TECHNICAL SPECIFICATION

ISO/TS 21219-23

First edition
2016-12-15

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

Part 23: Roads and multimodal routes (TPEG2-RMR)

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

Partie 23: Routes et routes multimodales (TPEG2-RMR)

Reference number
ISO/TS 21219-23:2016(E)

International Organization for Standardization

COPYRIGHT PROTECTED DOCUMENT

ISO copyright office
Ch. de Blandonnet 8 • CP 401
CH-1214 Vernier, Geneva, Switzerland
Tel. +41 22 749 01 11
Fax +41 22 749 09 47
copyright@iso.org
www.iso.org

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 on the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the WTO principles in the Technical Barriers to Trade (TBT) see the following URL: Foreword - Supplementary information

The committee responsible for this document is ISO/TC 204, *Intelligent transport systems*.

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

Introduction

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 may 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, 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, 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, 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 parts of the ISO/TS 18234- series to provide Location Referencing.

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

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 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/TS 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 is embodied in the ISO/TS 21219- series and it comprises many parts that cover 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/TS 21219-2, ISO/TS 21219-3, ISO/TS 21219-4 and the conversion to two current physical formats: binary and XML; others could 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, that forms the Annex for each physical format.

TPEG2 has a three container conceptual structure: Message Management (ISO/TS 21219-6), Application (several Parts) and Location Referencing (ISO/TS 21219-7[1)]). 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.

— Toolkit parts: TPEG2-INV (ISO/TS 21219-1), TPEG2-UML (ISO/TS 21219-2), TPEG2-UBCR (ISO/TS 21219-3), TPEG2-UXCR (ISO/TS 21219-4), TPEG2-SFW (ISO/TS 21219-5), TPEG2-MMC (ISO/TS 21219-6), TPEG2-LRC (ISO/TS 21219-7), TPEG2-LTE (ISO/TS 21219-24[2)]);

— Special applications: TPEG2-SNI (ISO/TS 21219-9), TPEG2-CAI (ISO/TS 21219-10);

— Location Referencing: TPEG2-ULR (ISO/TS 21219-11[3)]), TPEG2-GLR (ISO/TS 21219-21[4)]), TPEG2-OLR (ISO/TS 21219-22[5)]);

— Applications: TPEG2-PKI (ISO/TS 21219-14), TPEG2-TEC (ISO/TS 21219-15), TPEG2-FPI (ISO/TS 21219-16), TPEG2-TFP (ISO/TS 21219-18), TPEG2-WEA (ISO/TS 21219-19), TPEG2-RMR (ISO/TS 21219-23[6)]), TPEG2-EMI (ISO/TS 21219-25[7)]).

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 having 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:

SP13010/1.0/001

1) Under development.

2) To be published.

3) Under development.

4) Under development.

5) Under development.

6) To be published.

7) To be published.

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

Part 23:
Roads and multimodal routes (TPEG2-RMR)

1 Scope

New mobility services like car sharing, car rental or park and ride as well as the integration of different transport modes by multimodal or off-board navigation are gaining increasing importance. Furthermore, the cooperative management of the transport infrastructure requires the provision of precise information and guidance on dedicated routes from a central knowledge base to a traveller's mobile device.

Such use cases are addressed by the TPEG application defined in this document. The Road and Multimodal Routes (RMR) application enables the service provision for road routes as well as multimodal routes including more than one transport mode and parking. For example, an optimal multimodal route may include a drive by car to a train station with parking facility, a train connection to a station nearby the destination and a local public transport ride from the train station to the traveller's destination.

The standardized delivery, via TPEG technology, of routing information has some potential benefits for the users of an RMR TPEG service, for instance:

— Enabling of specialized routing services like scenic routing or Eco routing;

— The best use of the overall transport network, i.e. not only the road network;

— Cost and time savings to traveller;

— Harmonization of in-car navigation and traffic management, e.g. routing advices by variable message signs;

— Personalized service provisioning, i.e. information services considering the specific characteristics of a user.

Some of the use cases above, in particular personalized service, may require a Peer-to-Peer (P2P) communication while others may apply a broadcast communication approach, e.g. city routing.

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/TS 18234-11:2013, *Intelligent transport systems — Traffic and Travel Information (TTI) via transport protocol experts group, generation 1 (TPEG1) binary data format — Part 11: Location Referencing Container (TPEG1-LRC)*

ISO/TS 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)*

ISO/TS 21219-5, *Intelligent transport systems - Traffic and travel information (TTI) via transport protocol experts group, generation 2 (TPEG2) — Part 5: Service framework (TPEG2-SFW)*

ISO/TS 21219-7, *Intelligent transport systems — Traffic and travel information (TTI) via transport protocol experts group, generation 2 (TPEG2) — Part 7: Location Referencing Container (TPEG2-LRC)*

ISO/TS 21219-9, *Intelligent transport systems — Traffic and travel information (TTI) via transport protocol experts group, generation 2 (TPEG2) — Part 9: Service and network information (TPEG2-SNI)*

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

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

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

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

3.1
message management container
MMC
concept applied to the grouping of all message elements including Message Management Information of a TPEG-message together in one place

3.2
location referencing
means to provide information that allows a system to accurately identify a location

Note 1 to entry: The content of a location reference allows the location to be presented in a graphical or textual manner to the end-user (e.g. coloured network graphs) as well as to be used for navigational systems purposes.

3.3
location referencing container
concept applied to the grouping of all the location referencing elements, of a TPEG-message, together in one place

4 Abbreviated terms

ACID	Application and Content Identifier
AID	Application Identification
ADC	Application Data Container
GLR	TPEG LR method 'Geo Location Referencing'
CEN	Comité Européen de Normalisation
EBU	European Broadcasting Union
LR	Location Reference
LRC	Location Reference Container
MMC	Message Management Container
MMR	Multimodal Route
P2P	Peer-to-Peer (communication)
PKI	Parking Information application

RMR	Road and Multimodal Routes application
SFW	TPEG Service Framework: Modelling and Conversion Rules
TISA	Traveller Information Services Association
TPEG	Transport Protocol Expert Group
TTI	Traffic and Traveller Information
UML	Unified Modelling Language

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 message, for example parking information or road traffic information. Each TPEG application is assigned a unique number, called the Application Identification (AID). An AID is defined whenever a new application is developed and these are all listed in ISO/TS 21219-1.

The AID number is used within the TPEG-SNI application ISO/TS 21219-9 to indicate how to process TPEG content and 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 may have an impact on client devices.

The version numbering principle is defined in ISO/TS 21219-1 (TPEG-INV).

Table 1 shows the current version numbers for signalling RMR within the SNI application ISO/TS 21219-9:

Table 1 — Current version numbers for signalling of RMR

major version number	1
minor version number	0

5.3 Ordered components

TPEG2-RMR requires a fixed order of TPEG components. The order for the RMR message component is shown in Figure 1. The first component shall be the *Message Management Container (MMC).* This shall be the only component if the message is a cancellation message. Otherwise, the MMC component shall be followed by one or more *Application Data Container* (ADC) component(s) which includes the application-specific information.

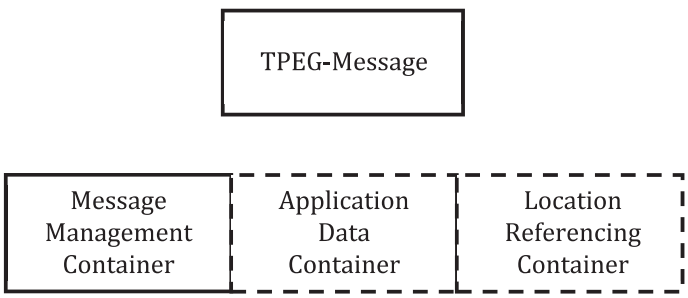

Figure 1 — Composition of TPEG messages

Note that the definition of the used Location Referencing methods is out-of-scope for the RMR specification and has to be agreed between the service provider and the client supplier.

5.4 Extendibility

The requirement of a fixed component order does not affect the extension of RMR. Future application extensions may insert new components or may replace existing components by new ones without losing backward compatibility. That means a RMR decoder shall be able to detect and skip unknown components.

5.5 TPEG Service Component Frame

RMR makes use of the "Service Component Frame with dataCRC and messageCount" according to specification ISO/TS 21219-5 (TPEG2-SFW).

6 RMR structure

6.1 Message structure

Figure 2 below shows the message structure of RMR. In case of a P2P communication this structure will also be used in the response of the TPEG server.

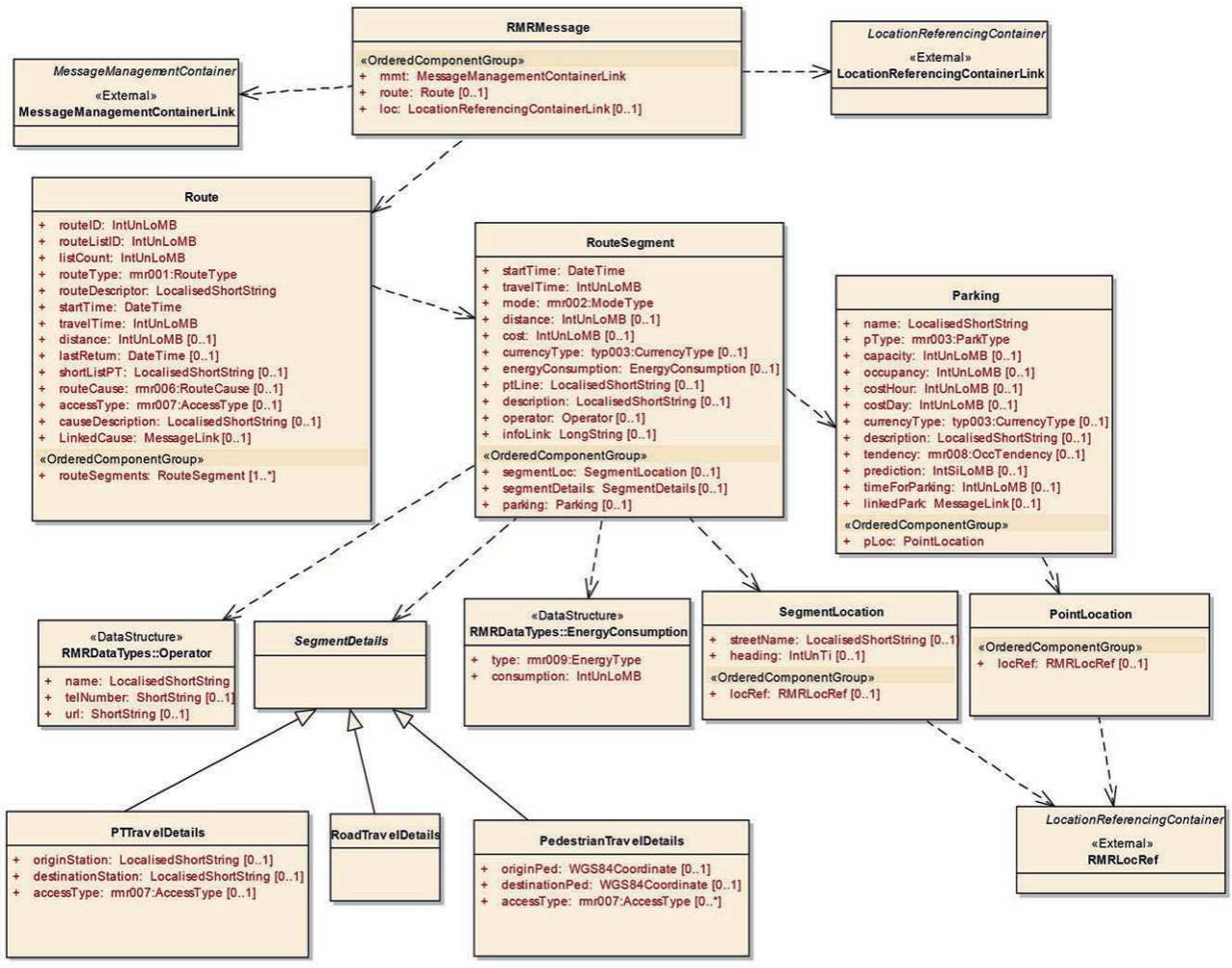

Figure 2 — RMR message structure

Figure 3 below shows the RMR request message structure. This message type will only be used for P2P communication to request routing information from the TPEG server.

