity	Description
tableVersion	IntUnTi	1	Incremented, if any of the entries are changed.
tableEntry	SameServiceEntry	1..255	Lines of LinkageToSameService table.

8.18 LinkageToRelatedService

The LinkageToRelatedService provides information that links a service component to the components of related services. It supports the receiver in finding similar service components on the same bearer in the same network. It also provides information on how to find related service components on the same or on another bearer. The encoding of the LinkageToRelatedService is shown in Table 17.

Table 17 — LinkageToRelatedService

Name	Type	Multiplicity	Description
tableVersion	IntUnTi	1	Incremented, if any of the entries are changed.
tableEntry	RelatedServiceEntry	1..255	Lines of LinkageToRelatedService table.

8.19 Reserved for future use

Void.

8.20 BearerLinkageInfoDAB

The BearerLinkageInfoDAB provides information about bearer type and linkage information for DAB. This information shall be as defined in ISO 21219-1 and ETSI/TS 101 759. The encoding of the BearerLinkageInfoDAB is shown in Table 18.

NOTE While being in a DAB ensemble, the TPEG service can be found by evaluating the FIC.

Table 18 — BearerLinkageInfoDAB

Name	Type	Multiplicity	Description
extendedCountryCode	IntUnTi	1	ECC.
ensembleIdentification	IntUnLi	1	EID.
dabFrequency	DABFrequency	1.*	DAB frequency information.

8.21 BearerLinkageInfoDARC

The BearerLinkageInfoDARC provides information about the DARC bearer and linkage information. The encoding of the BearerLinkageInfoDARC is shown in Table 19.

Table 19 — BearerLinkageInfoDARC

Name	Type	Multiplicity	Description
extendedCountryCode	IntUnTi	1	ECC.
DARCServiceID	IntUnLi	1	DARC service identification.
fmFrequency	FMFrequency	1..*	FM frequency information.

8.22 BearerLinkageInfoDVB

The BearerLinkageInfoDVB provides information about the bearer type and linkage info for DVB. The encoding of the BearerLinkageInfoDVB is shown in Table 20.

Table 20 — BearerLinkageInfoDVB

Name	Type	Multiplicity	Description
dvbFrequency	DVBFrequency	1..*	DVB frequency.

8.23 BearerLinkageInfoURL

The BearerLinkageInfoURL provides information about related internet services and linkage information. The encoding of the BearerLinkageInfoURL is shown in Table 21.

Table 21 — BearerLinkageInfoURL

Name	Type	Multiplicity	Description
uniformResourceLocator	LongString	1	URL shall be as specified in RFC 1738.

8.24 BearerLinkageInfoHDRadio

The BearerLinkageInfoHDRadio provides information about the HD Radio System (as described in Reference [10]) bearer and linkage information. Multiple hdFMBearerInfo and/or hdAMBearerInfo fields within one BearerLinkageInfoHDRadio component allow for coding of service following information to all adjacent stations of the station ID given by the transmitter field. Multiples of such BearerLinkageInfoHDRadio components may be used to allow for fast service following within a dense network of TPEG services. For any transmitter as given by the HD Radio station ID field, only one BearerLinkageInfoHDRadio is allowed with a single SNI service component. The encoding of the BearerLinkageInfoHDRadio is shown in Table 22.

Table 22 — BearerLinkageInfoHDRadio

Name	Type	Multiplicity	Description
hdRadioStationID	HDRadioStationID	1	Transmitter HD Radio station for which a list of applicable alternate stations is given by the following lists of HDFMBearerInfo and HDAMBearerInfo entries.
hdFMBearerInfo	HDFMBearerInfo	0..255	HD FM bearer information.
hdAMBearerInfo	HDAMBearerInfo	0..255	HD AM bearer information.

8.25 SIT_ServiceInformationTables

This subclause describes the service information tables (SITs).

The tables in this subclause have the same structure as the GSTs (every line in the table is identified by a SCID). The major difference compared to the GSTs is the definition and usage of the version number. The encoding of the SIT_ServiceInformationTables is shown in Figure 9.

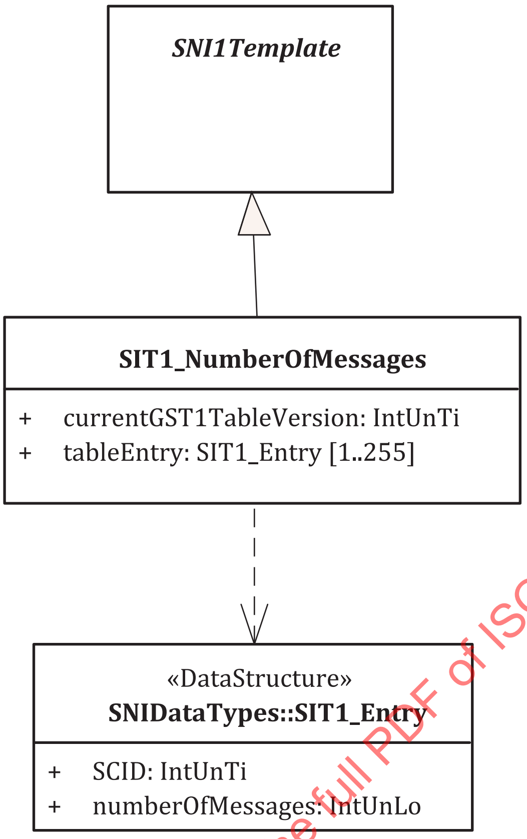

SNI1Template
SIT1_NumberOfMessages
+ currentGST1TableVersion: IntUnTi
+ tableEntry: SIT1_Entry [1..255]
«DataStructure»
SNIDataTypes::SIT1_Entry
+ SCID: IntUnTi
+ numberOfMessages: IntUnLo

Figure 9 — SNI SITs

8.26 SIT1_NumberOfMessages

SIT1 is optional. The same SCID shall not occur more than once within this table. The encoding of the SIT1_NumberOfMessages is shown in Table 23.

Table 23 — SIT1_NumberOfMessages

Name	Type	Multiplicity	Description
currentGST1TableVersion	IntUnTi	1	The version in this table shall equal the version number in the corresponding GST1 table. Consequently, the version number will only change when the version number in the GST1 table changes, either directly (change in GST1 table) or indirectly (change in another GST table). The version number in this SIT shall therefore NOT change when any of the contained total number of messages changes.
tableEntry	SIT1_Entry	1..255	Line of SIT1.

9 SNI datatypes

9.1 MaskedTime

The MaskedTime data type expresses fixed date and time information. Each byte may have a zero value meaning that the signalled event occurs periodically in this range. The value zero is reserved for indicating repetition; therefore, 1 has to be added to the hours, minutes and seconds. The start year is therefore the year 2000 (2000 − 1999 = 1). The first usage of the year element in the masked time function will thus not be before the year 2000. The next box shows the changed MaskedTime type. The right-hand column provides the formula for calculating the type element (e.g. hour) from the real value. Hence, two functions are available simultaneously: pointing to a specific time and indicating a repeating event. The encoding of the MaskedTime is shown in Table 24.

EXAMPLE 1 MaskedTime = 01 0C 00 0F 1F 01 hex. Meaning: The event starts every day in December 2000 at 14 h 30 min and 00 s.

EXAMPLE 2 MaskedTime = 00 00 0B 00 2E 38 hex. Meaning: The event starts every year, every month, on the 11th day of the month, every hour, 45 min and 55 s after the full hour.