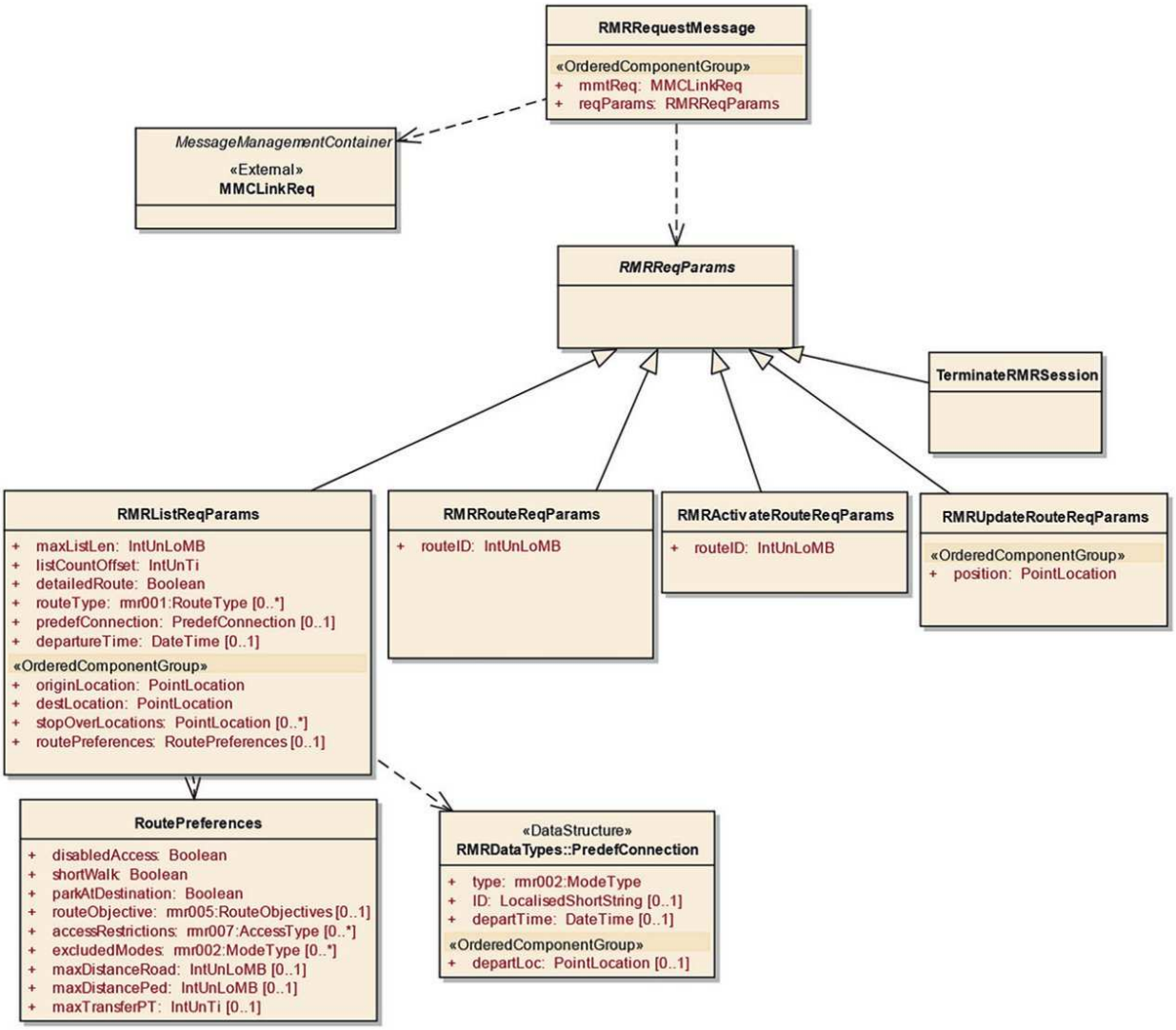

Figure 3 — RMR request message structure

NOTE Figure 4 below shows some examples for RMR.

— The first line describes a route consisting of just one route segment, in this example, travel on a road network;

— The second line shows travel on a road network, ending at a car park and followed by a pedestrian walk to the destination;

— The third line includes additionally a route segment using public transport.

Other combinations of route segments are possible. The RMR service has to take care that the sequence of routes makes sense, e.g. travel by private car is in general only possible for the first part of the travel segment (first route segment).

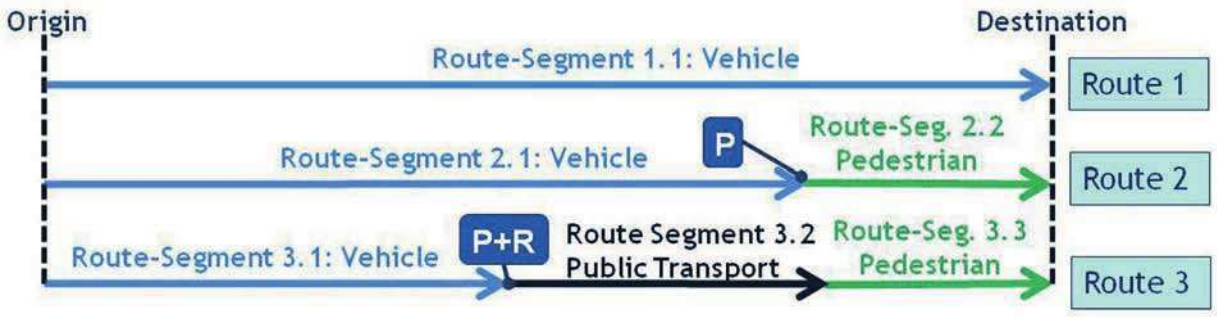

Figure 4 — Examples of RMR

6.2 Scenarios, features and requirements

6.2.1 General

An RMR message (see Figure 2) includes:

— One MMC only in case of a cancellation message;

— Otherwise:

  — one MMC, with a message ID unique for the route transmitted in this message;

  — one ADC with all required information of one route; and

  — optionally one Location Reference Container (LRC) describing the overall location of the route. As the particular route segments contain dedicated LRCs the general LRC on the message top level is of descriptive nature and may be used to provide a graphical representation of the overall route, e.g. by a GLR location reference.

6.2.2 Broadcast scenario

In a broadcast transmission, messages are sent from the TPEG server to a number of TPEG client devices. Thus, only the message structure described in Figure 2 is applicable for this scenario while requests are not possible. In this case, an RMR message shall include all service information required by the client.

6.2.3 P2P scenario

6.2.3.1 P2P transmission without request messages

In this scenario, the client only uses the general request mechanisms as defined in TPEG over HTTP 0 and does not send dedicated request messages on TPEG level (Figure 3). Thus, the same requirements as for the broadcast scenario apply.

6.2.3.2 P2P transmission with request messages

The request by dedicated TPEG RMR request messages enable a number of features, which are particularly important for personalized services:

1. Request for a route list:

  — The RMR client transmits with its request an RMR request message with an 'RMRListReqParams' component included. This component contains a number of parameters that define the user's characteristics and preferences concerning a road or multimodal route.

— The RMR server stores the request parameters of the client for this session and provides in its response a list of routes that fulfil the user's requirements delivered in the request message. The list enables the user to select a suitable route.

— The route list consists of a number of RMR messages but should provide restricted details. As a minimum requirement the components 'Route' and 'RouteSegment' shall be present in the messages delivered by the RMR server. Each list has a RouteId unique for this service.

— The client may request further route list entries by using the attribute 'maxListLen' and listCountOffset' (see also 7.1.3).

2. Request for detailed route information

— The RMR client may request detailed route information by one or several requests. For that, it transmits the 'RMRRouteReqParams' component with its RMR request message including the RouteId of the route of interest.

— The RMR server responds with an RMR message with the detailed information about the requested route. The RMR server shall use the same message ID as for the related route in the route list.

3. Selection of a route

— The RMR client may select or activate a route of the route list. For that, it transmits an 'RMRActivateRouteReqParams' component with its RMR request message including the RouteId of the selected route.

— The RMR server activates the route and is now able to keep the route up-to-date by sending updates. The RMR client can request this information by additional requests for detailed information (see list item 2 above).

4. Update of the route

— The RMR client may request updates of the information of the selected route information. For that, it transmits the 'RMRUpdateRouteReqParams' component with its RMR request message including its current position. The update may be requested iteratively.

— The RMR server responds with an RMR message with the detailed information about the currently activated route. The RMR server shall use the same message ID as for the related route in the route list.

5. Terminate a session

— The RMR client may actively terminate an RMR session by sending a 'TerminateRMRSession' component with its RMR request message.

— The RMR server may terminate the session and can release the user data transmitted by the route list request (see list item 1 above).

NOTE This document defines only a basic mandatory set of request parameters. An implementation can add vendor-specific attributes to the request components.

Figure 5 below shows a typical sequence of an RMR session.

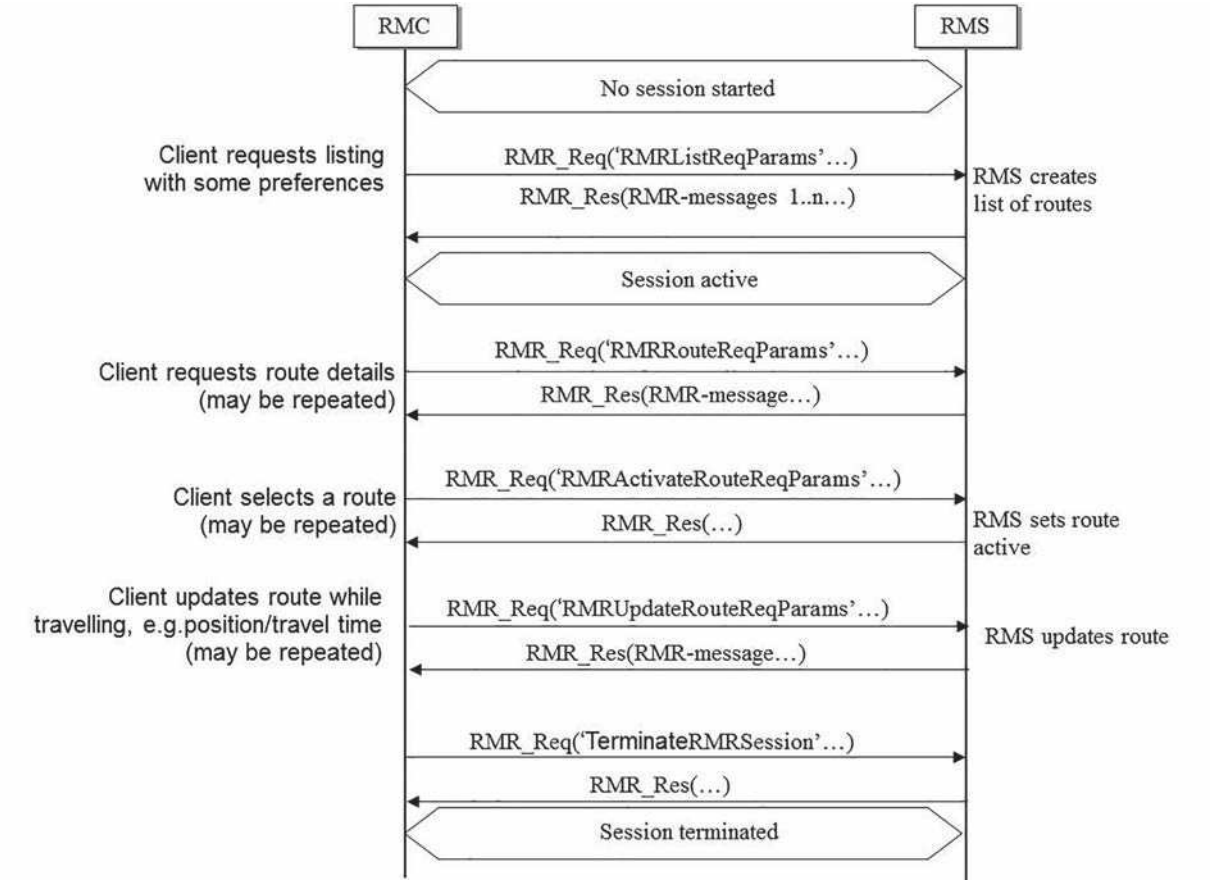

Figure 5 — Typical sequence of an RMR session (RMC/RMS = RMR client/server)

7 RMR Message components

7.1 RMRMessage

The description of the RMR message (see Table 2) contains a road route or a multimodal route.

The RMR message may be sent by an RMR server on request of a client device (P2P communication) or may be sent by broadcast if only a limited number of routes have to be transmitted.

Table 2 — RMRMessage

Name	Type	Multiplicity	Description
Ordered components			
mmt	MessageManagementContainerLink	1	The MMC of the RMR message. The RMR server shall use a service-wide unique message ID for each route and shall keep this ID as long as the route is valid to enable an easy update on the client side.
route	Route	0..1	A route delivered by the RMR server for a given origin and destination pair.
loc	LocationReferencingContainerLink	0..1	The overall location reference for the route. May be optionally used for supplemental information. As the specific routes are referenced by additional location references in its route-segments this LRC shall contain only a simple descriptive geographic outline of the route delivered by the RMR server (e.g. a GLR line reference). The LR shall be a linear location including the origin and the destination of the route.