Table 24 — MaskedTime

Name	Type	Multiplicity	Description
year	IntUnTi	1	Year, 0:any; 1..255: Year-1999
month	IntUnTi	1	Month, 0:any; 1..12:Month
day	IntUnTi	1	Day, 0:any; 1..31: Day
hour	IntUnTi	1	Hour, 0:any; 1..24:Hour+1
min	IntUnTi	1	Min, 0:any; 1..60; Min+1
sec	IntUnTi	1	Sec, 0:any; 1..60: Sec+1

9.2 DayMask

The DayMask is similar to the TPEG1 day_mask data type and to the TPEG2 DaySelector type. The encoding of the DayMask is shown in Table 25.

Table 25 — DayMask

Name	Type	Multiplicity	Description
everySunday	Boolean	1	n.a.
everyMonday	Boolean	1	n.a.
everyTuesday	Boolean	1	n.a.
everyWednesday	Boolean	1	n.a.
everyThursday	Boolean	1	n.a.
everyFriday	Boolean	1	n.a.
everySaturday	Boolean	1	n.a.

9.3 AppStartTime

This type is a compound element and is helpful to indicate the start time and also the repeatability of an event at the same time. The encoding of the AppStartTime is shown in Table 26.

IMPORTANT — All time information is absolute and always referenced to UTC. For example, an event in China is repeated every Tuesday and Friday at 06:00. This automatically leads to a change of the DayMask value, becoming Monday and Thursday. Also, the MaskedTime value shall change according to the China time zone offset in relation to UTC. All receiving clients shall "know their local time offset", thereby allowing them to convert all time information to the format that the end-user expect.

NOTE The day in the masked time can be set with the dayMask. The resulting start time is the first day of the day mask on or after the specified day in the masked time.

EXAMPLE Assuming 1st July is a Thursday, if the masked time indicates 2nd July and the dayMask is Monday and Tuesday, then the resulting start time is Monday 5th July. The next occurrence is then Tuesday 6th.

Table 26 — AppStartTime

Name	Type	Multiplicity	Description
maskedTime	MaskedTime	1	At what time and date.
dayMask	DayMask	1	At which day of the week.

9.4 TimeSlot

This compound type allows the indication of start time, repetition and duration of an event, all in one. The encoding of the TimeSlot is shown in Table 27.

Table 27 — TimeSlot

Name	Type	Multiplicity	Description
appStartTime	AppStartTime	1	At what time and date.
duration	Duration	1	How long it lasts.

9.5 OpTime

This time element consists of the start and stop time of a service component within a specific application. The start and stop time are transmitted as an absolute UTC value to be independent from any other vague time description (e.g. one hour after midday). The service provider can use this time information to announce the next occurrence of a component of a certain service.

The decoder can then tell the end-user what might be expected in the near future. To take full advantage of this function, six cases are distinguished in Table 28.

Table 28 — Operating time

#	Condition	Explanation	Meaning
1	$T_p \le T_s \le T_e$	Component of the service starts and ends in the future.	Default situation.
2	$T_s \le T_p \le T_e$	Component of the service has already started and ends in the future.	Programme is running.
3	$T_s \le T_e \le T_p$	Component of the service was transmitted, change to condition 1 expected.	Programme is over.
4	$T_p \le T_e \le T_s$	Same as condition 2, but next start time is already announced.	Programme is running.
5	$T_e \le T_p \le T_s$	This condition indicates that a new service will be established.	New programme in the future.
6	$T_e \le T_s \le T_p$	This condition indicates that a service has been abandoned.	Old programme dropped.

The time descriptors have the following meanings:

— T_p: present or current time, i.e. the actual time that changes continuously;

— T_s: start time of a component within an application, fixed by the service provider;

— T_e: end (stop) time of a component within an application, fixed by the service provider.

The encoding of the start time and stop time is shown in Table 29.

Table 29 — Start time and stop time

Name	Type	Multiplicity	Description
startTime	IntUnLo	1	Next start date and time.
stopTime	IntUnLo	1	Next stop date and time.

9.6 GeographicCoverage

This basic type is needed to define an area to which a specific service component is allocated. This feature only makes sense if the application that uses that service component has a relation to a certain coverage area. The encoding of the GeographicCoverage is shown in Table 30.

NOTE Notional rectangle on a flat map.

Table 30 — GeographicCoverage

Name	Type	Multiplicity	Description
pointNorthWest	CoordinatePair	1	Northwest corner of rectangle.
pointSouthEast	CoordinatePair	1	Southeast corner of rectangle.

9.7 CoordinatePair

CoordinatePair defines the corner of a rectangle. The encoding of the CoordinatePair is shown in Table 31, with its numerical representation in Table 32.

Table 31 — Numerical presentation of the coordinates

Range of type	Min. value	Max. value	Resolution (deg)	Resolution (km)	Remarks
Decimal range of <IntSiLi>	− 32 768	+ 32 767	n.a.	n.a.	
Decimal range for longitude	− 18 000	+ 18 000	n.a.	n.a.	
Range of longitude in degrees	− 180,00	+180,00	0,01	1,08	Resolution (km) constant
Decimal range for latitude	− 9 000	+ 9 000	n.a.	n.a.	
Range of latitude in degrees	− 90,00	+ 90,00	0,01	1,08	Resolution (km) variable

Table 32 — CoordinatePair

Name	Type	Multiplicity	Description
longitude	IntSiLi	1	WGS 84 longitude in units of 0,01 degrees.
latitude	IntSiLi	1	WGS 84 latitude in units of 0,01 degrees.

9.8 ByteField

The ByteField is a placeholder data type for any number of bytes.

9.9 GST1_Entry

The encoding of the GST1_Entry is shown in Table 33.

Table 33 — GST1_Entry

Name	Type	Multiplicity	Description
SCID	IntUnTi	1	Service component identification
originatorServiceID	ServiceIdentifier	0..1	The originator service identification needs to be specified when the carrier service provider is not the originator of the content of the related service component. If the carrier service provider is also the originator of the content of the related service component, then it is not necessary to indicate the service identification in this column. In this case, the default is the carrier service provider. Belongs to the ACID, if present.
contentID	IntUnTi	1	COID, belongs to the ACID.
applicationID	IntUnLi	1	AID, belongs to the ACID.
operatingTime	OpTime	0..1	Next operating time. From year 1970 to 2106.
encryptionIndicator	IntUnTi	0..1	In the service frame of the TPEG frame structure, a service encryption indicator (ServEncID) is already defined for encrypting at the service frame level. If this mechanism is used, all underlying levels including the SNI data are "hidden". There is another encryption possibility in the SNI application at the service component level. Individual service components can be encrypted or not encrypted. This is indicated in the fast tuning GST. The SNI service component (00) cannot be encrypted at this level. Where encryption is applied to a service component, then encryption shall not be applied to its SCID, field length and the header CRC of the service component frame, but to its body only. The encryption indicator (EncID) in the fast tuning GST is defined in this document.
safetyFlagIsSet	Boolean	1	Absent or False = Safety flag not set. A stream of service components is marked with the safety flag if it contains safety-related (LHW) messages only: Messages can be used in devices that do not have a map, but that have the position and (driving) direction available. Messages can be presented in short text or spoken form similar to "attention, in 500 metres on the M6, slippery road due to oil spillage".

9.10 GST2_Entry

The encoding of the GST2_Entry is shown in Table 34.

NOTE Although the periodic time information is optional, it is not necessary to use a bit switch function. If no time information exists, no line will be present.

Table 34 — GST2_Entry

Name	Type	Multiplicity	Description
SCID	IntUnTi	1	Service component identification (SCID).
timeInfo	TimeSlot	1	Periodic time information. Operating time. Indicates the start time, the repetition and the duration of any SCID.

9.11 GST3_Entry

The encoding of the GST3_Entry is shown in Table 35.

NOTE Although the content description is optional, it is not necessary to use a bit switch function. If no content description exists, no line will be present.

Table 35 — GST3_Entry

Name	Type	Multiplicity	Description
SCID	IntUnTi	1	Service component identification (SCID).
contentDescription	ShortString	1	Content description: Gives further information related to a service component. The general description of a service is signalled by the CurrentServiceInformation component.

9.12 GST4_Entry

The encoding of the GST4_Entry is shown in Table 36.

Table 36 — GST4_Entry

Name	Type	Multiplicity	Description
SCID	IntUnTi	1	Service component identification (SCID).
geographicCoverage	GeographicCoverage	1	Geographical coordinates. NOTE The compound type Geographical Coverage is defined in 8.11.

9.13 GST5_Entry

The encoding of the GST5_Entry is shown in Table 37.

NOTE Although the service component reset is optional, it is not necessary to use a bit switch function. If no service component reset exists, no line will be present.

Table 37 — GST5_Entry

Name	Type	Multiplicity	Description
SCID	IntUnTi	1	Service component identification (SCID).
resetTimeStamp	IntUnLo	1	Service component reset (SCR) timestamp. Explanation: The SCR can be used as a tool by a service provider to clear previously received data for a particular service component.
k	IntUnTi	1	Number of bytes following. If *k* = 0, nothing follows. Then *k* is used as a terminator.
applicationContent	ByteField	0..1	Byte field containing application specific data with exactly *k* bytes. Needs to be defined in each application.

9.14 GST6_Entry

The encoding of the GST6_Entry is shown in Table 38.

NOTE Although the content description is optional, it is not necessary to use a bit switch function. If no content description exists, no line will be present.

Table 38 — GST6_Entry

Name	Type	Multiplicity	Description
SCID	IntUnTi	1	Service component identification (SCID).
referencedCAISCID	IntUnTi	1	Reference to SCID of CAI-Component (CAI-SCID).

9.15 GST7_Entry

The encoding of the GST7_Entry is shown in Table 39.

Table 39 — GST7_Entry

Name	Type	Multiplicity	Description
SCID	IntUnTi	1	Service component identification (SCID).
majorVersionNumber	IntUnTi	1	Major version number.
minorVersionNumber	IntUnTi	1	Minor version number.

9.16 RelatedServiceEntry

The encoding of the RelatedServiceEntry is shown in Table 40.

The applications in the related services are not necessarily the same as those in the current service. The same SCID may occur more than once within this table to indicate more than one component of a related service. It is not permitted to use SCID = 00 here. SCID = 00 is only possible in the case for linkage to the components of the same service.

EXAMPLE A service provider references a weather application from a traffic application. There can potentially be multiple references from any one SCID to related services.

Table 40 — RelatedServiceEntry

Name	Type	Multiplicity	Description
SCID	IntUnTi	1	Service component identification of the current service. EXAMPLE 08
carrierSID	ServiceIdentifier	1	Service ID of the carrier of the linked service. EXAMPLE 43.51.252
originatorSID	ServiceIdentifier	1	Service ID of the originator of the linked service. This default to the carrier service ID. EXAMPLE 34.45.124
contentID	IntUnTi	1	Content ID of the linked application. EXAMPLE 34
applicationID	IntUnLi	1	Application ID of the linked application. EXAMPLE 0014
Ordered Components			
bearerInformation	BearerLinkageInfo	0..1	Bearer Information. EXAMPLES: Internet: URL DVB: tba DARC: ECC SI m*fc DAB: ECC EID m*fc HD Radio: SIS m1*(SIS,fm) m2*(SIS,am)
serviceName	ShortString	0..1	Name of the related service. EXAMPLE Swedish Radio
serviceDescription	ShortString	0..1	Description of the related service. EXAMPLE National Traffic Information

9.17 DABFrequency

The encoding of the DABFrequency is shown in Table 41.

Table 41 — DABFrequency

Name	Type	Multiplicity	Description
centreFrequency	IntUn24	1	Definition of the centre frequency 19 bits (b0 to b18) shall be as defined in ETSI EN 300-401, b19 to b23 filled with zeros. Carrier frequency = 0 Hz + (centreFrequency * 16 kHz)

9.18 DVBFrequency

The encoding of the DVBFrequency is shown in Table 42.

Table 42 — DVBFrequency

Name	Type	Multiplicity	Description
dvbFrequency	ByteField	1	DVB frequency information. Other information shall be added as specified in ETSI EN 300-468.

9.19 FMFrequency

The encoding of the FMFrequency is shown in Table 43.

Table 43 — FMFrequency

Name	Type	Multiplicity	Description
fmFrequency	IntUnTi	1	Frequency of the FM bearer carrying DARC. Frequency definition shall be as specified in IEC 62106:2015, Table 10.

9.20 AMFrequency

AMFrequency provides the frequency code for HD Radio Broadcast. The encoding of the AMFrequency is shown in Table 44.

Table 44 — AMFrequency

Name	Type	Multiplicity	Description
amFrequency	IntUnTi	1	Integer value in the range of 0 to 122 or 128 to 246, where: range 0 to 122 encodes AM frequencies for ITU region 1 and 3 as follows: f = n * 9kHz + 522kHz, and range 128 to 246 encodes AM frequencies for ITU region 2 as follows: f = (n-128) * 10kHz + 530kHz.

9.21 SameServiceEntry

In general, there are three instances where linkage will be used:

a) the same carrier service provider SID on a different bearer;

b) a different carrier service provider SID on the same bearer;

c) a different carrier service provider SID on a different bearer.

If the optional "bearer and linkage info" is not specified, then the linked service is on the same bearer. The carrier service identification of the linked service may be the same as that of the current service, in which case the "bearer and linkage info" will be used to specify an alternative bearer. It is not recommended to use the current carrier service provider SID in this linkage function without a "bearer and linkage info" specified, since this would lead to linking to itself. The following rules shall be applied.

— The same SCID may occur more than once within this table to indicate more than one alternative linkage on other services.

— Once a link to an alternative service is established, the ACID within the SNI of the linked service has to be compared to the ACID of the current service. If the ACIDs are identical, then the application's content is identical.

— The SCID of the current service and the SCID of the linked service are not necessarily the same. Once the ACID of the linked service has been confirmed as being the same as the current service, then the SCID of the linked service is found from the linked service's fast tuning GST.

Linkage of all service components at once is possible in two cases.

a) Exactly the same service.

   1) If all components of the current service exist as a whole within another service, then the SCID 00 (SNI application) is used to link to the other service with the same content. The regionalization flag shall be unset in this case.

   2) It is possible that the linked service has more components than the current service. What is important is that the linked service contains all components of the current service.

   3) Even though the components are present in both services, they can have different SCIDs.

b) Regionalized service.

   1) If all applications of the current service exist as a whole within another service, but their contents are carried with suitably regionalized components, then the SCID 00 (SNI application) is used to link to the other service with the same content. The regionalization flag shall be set in this case.