7.2 Route

A route object (see Table 3) contains the information of one RMR route consisting of one or several route segments.

Table 3 — Route

Name	Type	Multiplicity	Description
routeID	IntUnLoMB	1	A handle or ID for the route; shall be unique for the service.
routeListID	IntUnLoMB	1	A handle or ID for the route list. Shall be used if the route is grouped in a route list and the value of this parameter shall be unique for the service in this case. If the route is not part of a list the value shall be zero.
listCount	IntUnLoMB	1	The value describes the position of the route in the route list delivered by the RMR server for a given origin destination pair, according to the sorting order of the server (broadcast scenario) or the client's request-preferences (P2P scenario). First position shall be 1. For instance, if the server provides the 10 fastest routes for an origin destination pair, value 2 means that this is the 2nd best alternative. The value shall be zero if the route is not a part of a route list.
routeType	rmr001:RouteType	1	Type of the route
routeDescriptor	LocalisedShortString	1	A description or name of the route (e.g. 'P&R Garching')
startTime	DateTime	1	Start time of the route
travelTime	IntUnLoMB	1	Overall travel time [seconds] of the route
distance	IntUnLoMB	0..1	Overall length or distance [metres] of the route; may not be present if not available or relevant (e.g. public transport)
lastReturn	DateTime	0..1	Time of last possible return trip for a multimodal route. Shall be present only in multimodal routes including PT or other operated services.
shortListPT	LocalisedShortString	0..1	A short list with the description of the public transport lines used for a multimodal route (e.g. 'U1 B244')

Table 3 *(continued)*

Name	Type	Multiplicity	Description
routeCause	rmr006:RouteCause	0..1	Descriptive information why this route has been proposed by the server. Only a few popular reasons are listed in the rmr:006 RouteCause table while others may be delivered by the 'causeDescription' attribute.
accessType	rmr007:AccessType	0..1	Type of access of overall route (e.g. ramp, stairs, elevator, etc. along route); relevant in particular for persons with restricted mobility.
causeDescription	LocalisedShortString	0..1	Further description why this route has been proposed by the server.
LinkedCause	MessageLink	0..1	Link to a TPEG message with detailed information about the route cause.
Ordered Components			
routeSegments	RouteSegment	1..*	The list of segments of the route. A new segment shall be added if the transport mode changes, e.g. one segment for a journey by car on the road network, another one for travel by bus.

7.3 RouteSegment

A route shall consist of one or several route segments (see Table 4) that may have different characteristics (e.g. transport mode).

Table 4 — - RouteSegment

Name	Type	Multiplicity	Description
startTime	DateTime	1	Scheduled time for start of this segment
travelTime	IntUnLoMB	1	Duration or travel time of this segment [seconds]
mode	rmr002:ModeType	1	Type of transport mode
distance	IntUnLoMB	0..1	Distance of walk or drive on this segment [metres]
cost	IntUnLoMB	0..1	Cost for this segment in local currency (using the lowest value local currency units, such as cents); for example public transport ticket or costs for gasoline; note that the value may be a rough estimate, as the basis for a detailed calculation may be unknown at the server side.
currencyType	typ003:CurrencyType	0..1	Local Currency
energyConsumption	EnergyConsumption	0..1	Energy consumption for the route segment; note that the value may be a rough estimate, as the basis for a detailed calculation may be unknown at the server side.
ptLine	LocalisedShortString	0..1	Short string with description of public transport line (e.g. head-sign 'U5'); shall only be used if segment is of PT-type.
description	LocalisedShortString	0..1	Further description of the segment.
operator	Operator	0..1	Operator of the travel service, e.g. public transport, car-sharing etc.
infoLink	LongString	0..1	A URL with further information, if available.
Ordered components			
segmentLoc	SegmentLocation	0..1	Location description of the segment.
segmentDetails	SegmentDetails	0..1	Detailed information for this route segment.
parking	Parking	0..1	Parking or park and ride location at end of the segment.

7.4 Parking

Description of a car park, a parking garage or a park and ride station (see Table 5).

Table 5 — Parking

Name	Type	Multiplicity	Description
name	LocalisedShortString	1	Name of the parking facility or park and ride facility
pType	rmr003:ParkType	1	Type of parking facility
capacity	IntUnLoMB	0..1	Capacity of parking facility, given as the number of parking spaces
occupancy	IntUnLoMB	0..1	Number of currently occupied parking spaces
costHour	IntUnLoMB	0..1	Cost for parking per hour given in local currency; if there is more than one currency unit (e.g. Euro and Cent) the currency unit with lower value shall be used (e.g. Cent).
costDay	IntUnLoMB	0..1	Cost for parking per day in local currency
currencyType	typ003:CurrencyType	0..1	Description of local currency
description	LocalisedShortString	0..1	Additional description of parking facility
tendency	rmr008:OccTendency	0..1	Tendency of occupancy
prediction	IntSiLoMB	0..1	Prediction of free capacity at arrival time at parking facility [number of parking spaces]
timeForParking	IntUnLoMB	0..1	Estimated time [sec] to get a parking place at parking facility, e.g. queuing at car park or search
linkedPark	MessageLink	0..1	Link to a TPEG message with detailed information about the parking facility
Ordered Components			
pLoc	PointLocation	1	The location of the parking facility

7.5 SegmentLocation

Table 6 describes a SegmentLocation.

A SegmentLocation may be a GLR line-reference or a map-matchable dynamic location reference.

If GLR is used, the line reference shall include at least the coordinates of the start and end of the segment where start/end means the beginning or end respectively (in driving direction) of the related segment.

A street name and heading may be added for further information.

Table 6 — SegmentLocation

Name	Type	Multiplicity	Description
streetName	LocalisedShortString	0..1	Name of the street of road section. This shall be used only if the section is on one named street (e.g. short pedestrian segment).
heading	IntUnTi	0..1	Heading of the client vehicle in 360/256 degrees resolution. Value range is 0 to 255, counted in counter-clockwise direction. Resulting Range: 000 = North 0° 064 = West 90° 128 = South 180° 196 = East 270°
Ordered components			
locRef	RMRLocRef	0..1	The location reference of the segment. Shall be a map-matchable LR-method in case of a road route segment (e.g. AGORA-C)

7.6 PointLocation

Table 7 describes a PointLocation, e.g.:

— a car park entry ramp location;

— an origin or destination location;

— a stopover location, e.g. for public transport.

The PointLocation can be defined by:

— a simple GLR point location coordinate and with optional street name, address or direction;

— a map-matchable dynamic location reference.

Table 7 — PointLocation

Name	Type	Multiplicity	Description
Ordered components			
locRef	RMRLocRef	0..1	The location reference of the location

7.7 SegmentDetails

SegmentDetails is an Abstract class construct for detailed segment information, switching to one of the classes 'RoadTravelDetails', 'PTTravelDetails' or 'PedestrianTravelDetails'.

7.8 RoadTravelDetails

RoadTravelDetails contains detailed information for a road route segment.

7.9 PTTravelDetails

Table 8 lists detailed information for a public transport route segment (see Table 8).

Table 8 — PTTravelDetails

Name	Type	Multiplicity	Description
originStation	LocalisedShortString	0..1	Name of the start or origin station/stop of the PT connection.
destinationStation	LocalisedShortString	0..1	Name of the destination station/stop of the PT connection.
accessType	rmr007:AccessType	0..1	Type of access (e.g. ramp, stairs, elevator, etc.); relevant in particular for persons with restricted mobility.

7.10 PedestrianTravelDetails

Table 9 lists detailed information for a pedestrian/walk route segment (see Table 9).

Table 9 — PedestrianTravelDetails

Name	Type	Multiplicity	Description
originPed	WGS84Coordinate	0..1	Origin or start of pedestrian walk
destinationPed	WGS84Coordinate	0..1	Destination of pedestrian walk
accessType	rmr007:AccessType	0..*	Type of access (e.g. stairs, elevator, etc.); relevant in particular persons with restricted mobility. Several access types may be listed but the most critical one (e.g. stairs) should be added.

7.11 RMRRequestMessage

Table 10 describes the request message of the RMR client (see Table 10). This shall only be required in case of a P2P transmission scenario. The request can be alternatively for a route list, detailed information on a specific route, an update of a route, a route selection/activation or the termination of the RMR session (see also P2P scenarios in 6.2.3.)

Table 10 — RMRRequestMessage

Name	Type	Multiplicity	Description
Ordered components			
mmtReq	MMCLinkReq	1	The MMC of the request message. The RMR client may use any message ID but shall use the same message ID for all its requests within one session. An increment of a version shall be used to indicate modified client preferences or conditions (e.g. position or change of route preferences).
reqParams	RMRReqParams	1	request parameters for a route list

7.12 RMRReqParams

RMRReqParams is an abstract class construct for switching the different classes for RMR requests, i.e. 'RMRListReqParams', 'RMRRouteReqParams', 'RMRActivateRouteReqParams' or 'RMRUpdateRouteReqParams'.

7.13 RMRListReqParams

Table 11 lists parameters for the route list request. An implementation may add vendor specific information if required.

Table 11 — RMRListReqParams

Name	Type	Multiplicity	Description
maxListLen	IntUnLoMB	1	The maximum length of the route list to be delivered, i.e. the number of messages to be returned by the server each representing a route alternative
listCountOffset	IntUnTi	1	offset for the alternative route list; offset value zero means that the server shall deliver the best first routes in its response where the maximum length of the list is defined by the attribute 'MaxListLen'. A value > 0 means that the delivered list shall start (in sorting order) at position 'listCountOffset'
detailedRoute	Boolean	1	Yes = add all available details to route list No = return just an overview, i.e. the components Routes and (optional) RouteSegment
routeType	rmr001:RouteType	0..*	types of route requested
predefConnection	PredefConnection	0..1	used if there is a predefined connection to be considered in multimodal route, e.g. a connection for a flight, a train or a ferry
departureTime	DateTime	0..1	Desired departure time of route at origin
Ordered components			
originLocation	PointLocation	1	origin location of the route request; this may be the current position of the client device (e.g. on-trip information) or the start location for a route planner (e.g. pre-trip information)
destLocation	PointLocation	1	destination location of the route request
stopOverLocations	PointLocation	0..*	stop over locations (optional)
routePreferences	RoutePreferences	0..1	additional route preferences

7.14 RMRRouteReqParams

Table 12 lists parameters of a request for delivery of detailed information of a dedicated route. An implementation may add vendor specific information if required.

Table 12 — RMRRouteReqParams — Route

Name	Type	Multiplicity	Description
routeID	IntUnLoMB	1	unique ID of the route the detailed or updated information is requested for

Table 13 lists parameters of a request for delivery of updates of a route previously activated or selected by an 'RMRActivateRoute' request. An implementation may add vendor specific information if required.

Table 13 — RMRRouteReqParams — Position

Name	Type	Multiplicity	Description
Ordered components			
position	PointLocation	1	Current position of the RMR client

7.15 RMRActivateRouteReqParams

Table 14 lists parameters for the activation of a route on the server side. An implementation may add vendor specific information if required.

Table 14 — RMRActivateRouteReqParams

Name	Type	Multiplicity	Description
routeID	IntUnLoMB	1	Unique ID of the route that shall be selected or activated

7.16 TerminateRMRSession

TerminateRMRSession is the request for active termination of an RMR session.

7.17 RoutePreferences

Table 15 lists preferences and restrictions for the requested route.

Table 15 — RoutePreferences

Name	Type	Multiplicity	Description
disabledAccess	Boolean	1	The routes shall be suitable for disabled people, e.g. it shall include no stairs
shortWalk	Boolean	1	Walk not excluded but should be short due to a restricted mobility of the requesting user (e.g. maximum 500m)
parkAtDestination	Boolean	1	Parking place at destination required
routeObjective	rmr005:RouteObjectives	0..1	Objective of route search, e.g. shortest route; default if not present is fastest route
accessRestrictions	rmr007:AccessType	0..*	Detailed descriptions of access types restrictions of the user, i.e. which types are excluded
excludedModes	rmr002:ModeType	0..*	Tansport modes excluded for a multimodal route search
maxDistanceRoad	IntUnLoMB	0..1	Maximum distance [kilometres] for road travel, e.g. used for electric vehicles with restricted cruising range
maxDistancePed	IntUnLoMB	0..1	Maximum distance [metres] for pedestrian walk in overall route
maxTransferPT	IntUnTi	0..1	Maximum acceptable number of transfers along public transport connection; shall not be used if public transport excluded

7.18 MessageManagementContainerLink

This component is a placeholder for the MMC. Partial message management shall not be used for RMR. The RMR server shall provide a unique message ID for each client request for a new origin destination pair. Updates of the origin/destination relation (e.g. changes of routes or new routes) shall be indicated by an identical MID with an incremented version ID.

7.19 RMRLocRef

RMRLocRef is the location reference used by an RMR message. The following requirements apply:

— In case of a route segment the RMRLocRef component shall include only linear locations. The component is mandatory for road segments and only dynamic LR-methods with map-matching capabilities shall be used in this case. The component may be used for other modes, e.g. a geo-linear path in case of a pedestrian walk.

— If used for a location reference for a car park entry or a destination location of a request message, only dynamic LR-methods with map-matching capabilities shall be used.

— If used for a position of the client device or the origin of a route request, any dynamic LR method may be used.

The RMRLocRef component is a placeholder for the LocationReferencingContainer (LRC) as described in the LRC toolkit specification: ISO/TS 18234-11 (for TPEG1) and ISO/TS 21219-7 (for TPEG2). It assigns the RMR application a specific local component ID for the LRC container. All component IDs within the LRC container are local to the LRC toolkit.

7.20 LocationReferencingContainerLink

As an RMR application delivers the routes by location references for the particular route-sections, an additional LRC for the overall route is optional but may be used as descriptive information. Therefore, only simple, dynamic Location Referencing methods shall be used. Recommended method is GLR. If present the location reference shall include the origin and the destination of the route.

The LocationReferencingContainer component is a placeholder for the LRC as described in the LRC toolkit specification: ISO/TS 18234-11 (for TPEG1) and ISO/TS 21219-7 (for TPEG2). It assigns the RMR application a specific local component ID for the LRC container. All component IDs within the LRC container are local to the LRC toolkit.

7.21 MMCLinkReq

MMCLinkReq is the link to the MMC for RMR requests. Partial message management shall not be used. The RMR client shall provide a unique message ID for each new request, i.e. for new origin destination pairs.

8 RMR datatypes

8.1 EnergyConsumption

Table 16 describes the data structure containing a key/value-pair for overall absolute energy consumption of a route or a route segment. If the type of the engine (gasoline, diesel, electric, etc.) of the client-vehicle is unknown at the server side, the 'fuel' type shall be used as default. Otherwise the specific kinds of fuel/energy may be used.

Table 16 — EnergyConsumption

Name	Type	Multiplicity	Description
type	rmr009:EnergyType	1	type of consumed energy
consumption	IntUnLoMB	1	value of consumption in units according type

8.2 Operator

Table 17 describes the operator of public transport, ship, railway, car-sharing, rental bike etc.

Operator shall not be used in the case of a private car.

Table 17 — Operator

Name	Type	Multiplicity	Description
name	LocalisedShortString	1	Name of the operator
telNumber	ShortString	0..1	Telephone number of the operator, only numbers without any further characters (e.g. '+', '-' '#') shall be used. International direct dial numbers shall be converted to numeric values (e.g. "+44" – > "0044").
url	ShortString	0..1	The url of the operator

8.3 PredefConnection

Table 18 describes the attributes that define a pre-defined connection within route, e.g. a booked train-travel, a flight or a ferry.

Table 18 — PredefConnection

Name	Type	Multiplicity	Description
type	rmr002:ModeType	1	mode type of connection
ID	LocalisedShortString	0..1	ID of the connection, e.g. flight number
departTime	DateTime	0..1	departure time of connection, e.g. boarding time of flight
Ordered components			
departLoc	PointLocation	0..1	departure location of connection, e.g. an airport or train station

8.4 WGS84Coordinate

Table 19 describes the data-structure for WGS84 coordinates in absolute values.

Table 19 — WGS84Coordinate

Name	Type	Multiplicity	Description
long	IntSi24	1	Longitude in standard 24 bit accuracy stores coordinates in order of magnitude of 10 micro degrees resolution (five decimals). The value (unit degrees) is encoded as follows: long = int (sign(longitude)*0.5 + (longitude * (2^24))/360)
lat	IntSi24	1	WGS84 latitude in standard 24 bit accuracy stores co-ordinates in order of magnitude of 10 micro degrees resolution (five decimals). The value (unit degrees) is encoded as follows: lat = int (sign(latitude)*0.5 + (latitude * (2^24))/ 360)

8.5 MessageLink

Table 20 describes the data structure that may be used if required to link to a TPEG message with more details about a route cause or a parking facility.

A link to another message is uniquely specified by the combination of ServiceID, ContentID, ApplicationID and messageID.

Table 20 — MessageLink

Name	Type	Multiplicity	Description
messageID	IntUnLoMB	1	The related message ID
COID	IntUnTi	1	Content ID of the TPEG service component related to the linked message
AID	IntUnLi	1	Application ID of the TPEG service component related to the linked message; in case of a PKI message for further information on the parking facility the value is 3 (PKI), in case of a TEC message for further information about the route cause the value is 5 (TEC)
SID	ServiceIdentifier	0..1	The TPEG service ID related to the service of the linked message; this attribute may be omitted if the linked message is in the same TPEG service as this RMR message

9 RMR Tables

9.1 rmr001:RouteType

Type of the route (see Table 21).

Table 21 — RouteType

Code	Reference-English 'word'	Comment	Example
000	unknown		
001	roadRoute	A road route; may include also cost factors for a local navigation	
002	greenRoute	A route enabling reduced fuel or energy consumption	
003	multimodalRoute	A multimodal route, i.e. including more than one transport mode	

9.2 rmr002:ModeType

Transport mode of the related route segment (see Table 22).

Table 22 — ModeType

Code	Reference-English 'word'	Comment	Example
001	car	The segment describes a car journey on road network.	
002	pedestrian	The segment describes a pedestrian walk.	
003	bus	The segment describes public transport travel by bus.	
004	subway	The segment describes public transport travel by subway/underground railway.	
005	train	The segment describes public transport travel by train.	
006	tram	The segment describes public transport travel by tram or light train.	

Table 22 *(continued)*

Code	Reference-English 'word'	Comment	Example
007	STrain	The segment describes public transport travel by S-train or regional train.	
008	carRental	The segment describes travel by rental car or by car sharing.	
009	airplane	The segment describes travel by air-plane.	
010	ship	The segment describes travel by ship or ferry.	
011	platformChange	Pedestrian walk including change of platform at a station (e.g. PT or railway).	
012	lineNameChangePT	The PT line name changes although no change of PT-vehicle or train is required, so the ModeType remains the same as for the previous route segment.	
013	bike	The segment describes travel by bicycle.	
014	bikeRental	The segment describes travel by rental bicycle or bicycle on demand.	
015	carRental	The segment describes travel by rental car or car-sharing.	
099	others	The segment describes travel by any other transport mode.	

9.3 rmr003:ParkType

Type of parking facility (see Table 23).

Table 23 — ParkType

Code	Reference-English 'word'	Comment	Example
000	unknown		
001	CarPark	Car park or parking garage	
002	ParkAndRide	Park and ride facility	
003	OthersPark	Other parking options, e.g. along street	

9.4 rmr005:RouteObjectives

Type of route objective (see Table 24).

Table 24 — RouteObjectives

Code	Reference-English 'word'	Comment	Example
001	fastest	fastest route with minimal travel time requested	
002	shortest	shortest route with minimal travel distance requested	
003	minimalEnergy	route with minimal energy consump-tion requested	
004	scenic	most beautiful route requested, e.g. sight-seeing tour	

9.5 rmr006:RouteCause

RouteCause defines descriptive information to explain why this route has been proposed by the server. Only a few popular reasons are listed in this table while others may be delivered by the 'causeDescription' attribute (see Table 25).

Table 25 — RouteCause

Code	Reference-English 'word'	Comment	Example
000	unknown		
001	optimalRoute	route is optimal according to user's route preferences	
002	disturbancePT	operational disturbances on public transport	
003	roadClosure	parts of the road network have been closed or are blocked	
004	connectionInvalid	a given PT connection is no longer valid	

9.6 rmr007:AccessType

Table 26 provides additional information for persons with restricted mobility.

Table 26 — AccessType

Code	Reference-English 'word'	Comment	Example
000	unknown	No information about barriers	
001	accessible	The pedestrian route (or path) or the interchange between public transport lines is without mobility obstructions for elderly and disabled persons with restricted mobility	
002	escalator	There is an escalator on the pedestrian route (or path) or the interchange between public transport lines	
003	elevator	There is an elevator on the pedestrian route (or path) or the interchange between public transport lines	
004	stairs	There are stairs on the pedestrian route (or path) or the interchange between public transport lines	
005	ramp	There is a ramp on the pedestrian route (or path) or the interchange between public transport lines	
006	gap	There is a gap on the pedestrian route (or path) or the interchange between public transport lines (e.g. between train and station platform)	

9.7 rmr008:OccTendency

Tendency of the occupancy of a parking facility (see Table 27).

Table 27 — OccTendency

Code	Reference-English 'word'	Comment	Example
000	unknown	Tendency unknown	
001	constant	Forecast of occupancy is stable, i.e. remains as is	
002	increasing	Forecast for occupancy is increasing, higher occupancy expected	
003	decreasing	Forecast for occupancy is decreasing, lower occupancy expected	

9.8 rmr009:EnergyType

Type of energy or fuel (see Table 28).

Table 28 — EnergyType

Code	Reference-English 'word'	Comment	Example
000	unknown	Unknown fuel or engine type; the server shall assume fuel/gasoline type in this case	
001	fuel	Gasoline/fuel type; unit for consumption shall be [0.1 l]	
002	diesel	Diesel type; unit for consumption shall be [0.1 l]	
003	CNG	Compressed Natural Gas type; unit for consumption shall be [0.1 kg]	
004	LPG	Liquified Petroleum Gas type; unit for consumption shall be [0.1 l]	
005	H2	H2/hydrogen type; unit for consumption shall be [0.1 kg]	
006	electric	(Battery) electric type; unit for consumption shall be [0.1 kw/h]	

9.9 rmr099:ErrorCode

Error code delivered by the RMR server in case of a non-successful request of the client. Note that only errors related to the RMR application are handled here. Errors on lower protocol layers or general errors (e.g. time out of request, invalid URL, server overload) are handled by the adaptation layer profiles used for the transmission, e.g. THTTP (see Table 29).

Table 29 — ErrorCode

Code	Reference-English 'word'	Comment	Example
001	invalid location		

Annex A
(normative)

TPEG application — TPEG-Binary Representation

A.1 Message components

A.1.1 List of Generic Component Ids

Table A.1 defines the Generic Component Ids for use in the TPEG-Binary representation.

Table A.1 — Generic Component Ids

Name	Id
RMRMessage	0
MessageManagementContainerLink	1
LocationReferencingContainerLink	2
Route	3
RouteSegment	4
Parking	5
SegmentLocation	6
PointLocation	7
RMRLocRef	8
PedestrianTravelDetails	9
RoadTravelDetails	10
PTTravelDetails	11
RMRRequestMessage	20
MMCLinkReq	21
RMRListReqParams	22
RMRRouteReqParams	23
RMRActivateRouteReqParams	24
RMRUpdateRouteReqParams	25
TerminateRMRSession	26
RoutePreferences	27

A.1.2 RMRMessage

< RMRMessage(0) > : =	
< IntUnTi > (0),	: id of this component
< IntUnLoMB > (lengthComp),	: number of bytes in component
< IntUnLoMB > (lengthAttr),	: number of bytes in attributes
ordered {	
< MessageManagementContainerLink > (mmt),	: The MMC of the RMR message. The RMR server shall use a service-wide unique message ID for each route and shall keep this ID as long as the route is valid to enable an easy update on the client side.