   2) It is possible that the linked service has more components than the current service. What is important is that the linked service contains all applications of the current service with suitably regionalized components.

   3) Even though the components are present in both services, they may have different SCIDs.

Linkage to an application component distributed by the following alternative carriers:

- the GST always gives the SID (A-B-C) of the originator of the service (explicit or implicit);
- the linkage table gives the SIDs (A-B-C) of the alternative carriers;
- the alternative carriers will signal the SIDs (A-B-C) of the originator in their GSTs, to allow the link to be confirmed.

The encoding of the SameServiceEntry is shown in Table 45.

Table 45 — SameServiceEntry

Name	Type	Multiplicity	Description
SCID	IntUnTi	1	Service component identification (SCID). EXAMPLE 00
serviceID	ServiceIdentifier	1	Carrier service identification of the linked service. EXAMPLE 43.51.52
Ordered components			
bearerInformation	BearerLinkageInfo	0..1	Bearer and linkage information EXAMPLES Internet: URL DVB: tba DARC: ECC SI m*fc DAB: ECC EID m*fc HD Radio: SIS m1*(SIS,fm) m2*(SIS,am)
regionalizationFlag	Boolean	1	True if regionalization flag is set.

9.22 SIT1_Entry

The encoding of the SIT1_Entry is shown in Table 46.

Table 46 — SIT1_Entry

Name	Type	Multiplicity	Description
SCID	IntUnTi	1	Service component identification.
numberOfMessages	IntUnLo	1	Total number of messages for service component (including cancellations). The total number of messages in the service component shall be the total that is currently in the notional carousel being transmitted in the service component stream, which a client device can be expected to receive before the service provider changes the carousel.

9.23 HDRadioStationID

The HDRadioStationID datatype is defined by the HD Radio SIS message "Station ID number" as shown in Table 47. This SIS message with MSG ID 0000 is the HD Radio system equivalent of the DARC service identification (see Reference [10]).

The HD Radio SIS message "Station ID number" (SIS MSG ID 0000) is the HD Radio identification of a transmitter facility/frequency allocation, and as such, it is more specific than, for example, the DARC service identification (or an RDS PI code) which may be the same on different (alternative) frequencies. A receiver, having obtained this information, can inspect the given frequencies and compare the station identification parameter as provided with the HDFMBearerInfo or HDAMBearerInfo with the actually transmitted station ID number on this other FM or AM frequency. If they are the same, then the receiver can look for the invariably more slowly transmitted TPEG service parameters. The encoding of the HDRadioStationID is shown in Table 47.

Table 47 — HDRadioStationID

Name	Type	Multiplicity	Description
stationIDNumber	IntUnLo	1	HD Radio 32 bit SIS message "Station ID number" (SIS MSG ID 0000).

9.24 HDFMBearerInfo

The HD_FM_bearer_info and HD_AM_bearer_info datatypes provide information on possible alternate HD radio stations on FM or AM frequencies respectively.

The <HD_FM_bearer_info> consists of an HD radio station ID number and an FM frequency code as defined in the RDS standard (see specification of <FMFrequency> in 8.21). The encoding of the HDFMBearerInfo is shown in Table 48.

Table 48 — HDFMBearerInfo

Name	Type	Multiplicity	Description
hdRadioStationID	HDRadioStationID	1	Station ID of alternate HD radio station.
fmFrequency	FMFrequency	1	FM frequency of alternate HD radio station.

9.25 HDAMBearerInfo

The HDFMBearerInfo and HDAMBearerInfo datatypes provide information on possible alternate HD radio stations on FM or AM frequencies, respectively.

The <HDAMBearerInfo> consists of an HD radio station ID number and now an AM frequency code as defined below. The encoding of the HDAMBearerInfo is shown in Table 49.

Table 49 — HDAMBearerInfo

Name	Type	Multiplicity	Description
hdRadioStationID	HDRadioStationID	1	Station ID of alternate HD radio station.
amFrequency	AMFrequency	1	AM frequency of alternate HD radio station.

10 SNI tables

10.1 sni001:GraphicType

sni001:GraphicType allows service providers to specify the type of a transmitted image. The encoding of the sni001:GraphicType is shown in Table 50.

Table 50 — sni001:GraphicType

Code	Reference-English "word"	Comment	Example
0	BMP		
1	PNG		
2	JPG		

10.2 sni002:CharacterEncoding

sni002:CharacterEncoding allows service providers to specify the character encoding used for texts in SNI table entries. The only character encoding method allowed is UTF-8 (value 125 decimal).

Annex A
(normative)

TPEG SNI and TPEG-binary representation

A.1 General

This annex provides the TPEG-binary representation derived via application of the UML to binary conversion rules specified in ISO 21219-3.

The following changes have been made after the application of the UML to binary conversion rules (ISO 21219-3).

1) Changed LengthIndicator type to IntUnLi.

2) Removed Line with lengthAttr.

3) Removed counters for Table line entries.

4) Changed type for selectors from bit_switch to BitArray and modified numbering of bits accordingly. Continuity indicator always set to “0”/False.

EXAMPLE bit_switch bit 0 is now BitArray bit 6 (the least significant bit of the first byte), bit_switch bit 1 is now BitArray bit 5, and so on. bit_switch bit 6 is BitArray bit 0. See ISO 21219-3 for encoding details.

5) Changed types for counters in BearerLinkageInfoHDRadio with limit 0.255 from IntUnLoMB to IntUnTi:

Table A.1 shows the encoding of the BearerLinkageInfoHDRadio as a data type IntUnTi.

Table A.1 — BearerLinkageInfoHDRadio

<IntUnTi>(n),	— : {0 <= *n* <= 255}

6) Subtracted 100 from GCIds larger than 99 to avoid duplicate ids in ISO 21219-2.

7) Changed bit ordering in selector.

8) Removed selector from GST5_Entry.

9) Removed selectors from HDBearerLinkageInfo.

10) Moved selector to first table column after SCID in “Same Service Entry”.

11) Moved selector to first table column after SCID in “RelatedServiceEntry”.

A.2 Message components

A.2.1 List of generic component Ids

Table A.2 shows the list of generic component Ids.

Table A.2 — List of generic component Ids

Name	Id
CurrentServiceInformation	0
GST1_FastTuningTable	1
GST2_TimeScheduleTable	2
GST3_ContentDescription	3
GST4_GeographicalCoverage	4
GST5_ServiceComponentReset	5
GST_ServiceTableAccelerator	6
ServiceLogo	7
LinkageToSameService	8
LinkageToRelatedService	9
SubscriberInformation	10
FreeTextInformation	11
HelpInformation	12
GST6_ConditionalAccessInformationReference	13
GST7_Versioning	14
BearerLinkageInfoHDRadio	15
SIT1_NumberOfMessages	33

A.2.2 Non unique component IDs

Table A.3 shows BearerLinkageInfo values.

Table A.3 — Non unique component IDs

Name	Id
BearerLinkageInfoDAB	0
BearerLinkageInfoURL	1
BearerLinkageInfoDARC	2
BearerLinkageInfoDVB	3

A.2.3 SNI1Template

Table A.4 shows the SNI1Template.

Table A.4 — SNI1Template

<SNI1Template(x)>:=	
<IntUnTi>(x),	Id of this component.
<IntUnLi>(lengthComp),	Number of bytes in component.

A.2.4 CurrentServiceInformation

Table A.5 shows the CurrentServiceInformation.

Table A.5 — CurrentServiceInformation

<CurrentServiceInformation(0)<SNI1Template(0)>>:=	
<IntUnTi>(0),	Id of this component.
<IntUnLi>(lengthComp),	Number of bytes in component.
<ShortString>(serviceName),	Identifies the service to a human being. Identifies the service by a label, comparable to PS in RDS. EXAMPLE "BBC 2 - TPEG Service"
<ShortString>(serviceDescription);	Identifies the applications and scope thereof within a service. Describes in more detail the content of a service. EXAMPLE "Local and interurban road traffic information combined with public transport information for South-East England"

A.2.5 ServiceLogo

Table A.6 shows the ServiceLogo.

Table A.6 — ServiceLogo

<ServiceLogo(7)<SNI1Template(7)>>:=	
<IntUnTi>(7),	Id of this component.
<IntUnLi>(lengthComp),	Number of bytes in component.
<sni001:GraphicType>(graphicType),	
<ByteField>(graphicData);	

A.2.6 SubscriberInformation

Table A.7 shows the SubscriberInformation.

Table A.7 — SubscriberInformation

<SubscriberInformation(10)<SNI1Template(10)>>:=	
<IntUnTi>(10),	Id of this component.
<IntUnLi>(lengthComp),	Number of bytes in component.
<ByteField>(subscriberData);	Contents defined by the service provider. Gives information about payment and tariffs for restricted service components.

A.2.7 FreeTextInformation

Table A.8 shows the FreeTextInformation.

Table A.8 — FreeTextInformation

<FreeTextInformation(11)<SNI1Template(11)>>:=	
<IntUnTi>(11),	Id of this component.
<IntUnLi>(lengthComp),	Number of bytes in component.
<ShortString>(freeText);	Additional information that is not coded and is therefore language dependent.

A.2.8 HelpInformation

Table A.9 shows the HelpInformation.

Table A.9 — HelpInformation

<HelpInformation(12)<SNI1Template(12)>>:=	
<IntUnTi>(12),	Id of this component.
<IntUnLi>(lengthComp),	Number of bytes in component.
<ShortString>(helpText);	Additional information that gives addresses to which the user can apply. A link between the user and the service provider for feedback.

A.2.9 GST_GuideToServiceTables

Table A.10 shows the GST_GuideToServiceTables.

Table A.10 — GST_GuideToServiceTables

<GST_GuideToServiceTables(x)>:=	
<IntUnTi>(x),	Id of this component.
<IntUnLi>(lengthComp),	Number of bytes in component.

A.2.10 GST1_FastTuningTable

Table A.11 shows the GST1_FastTuningTable.

Table A.11 — GST1_FastTuningTable

<GST1_FastTuningTable(1)<SNI1Template(1)>>:=	
<IntUnTi>(1),	Id of this component.
<IntUnLi>(lengthComp),	Number of bytes in component.
<IntUnTi>(tableVersion),	Incremented, if any of the entries are changed.
<sni002:CharacterEncoding>(characterEncoding),	Default character table for the current service. The one and only encoding to be used in TPEG is UTF-8. All other possible values are deprecated. The character table identifier is valid for the whole service including the SNI application itself. The character table identifier belongs to the basic service features and is therefore integrated into the guide to the Service Table. If the <characterEncoding> is invalid or unknown to the receiver, it should assume that the <characterEncoding> equals 125 (UTF-8).
n *<GST1_Entry>(tableEntry);	Lines of GST1.

A.2.11 GST2_TimeScheduleTable

Table A.12 shows the GST2_TimeScheduleTable.

Table A.12 — GST2_TimeScheduleTable

<GST2_TimeScheduleTable(2)<SNI1Template(2)>>:=	
<IntUnTi>(2),	Id of this component.
<IntUnLi>(lengthComp),	Number of bytes in component.
<IntUnTi>(tableVersion),	Incremented if any of the entries are changed.
n *<GST2_Entry>(tableEntry);	Lines of GST2 (time schedule).

A.2.12 GST3_ContentDescription

Table A.13 shows the GST3_ContentDescription.

Table A.13 — GST3_ContentDescription

<GST3_ContentDescription(3)<SNI1Template(3)>>:=	
<IntUnTi>(3),	Id of this component.
<IntUnLi>(lengthComp),	Number of bytes in component.
<IntUnTi>(tableVersion),	Incremented, if any of the entries are changed.
n *<GST3_Entry>(tableEntry);	Lines of GST3.

A.2.13 GST4_GeographicalCoverage

Table A.14 shows the GST4_GeographicalCoverage.

Table A.14 — GST4_GeographicalCoverage

<GST4_GeographicalCoverage(4)<SNI1Template(4)>>:=	
<IntUnTi>(4),	Id of this component.
<IntUnLi>(lengthComp),	Number of bytes in component.
<IntUnTi>(tableVersion),	Incremented, if any of the entries are changed.
n *<GST4_Entry>(tableEntry);	Lines of GST4.

A.2.14 GST5_ServiceComponentReset

Table A.15 shows the GST5_ServiceComponentReset.

Table A.15 — GST5_ServiceComponentReset

<GST5_ServiceComponentReset(5)<SNI1Template(5)>>:=	
<IntUnTi>(5),	Id of this component.
<IntUnLi>(lengthComp),	Number of bytes in component.
<IntUnTi>(tableVersion),	Incremented, if any of the entries are changed.
n *<GST5_Entry>(tableEntry);	Lines for GST5.

A.2.15 GST6_ConditionalAccessInformationReference

Table A.16 shows the GST6_ConditionalAccessInformationReference.

Table A.16 — GST6_ConditionalAccessInformationReference

<GST6_ConditionalAccessInformationReference(13) <SNI1Template(13)>>:=	
<IntUnTi>(13),	Id of this component
<IntUnLi>(lengthComp),	Number of bytes in component.
<IntUnTi>(tableVersion),	Incremented, if any of the entries are changed.
n *<GST6_Entry>(tableEntry);	Lines of GST6.

A.2.16 GST7_Versioning

Table A.17 shows the GST7_Versioning.

Table A.17 — GST7_Versioning

<GST7_Versioning(14)<SNI1Template(14)>>:=	
<IntUnTi>(14),	Id of this component.
<IntUnLi>(lengthComp),	Number of bytes in component.
<IntUnTi>(tableVersion),	Incremented, if any of the entries are changed.
n *<GST7_Entry>(tableEntry);	Lines of GST7.

A.2.17 GST_ServiceTableAccelerator

Table A.18 shows the GST_ServiceTableAccelerator.

Table A.18 — GST_ServiceTableAccelerator

<GST_ServiceTableAccelerator(6)<SNI1Template(6)>>:=	
<IntUnTi>(6),	Id of this component.
<IntUnLi>(lengthComp),	Number of bytes in component.
<IntUnTi>(tableVersion);	Table incremental version number. Incremented each time a change to the versioning of the GST tables occurs.

A.2.18 LinkageToSameService

Table A.19 shows the LinkageToSameService.

Table A.19 — LinkageToSameService

<LinkageToSameService(8)<SNI1Template(8)>>:=	
<IntUnTi> (8),	Id of this component.
<IntUnLi>(lengthComp),	Number of bytes in component.
<IntUnTi>(tableVersion),	Incremented, if any of the entries are changed.
n *<SameServiceEntry>(tableEntry);	Lines of LinkageToSameService table.