Table A.1 *(continued)*

n * < Route > (route)[0..1],	: A route delivered by the RMR server for a given origin and destination pair.
n * < LocationReferencingContainerLink > (loc) [0..1]	: The overall location reference for the route. May be optionally used for supplemental information. As the specific routes are referenced by additional location references in its route-segments this LRC shall contain only a simple descriptive geographic outline of the route delivered by the RMR server (e.g. a GLR line reference). The LR shall be a linear location including the origin and the destination of the route.
};	

A.1.3 Route

< Route(3) > : =	
< IntUnTi > (3),	: id of this component
< IntUnLoMB > (lengthComp),	: number of bytes in component
< IntUnLoMB > (lengthAttr),	: number of bytes in attributes
< IntUnLoMB > (routeID),	: A handle or ID for the route; shall be unique for the service.
< IntUnLoMB > (routeListID),	: A handle or ID for the route list. Shall be used if the route is grouped in a route list and the value of this parameter shall be unique for the service in this case. If the route is not part of a list the value shall be zero.
< IntUnLoMB > (listCount),	: The value describes the position of the route in the route list delivered by the RMR server for a given origin destination pair, according to the sorting order of the server (broadcast scenario) or the client's request-preferences (P2P scenario). First position shall be 1. For instance, if the server provides the 10 fastest routes for an origin destination pair, value 2 means that this is the 2nd best alternative. The value shall be zero if the route is not a part of a route list.
< rmr001:RouteType > (routeType),	: Type of the route
< LocalisedShortString > (routeDescriptor),	: A description or name of the route (e.g. 'P&R Garching')
< DateTime > (startTime),	: Start time of the route
< IntUnLoMB > (travelTime),	: overall travel time [seconds] of the route
BitArray(selector),	
if (bit 0 of selector is set)	
< IntUnLoMB > (distance),	: overall length or distance [metres] of the route; may not be present if not available or relevant (e.g. public transport)
if (bit 1 of selector is set)	
< DateTime > (lastReturn),	: Time of last possible return trip for a multimodal route. Shall be present only in multimodal routes including PT or other operated services, accordingly.
if (bit 2 of selector is set)	
< LocalisedShortString > (shortListPT),	: A short list with the description of the public transport lines used for a multimodal route (e.g. 'U1 B244')
if (bit 3 of selector is set)	

< rmr006:RouteCause > (routeCause),	: Descriptive information why this route has been proposed by the server. Only a few popular reasons are listed in the rmr:006 RouteCause table while others may be delivered by the 'causeDescription' attribute.
if (bit 4 of selector is set)	
< rmr007:AccessType > (accessType),	: Type of access of overall route (e.g. ramp, stairs, elevator, etc. along route); relevant in particular for persons with restricted mobility.
if (bit 5 of selector is set)	
< LocalisedShortString > (causeDescription),	: Further description why this route has been proposed by the server.
if (bit 6 of selector is set)	
< MessageLink > (LinkedCause),	: Link to a TPEG message with detailed information about the route cause
ordered {	
n * < RouteSegment > (routeSegments)	: The list of segments of the route. A new segment shall be added if the transport mode changes, e.g. one segment for a journey by car on the road network, another one for travel by bus.
};	

A.1.4 RouteSegment

< RouteSegment(4) > : =	
< IntUnTi > (4),	: id of this component
< IntUnLoMB > (lengthComp),	: number of bytes in component
< IntUnLoMB > (lengthAttr),	: number of bytes in attributes
< DateTime > (startTime),	: Scheduled time for start of this segment
< IntUnLoMB > (travelTime),	: Duration or travel time of this segment [seconds]
< rmr002:ModeType > (mode),	: Type of transport mode
BitArray(selector),	
if (bit 0 of selector is set)	
< IntUnLoMB > (distance),	: Distance of walk or drive on this segment [metres]
if (bit 1 of selector is set)	
< IntUnLoMB > (cost),	: Cost for this segment in local currency (using the lowest value local currency units, such as cents); for example public transport ticket or costs for gasoline; note that the value may be a rough estimate as the basis for a detailed calculation may be unknown at the server side.
if (bit 2 of selector is set)	
< typ003:CurrencyType > (currencyType),	: Local Currency type
if (bit 3 of selector is set)	
< EnergyConsumption > (energyConsumption),	: Energy consumption for the route segment; note that the value may be a rough estimate as the basis for a detailed calculation may be unknown at the server side.
if (bit 4 of selector is set)	
< LocalisedShortString > (ptLine),	: Short string with description of public transport line (e.g. head-sign 'U5'); shall only be used if segment is of PT-type.
if (bit 5 of selector is set)	

< LocalisedShortString > (description),	: Further description of the segment.
if (bit 6 of selector is set)	
< Operator > (operator),	: Operator of the travel service, e.g. public transport, car-sharing etc.
if (bit 7 of selector is set)	
< LongString > (infoLink),	: A URL with further information, if available.
ordered {	
n * < SegmentLocation > (segmentLoc)[0..1],	: Location description of the segment
n * < SegmentDetails > (segmentDetails)[0..1],	: Detailed information for this route segment
n * < Parking > (parking)[0..1]	: Parking or park and ride location at end of the segment
};	

A.1.5 Parking

< Parking(5) > : =	
< IntUnTi > (5),	: id of this component
< IntUnLoMB > (lengthComp),	: number of bytes in component
< IntUnLoMB > (lengthAttr),	: number of bytes in attributes
< LocalisedShortString > (name),	: Name of the parking facility or park and ride facility
< rmr003:ParkType > (pType),	: Type of parking facility
BitArray(selector),	
if (bit 0 of selector is set)	
< IntUnLoMB > (capacity),	: Capacity of parking facility, given as the number of parking spaces
if (bit 1 of selector is set)	
< IntUnLoMB > (occupancy),	: Number of currently occupied parking spaces
if (bit 2 of selector is set)	
< IntUnLoMB > (costHour),	: Cost for parking per hour given in local currency; if there is more than one currency unit (e.g. Euro and Cent) the currency unit with the lower value shall be used (e.g. Cent).
if (bit 3 of selector is set)	
< IntUnLoMB > (costDay),	: Cost for parking per day in local currency
if (bit 4 of selector is set)	
< typ003:CurrencyType > (currencyType),	: Description of local currency
if (bit 5 of selector is set)	
< LocalisedShortString > (description),	: Additional description of parking facility
if (bit 6 of selector is set)	
< rmr008:OccTendency > (tendency),	: Tendency of occupancy
if (bit 7 of selector is set)	
< IntSiLoMB > (prediction),	: Prediction of free capacity at arrival time at parking facility [number of parking spaces]
if (bit 8 of selector is set)	
< IntUnLoMB > (timeForParking),	: Estimated time [seconds] to get a parking place at parking facility, e.g. queuing at car park or search
if (bit 9 of selector is set)	
< MessageLink > (linkedPark),	: Link to a TPEG message with detailed information about the parking facility

ordered {	
< PointLocation > (pLoc)	: The location of the parking facility
};	

A.1.6 SegmentLocation

< SegmentLocation(6) > : =	
< IntUnTi > (6),	: id of this component
< IntUnLoMB > (lengthComp),	: number of bytes in component
< IntUnLoMB > (lengthAttr),	: number of bytes in attributes
BitArray(selector),	
if (bit 0 of selector is set)	
< LocalisedShortString > (streetName),	: Name of the street of road section. This shall be used only if the section is on one named street (e.g. short pedestrian segment).
if (bit 1 of selector is set)	
< IntUnTi > (heading),	: Heading of the client vehicle in 360/256 degree resolution. Value range is 0 to 255, counted in counter-clockwise direction. Resulting Range: 000 = North 0° 064 = West 90° 128 = South 180° 196 = East 270°
ordered {	
n * < RMRLocRef > (locRef)[0..1]	: The location reference of the segment. Shall be a map-matchable LR-method in case of a road route segment (e.g. AGORA-C)
};	

A.1.7 PointLocation

< PointLocation(7) > : =	
< IntUnTi > (7),	: id of this component
< IntUnLoMB > (lengthComp),	: number of bytes in component
< IntUnLoMB > (lengthAttr),	: number of bytes in attributes
ordered {	
n * < RMRLocRef > (locRef)[0..1]	: The location reference of the location
};	

A.1.8 SegmentDetails

< SegmentDetails(x) > : =	
< IntUnTi > (x),	: id of this component
< IntUnLoMB > (lengthComp),	: number of bytes in component
< IntUnLoMB > (lengthAttr);	: number of bytes in attributes

A.1.9 RoadTravelDetails

< RoadTravelDetails(10) < SegmentDetails(10) > > : =	
< IntUnTi > (10),	: id of this component
< IntUnLoMB > (lengthComp),	: number of bytes in component
< IntUnLoMB > (lengthAttr),	: number of bytes in attributes

A.1.10 PTTravelDetails

< PTTravelDetails(11) < SegmentDetails(11) > > : =	
< IntUnTi > (11),	: id of this component
< IntUnLoMB > (lengthComp),	: number of bytes in component
< IntUnLoMB > (lengthAttr),	: number of bytes in attributes
BitArray(selector),	
if (bit 0 of selector is set)	
< LocalisedShortString > (originStation),	: Name of the start origin station/stop of the PT connection
if (bit 1 of selector is set)	
< LocalisedShortString > (destinationStation),	: Name of the destination station/stop of the PT connection
if (bit 2 of selector is set)	
< rmr007:AccessType > (accessType);	: Type of access (e.g. ramp, stairs, elevator, etc.); relevant in particular for persons with restricted mobility.

A.1.11 PedestrianTravelDetails

< PedestrianTravelDetails(9) < SegmentDetails(9) > > : =	
< IntUnTi > (9),	: id of this component
< IntUnLoMB > (lengthComp),	: number of bytes in component
< IntUnLoMB > (lengthAttr),	: number of bytes in attributes
BitArray(selector),	
if (bit 0 of selector is set)	
< WGS84Coordinate > (originPed),	: Origin or start of pedestrian walk
if (bit 1 of selector is set)	
< WGS84Coordinate > (destinationPed),	: Destination of pedestrian walk
if (bit 2 of selector is set)	
{	
< IntUnLoMB > (n),	
n * < rmr007:AccessType > (accessType)	: Type of access (e.g. stairs, elevator, etc.); relevant in particular for persons with restricted mobility. Several access types may be listed but the most critical one (e.g. stairs) should be added.
};	

A.1.12 RMRRequestMessage

< RMRRequestMessage(20) > : =	
< IntUnTi > (20),	: id of this component
< IntUnLoMB > (lengthComp),	: number of bytes in component

< IntUnLoMB > (lengthAttr),	: number of bytes in attributes
ordered {	
< MMCLinkReq > (mmtReq),	: The MMC of the request message. The RMR client may use any message ID but shall use the same message ID for all its requests within one session. An increment of a version shall be used to indicate modified client preferences or conditions (e.g. position or change of route preferences).
< RMRReqParams > (reqParams)	: request parameters for route list
};	

A.1.13 RMRReqParams

< RMRReqParams(x) > : =	
< IntUnTi > (x),	: id of this component
< IntUnLoMB > (lengthComp),	: number of bytes in component
< IntUnLoMB > (lengthAttr);	: number of bytes in attributes

A.1.14 RMRListReqParams

< RMRListReqParams(22) < RMRReqParams(22) > > : =	
< IntUnTi > (22),	: id of this component
< IntUnLoMB > (lengthComp),	: number of bytes in component
< IntUnLoMB > (lengthAttr),	: number of bytes in attributes
< IntUnLoMB > (maxListLen),	: The maximum length of route list to be delivered, i.e. the number of messages to be returned by the server each representing a route alternative
< IntUnTi > (listCountOffset),	: offset for the alternative route list; offset value zero means that the server shall deliver the best first routes in its response where the maximum length of the list is defined by the attribute 'MaxListLen'. A value > 0 means that the delivered list shall start (in sorting order) at position 'listCount'Offset'
BitArray(selector),	
if (bit 0 of selector is set)	
< Boolean > (detailedRoute),	: Yes = add all available details to route list No = return just an overview, i.e. the components Routes and (optional) RouteSegment
if (bit 1 of selector is set)	
{	
< IntUnLoMB > (n),	
n * < rmr001:RouteType > (routeType),	: types of route requested
}	
if (bit 2 of selector is set)	
< PredefConnection > (predefConnection),	: used if there is a predefined connection to be considered in multimodal route, e.g. a flight, a train or a ferry
if (bit 3 of selector is set)	
< DateTime > (departureTime),	: Desired departure time of route at origin
ordered {	

< PointLocation > (originLocation),	: origin location of the route request; this may be the current position of the client device (e.g. on-trip information) or the start location for a route planner (e.g. pre-trip information)
< PointLocation > (destLocation),	: destination location of the route request
n * < PointLocation > (stopOverLocations),	: stop over locations (optional)
n * < RoutePreferences > (routePreferences)[0..1]	: additional route preferences
};	

A.1.15 RMRRouteReqParams

< RMRRouteReqParams(23) < RMRReqParams(23) > > : =	
< IntUnTi > (23),	: id of this component
< IntUnLoMB > (lengthComp),	: number of bytes in component
< IntUnLoMB > (lengthAttr),	: number of bytes in attributes
< IntUnLoMB > (routeID);	: unique ID of the route the detailed or update information is requested for