A.2.19 LinkageToRelatedService

Table A.20 shows the LinkageToRelatedService.

Table A.20 — LinkageToRelatedService

<LinkageToRelatedService(9)<SNI1Template(9)>>:=	
<IntUnTi>(9),	Id of this component.
<IntUnLi>(lengthComp),	Number of bytes in component.
<IntUnTi>(tableVersion),	Incremented, if any of the entries are changed.
n *<RelatedServiceEntry>(tableEntry);	Lines of LinkageToRelatedService table.

A.2.20 BearerLinkageInfo

Table A.21 shows the BearerLinkageInfo.

Table A.21 — BearerLinkageInfo

<BearerLinkageInfo(x)>:=	
<IntUnTi>(x),	Id of this component.
<IntUnLi>(lengthComp),	Number of bytes in component.

A.2.21 BearerLinkageInfoDAB

Table A.22 shows the BearerLinkageInfoDAB.

Table A.22 — BearerLinkageInfoDAB

<BearerLinkageInfoDAB(0)<BearerLinkageInfo(0)>>:=	
<IntUnTi>(0),	Id of this component.
<IntUnLi>(lengthComp),	Number of bytes in component.
<IntUnTi>(extendedCountryCode),	ECC.
<IntUnLi>(ensembleIdentification),	EID.
n *<DABFrequency>(dabFrequency);	DAB frequency information.

A.2.22 BearerLinkageInfoDARC

Table A.23 shows the BearerLinkageInfoDARC.

Table A.23 — BearerLinkageInfoDARC

<BearerLinkageInfoDARC(2)<BearerLinkageInfo(2)>>:=	
<IntUnTi>(2),	Id of this component.
<IntUnLi>(lengthComp),	Number of bytes in component.
<IntUnTi>(extendedCountryCode),	ECC.
<IntUnLi>(DARCServiceID),	DARC service identification.
n *<FMFrequency>(fmFrequency);	FM frequency information.

A.2.23 BearerLinkageInfoDVB

Table A.24 shows the BearerLinkageInfoDVB.

Table A.24 — BearerLinkageInfoDVB

<BearerLinkageInfoDVB(3)<BearerLinkageInfo(3)>>:=	
<IntUnTi>(3),	Id of this component.
<IntUnLi>(lengthComp),	Number of bytes in component.
n *<DVBFrequency>(dvbFrequency);	DVB frequency according to 0.

A.2.24 BearerLinkageInfoURL

Table A.25 shows the BearerLinkageInfoURL.

Table A.25 — BearerLinkageInfoURL

<BearerLinkageInfoURL(1)<BearerLinkageInfo(1)>>:=	
<IntUnTi>(1),	Id of this component.
<IntUnLi>(lengthComp),	Number of bytes in component.
<LongString>(uniformResourceLocator);	URL

A.2.25 BearerLinkageInfoHDRadio

Table A.26 shows the BearerLinkageInfoHDRadio.

Table A.26 — BearerLinkageInfoHDRadio

<BearerLinkageInfoHDRadio(15) < BearerLinkageInfo(15) > >: =	
<IntUnTi>(15),	Id of this component
<IntUnLi>(lengthComp),	Number of bytes in component.
<HDRadioStationID>(hdRadioStationID),	Transmitter HD radio station for which a list of applicable alternate stations is given by the following lists of HDFMBearerInfo and HDAM-BearerInfo entries.
<IntUnTi>(n),	{0 <= n <= 255}
n *<HDFMBearerInfo>(hdFMBearerInfo),	HD FM bearer information.
<IntUnTi>(n),	{0 <= n <= 255}
n *<HDAMBearerInfo>(hdAMBearerInfo);	HD AM bearer information.

A.2.26 SIT_ServiceInformationTables

Table A.27 shows the SIT_ServiceInformationTables.

Table A.27 — SIT_ServiceInformationTables

<SIT_ServiceInformationTables(x)>:=	
<IntUnTi>(x),	Id of this component.
<IntUnLi>(lengthComp),	Number of bytes in component.

A.2.27 SIT1_NumberOfMessages

Table A.28 shows the SIT1_NumberOfMessages.

Table A.28 — SIT1_NumberOfMessages

<SIT1_NumberOfMessages(33)<SNI1Template(33)>>:=	
<IntUnTi>(33),	Id of this component.
<IntUnLi>(lengthComp),	Number of bytes in component.
<IntUnTi>(currentGST1TableVersion),	The version in this table shall equal the version number in the corresponding GST1. Consequently, the version number will only change when the version number in the GST1 changes, either directly (change in GST1) or indirectly (change in another GST). The version number in this SIT shall therefore not change when any of the contained total number of messages changes.
n *<SIT1_Entry>(tableEntry);	Line of SIT1.

A.3 SNI Datatypes

A.3.1 MaskedTime

Table A.29 shows the MaskedTime.

Table A.29 — MaskedTime

<MaskedTime>:=	
<IntUnTi>(year),	Year, 0:any; 1..255: Year-1999
<IntUnTi>(month),	Month, 0:any; 1..12:Month
<IntUnTi>(day),	Day, 0:any; 1..31: Day
<IntUnTi>(hour),	Hour, 0:any; 1..24:Hour+1
<IntUnTi>(min),	Min, 0:any; 1..60; Min+1
<IntUnTi>(sec);	Sec, 0:any; 1..60: Sec+1

A.3.2 DayMask

Table A.30 shows the DayMask.

Table A.30 — DayMask

<DayMask>:=	
BitArray(selector),	
if (bit 6 of selector is set)	
<Boolean>(everySunday),	
if (bit 5 of selector is set)	
<Boolean>(everyMonday),	
if (bit 4 of selector is set)	
<Boolean>(everyTuesday),	
if (bit 3 of selector is set)	
<Boolean>(everyWednesday),	
if (bit 2 of selector is set)	
<Boolean>(everyThursday),	
if (bit 1 of selector is set)	
<Boolean>(everyFriday),	
if (bit 0 of selector is set)	
<Boolean>(everySaturday);	

A.3.3 AppStartTime

Table A.31 shows the AppStartTime.

Table A.31 — AppStartTime

<AppStartTime>:=	
<MaskedTime>(maskedTime),	At what time and date.
<DayMask>(dayMask);	At which day of the week.

A.3.4 TimeSlot

Table A.32 shows the TimeSlot.

Table A.32 — TimeSlot

<TimeSlot>:=	
<AppStartTime>(appStartTime),	At what time and date.
<Duration>(duration);	How long it lasts.

A.3.5 OpTime

Table A.33 shows the OpTime.

Table A.33 — OpTime

<OpTime>:=	
<IntUnLo>(startTime),	Next start date and time.
<IntUnLo>(stopTime);	Next stop date and time.

A.3.6 GeographicCoverage

Table A.34 shows the GeographicCoverage.

Table A.34 — GeographicCoverage

<GeographicCoverage>:=	
<CoordinatePair>(pointNorthWest),	North-West corner of rectangle.
<CoordinatePair>(pointSouthEast);	South-East corner of rectangle.

A.3.7 CoordinatePair

Table A.35 shows the CoordinatePair.

Table A.35 — CoordinatePair

<CoordinatePair>:=	
<IntSiLi>(longitude),	WGS 84 Longitude in units of 0,01°.
<IntSiLi>(latitude);	WGS 84 Latitude in units of 0,01°.