A.1.16 RMRUpdateRouteReqParams

< RMRUpdateRouteReqParams(25) < RMRReqParams(25) > > : =	
< IntUnTi > (25),	: id of this component
< IntUnLoMB > (lengthComp),	: number of bytes in component
< IntUnLoMB > (lengthAttr),	: number of bytes in attributes
ordered {	
< PointLocation > (position)	: Current position of the RMR client
};	

A.1.17 RMRActivateRouteReqParams

< RMRActivateRouteReqParams(24) < RMRReqParams(24) > > : =	
< IntUnTi > (24),	: id of this component
< IntUnLoMB > (lengthComp),	: number of bytes in component
< IntUnLoMB > (lengthAttr),	: number of bytes in attributes
< IntUnLoMB > (routeID);	: Unique ID of the route that shall be selected or activated

A.1.18 TerminateRMRSession

< TerminateRMRSession(26) < RMRReqParams(26) > > : =	
< IntUnTi > (26),	: id of this component
< IntUnLoMB > (lengthComp),	: number of bytes in component
< IntUnLoMB > (lengthAttr),	: number of bytes in attributes

A.1.19 RoutePreferences

< RoutePreferences(27) > : =	
< IntUnTi > (27),	: id of this component
< IntUnLoMB > (lengthComp),	: number of bytes in component
< IntUnLoMB > (lengthAttr),	: number of bytes in attributes
BitArray(selector),	
if (bit 0 of selector is set)	
< Boolean > (disabledAccess),	: The routes shall be suitable for disabled people, e.g. it shall include no stairs
if (bit 1 of selector is set)	
< Boolean > (shortWalk),	: Walk not excluded but should be short due to a restricted mobility of the requesting user (e.g. maximum 500m)
if (bit 2 of selector is set)	
< Boolean > (parkAtDestination),	: Parking place at destination required
if (bit 3 of selector is set)	
< rmr005:RouteObjectives > (routeObjective),	: objective of route search, e.g. shortest route; default if not present is fastest route
if (bit 4 of selector is set)	
{	
< IntUnLoMB > (n),	
n * < rmr007:AccessType > (accessRestrictions),	: Detailed descriptions of access type restrictions of the user, i.e. which types are excluded
}	
if (bit 5 of selector is set)	
{	
< IntUnLoMB > (n),	
n * < rmr002:ModeType > (excludedModes),	: transport modes excluded for a multimodal route search
}	
if (bit 6 of selector is set)	
< IntUnLoMB > (maxDistanceRoad),	: Maximum distance [kilometres] for road travel, e.g. used for electric vehicles with restricted cruising range
if (bit 7 of selector is set)	
< IntUnLoMB > (maxDistancePed),	: Maximum distance [metres] for pedestrian walk in overall route
if (bit 8 of selector is set)	
< IntUnTi > (maxTransferPT);	: Maximum acceptable number of transfers along public transport connection; shall not be used if public transport excluded

A.1.20 MessageManagementContainerLink

< MessageManagementContainerLink(1) > : =	
External < MessageManagementContainer(1) > ;	: see MessageManagementContainer specification

A.1.21 RMRLocRef

< RMRLocRef(8) > : =	
External < LocationReferencingContainer(8) > ;	: see LocationReferencingContainer specification

A.1.22 LocationReferencingContainerLink

< LocationReferencingContainerLink(2) > : =	
External < LocationReferencingContainer(2) > ;	: see LocationReferencingContainer specification

A.1.23 MMCLinkReq

< MMCLinkReq(21) > : =	
External < MessageManagementContainer(21) > ;	: see MessageManagementContainer specification

A.2 RMR Datatypes

A.2.1 EnergyConsumption

< EnergyConsumption > : =	
< rmr009:EnergyType > (type),	: type of consumed energy
< IntUnLoMB > (consumption);	: value of consumption in units according type

A.2.2 Operator

< Operator > : =	
< LocalisedShortString > (name),	: Name of the operator
BitArray(selector),	
if (bit 0 of selector is set)	
< ShortString > (telNumber),	: Telephone number of the operator; only numbers without any further characters (e.g. '+', '-' '#') shall be used
if (bit 1 of selector is set)	
< ShortString > (url);	: The url of the operator

A.2.3 PredefConnection

< PredefConnection > : =	
< rmr002:ModeType > (type),	: mode type of connection
BitArray(selector),	
if (bit 0 of selector is set)	
< LocalisedShortString > (ID),	: ID of the connection, e.g. flight number
if (bit 1 of selector is set)	
< DateTime > (departTime),	: departure time of connection, e.g. boarding time of flight
if (bit 2 of selector is set)	
< PointLocation > (departLoc);	: departure location of connection, e.g. an airport or train station

A.2.4 WGS84Coordinate

< WGS84Coordinate > : =	
< IntSi24 > (long),	: Longitude in standard 24 bit accuracy stores coordinates in order of magnitude of 10 micro degrees resolution (five decimals). The value (unit degrees) is encoded as follows: long = int (sign(longitude)*0.5 + (longitude * (2^24))/360)
< IntSi24 > (lat);	: WGS84 latitude in standard 24 bit accuracy stores coordinates in order of magnitude of 10 micro degrees resolution (five decimals). The value (unit degrees) is encoded as follows: lat = int (sign(latitude)*0.5 + (latitude * (2^24))/ 360)

A.2.5 MessageLink

< MessageLink > : =	
< IntUnLoMB > (messageID),	: The related message ID
< IntUnTi > (COID),	: Content ID of the TPEG service component related to the linked message
< IntUnLi > (AID),	: Application ID of the TPEG service component related to the linked message; in case of a PKI message for further information on the parking facility the value is 3 (PKI), in case of a TEC message for further information about the route-cause the value is 5 (TEC),
BitArray(selector),	
if (bit 0 of selector is set)	
< ServiceIdentifier > (SID);	: The TPEG service ID related to the service of the linked message; this attribute may be omitted if the linked message is in the same TPEG service as this RMR message

Annex B
(normative)

TPEG application — TPEG-ML Representation

B.1 Message Components

B.1.1 RMRMessage

```
<xs:element name="RMRMessage" type="RMRMessage"/>
<xs:complexType name="RMRMessage">
 <xs:complexContent>
 <xs:extension base="tsf:ApplicationRootMessageML">
 <xs:sequence>
 <xs:element name="mmt" type="mmc:MessageManagementContainer"/>
 <xs:element name="route" type="Route" minOccurs="0"/>
 <xs:element name="loc" type="lrc:LocationReferencingContainer" minOccurs="0"/>
 </xs:sequence>
 </xs:extension>
 </xs:complexContent>
</xs:complexType>
```

B.1.2 Route

```
<xs:complexType name="Route">
 <xs:sequence>
 <xs:element name="routeID" type="tdt:IntUnLoMB"/>
 <xs:element name="routeListID" type="tdt:IntUnLoMB"/>
 <xs:element name="listCount" type="tdt:IntUnLoMB"/>
 <xs:element name="routeType" type="rmr001_RouteType"/>
 <xs:element name="routeDescriptor" type="tdt:LocalisedShortString"/>
 <xs:element name="startTime" type="tdt:DateTime"/>
 <xs:element name="travelTime" type="tdt:IntUnLoMB"/>
 <xs:element name="distance" type="tdt:IntUnLoMB" minOccurs="0"/>
 <xs:element name="lastReturn" type="tdt:DateTime" minOccurs="0"/>
 <xs:element name="shortListPT" type="tdt:LocalisedShortString" minOccurs="0"/>
 <xs:element name="routeCause" type="rmr006_RouteCause" minOccurs="0"/>
 <xs:element name="accessType" type="rmr007_AccessType" minOccurs="0"/>
 <xs:element name="causeDescription" type="tdt:LocalisedShortString" minOccurs="0"/>
 <xs:element name="LinkedCause" type="MessageLink" minOccurs="0"/>
 <xs:element name="routeSegments" type="RouteSegment" maxOccurs="unbounded"/>
 </xs:sequence>
</xs:complexType>
```

B.1.3 RouteSegment

```
<xs:complexType name="RouteSegment">
 <xs:sequence>
 <xs:element name="startTime" type="tdt:DateTime"/>
 <xs:element name="travelTime" type="tdt:IntUnLoMB"/>
 <xs:element name="mode" type="rmr002_ModeType"/>
 <xs:element name="distance" type="tdt:IntUnLoMB" minOccurs="0"/>
 <xs:element name="cost" type="tdt:IntUnLoMB" minOccurs="0"/>
 <xs:element name="currencyType" type="tdt:typ003_CurrencyType" minOccurs="0"/>
 <xs:element name="energyConsumption" type="EnergyConsumption" minOccurs="0"/>
 <xs:element name="ptLine" type="tdt:LocalisedShortString" minOccurs="0"/>
 <xs:element name="description" type="tdt:LocalisedShortString" minOccurs="0"/>
 <xs:element name="operator" type="Operator" minOccurs="0"/>
 <xs:element name="infoLink" type="tdt:LongString" minOccurs="0"/>
```

```
 <xs:element name="segmentLoc" type="SegmentLocation" minOccurs="0"/>
 <xs:element name="segmentDetails" type="SegmentDetails" minOccurs="0"/>
 <xs:element name="parking" type="Parking" minOccurs="0"/>
 </xs:sequence>
</xs:complexType>
```

B.1.4 Parking

```
<xs:complexType name="Parking">
 <xs:sequence>
 <xs:element name="name" type="tdt:LocalisedShortString"/>
 <xs:element name="pType" type="rmr003_ParkType"/>
 <xs:element name="capacity" type="tdt:IntUnLoMB" minOccurs="0"/>
 <xs:element name="occupancy" type="tdt:IntUnLoMB" minOccurs="0"/>
 <xs:element name="costHour" type="tdt:IntUnLoMB" minOccurs="0"/>
 <xs:element name="costDay" type="tdt:IntUnLoMB" minOccurs="0"/>
 <xs:element name="currencyType" type="tdt:typ003_CurrencyType" minOccurs="0"/>
 <xs:element name="description" type="tdt:LocalisedShortString" minOccurs="0"/>
 <xs:element name="tendency" type="rmr008_OccTendency" minOccurs="0"/>
 <xs:element name="prediction" type="tdt:IntSiLoMB" minOccurs="0"/>
 <xs:element name="timeForParking" type="tdt:IntUnLoMB" minOccurs="0"/>
 <xs:element name="linkedPark" type="MessageLink" minOccurs="0"/>
 <xs:element name="pLoc" type="PointLocation"/>
 </xs:sequence>
</xs:complexType>
```

B.1.5 SegmentLocation

```
<xs:complexType name="SegmentLocation">
 <xs:sequence>
 <xs:element name="streetName" type="tdt:LocalisedShortString" minOccurs="0"/>
 <xs:element name="heading" type="tdt:IntUnTi" minOccurs="0"/>
 <xs:element name="locRef" type="lrc:LocationReferencingContainer" minOccurs="0"/>
 </xs:sequence>
</xs:complexType>
```

B.1.6 PointLocation

```
<xs:complexType name="PointLocation">
 <xs:sequence>
 <xs:element name="locRef" type="lrc:LocationReferencingContainer" minOccurs="0"/>
 </xs:sequence>
</xs:complexType>
```

B.1.7 SegmentDetails

```
<xs:complexType name="SegmentDetails">
 <xs:sequence>
 <xs:choice minOccurs="1" maxOccurs="1">
 <xs:element name="optionRoadTravelDetails" type="RoadTravelDetails" minOccurs="1"
maxOccurs="1"/>
 <xs:element name="optionPTTravelDetails" type="PTTravelDetails" minOccurs="1"
maxOccurs="1"/>
 <xs:element name="optionPedestrianTravelDetails" type="PedestrianTravelDetails"
minOccurs="1" maxOccurs="1"/>
 </xs:choice>
 </xs:sequence>
</xs:complexType>
```

B.1.8 RoadTravelDetails

```
<xs:complexType name="RoadTravelDetails">
 <xs:sequence>
 </xs:sequence>
</xs:complexType>
```

B.1.9 PTTravelDetails

```
<xs:complexType name="PTTravelDetails">
 <xs:sequence>
 <xs:element name="originStation" type="tdt:LocalisedShortString" minOccurs="0"/>
 <xs:element name="destinationStation" type="tdt:LocalisedShortString" minOccurs="0"/>
 <xs:element name="accessType" type="rmr007_AccessType" minOccurs="0"/>
 </xs:sequence>
</xs:complexType>
```

B.1.10 PedestrianTravelDetails

```
<xs:complexType name="PedestrianTravelDetails">
 <xs:sequence>
 <xs:element name="originPed" type="WGS84Coordinate" minOccurs="0"/>
 <xs:element name="destinationPed" type="WGS84Coordinate" minOccurs="0"/>
 <xs:element name="accessType" type="rmr007_AccessType" minOccurs="0"
maxOccurs="unbounded"/>
 </xs:sequence>
</xs:complexType>
```