A.3.8 ByteField

Table A.36 shows the ByteField.

Table A.36 — ByteField

<ByteField>:=	Not otherwise specified binary data.

A.3.9 GST1_Entry

Table A.37 shows the GST1_Entry.

Table A.37 — GST1_Entry

<GST1_Entry>:=	
<IntUnTi>(SCID),	Service Component Identification
BitArray(selector),	
if (bit 6 of selector is set)	
<ServiceIdentifier>(originatorServiceID),	The originator service identification needs to be specified when the carrier service provider is not the originator of the content of the related service component. If the carrier service provider is also the originator of the content of the related service component, then it is not necessary to indicate the service identification in this column. In this case, the default is the carrier service provider. Belongs to the ACID, if present.
<IntUnTi>(contentID),	COID, belongs to the ACID.
<IntUnLi>(applicationID),	AID, belongs to the ACID.
if (bit 4 of selector is set)	
<OpTime>(operatingTime),	Next operating time. From year 1970 to 2106.
if (bit 3 of selector is set)	
<IntUnTi>(encryptionIndicator),	In the service frame of the TPEG frame structure, a service encryption indicator (ServEncID) is already defined for encrypting at the service frame level. If this mechanism is used, all underlying levels including the SNI data are "hidden". There is another encryption possibility in the SNI application at the service component level. Individual service components can be encrypted or not encrypted. This is indicated in the fast tuning GST. The SNI service component (00) cannot be encrypted at this level. Where encryption is applied to a service component, then encryption shall not be applied to its SCID, field length and the header CRC of the service component frame, but to its body only. The encryption indicator (EncID) in the fast tuning GST is defined in this document.
if (bit 2 of selector is set)	
<Boolean>(safetyFlagIsSet);	Absent or False = Safety flag not set A stream of service components is marked with the safety flag if it contains safety-related (LHW) messages only: Messages can be used in devices that do not have a map, but that have the position and (driving) direction available. Messages can be presented in short text or spoken form, similar to "attention, in 500 metres on the M6, slippery road due to oil spillage".

A.3.10 GST2_Entry

Table A.38 shows the GST2_Entry.

Table A.38 — GST2_Entry

<GST2_Entry>:=	
<IntUnTi>(SCID),	Service component identification (SCID).
<TimeSlot>(timeInfo);	Periodic time information. Operating time. Indicates the start time, the repetition and the duration of any SCID.

A.3.11 GST3_Entry

Table A.39 shows the GST3_Entry.

Table A.39 — GST3_Entry

<GST3_Entry>:=	
<IntUnTi>(SCID),	Service component identification (SCID).
<ShortString>(contentDescription);	Content description: Gives further information related to a service component. The general description of a service is signalled by the CurrentServiceInformation component.

A.3.12 GST4_Entry

Table A.40 shows the GST4_Entry.

Table A.40 — GST4_Entry

<GST4_Entry>:=	
<IntUnTi>(SCID),	SCID.
<GeographicCoverage>(geographicCoverage);	Geographical coordinates. NOTE The compound type Geographical Coverage is defined in 8.11.

A.3.13 GST5_Entry

Table A.41 shows the GST5_Entry.

Table A.41 — GST5_Entry

<GST5_Entry>:=	
<IntUnTi>(SCID),	Service component identifier (SCID).
<IntUnLo>(resetTimeStamp),	Service component reset (SCR) timestamp. Explanation: The SCR can be used as a tool by a service provider to clear previously received data for a particular service component.
<IntUnTi>(k),	Number of bytes following. If $k = 0$, nothing follows. Then k is used as a terminator.
<ByteField>(applicationContent);	Byte field containing application specific data with exactly k bytes. Shall be defined in each application.

A.3.14 GST6_Entry

Table A.42 shows the GST6_Entry.

Table A.42 — GST6_Entry

<GST6_Entry>:=	
<IntUnTi>(SCID),	Service component identifier (SCID).
<IntUnTi>(referencedCAISCID);	Reference to SCID of CAI-Component (CAI-SCID).

A.3.15 GST7_Entry

Table A.43 shows the GST7_Entry.

Table A.43 — GST7_Entry

<GST7_Entry>:=	
<IntUnTi>(SCID),	Service component identification (SCID)
<IntUnTi>(majorVersionNumber),	Major version number.
<IntUnTi>(minorVersionNumber);	Minor version number.

A.3.16 RelatedServiceEntry

Table A.44 shows the RelatedServiceEntry.

Table A.44 — RelatedServiceEntry

<RelatedServiceEntry>:=	
<IntUnTi > (SCID),	Service component ID of the current service. EXAMPLE 08
BitArray(selector),	
<ServiceIdentifier>(carrierSID),	Service ID of the carrier of the linked service. EXAMPLE 43.51.252
<ServiceIdentifier>(originatorSID),	Service ID of the originator of the linked service. This defaults to the carrier service ID. EXAMPLE 34.45.124
<IntUnTi>(contentID),	Content ID of the linked application. EXAMPLE 34
<IntUnLi>(applicationID),	Application ID of the linked application. EXAMPLE 0014
if (bit 6 of selector is set)	
<BearerLinkageInfo>(bearerInformation),	Bearer information. EXAMPLES Internet: URL DVB: tba DARC: ECC SI m*fc DAB: ECC EID m*fc HD Radio: SIS m1*(SIS,fm) m2*(SIS,am)
if (bit 5 of selector is set)	
<ShortString>(serviceName),	Name of the related service. EXAMPLE Swedish Radio
if (bit 4 of selector is set)	
<ShortString>(serviceDescription);	Description of the related service. EXAMPLE National Traffic Information

A.3.17 DABFrequency

Table A.45 shows the DABFrequency.

Table A.45 — DABFrequency

<DABFrequency>:=	
<IntUn24>(centreFrequency);	Definition of the centre frequency 19 bits (b0 to b18) shell be as defined in ETSI EN 300-401, b19 to b23 filled with zeros. Carrier frequency = 0 Hz + (centreFrequency * 16 kHz).

A.3.18 DVBFrequency

Table A.46 shows the DVBFrequency.

Table A.46 — DVBFrequency

<DVBFrequency>:=	
<ByteField>(dvbFrequency);	Placeholder, tba. DVB frequency information. Other information shall be added as specified in ETSI EN 300-468.

A.3.19 FMFrequency

Table A.47 shows the FMFrequency.

Table A.47 — FMFrequency

<FMFrequency>:=	
<IntUnTi>(fmFrequency);	Frequency of the FM bearer carrying DARC. Frequency definition shall be as specified in IEC 62106:2015, Table 10.

A.3.20 AMFrequency

Table A.48 shows the AMFrequency.

Table A.48 — AMFrequency

<AMFrequency>:=	
<IntUnTi>(amFrequency);	Integer value in the range of 0-122 or 128-246, where: A value of 0 to 122 encodes AM frequencies for ITU region 1 and 3 as follows: f = value * 9kHz + 522kHz, and, A value of 128 to 246 encodes AM frequencies for ITU region 2 as follows: f = (value-128) * 10kHz + 530kHz.

A.3.21 SameServiceEntry

Table A.49 shows the SameServiceEntry.