B.1.11 RMRRequestMessage

```
<xs:complexType name="RMRRequestMessage">
 <xs:sequence>
 <xs:element name="mmtReq" type="mmy:MessageManagementContainer"/>
 <xs:element name="reqParams" type="RMRReqParams"/>
 </xs:sequence>
</xs:complexType>
```

B.1.12 RMRReqParams

```
<xs:complexType name="RMRReqParams">
 <xs:sequence>
 <xs:choice minOccurs="1" maxOccurs="1">
 <xs:element name="optionRMRListReqParams" type="RMRListReqParams" minOccurs="1"
maxOccurs="1"/>
 <xs:element name="optionRMRRouteReqParams" type="RMRRouteReqParams" minOccurs="1"
maxOccurs="1"/>
 <xs:element name="optionRMRUpdateRouteReqParams" type="RMRUpdateRouteReqParams"
minOccurs="1" maxOccurs="1"/>
 <xs:element name="optionRMRActivateRouteReqParams" type="RMRActivateRouteReqParams"
minOccurs="1" maxOccurs="1"/>
 <xs:element name="optionTerminateRMRSession" type="TerminateRMRSession" minOccurs="1"
maxOccurs="1"/>
 </xs:choice>
 </xs:sequence>
</xs:complexType>
```

B.1.13 RMRListReqParams

```
<xs:complexType name="RMRListReqParams">
 <xs:sequence>
 <xs:element name="maxListLen" type="tdt:IntUnLoMB"/>
 <xs:element name="listCountOffset" type="tdt:IntUnTi"/>
 <xs:element name="detailedRoute" type="tdt:Boolean"/>
 <xs:element name="routeType" type="rmr001_RouteType" minOccurs="0"
```

```
maxOccurs="unbounded"/>
 <xs:element name="predefConnection" type="PredefConnection" minOccurs="0"/>
 <xs:element name="departureTime" type="tdt:DateTime" minOccurs="0"/>
 <xs:element name="originLocation" type="PointLocation"/>
 <xs:element name="destLocation" type="PointLocation"/>
 <xs:element name="stopOverLocations" type="PointLocation" minOccurs="0"
maxOccurs="unbounded"/>
 <xs:element name="routePreferences" type="RoutePreferences" minOccurs="0"/>
 </xs:sequence>
</xs:complexType>
```

B.1.14 RMRRouteReqParams

```
<xs:complexType name="RMRRouteReqParams">
 <xs:sequence>
 <xs:element name="routeID" type="tdt:IntUnLoMB"/>
 </xs:sequence>
</xs:complexType>
```

B.1.15 RMRUpdateRouteReqParams

```
<xs:complexType name="RMRUpdateRouteReqParams">
 <xs:sequence>
 <xs:element name="position" type="PointLocation"/>
 </xs:sequence>
</xs:complexType>
```

B.1.16 RMRActivateRouteReqParams

```
<xs:complexType name="RMRActivateRouteReqParams">
 <xs:sequence>
 <xs:element name="routeID" type="tdt:IntUnLoMB"/>
 </xs:sequence>
</xs:complexType>
```

B.1.17 TerminateRMRSession

```
<xs:complexType name="TerminateRMRSession">
 <xs:sequence>
 </xs:sequence>
</xs:complexType>
```

B.1.18 RoutePreferences

```
<xs:complexType name="RoutePreferences">
 <xs:sequence>
 <xs:element name="disabledAccess" type="tdt:Boolean"/>
 <xs:element name="shortWalk" type="tdt:Boolean"/>
 <xs:element name="parkAtDestination" type="tdt:Boolean"/>
 <xs:element name="routeObjective" type="rmr005_RouteObjectives" minOccurs="0"/>
 <xs:element name="accessRestrictions" type="rmr007_AccessType" minOccurs="0"
maxOccurs="unbounded"/>
 <xs:element name="excludedModes" type="rmr002_ModeType" minOccurs="0"
maxOccurs="unbounded"/>
 <xs:element name="maxDistanceRoad" type="tdt:IntUnLoMB" minOccurs="0"/>
 <xs:element name="maxDistancePed" type="tdt:IntUnLoMB" minOccurs="0"/>
 <xs:element name="maxTransferPT" type="tdt:IntUnTi" minOccurs="0"/>
 </xs:sequence>
</xs:complexType>
```

B.2 Datatypes

B.2.1 EnergyConsumption

```
<xs:complexType name="EnergyConsumption">
 <xs:sequence>
 <xs:element name="type" type="rmr009_EnergyType"/>
 <xs:element name="consumption" type="tdt:IntUnLoMB"/>
 </xs:sequence>
</xs:complexType>
```

B.2.2 Operator

```
<xs:complexType name="Operator">
 <xs:sequence>
 <xs:element name="name" type="tdt:LocalisedShortString"/>
 <xs:element name="telNumber" type="tdt:ShortString" minOccurs="0"/>
 <xs:element name="url" type="tdt:ShortString" minOccurs="0"/>
 </xs:sequence>
</xs:complexType>
```

B.2.3 PredefConnection

```
<xs:complexType name="PredefConnection">
 <xs:sequence>
 <xs:element name="type" type="rmr002_ModeType"/>
 <xs:element name="ID" type="tdt:LocalisedShortString" minOccurs="0"/>
 <xs:element name="departTime" type="tdt:DateTime" minOccurs="0"/>
 <xs:element name="departLoc" type="PointLocation" minOccurs="0"/>
 </xs:sequence>
</xs:complexType>
```

B.2.4 WGS84Coordinate

```
<xs:complexType name="WGS84Coordinate">
 <xs:sequence>
 <xs:element name="long" type="tdt:IntSi24"/>
 <xs:element name="lat" type="tdt:IntSi24"/>
 </xs:sequence>
</xs:complexType>
```

B.2.5 MessageLink

```
<xs:complexType name="MessageLink">
 <xs:sequence>
 <xs:element name="messageID" type="tdt:IntUnLoMB"/>
 <xs:element name="COID" type="tdt:IntUnTi"/>
 <xs:element name="AID" type="tdt:IntUnLi"/>
 <xs:element name="SID" type="tdt:ServiceIdentifier" minOccurs="0"/>
 </xs:sequence>
</xs:complexType>
```

B.3 Full RMR Schema Definition

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- This XML schema is generated with tpegUMLconverter V2.3 -->
<xs:schema xmlns="http://www.tisa.org/TPEG/RMR_1_0"
   targetNamespace="http://www.tisa.org/TPEG/RMR_1_0"
   xmlns:xs="http://www.w3.org/2001/XMLSchema"
   xmlns:tsf="http://www.tisa.org/TPEG/SFW_1_1"
```

```
   xmlns:tdt="http://www.tisa.org/TPEG/TPEGDataTypes_2_0"
   xmlns:mmc="http://www.tisa.org/tpeg/mmc_1_1"
   xmlns:lrc="http://www.tisa.org/TPEG/lrc_2_1"
   xmlns:mmy="http://www.tisa.org/tpeg/mmc_1_1"
   elementFormDefault="qualified"
   attributeFormDefault="qualified">
   <xs:import namespace="http://www.tisa.org/TPEG/SFW_1_1" schemaLocation="SFW_1_1.xsd"/>
   <xs:import namespace="http://www.tisa.org/TPEG/TPEGDataTypes_2_0"
schemaLocation="TDT_2_0.xsd"/>
   <xs:import namespace="http://www.tisa.org/tpeg/mmc_1_1" schemaLocation="MMC_1_1.xsd"/>
   <xs:import namespace="http://www.tisa.org/TPEG/lrc_2_1" schemaLocation="LRC_2_1.xsd"/>
   <xs:element name="RMRMessage" type="RMRMessage"/>
   <xs:complexType name="RMRMessage">
      <xs:complexContent>
         <xs:extension base="tsf:ApplicationRootMessageML">
            <xs:sequence>
               <xs:element name="mmt" type="mmc:MessageManagementContainer"/>
               <xs:element name="route" type="Route" minOccurs="0"/>
               <xs:element name="loc" type="lrc:LocationReferencingContainer"
minOccurs="0"/>
            </xs:sequence>
         </xs:extension>
      </xs:complexContent>
   </xs:complexType>
   <xs:complexType name="Route">
      <xs:sequence>
         <xs:element name="routeID" type="tdt:IntUnLoMB"/>
         <xs:element name="routeListID" type="tdt:IntUnLoMB"/>
         <xs:element name="listCount" type="tdt:IntUnLoMB"/>
         <xs:element name="routeType" type="rmr001_RouteType"/>
         <xs:element name="routeDescriptor" type="tdt:LocalisedShortString"/>
         <xs:element name="startTime" type="tdt:DateTime"/>
         <xs:element name="travelTime" type="tdt:IntUnLoMB"/>
         <xs:element name="distance" type="tdt:IntUnLoMB" minOccurs="0"/>
         <xs:element name="lastReturn" type="tdt:DateTime" minOccurs="0"/>
         <xs:element name="shortListPT" type="tdt:LocalisedShortString" minOccurs="0"/>
         <xs:element name="routeCause" type="rmr006_RouteCause" minOccurs="0"/>
         <xs:element name="accessType" type="rmr007_AccessType" minOccurs="0"/>
         <xs:element name="causeDescription" type="tdt:LocalisedShortString"
minOccurs="0"/>
         <xs:element name="LinkedCause" type="MessageLink" minOccurs="0"/>
         <xs:element name="routeSegments" type="RouteSegment" maxOccurs="unbounded"/>
      </xs:sequence>
   </xs:complexType>
   <xs:complexType name="RouteSegment">
      <xs:sequence>
         <xs:element name="startTime" type="tdt:DateTime"/>
         <xs:element name="travelTime" type="tdt:IntUnLoMB"/>
         <xs:element name="mode" type="rmr002_ModeType"/>
         <xs:element name="distance" type="tdt:IntUnLoMB" minOccurs="0"/>
         <xs:element name="cost" type="tdt:IntUnLoMB" minOccurs="0"/>
         <xs:element name="currencyType" type="tdt:typ003_CurrencyType" minOccurs="0"/>
         <xs:element name="energyConsumption" type="EnergyConsumption" minOccurs="0"/>
         <xs:element name="ptLine" type="tdt:LocalisedShortString" minOccurs="0"/>
         <xs:element name="description" type="tdt:LocalisedShortString" minOccurs="0"/>
         <xs:element name="operator" type="Operator" minOccurs="0"/>
         <xs:element name="infoLink" type="tdt:LongString" minOccurs="0"/>
         <xs:element name="segmentLoc" type="SegmentLocation" minOccurs="0"/>
         <xs:element name="segmentDetails" type="SegmentDetails" minOccurs="0"/>
         <xs:element name="parking" type="Parking" minOccurs="0"/>
      </xs:sequence>
   </xs:complexType>
   <xs:complexType name="Parking">
      <xs:sequence>
         <xs:element name="name" type="tdt:LocalisedShortString"/>
         <xs:element name="pType" type="rmr003_ParkType"/>
         <xs:element name="capacity" type="tdt:IntUnLoMB" minOccurs="0"/>
         <xs:element name="occupancy" type="tdt:IntUnLoMB" minOccurs="0"/>
         <xs:element name="costHour" type="tdt:IntUnLoMB" minOccurs="0"/>
         <xs:element name="costDay" type="tdt:IntUnLoMB" minOccurs="0"/>
         <xs:element name="currencyType" type="tdt:typ003_CurrencyType" minOccurs="0"/>
```