Table A.49 — SameServiceEntry

<SameServiceEntry>:=	
<IntUnTi>(SCID),	Service component identifier (SCID) EXAMPLE 00
BitArray(selector),	
<ServiceIdentifier>(serviceID),	Carrier service identification of the linked service. EXAMPLE 43.51.52
if (bit 6 of selector is set)	
<BearerLinkageInfo>(bearerInformation),	Bearer and linkage information EXAMPLES Internet: URL DVB: tba DARC: ECC SI m*fc DAB: ECC EID m*fc HD Radio: SIS m1*(SIS,fm) m2*(SIS,am)
if (bit 5 of selector is set)	
<Boolean>(regionalisationFlag);	True if regionalization flag is set.

A.3.22 SIT1_Entry

Table A.50 shows the SIT1_Entry.

Table A.50 — SIT1_Entry

<SIT1_Entry>:=	
<IntUnTi>(SCID),	Service component ID
<IntUnLo>(numberOfMessages);	Total number of messages for service component (including cancellations). NOTE The total number of messages in the service component shall be the total that is currently in the notional carousel being transmitted in the service component stream, which a client device can be expected to receive before the service provider changes the carousel.

A.3.23 HDRadioStationID

Table A.51 shows the HDRadioStationID.

Table A.51 — HDRadioStationID

<HDRadioStationID>:=	
<IntUnLo>(stationIDNumber);	HD radio 32 bit SIS message “Station ID number” (SIS MSG ID 0000)

A.3.24 HDFMBearerInfo

Table A.52 shows the HDFMBearerInfo.

Table A.52 — HDFMBearerInfo

<HDFMBearerInfo>:=	
<HDRadioStationID>(hdRadioStationID),	Station ID of alternate HD radio station.
<FMFrequency>(fmFrequency);	FM frequency of alternate HD radio station.

A.3.25 HDAMBearerInfo

Table A.53 shows the HDAMBearerInfo.

Table A.53 — HDAMBearerInfo

<HDAMBearerInfo>:=	
<HDRadioStationID>(hdRadioStationID),	Station ID of alternate HD radio station.
<AMFrequency>(amFrequency);	AM frequency of alternate HD radio station.

Annex B
(normative)

TPEG SNI, tpegML representation

B.1 General

This annex provides the XML representation derived via application of the UML to XML conversion rules specified in ISO 21219-4.

B.2 Message components

B.2.1 SNI1Template

```
<xs:element name="SNI1Template" type="SNI1Template"/>
<xs:complexType name="SNI1Template">
<xs:complexContent>
<xs:extension base="tsf:ApplicationRootMessageML">
 <xs:sequence>
 <xs:choice minOccurs="1" maxOccurs="1">
 <xs:element name="optionCurrentServiceInformation" type="CurrentServiceInformation"
minOccurs="1" maxOccurs="1"/>
 <xs:element name="optionServiceLogo" type="ServiceLogo" minOccurs="1" maxOccurs="1"/>
 <xs:element name="optionSubscriberInformation" type="SubscriberInformation" minOccurs="1"
maxOccurs="1"/>
 <xs:element name="optionFreeTextInformation" type="FreeTextInformation" minOccurs="1"
maxOccurs="1"/>
 <xs:element name="optionHelpInformation" type="HelpInformation" minOccurs="1"
maxOccurs="1"/>
 <xs:element name="optionGST1_FastTuningTable" type="GST1_FastTuningTable" minOccurs="1"
maxOccurs="1"/>
 <xs:element name="optionGST2_TimeScheduleTable" type="GST2_
TimeScheduleTable" minOccurs="1" maxOccurs="1"/>
 <xs:element name="optionGST3_ContentDescription" type="GST3_ContentDescription"
minOccurs="1" maxOccurs="1"/>
 <xs:element name="optionGST4_GeographicalCoverage" type="GST4_GeographicalCoverage"
minOccurs="1" maxOccurs="1"/>
 <xs:element name="optionGST5_ServiceComponentReset" type="GST5_ServiceComponentReset"
minOccurs="1" maxOccurs="1"/>
 <xs:element name="optionGST6_ConditionalAccessInformationReference" type="GST6_
ConditionalAccessInformationReference" minOccurs="1" maxOccurs="1"/>
 <xs:element name="optionGST7_Versioning" type="GST7_Versioning" minOccurs="1"
maxOccurs="1"/>
 <xs:element name="optionGST_ServiceTableAccelerator" type="GST_ServiceTableAccelerator"
minOccurs="1" maxOccurs="1"/>
 <xs:element name="optionLinkageToSameService" type="LinkageToSameService" minOccurs="1"
maxOccurs="1"/>
 <xs:element name="optionLinkageToRelatedService" type="LinkageToRelatedService"
minOccurs="1" maxOccurs="1"/>
 <xs:element name="optionSIT1_NumberOfMessages" type="SIT1_NumberOfMessages" minOccurs="1"
maxOccurs="1"/>
 </xs:choice>
 </xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
```

B.2.2 CurrentServiceInformation

```
<xs:complexType name="CurrentServiceInformation">
 <xs:sequence>
 <xs:element name="serviceName" type="tdt:ShortString"/>
 <xs:element name="serviceDescription" type="tdt:ShortString"/>
 </xs:sequence>
</xs:complexType>
```

B.2.3 ServiceLogo

```
<xs:complexType name="ServiceLogo">
 <xs:sequence>
 <xs:element name="graphicType" type="sni001_GraphicType"/>
 <xs:element name="graphicData" type="ByteField"/>
 </xs:sequence>
</xs:complexType>
```

B.2.4 SubscriberInformation

```
<xs:complexType name="SubscriberInformation">
 <xs:sequence>
 <xs:element name="subscriberData" type="ByteField"/>
 </xs:sequence>
</xs:complexType>
```

B.2.5 FreeTextInformation

```
<xs:complexType name="FreeTextInformation">
 <xs:sequence>
 <xs:element name="freeText" type="tdt:ShortString"/>
 </xs:sequence>
</xs:complexType>
```

B.2.6 HelpInformation

```
<xs:complexType name="HelpInformation">
 <xs:sequence>
 <xs:element name="helpText" type="tdt:ShortString"/>
 </xs:sequence>
</xs:complexType>
```

B.2.7 GST_GuideToServiceTables

```
<xs:complexType name="GST_GuideToServiceTables">
 <xs:sequence>
 <xs:choice minOccurs="1"  maxOccurs="1">
 </xs:choice>
 </xs:sequence>
</xs:complexType>
```

B.2.8 GST1_FastTuningTable

```
<xs:complexType name="GST1_FastTuningTable">
 <xs:sequence>
 <xs:element name="tableVersion" type="tdt:IntUnTi"/>
 <xs:element name="characterEncoding" type="sni002_CharacterEncoding"/>
 <xs:element name="tableEntry" type="GST1_Entry" maxOccurs="255"/>
 </xs:sequence>
</xs:complexType>
```

B.2.9 GST2_TimeScheduleTable

```
<xs:complexType name="GST2_TimeScheduleTable">
 <xs:sequence>
 <xs:element name="tableVersion" type="tdt:IntUnTi"/>
 <xs:element name="tableEntry" type="GST2_Entry" maxOccurs="255"/>
 </xs:sequence>
</xs:complexType>
```

B.2.10 GST3_ContentDescription

```
<xs:complexType name="GST3_ContentDescription">
 <xs:sequence>
 <xs:element name="tableVersion" type="tdt:IntUnTi"/>
 <xs:element name="tableEntry" type="GST3_Entry" maxOccurs="255"/>
 </xs:sequence>
</xs:complexType>
```