```
         <xs:element name="description" type="tdt:LocalisedShortString" minOccurs="0"/>
         <xs:element name="tendency" type="rmr008_OccTendency" minOccurs="0"/>
         <xs:element name="prediction" type="tdt:IntSiLoMB" minOccurs="0"/>
         <xs:element name="timeForParking" type="tdt:IntUnLoMB" minOccurs="0"/>
         <xs:element name="linkedPark" type="MessageLink" minOccurs="0"/>
         <xs:element name="pLoc" type="PointLocation"/>
      </xs:sequence>
   </xs:complexType>
   <xs:complexType name="SegmentLocation">
      <xs:sequence>
         <xs:element name="streetName" type="tdt:LocalisedShortString" minOccurs="0"/>
         <xs:element name="heading" type="tdt:IntUnTi" minOccurs="0"/>
         <xs:element name="locRef" type="lrc:LocationReferencingContainer" minOccurs="0"/>
      </xs:sequence>
   </xs:complexType>
   <xs:complexType name="PointLocation">
      <xs:sequence>
         <xs:element name="locRef" type="lrc:LocationReferencingContainer" minOccurs="0"/>
      </xs:sequence>
   </xs:complexType>
   <xs:complexType name="SegmentDetails">
      <xs:sequence>
         <xs:choice minOccurs="1" maxOccurs="1">
            <xs:element name="optionRoadTravelDetails" type="RoadTravelDetails"
minOccurs="1" maxOccurs="1"/>
            <xs:element name="optionPTTravelDetails" type="PTTravelDetails" minOccurs="1"
maxOccurs="1"/>
            <xs:element name="optionPedestrianTravelDetails"
type="PedestrianTravelDetails" minOccurs="1" maxOccurs="1"/>
         </xs:choice>
      </xs:sequence>
   </xs:complexType>
   <xs:complexType name="RoadTravelDetails">
      <xs:sequence>
      </xs:sequence>
   </xs:complexType>
   <xs:complexType name="PTTravelDetails">
      <xs:sequence>
         <xs:element name="originStation" type="tdt:LocalisedShortString" minOccurs="0"/>
         <xs:element name="destinationStation" type="tdt:LocalisedShortString"
minOccurs="0"/>
         <xs:element name="accessType" type="rmr007_AccessType" minOccurs="0"/>
      </xs:sequence>
   </xs:complexType>
   <xs:complexType name="PedestrianTravelDetails">
      <xs:sequence>
         <xs:element name="originPed" type="WGS84Coordinate" minOccurs="0"/>
         <xs:element name="destinationPed" type="WGS84Coordinate" minOccurs="0"/>
         <xs:element name="accessType" type="rmr007_AccessType" minOccurs="0"
maxOccurs="unbounded"/>
      </xs:sequence>
   </xs:complexType>
   <xs:complexType name="RMRRequestMessage">
      <xs:sequence>
         <xs:element name="mmtReq" type="mmy:MessageManagementContainer"/>
         <xs:element name="reqParams" type="RMRReqParams"/>
      </xs:sequence>
   </xs:complexType>
   <xs:complexType name="RMRReqParams">
      <xs:sequence>
         <xs:choice minOccurs="1" maxOccurs="1">
            <xs:element name="optionRMRListReqParams" type="RMRListReqParams"
minOccurs="1" maxOccurs="1"/>
            <xs:element name="optionRMRRouteReqParams" type="RMRRouteReqParams"
minOccurs="1" maxOccurs="1"/>
            <xs:element name="optionRMRUpdateRouteReqParams"
type="RMRUpdateRouteReqParams" minOccurs="1" maxOccurs="1"/>
            <xs:element name="optionRMRActivateRouteReqParams"
type="RMRActivateRouteReqParams" minOccurs="1" maxOccurs="1"/>
            <xs:element name="optionTerminateRMRSession" type="TerminateRMRSession"
minOccurs="1" maxOccurs="1"/>
```

```
         </xs:choice>
      </xs:sequence>
   </xs:complexType>
   <xs:complexType name="RMRListReqParams">
      <xs:sequence>
         <xs:element name="maxListLen" type="tdt:IntUnLoMB"/>
         <xs:element name="listCountOffset" type="tdt:IntUnTi"/>
         <xs:element name="detailedRoute" type="tdt:Boolean"/>
         <xs:element name="routeType" type="rmr001_RouteType" minOccurs="0"
maxOccurs="unbounded"/>
         <xs:element name="predefConnection" type="PredefConnection" minOccurs="0"/>
         <xs:element name="departureTime" type="tdt:DateTime" minOccurs="0"/>
         <xs:element name="originLocation" type="PointLocation"/>
         <xs:element name="destLocation" type="PointLocation"/>
         <xs:element name="stopOverLocations" type="PointLocation" minOccurs="0"
maxOccurs="unbounded"/>
         <xs:element name="routePreferences" type="RoutePreferences" minOccurs="0"/>
      </xs:sequence>
   </xs:complexType>
   <xs:complexType name="RMRRouteReqParams">
      <xs:sequence>
         <xs:element name="routeID" type="tdt:IntUnLoMB"/>
      </xs:sequence>
   </xs:complexType>
   <xs:complexType name="RMRUpdateRouteReqParams">
      <xs:sequence>
         <xs:element name="position" type="PointLocation"/>
      </xs:sequence>
   </xs:complexType>
   <xs:complexType name="RMRActivateRouteReqParams">
      <xs:sequence>
         <xs:element name="routeID" type="tdt:IntUnLoMB"/>
      </xs:sequence>
   </xs:complexType>
   <xs:complexType name="TerminateRMRSession">
      <xs:sequence>
      </xs:sequence>
   </xs:complexType>
   <xs:complexType name="RoutePreferences">
      <xs:sequence>
         <xs:element name="disabledAccess" type="tdt:Boolean"/>
         <xs:element name="shortWalk" type="tdt:Boolean"/>
         <xs:element name="parkAtDestination" type="tdt:Boolean"/>
         <xs:element name="routeObjective" type="rmr005_RouteObjectives" minOccurs="0"/>
         <xs:element name="accessRestrictions" type="rmr007_AccessType" minOccurs="0"
maxOccurs="unbounded"/>
         <xs:element name="excludedModes" type="rmr002_ModeType" minOccurs="0"
maxOccurs="unbounded"/>
         <xs:element name="maxDistanceRoad" type="tdt:IntUnLoMB" minOccurs="0"/>
         <xs:element name="maxDistancePed" type="tdt:IntUnLoMB" minOccurs="0"/>
         <xs:element name="maxTransferPT" type="tdt:IntUnTi" minOccurs="0"/>
      </xs:sequence>
   </xs:complexType>
   <xs:complexType name="EnergyConsumption">
      <xs:sequence>
         <xs:element name="type" type="rmr009_EnergyType"/>
         <xs:element name="consumption" type="tdt:IntUnLoMB"/>
      </xs:sequence>
   </xs:complexType>
   <xs:complexType name="Operator">
      <xs:sequence>
         <xs:element name="name" type="tdt:LocalisedShortString"/>
         <xs:element name="telNumber" type="tdt:ShortString" minOccurs="0"/>
         <xs:element name="url" type="tdt:ShortString" minOccurs="0"/>
      </xs:sequence>
   </xs:complexType>
   <xs:complexType name="PredefConnection">
      <xs:sequence>
         <xs:element name="type" type="rmr002_ModeType"/>
         <xs:element name="ID" type="tdt:LocalisedShortString" minOccurs="0"/>
         <xs:element name="departTime" type="tdt:DateTime" minOccurs="0"/>
```

```
         <xs:element name="departLoc" type="PointLocation" minOccurs="0"/>
      </xs:sequence>
   </xs:complexType>
   <xs:complexType name="WGS84Coordinate">
      <xs:sequence>
         <xs:element name="long" type="tdt:IntSi24"/>
         <xs:element name="lat" type="tdt:IntSi24"/>
      </xs:sequence>
   </xs:complexType>
   <xs:complexType name="MessageLink">
      <xs:sequence>
         <xs:element name="messageID" type="tdt:IntUnLoMB"/>
         <xs:element name="COID" type="tdt:IntUnTi"/>
         <xs:element name="AID" type="tdt:IntUnLi"/>
         <xs:element name="SID" type="tdt:ServiceIdentifier" minOccurs="0"/>
      </xs:sequence>
   </xs:complexType>
   <xs:complexType name="rmr001_RouteType">
      <xs:attribute name="table" type="xs:string" fixed="rmr001_RouteType" use="required"/>
      <xs:attribute name="code" use="required">
         <xs:simpleType>
            <xs:restriction base="xs:unsignedByte">
               <xs:minInclusive value="0"/>
               <xs:maxInclusive value="3"/>
            </xs:restriction>
         </xs:simpleType>
      </xs:attribute>
   </xs:complexType>
   <xs:complexType name="rmr002_ModeType">
      <xs:attribute name="table" type="xs:string" fixed="rmr002_ModeType" use="required"/>
      <xs:attribute name="code" use="required">
         <xs:simpleType>
            <xs:restriction base="xs:unsignedByte">
               <xs:minInclusive value="1"/>
               <xs:maxInclusive value="99"/>
            </xs:restriction>
         </xs:simpleType>
      </xs:attribute>
   </xs:complexType>
   <xs:complexType name="rmr003_ParkType">
      <xs:attribute name="table" type="xs:string" fixed="rmr003_ParkType" use="required"/>
      <xs:attribute name="code" use="required">
         <xs:simpleType>
            <xs:restriction base="xs:unsignedByte">
               <xs:minInclusive value="0"/>
               <xs:maxInclusive value="3"/>
            </xs:restriction>
         </xs:simpleType>
      </xs:attribute>
   </xs:complexType>
   <xs:complexType name="rmr005_RouteObjectives">
      <xs:attribute name="table" type="xs:string" fixed="rmr005_RouteObjectives"
use="required"/>
      <xs:attribute name="code" use="required">
         <xs:simpleType>
            <xs:restriction base="xs:unsignedByte">
               <xs:minInclusive value="1"/>
               <xs:maxInclusive value="4"/>
            </xs:restriction>
         </xs:simpleType>
      </xs:attribute>
   </xs:complexType>
   <xs:complexType name="rmr006_RouteCause">
      <xs:attribute name="table" type="xs:string" fixed="rmr006_RouteCause"
use="required"/>
      <xs:attribute name="code" use="required">
         <xs:simpleType>
            <xs:restriction base="xs:unsignedByte">
               <xs:minInclusive value="0"/>
               <xs:maxInclusive value="4"/>
            </xs:restriction>
```

```
         </xs:simpleType>
      </xs:attribute>
   </xs:complexType>
   <xs:complexType name="rmr007_AccessType">
      <xs:attribute name="table" type="xs:string" fixed="rmr007_AccessType"
use="required"/>
      <xs:attribute name="code" use="required">
         <xs:simpleType>
            <xs:restriction base="xs:unsignedByte">
               <xs:minInclusive value="0"/>
               <xs:maxInclusive value="6"/>
            </xs:restriction>
         </xs:simpleType>
      </xs:attribute>
   </xs:complexType>
   <xs:complexType name="rmr008_OccTendency">
      <xs:attribute name="table" type="xs:string" fixed="rmr008_OccTendency"
use="required"/>
      <xs:attribute name="code" use="required">
         <xs:simpleType>
            <xs:restriction base="xs:unsignedByte">
               <xs:minInclusive value="0"/>
               <xs:maxInclusive value="3"/>
            </xs:restriction>
         </xs:simpleType>
      </xs:attribute>
   </xs:complexType>
   <xs:complexType name="rmr009_EnergyType">
      <xs:attribute name="table" type="xs:string" fixed="rmr009_EnergyType"
use="required"/>
      <xs:attribute name="code" use="required">
         <xs:simpleType>
            <xs:restriction base="xs:unsignedByte">
               <xs:minInclusive value="0"/>
               <xs:maxInclusive value="6"/>
            </xs:restriction>
         </xs:simpleType>
      </xs:attribute>
   </xs:complexType>
   <xs:complexType name="rmr099_ErrorCode">
      <xs:attribute name="table" type="xs:string" fixed="rmr099_ErrorCode" use="required"/>
      <xs:attribute name="code" use="required">
         <xs:simpleType>
            <xs:restriction base="xs:unsignedByte">
               <xs:minInclusive value="1"/>
               <xs:maxInclusive value="1"/>
            </xs:restriction>
         </xs:simpleType>
      </xs:attribute>
   </xs:complexType>
</xs:schema>
```

Bibliography

[1] ISO/TS 21219-2, *Intelligent transport systems — Traffic and travel information (TTI) via transport protocol experts group, generation 2 (TPEG2) — Part 2: UML modelling rules*

[2] ISO/TS 21219-3, *Intelligent transport systems - Traffic and travel information (TTI) via transport protocol experts group, generation 2 (TPEG2) — Part 3: UML to binary conversion rules*

[3] ISO/TS 21219-4, *Intelligent transport systems — Traffic and travel information (TTI) via transport protocol experts group, generation 2 (TPEG2) — Part 4: UML to XML conversion rules*

[4] ISO/TS 21219-6, *Intelligent transport systems - Traffic and travel information(TTI) via transport protocol experts group, generation 2 (TPEG2) — Part 6: Message management container (TPEG2-MMC)*

[5] ISO/TS 21219-14, *Intelligent transport systems — Traffic and travel information (TTI) via transport protocol experts group, generation 2 (TPEG2) — Part 14: Parking information application (TPEG2-PKI)*

[6] Specification TISA SP10022 Traffic and travel information (TTI) via transport protocol experts group, generation 2 (TPEG2) – Adaptation Layer Profiles, Part 3: Transmission via HTTP (TPEG-ALP3-HTTP)

ICS 03.220.01; 35.240.60

Price based on 43 pages